

科学技術計算ライブラリ  
ASL ユーザーズガイド  
＜基本機能編 第3分冊＞

# はしがき

本書は、科学技術計算ライブラリ ASL (Advanced Scientific Library) の概念、機能、利用方法などについて説明したものです。

当製品に対応する説明書は7分冊からなっており、構成は次のとおりです。このうち本書は、基本機能第3分冊について記述したものです。

## 基本機能 第1分冊

章	タイトル	内 容
1	使用の手引き	本説明書の構成、各項目の見方、および使用上の制限事項などの説明
2	格納モードの変換	配列データの格納モードの変換に関するサブルーチンのアルゴリズム、使用方法および使用例の説明
3	基本行列演算	行列の基本演算に関するサブルーチンのアルゴリズム、使用方法および使用例の説明
4	固有値・固有ベクトル	実行列、複素行列、実対称行列、エルミート行列、実対称バンド行列、実対称3重対角行列、実対称スパース行列、エルミートスパース行列の標準固有値問題および実行列、実対称行列、エルミート行列、実対称バンド行列の一般化固有値問題に関するサブルーチンのアルゴリズム、使用方法および使用例の説明

## 基本機能 第2分冊

章	タイトル	内 容
1	使用の手引き	本説明書の構成、各項目の見方、および使用上の制限事項などの説明
2	連立1次方程式(直接法)	実行列、複素行列、正値対称行列、実対称行列、エルミート行列、実バンド行列、正値対称バンド行列、実3重対角行列、実上三角行列、実下三角行列の連立1次方程式に関するサブルーチンのアルゴリズム、使用方法および使用例の説明

基本機能 第3分冊

章	タイトル	内 容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	フーリエ変換とその応用	1次元, 2次元および3次元の複素ならびに実フーリエ変換, 1次元, 2次元および3次元の畳み込み, 相関, パワー・スペクトル解析, ウェーブレット変換およびラプラス逆変換に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第4分冊

章	タイトル	内 容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	微分方程式とその応用	〔常微分方程式初期値問題〕 連立高階, 陰的連立, 行列型, スティフ問題の連立高階, 連立1階, 高階常微分方程式 〔常微分方程式境界値問題〕 連立高階, 連立1階, 高階, 線形高階, 線形2階常微分方程式 〔積分方程式〕 第2種フレドホルム型, 第1種ボルテラ型積分方程式 〔偏微分方程式〕 2次元および3次元の非同次ヘルムホルツ方程式 に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
3	数値微分	1変数関数および多変数関数の数値微分に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
4	数値積分	有限区間, 半無限区間, 全無限区間, 2次元有限区間, 多次元有限区間の数値積分に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
5	補間・近似	補間, 曲面補間, 最小二乗近似, 最小二乗曲面近似, チェビシェフ近似に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
6	スプライン関数	3次スプライン, 双3次スプラインおよびB-スプラインを用いた補間, 平滑化, 数値微分, 数値積分に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第 5 分冊

章	タイトル	内 容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	特殊関数	ベッセル関数, 変形ベッセル関数, 球ベッセル関数, ベッセル関数に関連した関数, ガンマ関数, ガンマ関数に関連した関数, 楕円関数, 初等関数の不定積分, ルジャンドル陪関数, 直交多項式, その他の特殊関数に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
3	ソート・順位付け	ソート, 順位付けに関するサブルーチンの使用方法および使用例の説明
4	方程式の根	代数方程式, 非線形方程式, 連立非線形方程式の根に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
5	極値問題・最適化	制約なし関数の極小化, 制約なし関数二乗和の極小化, 制約付き 1 変数関数の極小化, 制約付き多変数関数の最小化, 最短路問題に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明

基本機能 第 6 分冊

章	タイトル	内 容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	乱数の検定	一様乱数の検定, 分布乱数の検定に関するサブルーチンの使用方法および使用例の説明
3	確率分布	連続分布, 離散分布に関するサブルーチンの使用方法および使用例の説明
4	基礎統計量	基礎統計量, 分散共分散, 相関係数に関するサブルーチンの使用方法および使用例の説明
5	推定と検定	区間推定, 検定に関するサブルーチンの使用方法および使用例の説明
6	分散分析・実験計画	1 元配置, 2 元配置, 多元配置, 乱塊法, グレコ・ラテン方格法, 累積法に関するサブルーチンの使用方法および使用例の説明
7	ノンパラメトリック検定	$\chi^2$ 分布による検定, その他分布による検定に関するサブルーチンの使用方法および使用例の説明
8	多変量解析	主成分分析, 因子分析, 正準相関分析, 判別分析, クラスタ分析に関するサブルーチンの使用方法および使用例の説明
9	時系列分析	自己相関・相互相関, 自己共分散・相互共分散, 平滑化・需要予測に関するサブルーチンの使用方法および使用例の説明
10	回帰分析	線形回帰, 非線形回帰に関するサブルーチンの使用方法および使用例の説明

## 共有メモリ並列機能

章	タイトル	内 容
1	使用の手引き	本説明書の構成, 各項目の見方, および使用上の制限事項などの説明
2	基本行列演算	実行列および複素行列の積を求めるサブルーチンのアルゴリズム, 使用方法の説明
3	連立 1 次方程式 (直接法)	実行列, 複素行列, 実対称行列, エルミート行列の連立 1 次方程式 (直接法) に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
4	連立 1 次方程式 (反復法)	実正値対称スパース行列, 実対称スパース行列, 実非対称スパース行列の連立 1 次方程式 (反復法) に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
5	固有値・固有ベクトル	実対称行列およびエルミート行列の固有値問題に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
6	フーリエ変換とその応用	1 次元, 2 次元および 3 次元の複素ならびに実フーリエ変換, 2 次元および 3 次元の畳み込み, 相関, パワー・スペクトル解析に関するサブルーチンのアルゴリズム, 使用方法および使用例の説明
7	ソート	ソートに関するサブルーチンの使用方法および使用例の説明

2023 年 3 月 ASL 付属説明書 3.0.0-230301

- 備考 (1) 本書に説明しているすべての機能は, プログラムプロダクトであり, ASL 1.1 に対応しています.
- (2) 製品名などの固有名詞は, 各メーカーの登録商標または商標です.
- (3) 本ライブラリは, 最新の数値計算技法を取り入れ, 開発されたものです. 従って, 最新の技術を維持する目的から, 改良または新しく追加されたサブルーチンが, 既存のサブルーチンの機能を包含し, かつ, これまで以上の高速性能が得られる場合には, 既存のサブルーチンを削除することもあります.

# 目次

第 1 章	使用の手引	1
1.1	概説	1
1.1.1	科学技術計算ライブラリ ASL の概要	1
1.1.2	ASL の特長	1
1.2	ライブラリの種類	2
1.3	マニュアルについて	3
1.3.1	『概要』	3
1.3.2	サブルーチン説明文の構成	3
1.3.3	各項目の内容	3
1.4	サブルーチン名	7
1.5	注意事項	9
第 2 章	フーリエ変換とその応用	11
2.1	概要	11
2.1.1	使用上の注意	11
2.1.2	使用しているアルゴリズム	13
2.1.2.1	1次元(連続)フーリエ変換	13
2.1.2.2	多次元(連続)フーリエ変換	16
2.1.2.3	1次元フーリエ変換	16
2.1.2.4	多次元フーリエ変換	19
2.1.2.5	高速フーリエ変換	21
2.1.2.6	1次元(連続)畳み込みと1次元(連続)相関	23
2.1.2.7	1次元離散畳み込みと1次元離散相関	25
2.1.2.8	多次元(連続)畳み込みと多次元(連続)相関	30
2.1.2.9	パワー・スペクトル	30
2.1.2.10	ラプラス変換	35
2.1.2.11	ウェーブレット変換	38
2.1.3	参考文献	42
2.2	1次元複素フーリエ変換(実数引数型)	43
2.2.1	[非推奨]DFC1FB, RFC1FB 1次元複素フーリエ変換(初期化を含む変換)	43
2.2.2	[非推奨]DFC1BF, RFC1BF 1次元複素フーリエ変換(初期化後の変換)	46
2.3	1次元複素フーリエ変換(複素引数型)	50
2.3.1	[非推奨]ZFC1FB, CFC1FB 1次元複素フーリエ変換(初期化を含む変換)	50

2.3.2	[非推奨]ZFC1BF, CFC1BF	
	1次元複素フーリエ変換 (初期化後の変換)	53
2.4	1次元実フーリエ変換	57
2.4.1	[非推奨]DFR1FB, RFR1FB	
	1次元実フーリエ変換 (初期化を含む変換)	57
2.4.2	[非推奨]DFR1BF, RFR1BF	
	1次元実フーリエ変換 (初期化後の変換)	61
2.5	多重1次元複素フーリエ変換 (実数指数型)	66
2.5.1	[非推奨]DFCMFB, RFCMFB	
	多重1次元複素フーリエ変換 (初期化を含む変換)	66
2.5.2	[非推奨]DFCMBF, RFCMBF	
	多重1次元複素フーリエ変換 (初期化後の変換)	70
2.6	多重1次元複素フーリエ変換 (複素指数型)	76
2.6.1	[非推奨]ZFCMFB, CFCMFB	
	多重1次元複素フーリエ変換 (初期化を含む変換)	76
2.6.2	[非推奨]ZFCMBF, CFCMBF	
	多重1次元複素フーリエ変換 (初期化後の変換)	79
2.7	多重1次元実フーリエ変換	84
2.7.1	[非推奨]DFRMFB, RFRMFB	
	多重1次元実フーリエ変換 (初期化を含む変換)	84
2.7.2	[非推奨]DFRMBF, RFRMBF	
	多重1次元実フーリエ変換 (初期化後の変換)	88
2.8	2次元複素フーリエ変換 (実数指数型)	93
2.8.1	[非推奨]DFC2FB, RFC2FB	
	2次元複素フーリエ変換 (初期化を含む変換)	93
2.8.2	[非推奨]DFC2BF, RFC2BF	
	2次元複素フーリエ変換 (初期化後の変換)	96
2.9	2次元複素フーリエ変換 (複素指数型)	100
2.9.1	[非推奨]ZFC2FB, CFC2FB	
	2次元複素フーリエ変換 (初期化を含む変換)	100
2.9.2	[非推奨]ZFC2BF, CFC2BF	
	2次元複素フーリエ変換 (初期化後の変換)	103
2.10	2次元実フーリエ変換	107
2.10.1	[非推奨]DFR2FB, RFR2FB	
	2次元実フーリエ変換 (初期化を含む変換)	107
2.10.2	[非推奨]DFR2BF, RFR2BF	
	2次元実フーリエ変換 (初期化後の変換)	111
2.11	3次元複素フーリエ変換 (実数指数型)	116
2.11.1	[非推奨]DFC3FB, RFC3FB	
	3次元複素フーリエ変換 (初期化を含む変換)	116
2.11.2	[非推奨]DFC3BF, RFC3BF	
	3次元複素フーリエ変換 (初期化後の変換)	120
2.12	3次元複素フーリエ変換 (複素指数型)	125

2.12.1	[非推奨]ZFC3FB, CFC3FB 3次元複素フーリエ変換 (初期化を含む変換)	125
2.12.2	[非推奨]ZFC3BF, CFC3BF 3次元複素フーリエ変換 (初期化後の変換)	128
2.13	3次元実フーリエ変換	133
2.13.1	[非推奨]DFR3FB, RFR3FB 3次元実フーリエ変換 (初期化を含む変換)	133
2.13.2	[非推奨]DFR3BF, RFR3BF 3次元実フーリエ変換 (初期化後の変換)	137
2.14	畳み込み	143
2.14.1	DFCN1D, RFCN1D 1次元畳み込み	143
2.14.2	DFCN2D, RFCN2D 2次元畳み込み	152
2.14.3	DFCN3D, RFCN3D 3次元畳み込み	159
2.15	相関	169
2.15.1	DFCR1D, RFCR1D 1次元相関	169
2.15.2	DFCR2D, RFCR2D 2次元相関	177
2.15.3	DFCR3D, RFCR3D 3次元相関	184
2.16	パワー・スペクトル解析	194
2.16.1	DFPS1D, RFPS1D 1次元フーリエ・ピリオドグラム	194
2.16.2	DFPS2D, RFPS2D 2次元フーリエ・ピリオドグラム	201
2.16.3	DFPS3D, RFPS3D 3次元フーリエ・ピリオドグラム	208
2.17	ラプラス変換	220
2.17.1	DFLARA, RFLARA ラプラス逆変換 (有理関数)	220
2.17.2	DFLAGE, RFLAGE ラプラス逆変換 (一般関数)	225
2.18	ウェーブレット変換	228
2.18.1	DFWTH1, RFWTH1 Haar 関数の生成	228
2.18.2	DFWTHR, RFWTHR Haar 関数によるウェーブレット変換	230
2.18.3	DFWTHS, RFWTHS Haar 関数による逆ウェーブレット変換	233
2.18.4	DFWTH2, RFWTH2 Haar 関数の生成 (等間隔サンプリングデータ)	236



2.18.5	DFWTHT, RFWTHT	
	Haar 関数によるウェーブレット変換 (等間隔サンプリングデータ) . . . . .	239
2.18.6	DFWTHI, RFWTHI	
	Haar 関数による逆ウェーブレット変換 (等間隔サンプリングデータ) . . . . .	242
2.18.7	DFWTMF, RFWTMF	
	メキシカンハット関数の計算 . . . . .	246
2.18.8	DFWTMT, RFWTMT	
	メキシカンハット関数によるウェーブレット変換 . . . . .	248
2.18.9	DFWTFF, RFWTFF	
	フレンチハット関数の計算 . . . . .	250
2.18.10	DFWTFT, RFWTFT	
	フレンチハット関数によるウェーブレット変換 . . . . .	252
<b>付録 A</b>	<b>ASL で使用している計算機依存定数</b>	<b>255</b>
A.1	誤差判定のための単位 . . . . .	255
A.2	浮動小数点データの値の最大値・最小値 . . . . .	255

# 第 1 章 使用の手引

## 1.1 概説

### 1.1.1 科学技術計算ライブラリ ASL の概要

科学技術計算ライブラリ ASL (Advanced Scientific Library) は、数値解析プログラムの作成を強力に支援する数学ライブラリである。ASL では広範な数値解析分野で頻出するプログラムを提供しており、それらは VE(Vector Engine) 上で優れた実行速度と精度を実現するための高度な最適化が適用されている。ASL を用いることによって、難解な数値計算アルゴリズムの詳細に煩わされることなく高度な数値解析プログラムを作成することができ、数値解析プログラム開発の生産性を大幅に改善することができる。

ASL は、基本機能、共有メモリ並列機能で構成される。機能分類と本マニュアルの分冊との対応を表 1-1 に示す。

表 1-1 ASL の機能分類

機能分類	分冊
基本機能	第 1～6 分冊
共有メモリ並列機能	第 7 分冊

### 1.1.2 ASL の特長

ASL の特長は、次のとおりである。

- (1) ハードウェア性能を十分発揮できるように設計しており、コンパイラの最適化機能を用いて作成した。
- (2) 行列を扱うサブルーチンでは、行列の種類 (対称行列、エルミート行列など) に応じて最適に処理を行えるように、専用のサブルーチンをそれぞれ提供している。一般に、専用のサブルーチンを用いて処理を行った方が、処理性能を向上したり、必要なメモリ容量を節約したりすることができる。
- (3) 処理手順に従ってモジュール化を行い、コンポーネントサブルーチンごとの信頼性向上に努めるとともに、システム全体の効率化、信頼性向上を図った。
- (4) サブルーチンを利用した後の エラーインディケータの番号が体系的に決めてあるので、エラー情報を把握しやすい。

## 1.2 ライブラリの種類

ASL には、32 ビット整数型ライブラリと 64 ビット整数型ライブラリがある。32 ビット整数型ライブラリに含まれるサブルーチンの整数型の引数は、32 ビット (4 バイト) 整数型である。一方、64 ビット整数型ライブラリに含まれるサブルーチンの整数型の引数は、64 ビット (8 バイト) 整数型である。また、サブルーチンの実数型の引数によってサブルーチン名が異なる。サブルーチン名については、1.4 を参照のこと。

表 1-2 ASL で提供しているライブラリの種類

変数の大きさ (バイト)		引数の型宣言文	通称	ライブラリの種類
整数型	実数型			
4	8	INTEGER(4) REAL(8)	32 ビット整数型倍精度 サブルーチン	32 ビット整数型ライブラリ (リンクオプション: -lasl_sequential)
4	4	INTEGER(4) REAL(4)	32 ビット整数型単精度 サブルーチン	
8	8	INTEGER(8) REAL(8)	64 ビット整数型倍精度 サブルーチン	64 ビット整数型ライブラリ (リンクオプション: -lasl_sequential_i64)
8	4	INTEGER(8) REAL(4)	64 ビット整数型単精度 サブルーチン	

(注 1) 機能によっては、4 種類全てをサポートしているとは限らない。その場合、個別の説明の注意事項の欄に記述するので注意されたい。

(注 2) INTEGER(4) および REAL(4) で型宣言する場合、“(4)” は省略可。

---

## 1.3 マニュアルについて

ここでは本マニュアルの第2章以降の構成について述べる。

第2章以降は ASL で用いられるサブルーチンとその機能, 使用方法の説明を行う。

### 1.3.1 『概要』

各章の第1節では, 概要として各サブルーチンの効果的な使用法, 採用した手法およびそのアルゴリズム, 注意事項などについて述べてある。

### 1.3.2 サブルーチン説明文の構成

各章の第2節では, サブルーチンごとに以下の順で説明している。

- (1) 機能
- (2) 使用法
- (3) 引数
- (4) 制限条件
- (5) エラーインディケータ
- (6) 注意事項
- (7) 使用例

各項目は次に述べる原則に従って記述されている。

### 1.3.3 各項目の内容

#### (1) 機能

この項目では, サブルーチンの目的とする機能について簡単に述べてある。

#### (2) 使用法

この項目では, サブルーチン名とその引数の順序について記述してある。  
引数の並べ方は, 原則として次のように決められている。

CALL サブルーチン名 (入力引数, 入出力引数, 出力引数, ISW, ワーク, IERR)

ここで, ISW は処理の手順を指定するための入力引数であり, IERR は エラーインディケータである。ただし, 入力引数と入出力引数の順序が逆の場合もある。さらに次の規則にしたがっている。

- 配列は重要度に応じてできるだけ左方によせる。
- 配列名に続けて配列の大きさをそえる。同じ大きさをもつ配列が複数個あるときは, その最初の配列名に続けてその大きさを引数として与え, 2 番目以降の配列からは, その大きさは引数として与えない。

## (3) 引数

(2) 項で記述された引数について、順番に説明されている。その形式は以下のように統一されている。

引数	型	大きさ	入出力	内容
(a)	(b)	(c)	(d)	(e)

## (a) 引数

引数が記載されている。

## (b) 型

引数のデータの型を示す。次の略記号のいずれかに示されている。

I : 整数型

D : 倍精度実数型

R : 単精度実数型

Z : 倍精度複素数型

C : 単精度複素数型

整数型の引数には 64 ビット整数型と 32 ビット整数型とがある。サブルーチンの整数型引数が 64 ビット整数型であるのか 32 ビット整数型であるのかは、そのサブルーチンが 64 ビット整数型であるか 32 ビット整数型であるか、つまりライブラリの種類によって決められる (1.2 参照)。ユーザプログラムにおいて引数の型を宣言する際は、32 ビット整数型の引数は INTEGER(4)、64 ビット整数型の引数は INTEGER(8) を用いて宣言する必要がある。

## (c) 大きさ

指定された引数の必要な大きさを示す。2 以上を指定した場合には、このサブルーチンを利用したプログラム側で、その必要な領域を確保しなければならない。

1 : 変数であることを示す。

N : 要素が N 個の 1 次元配列であることを示す。この配列が指定された直後にその大きさを示す引数 N が定義される。ただし大きさ N が以前に定義された配列の大きさを規定している場合には省略される。このほかに数値のみにて指定する場合や、 $3 \times N$  や  $N + M$  のように、積または和の形で表記する場合もある。

M, N : M 行 N 列の 2 次元の配列であることを示す。この配列が指定される前にこの M と N が定義されていない場合は、この配列の直後にその大きさを示す引数 M または N が定義される。

## (d) 入出力

引数の内容説明が入力時であるか出力時であるかを示す。

## i. 「入力」とだけある場合 :

このサブルーチンを利用したプログラムに制御がもどったときに、引数の入力時の情報は保存されている。入力時の情報は特に断らない限り、利用者が与えなければならない。

## ii. 「出力」とだけある場合 :

引数には、サブルーチン内で計算された結果が出力される。入力時には何も入れなくてよい。

## iii. 「入力」と「出力」の両方に説明がある場合 :

サブルーチンに制御がわたる前とサブルーチンから制御がもどった後で、この引数の内容に変化がある場合である。入力時の情報は特に断らない限り、利用者が与えなければならない。

## iv. 「ワーク」とある場合 :

サブルーチン内で演算を行うときに利用する領域であることを示す。サブルーチンを利用するプログラム側で、指定された大きさの作業領域を確保しなければならない。なお、次の計算に流用するために、作業領域の内容を保存しておく必要がある場合がある。

## (e) 内容

入力時あるいは出力時に、引数が保持している情報について説明される。

- 「引数」の説明の例を次に示す。

例 実行列の LU 分解と条件数を求めるサブルーチン (DBGMLC, RBGMLC) の使用法は以下のとおりである。

倍精度サブルーチン:

CALL DBGMLC (A, LNA, N, IPVT, COND, W1, IERR)

単精度サブルーチン:

CALL RBGMLC (A, LNA, N, IPVT, COND, W1, IERR)

この場合の引数の説明は次のようになる。

表 1-3 引数の例

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$ 注	LNA, N	入力	実行列 A(2次元配列型)
				出力	A = LU と分解した時の単位上三角行列 U および下三角行列 L
2	LNA	I	1	入力	配列 A の整合寸法
3	N	I	1	入力	行列 A の次数 n
4	IPVT	I	N	出力	ピボット情報 IPVT(i): i 段目の処理において行 i と交換した行の番号
5	COND	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出力	条件の逆数
6	W1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	ワーク	作業領域
7	IERR	I	1	出力	エラーインディケータ

このサブルーチンを利用するには、まず、引数として使用する配列 A, IPVT および W1 を、呼び出し元の利用者プログラム側でアロケートする必要がある。それらはそれぞれ、 $\begin{Bmatrix} \text{倍精度} \\ \text{単精度} \end{Bmatrix}$ 注 実数型で大きさ

(LNA, N), 整数型で大きさ N,  $\begin{Bmatrix} \text{倍精度} \\ \text{単精度} \end{Bmatrix}$ 実数型で大きさ N の配列である。

また、64 ビット整数版を利用する場合には、整数型引数 (LNA, N, IPVT, IERR) はすべて INTEGER ではなく INTEGER(8) を用いて宣言する必要がある。

注 DBGMLC のときには倍精度実数型 (略記号 D), RBGMLC のときには実数型 (略記号 R) で宣言することを意味する。以下、本文中で特に断らない限り中括弧 {} 等の使用法は、同様の扱いとする。

このサブルーチンを使用するときには、A、LNA および N にデータを格納しておかなければならない。サブルーチン内では、与えられた行列の LU 分解と条件数の算出が行われ、結果が配列 A と変数 COND に格納される。また、後続サブルーチンで利用するため、ピボット情報 IPVT に格納される。

IERR は、入力データや処理途中の異常を利用者に知らせるための引数であり、正常の場合は 0 にセットされる。

なお、W1 はサブルーチン内でのみ使用する作業領域であるので、入力時および出力時の内容は特に意味をもたない。

#### (4) 制限条件

サブルーチンの引数の制限範囲を明確にしてある。

#### (5) エラーインディケータ

各サブルーチンには、エラーインディケータが出力引数として設けられている。このエラーインディケータは、IERR という変数名に統一されており、引数表の最後におかれている。各サブルーチンはサブルーチン内でエラー検出を行い、その結果を IERR に設定する。IERR の値の意味は、次の 5 段階に分かれている。

表 1-4 エラーインディケータの出力値区分

レベル	IERR の値	意 味	処 理 内 容
正 常	0	正常終了した。	結果は保証される。
警 告	1000 ~ 2999	ある条件のもとで一応の処理が終了した。	条件付きで結果は保証される。
異 常	3000 ~ 3499	引数が制限条件に違反したために処理が打ち切られた。	結果は保証されない。
	3500 ~ 3999	得られた結果がある検定条件を満足しなかった。	得られた結果を返す (結果は保証されない)。
	4000 以上	処理の途中で致命的なエラーが発見された。通常は処理を打ち切る。	結果は保証されない。

#### (6) 注意事項

サブルーチンを使用するときの注意点およびあいまいな点を明確にしてある。

#### (7) 使用例

サブルーチンの使い方の一例を載せてある。なお複数のサブルーチンを組み合わせて一つの例としてある場合もあるので注意されたい。出力結果は、32 ビット整数版での結果であり、コンパイラや組み込み関数の変更などにより丸め誤差の範囲で異なる場合がある。

本説明書に記載されている使用例のプログラムはソースコードの形で「ASL ユーザーズガイド」に収録されている。入力データも (もし存在する場合は) 「ASL ユーザーズガイド」に収録されている。コンパイラを用いて使用例のソースコードから実行形式ファイルを作成する場合には、ライブラリ本体とリンクする必要がある。

## 1.4 サブルーチン名

ASL の基本機能のサブルーチン名は、6桁のアルファニューメリック記号の集まりである。また、サブルーチン名の各記号にはそれぞれ意味を持ち、図 1-1 で表される。利用時には、計算用途に合わせてサブルーチン名を指定する必要がある。

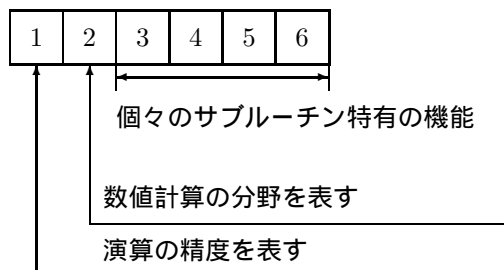


図 1-1 サブルーチン名の構成要素

図 1-1 の“1”：演算の精度を表す。基本機能編で使用される文字は、次の 8 種類である。

- D, W 倍精度実数型演算
- R, V 単精度実数型演算
- Z, J 倍精度複素数型演算
- C, I 単精度複素数型演算

ただし、上記の複素数型とは必ずしも引数の型が複素数型であることを意味しない。

図 1-1 の“2”：計算の分野を表す。現在、ASL では次の文字が使用されている。

文字	計算の分野	分冊
A	格納モードの変換	1
	基本行列演算	1, 7
B	連立 1 次方程式 (直接法)	2, 7
C	固有値・固有ベクトル	1, 7
F	フーリエ変換とその応用	3, 7
	時系列分析	6
G	スプライン関数	4
H	数値積分	4
I	特殊関数	5
J	乱数の検定	6
K	常微分方程式初期値問題	4
L	方程式の根	5
M	極値問題・最適化	5
N	近似・回帰分析	4, 6
O	常微分方程式境界値問題, 積分方程式, 偏微分方程式	4
P	補間	4
Q	数値微分	4



---

文字	計算の分野	分冊
S	ソート・順位付け	5, 7
X	基本行列演算	1
	連立1次方程式(反復法)	7
1	確率分布	6
2	標本統計	6
3	推定と検定	6
4	分散分析・実験計画	6
5	ノンパラメトリック検定	6
6	多変量解析	6

図1-1の“3”～“6”：これらの文字で、個々のサブルーチンに特有の機能を表す。

---

## 1.5 注意事項

- (1) 単精度版ではなく、倍精度版を標準として利用する方がよい。精度が高いことに加え、倍精度版の方が単精度版に比べて安定的に解が求まる場合 (特に固有値・固有ベクトル) が多い。
- (2) 演算例外の抑止はメインプログラム側で行う必要がある。ASL のサブルーチンでは、コンパイラの演算例外の抑止に関して、ユーザのメインプログラムのコンパイルパラメータの指示に従うように設定してある。
- (3) 扱う演算桁数を越える精度を期待することはできない。たとえば倍精度演算の (仮数部の) 演算桁数は 10 進 15 桁程度であるが、ここで数学的に 1 となるような値を計算した場合、 $10^{-15}$  程度の誤差は必ず発生する。これを抑制する方法として、任意桁数演算のような多倍長演算のエミュレートが考えられるが、この場合、たとえば円周率のような定数や関数近似の定数なども都度計算する必要が生じるので、通常の演算と比較して計算効率は悪くなる。
- (4) 数学的に解が存在しないような問題の解を得ることはできない。たとえば、数学的に特異な (または特異に近い) 行列を係数に持つ連立 1 次方程式の解を精度良く求めることは原理的にできない。なお、数値計算上は、数学的に特異な行列と特異に近い行列とを厳密に区別することはできない。もちろん、たとえば、条件数の計算値が設定した基準値以上であれば特異とみなすというようなことはいつでも可能である。
- (5) 浮動小数点例外 (オーバフローなど) をおこすようなデータを与えた場合、正常な計算結果を期待することはできない。ただし、反復計算で残差の加算等を行った場合に発生する浮動小数点アンダフローなどはこの限りではない。
- (6) 数値計算で扱う問題 (特に反復法を計算手法とする問題) では、与えるデータによっては解が精度良く求められない場合や全く求まらない場合がある。このような場合は、問題自体を見直して、解が求まるような問題に変更するなどの処置を講じる必要がある。たとえば、スパース行列を係数とする連立 1 次方程式を解く場合に、専用のサブルーチンで解が得られないときでも、密行列用のサブルーチンを用いることで解が得られる場合がある。
- (7) 解が複数ある問題を解く場合、実行するマシンや OS、用いるコンパイラ等で実行結果が見掛け上異なる場合がある。たとえば、固有値問題を解いた場合に得られる固有ベクトルがこれに相当する。
- (8) “[非推奨]” と表示のあるサブルーチンは、今後廃止予定の機能である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを利用されたい。



## 第 2 章 フーリエ変換とその応用

### 2.1 概要

本章では高速フーリエ変換 FFT(Fast Fourier Transform), 畳み込み, 相関, パワー・スペクトル解析, ウェーブレット変換およびラプラス逆変換を行うサブルーチンについて説明する.

離散型フーリエ変換では, 入力データの性質に応じて, 次のようなサブルーチンが用意されている.

- (1) 複素フーリエ変換  
入力データが複素数である.
- (2) 実フーリエ変換  
入力データが実数のみである.

本章で扱う離散型フーリエ変換は, 高速フーリエ変換を用いて実現している. 高速フーリエ変換は, 入力データの等分数 ( $n$ ) がどのような素数を基数としても変換が可能であるが, 大きな素数からなる数列では演算効率が減少する. 従って等分数 ( $n$ ) は小さな基数 (2, 3, 5 など) に因数分解できる数であることが望ましい.

畳み込み, 相関, パワー・スペクトル解析では, 次のようなデータを扱うサブルーチンが用意されている.

- (1) 1次元データ
- (2) 2次元データ
- (3) 3次元データ

ラプラス逆変換では, 像関数の型に応じて次のようなサブルーチンが用意されている.

- (1) 像関数有理関数の場合
- (2) 像関数が一般関数の場合

ウェーブレット変換では, 基底関数の型に応じて次のようなサブルーチンが用意されている.

- (1) 基底関数が Haar 関数である場合.
- (2) 基底関数がメキシカンハット関数である場合.
- (3) 基底関数がフレンチハット関数である場合.

#### 2.1.1 使用上の注意

- (1) 利用者は入力データのタイプにより適切なサブルーチン群を選ばばよい.
- (2) 一般に入力データが実数の場合は, 実フーリエ変換のサブルーチンが利用できる.
- (3) 変換の対象となる入力データは, 1 周期分  $[0, 2\pi]$  の入力データである.
- (4) 利用者はまず初期化を含む変換のサブルーチンを呼び出す必要がある. このサブルーチンでは, 三角関数テーブルの作成や因数分解がなされるので, 同じ入力データ数  $n$  で同じフーリエ変換を続ける場合には, これ以後, IFAX, TRIGS の内容を保存し, 初期化後の変換サブルーチンを利用することにより効率のよい処理ができる.

- (5) 一般関数のラプラス逆変換では、像関数  $F(s)$  の虚数部を計算する関数副プログラムを利用者が用意する。  $F(s)$  が多価関数の場合は、関数副プログラムの中で正しい枝を計算する様に注意しなければならない。また、像関数  $F(s)$  は無限遠で収束することが望ましい。 ( $\lim_{|s| \rightarrow \infty} F(s) = 0$ )

## 2.1.2 使用しているアルゴリズム

### 2.1.2.1 1次元(連続)フーリエ変換

次式で定義される積分  $F(\xi)$  を  $f(x)$  のフーリエ積分 (Fourier integral) という。

$$F(\xi) = a \int_{-\infty}^{\infty} f(x) e^{-\sqrt{-1}\xi x} dx$$

ここで  $a$  は定数であり、文献によって、 $a = 1$  ととる場合もあれば、 $a = \frac{1}{2\pi}$  や  $a = \frac{1}{\sqrt{2\pi}}$  ととる場合もある。なお、虚数単位  $\sqrt{-1}$  を表すために通常  $i$  や  $j$  が用いられるが、添え字等に用いる整数  $i, j$  と紛らわしいのであえて  $\sqrt{-1}$  を用いることにする。

もしこの積分がパラメータ  $\xi$  のあらゆる値に対して存在するとき  $F(\xi)$  を関数  $f(x)$  の(連続)フーリエ変換 (Fourier transform) と呼ぶ。この場合、 $F(\xi)$  は  $\mathcal{F}\{f(x)\}$  と表記される。例えば、 $x$  として時間  $t$  を考える場合には  $\xi$  は角周波数または角振動数 (angular frequency)  $\omega$  に対応し、 $x$  として空間座標  $r$  の成分を考える場合には  $\xi$  は波数ベクトル (wave vector)  $k$  の成分に対応する。また、 $x$  として時間  $t$  を考える場合には角周波数  $\omega$  の代わりに、 $f = \frac{\omega}{2\pi}$  で定義される周波数または振動数 (frequency)  $f$  も良く用いられる。なお、振動数を表す場合、文字  $f$  の代わりに文字  $\nu$  が利用される。 $f$  に対応する空間での量  $\frac{k}{2\pi}$  は空間周波数と呼ばれることもある。 $F(\xi)$  の逆変換は

$$f(x) = \frac{1}{2\pi a} \int_{-\infty}^{\infty} F(\xi) e^{\sqrt{-1}\xi x} d\xi$$

と定義される。この式が逆変換であることの形式的な証明は以下のとおり。

$$\begin{aligned} \frac{1}{2\pi a} \int_{-\infty}^{\infty} F(\xi) e^{\sqrt{-1}\xi x} d\xi &= \frac{1}{2\pi a} \int_{-\infty}^{\infty} \left\{ a \int_{-\infty}^{\infty} f(\eta) e^{-\sqrt{-1}\xi \eta} d\eta \right\} e^{\sqrt{-1}\xi x} d\xi \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\eta) \left\{ \int_{-\infty}^{\infty} e^{\sqrt{-1}\xi(x-\eta)} d\xi \right\} d\eta \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\eta) 2\pi \delta(x-\eta) d\eta \\ &= f(x) \end{aligned}$$

ここで、 $\delta(x)$  は超関数 (distribution) として定義される Dirac の  $\delta$  関数である。

最後の等号は  $f(x)$  が連続であるときに成立するが、不連続点  $x = x_0$  での関数  $f(x)$  の値を

$$f(x_0) = \lim_{h \rightarrow +0} \frac{f(x_0 + h) + f(x_0 - h)}{2}$$

と再定義すれば成立する。なお、連続フーリエ変換は実際上あらわれる大部分の関数に対して存在すると考えて良いが、(無限積分を扱うので) あらゆる関数について存在するわけではないということに注意しておく必要がある。フーリエ変換の存在条件については専門書を参照されたい。

また、フーリエ変換 (以降フーリエ順変換と呼ぶ) とその逆変換の定義を逆にする場合もあるので注意が必要である。例えば、時間と空間の両方を考える場合、複素平面波は角周波数  $\omega$  と波数ベクトル  $k$  を用いて  $e^{\sqrt{-1}(k \cdot r - \omega t)}$  と表せるが、この場合、時間についてのフーリエ変換と空間についてのフーリエ変換とでフーリエ変換の定義における指数関数のベキの符号を逆にしておく方が都合が良い。なお、この場合、フーリエ変換は関数の平面波展開に相当する。

$f(x)$  が実関数であれば、次のオイラーの公式 (Euler's formula) から明らかなように指数関数のベキの符号の変更はフーリエ変換の虚部の符号を変更することに対応する。

$$e^{\sqrt{-1}x} = \cos x + \sqrt{-1} \sin x$$

$f(x)$  と  $F(\xi)$  を組みにしてフーリエ変換対 (Fourier transform pair) と呼ぶ。

フーリエ変換対に対して次のようなパーセバルの定理 (Parseval's Theorem) が成立する。

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \frac{1}{2\pi a} \int_{-\infty}^{\infty} |F(\xi)|^2 d\xi$$

これは、時間 (または空間) 領域と周波数領域での全エネルギーの対応を与えるものであると解釈される。特に  $f(x)(=f(t))$  として実時間関数を考える場合には、負の周波数の出現を避けるために

$$\int_{-\infty}^{\infty} \{f(t)\}^2 dt = \frac{1}{\pi a} \int_0^{\infty} |F(\omega)|^2 d\omega \quad [\because f(t) \in \mathbf{R} \Rightarrow F(-\omega) = F(\omega)^*]$$

と変形する。

たとえば、電気回路で、理想抵抗  $R$ (単位:オーム  $[\Omega]$ ) に流れる電流が  $i(t)$ (単位:アンペア  $[A]$ ) の場合、 $|f(t)|^2$  に対応する電力は  $R(i(t))^2$ (単位:ワット  $[W]$ )、 $\int_{-\infty}^{\infty} |f(t)|^2 dt$  に対応する電力量は  $\int_{-\infty}^{\infty} R(i(t))^2 dt$ (単位:ジュール  $[J]$ ) となるが、電流  $i(t)$  の周波数スペクトル  $\hat{i}(f)$  についてはその単位を  $[A \cdot \text{sec}] = [A/Hz]$  として、その大きさを、

$$\int_{-\infty}^{\infty} \{R(i(t))^2\} dt = 2 \int_0^{\infty} R|\hat{i}(f)|^2 df$$

が成立するように正規化すればよい。もちろん、単位系を変えたい場合には、用いる単位に応じて、定数をかければよい。なお、多くの場合、パワー・スペクトル解析では、パワー・スペクトルに現れるピーク波形の半値全幅 (full width half maximum; FWHM) 等を問題にするので、このような場合には、特にスペクトルの絶対値を換算する必要はなく、便利な任意単位 (arbitrary unit) を用いればよい。

実関数  $f(x)$  が周期  $T_0$  の周期関数すなわち、任意の整数  $j$  について  $f(x) = f(x + jT_0)$  であれば関数を以下のようにフーリエ級数 (Fourier series) に展開できる (ただし、連続フーリエ変換の場合と同様、不連続点で関数はその点の左右両極限値の平均値を値としてとるものとする)。

$$\begin{aligned} f(x) &= \frac{a_0}{2} + \sum_{j=1}^{\infty} \{a_j \cos(j\omega_0 x) + b_j \sin(j\omega_0 x)\} \\ &= \sum_{j=-\infty}^{\infty} c_j e^{\sqrt{-1}j\omega_0 x} \quad (\omega_0 = \frac{2\pi}{T_0}) \end{aligned}$$

ここで、展開係数  $a_j, b_j$ (実数) ならびに  $c_j$ (複素数) はフーリエ係数 (Fourier coefficients) と呼ばれ、次式で与えられる。

$$\begin{aligned} a_j &= \frac{2}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(x) \cos(j\omega_0 x) dx \quad (j = 0, 1, 2, \dots) \\ b_j &= \frac{2}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(x) \sin(j\omega_0 x) dx \quad (j = 1, 2, \dots) \\ c_j &= \frac{1}{2}(a_j - \sqrt{-1}b_j) = \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(x) e^{-\sqrt{-1}j\omega_0 x} dx \quad (j = 0, \pm 1, \pm 2, \dots) \end{aligned}$$

なお、 $f(x)$  が複素値関数 ( $x$  は実数) の場合には、フーリエ係数  $a_j, b_j$  も複素数となる。 $f(x)$  の 1 周期分だけを取りだし、残りを 0 とした関数  $g(x)$  すなわち

$$g(x) = \begin{cases} f(x) & |x| \leq \frac{T_0}{2} \\ 0 & \text{それ以外} \end{cases}$$

を考えると、

$$\begin{aligned} c_j &= \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(x) e^{-\sqrt{-1}j\omega_0 x} dx \\ &= \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} g(x) e^{-\sqrt{-1}j\omega_0 x} dx \\ &= \frac{1}{T_0} \int_{-\infty}^{\infty} g(x) e^{-\sqrt{-1}j\omega_0 x} dx \\ &= \frac{1}{aT_0} G(\xi)_{\xi=j\omega_0} \end{aligned}$$

したがって定数因子を除けばフーリエ係数は周期関数  $(f(x))$  を 1 周期分で帯域制限した関数  $(g(x))$  のフーリエ変換  $G(\xi)$  の  $\xi = j\omega_0$  での値に等しい。逆に、帯域制限された関数  $(g(x))$  の離散点でのフーリエ変換  $(G(\xi)_{\xi=j\omega_0})$  は関数をその帯域を 1 周期とする周期関数  $(f(x))$  に拡張し、そのフーリエ係数  $(c_j)$  を求めることによって計算できる。なお、連続関数の標準化を行うに当たっては次の標準化定理がある。

標準化定理 (sampling theorem): 周波数  $f_c$  で帯域制限された時間関数  $g(t)$  の場合、すなわち、 $g(t)$  のフーリエ変換  $G(\omega)$  が  $|\omega| \leq 2\pi f_c$  でのみ非ゼロ値をとる場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下のように標準値列  $\{g(iT)\}$  だけの知識から  $g(t)$  を復元できる。

$$g(t) = T \sum_{i=-\infty}^{\infty} g(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

周波数帯域幅  $2f_c$  はナイキストの標準化周波数 (Nyquist sampling rate) と呼ばれる。

現実には、標準化定理の条件が満たされることはほとんどないが、離散フーリエ変換の応用においては、ナイキストの標準化周波数をもとに標準化が行われる。

離散フーリエ変換 (discrete Fourier transform) では、計算機で計算可能となるように、フーリエ級数における積分を以下のように方形近似して計算する。

$$\begin{aligned} c_j &= \frac{1}{T_0} \int_{-\frac{T_0}{2}}^{\frac{T_0}{2}} f(x) e^{-\sqrt{-1}j \frac{2\pi}{T_0} x} dx \\ &\simeq \frac{1}{T_0} \left\{ \sum_{k=-\lceil \frac{n}{2} \rceil - 1}^{-1} f_k e^{-\sqrt{-1}j \frac{2\pi}{T_0} \frac{kT_0}{n}} + \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} f_k e^{-\sqrt{-1}j \frac{2\pi}{T_0} \frac{kT_0}{n}} \right\} \frac{T_0}{n} \\ &= \frac{1}{n} \left\{ \sum_{k=\lceil \frac{n}{2} \rceil + 1}^{n-1} f_{k-n} e^{-2\pi\sqrt{-1} \frac{j(k-n)}{n}} + \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} f_k e^{-2\pi\sqrt{-1} \frac{jk}{n}} \right\} \\ &= \frac{1}{n} \left\{ \sum_{k=0}^{n-1} f_k e^{-2\pi\sqrt{-1} \frac{jk}{n}} \right\} \end{aligned}$$

ここで、 $f_k = f(x)|_{x=\frac{kT_0}{n}}$  ( $k = 0, \pm 1, \pm 2, \dots$ ) である。また  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数、 $\lceil x \rceil$  は  $x$  以上の最小の整数をそれぞれ表す。最後の等式では  $f(x)$  が周期  $T_0$  を周期とする周期関数であるから、 $f_k$  は  $n$  を周期とする離散周期関数となるという事実を用いている。この式から  $c_j$  も  $n$  を周期とする離散周期関数であることがわかる。したがって、離散フーリエ変換では、時間 (または空間) 領域での  $n$  個の標本と周波数領域の  $n$  個の対応する標本とが時間領域と周波数領域における波形の 1 周期を表す。なお、離散フーリエ変換はその周期性から、フーリエ級数における積分の台形公式による近似にもなっている。

$$\frac{1}{n} \left\{ \sum_{k=0}^{n-1} f_k e^{-2\pi\sqrt{-1} \frac{jk}{n}} \right\} = \frac{1}{n} \left[ \sum_{k=0}^{n-1} \frac{1}{2} \left\{ f_k e^{-2\pi\sqrt{-1} \frac{jk}{n}} + f_{k+1} e^{-2\pi\sqrt{-1} \frac{j(k+1)}{n}} \right\} \right]$$

$[\because f_n e^{-2\pi\sqrt{-1} \frac{jn}{n}} = f_0]$

本書では、以下特に断らない限り、フーリエ変換は「離散フーリエ変換」を指すものとする。離散フーリエ変換は高速フーリエ変換 (Fast Fourier Transform; FFT) アルゴリズムを用いることで効率良く計算できる。データ数  $n$  の離散フーリエ変換の計算に必要な複素数の乗算回数は、定義通り考えれば、 $n^2$  のオーダーとなるが、高速フーリエ変換では、 $n = r_1 r_2 \dots r_m$  と因数分解できるとき、 $n(r_1 + r_2 + \dots + r_m)$  のオーダーになる。特に、 $n = 2^m$  であれば、乗算回数は  $2n \log_2 n$  となる。したがって、通常  $n$  がそれほど大きくなくても、高速フーリエ変換アルゴリズムで計算を行う場合とそうでない場合の計算効率の違いは大きい。

なお、連続フーリエ変換では、もとの波形が不連続点を持てば、フーリエ級数の部分和はもとの波形に一樣には収束せず、ギブスの現象 (Gibbs phenomenon) として知られる振動が発生するが、離散フーリエ変換そのものにはギブスの現象はない。(離散フーリエ変換では変換後、逆変換を行えば、もとの系列に一致した系列が得られる。) ただし、連続フーリエ変換に対応する (無限個の) フーリエ係数のうち高次の項を打ち切った有限個が与えられた場合、調和合成



(harmonic synthesis) を行うことによって、ギブスの現象を再現することはできる。また、連続フーリエ変換では、逆変換式が成立するためには、不連続点における関数値を、その点の両側 (近傍) の値の平均値と定義しなければならなかったが、離散フーリエ変換では、必ずしもこのようにデータを選ぶ必要性はない (サンプリング間隔をそれが数学的な意味で近傍とみなせるほど十分小さくすることは、通常、現実的でない)。

### 2.1.2.2 多次元 (連続) フーリエ変換

フーリエ変換は多次元の場合に拡張することができる。例えば、4次元時空間の場合には位置ベクトル  $\mathbf{r} = (x, y, z)$ 、波数ベクトル (wave vector)  $\mathbf{k} = (k_x, k_y, k_z)$ 、時間  $t$ 、角周波数  $\omega$  を用いて4重積分として以下のように定義される。

$$\hat{f}(\mathbf{k}, \omega) = a \int_{-\infty}^{\infty} f(\mathbf{r}, t) e^{\sqrt{-1}(\mathbf{k} \cdot \mathbf{r} - \omega t)} d\mathbf{r} dt$$

$\hat{f}(\mathbf{k}, \omega)$  の逆変換は

$$f(\mathbf{r}, t) = \frac{1}{(2\pi)^4 a} \int_{-\infty}^{\infty} \hat{f}(\mathbf{k}, \omega) e^{-\sqrt{-1}(\mathbf{k} \cdot \mathbf{r} - \omega t)} d\mathbf{k} d\omega$$

と表せる。ここで  $a$  は定数であり、文献によって、 $a = 1$  とする場合もあれば、 $a = \frac{1}{(2\pi)^4}$  とする場合もある。なお、指数関数のべきの符号を逆にした定義をとる場合もあるので注意されたい。1次元の場合と同様に対応する離散フーリエ変換を多次元にも拡張できる。

### 2.1.2.3 1次元フーリエ変換

任意の整数  $k$  に対して  $\hat{c}_k = \hat{c}_{k+n}$  を満たす複素周期データ  $\hat{c}_k$  の1周期分  $c_k$  ( $k = 0, \dots, n-1$ ) に対して、複素フーリエ順変換  $C_j$  は次式で定義される。

$$C_j = \frac{1}{\alpha} \sum_{k=0}^{n-1} c_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1)$$

$\alpha$  は任意の定数であり通常1や  $n$  が選ばれる。このとき変換後の複素データ  $C_j$  ( $j = 0, \dots, n-1$ ) も任意の整数  $j$  に対して  $\hat{C}_j = \hat{C}_{j+n}$  を満たす複素周期データ  $\hat{C}_j$  の1周期分に相当する。対応する逆変換は

$$c_k = \frac{1}{n} \sum_{j=0}^{n-1} (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (k = 0, \dots, n-1)$$

である。この式が逆変換であることは、

$$\begin{aligned} \frac{1}{n} \sum_{j=0}^{n-1} (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} &= \frac{1}{n} \sum_{j=0}^{n-1} \left( \sum_{l=0}^{n-1} c_l e^{-2\pi\sqrt{-1}\frac{jl}{n}} \right) e^{2\pi\sqrt{-1}\frac{jk}{n}} \\ &= \frac{1}{n} \sum_{l=0}^{n-1} c_l \left\{ \sum_{j=0}^{n-1} e^{2\pi\sqrt{-1}\frac{j(k-l)}{n}} \right\} \\ &= \frac{1}{n} \sum_{l=0}^{n-1} c_l \{n\delta_{k,l}\} \\ &= c_k \end{aligned}$$

からわかる。ここで、

$$\sum_{j=0}^{n-1} e^{2\pi\sqrt{-1}\frac{j(k-l)}{n}} = n\delta_{k,l}$$

は選点直交関係として知られ、初項 1, 公比  $r = e^{2\pi\sqrt{-1}\frac{(k-l)}{n}}$ , 項数  $n$  の等比級数の和を考えれば容易に証明できる。 $\delta_{i,j}$  はクロネッカーのデルタで以下のように定義される。

$$\delta_{i,j} = \begin{cases} 1 & (i = j) \\ 0 & (\text{その他}) \end{cases}$$

離散フーリエ変換は、離散関数が値を取ればいつでも存在し、また、順変換に引き続き逆変換を行った結果は (数値計算上の誤差を除いて) もとの離散関数の値に一致する。なお、選点直交関係は良く知られた複素三角関数系  $\{e^{\sqrt{-1}kt}\}$  ( $k$ : 整数) の直交関係

$$\frac{1}{2\pi} \int_0^{2\pi} e^{\sqrt{-1}kt} e^{\sqrt{-1}lt} dt = \delta_{k,l}$$

の方形近似 (または台形公式による近似) に対応する。なお、本書のサブルーチンでは通常フーリエ変換の定義から定数倍を除いた  $\alpha C_j, n c_k$  を求める。

連続フーリエ変換では、通常関数の変域を原点に対して左右対称に取るので、これと対応させる場合には

$$\{\tilde{C}_k\}_{k=-(n-m-2), \dots, -1, 0, 1, \dots, m} = \{C_{m+1}, C_{m+2}, \dots, C_{n-1}, C_0, C_1, \dots, C_m\}$$

(ただし、 $m = \lfloor \frac{n}{2} \rfloor$  であり、 $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す) として半周期分のデータを 1 周期ずらして考えた方がよい (両側スペクトル (two-sided spectrum))。このとき  $C_0$  がゼロ周波数に対応する要素である。また、特に時間関数を考える場合には、負の周波数を避けるために、

$$\{\tilde{C}_k\}_{k=0, 1, \dots, m} = \begin{cases} \{C_0, 2C_1, \dots, 2C_{m-1}, C_m\} & n: \text{偶数} \\ \{C_0, 2C_1, \dots, 2C_m\} & n: \text{奇数} \end{cases}$$

として考える場合もある (片側スペクトル (one-sided spectrum))。

(1) 1 次元複素フーリエ変換と多重 1 次元複素フーリエ変換

本書では、フーリエ変換を行うデータの格納法の違いにより 1 次元複素フーリエ変換と多重 1 次元複素フーリエ変換をわけて考える。1 次元複素フーリエ変換ではデータ  $c_k$  または  $C_j$  は配列に連続に格納するのに対して、多重 1 次元複素フーリエ変換では一定の間隔をおいて格納する。多重フーリエ変換は通常多次元フーリエ変換やフーリエ変換と他の変換との混合による多次元変換を行う場合に利用できる。

(2) 1 次元実フーリエ変換と多重 1 次元実フーリエ変換

フーリエ順変換を行うデータが実数の場合、

$$\begin{aligned} \alpha C_{n-j}^* &= \sum_{k=0}^{n-1} c_k^* e^{2\pi\sqrt{-1}\frac{(n-j)k}{n}} \\ &= \sum_{k=0}^{n-1} c_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad [\because e^{-2\pi k\sqrt{-1}} = 1] \\ &= \alpha C_j \end{aligned}$$

を満たす。ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。特に、 $C_0$  は実数で、 $n$  が偶数のときは、 $C_{\frac{n}{2}}$  も実数となる。また、逆変換は

$$\begin{aligned}
 nc_k &= \sum_{j=0}^{n-1} (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} \\
 &= \alpha(C_0 + (-1)^k \hat{C}_{\frac{n}{2}}) + \sum_{j=1}^{\lceil \frac{n}{2} \rceil - 1} (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} + \sum_{j=\lfloor \frac{n}{2} \rfloor + 1}^{n-1} (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} \\
 &= \alpha(C_0 + (-1)^k \hat{C}_{\frac{n}{2}}) + \sum_{j=1}^{\lceil \frac{n}{2} \rceil - 1} \left\{ (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} + (\alpha C_{n-j}) e^{2\pi\sqrt{-1}\frac{(n-j)k}{n}} \right\} \\
 &= \alpha(C_0 + (-1)^k \hat{C}_{\frac{n}{2}}) + \sum_{j=1}^{\lceil \frac{n}{2} \rceil - 1} \left[ (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} + \{ (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} \}^* \right] \\
 &= \alpha(C_0 + (-1)^k \hat{C}_{\frac{n}{2}}) + 2 \sum_{j=1}^{\lceil \frac{n}{2} \rceil - 1} \Re\{ (\alpha C_j) e^{2\pi\sqrt{-1}\frac{jk}{n}} \} \\
 &= \alpha(C_0 + (-1)^k \hat{C}_{\frac{n}{2}}) + 2 \sum_{j=1}^{\lceil \frac{n}{2} \rceil - 1} \left[ (\alpha \Re\{C_j\}) \cos(2\pi\frac{jk}{n}) - (\alpha \Im\{C_j\}) \sin(2\pi\frac{jk}{n}) \right]
 \end{aligned}$$

ここで  $[x]$  は  $x$  を超えない最大の整数、 $\lceil x \rceil$  は  $x$  以上の最小の整数をそれぞれ表す。また、 $n$  が奇数のとき  $\hat{C}_{\frac{n}{2}} = 0$ 、 $n$  が偶数のとき  $\hat{C}_{\frac{n}{2}} = C_{\frac{n}{2}}$ 。したがって、フーリエ変換は一般の複素データの場合の半分のデータ ( $c_k$  についてはその実部のみ、 $C_j$  についてはその半周期分) で計算を実行できる。

たとえば、図 2-1 のような実数データ  $r_i$  を複素フーリエ変換すると図 2-2 のようになるが、実フーリエ変換では図 2-3 のように実部と虚部それぞれ半周期分が計算される。(なお、実フーリエ変換では通常、変換結果の実部と虚部を配列に交互に格納するので注意のこと。)

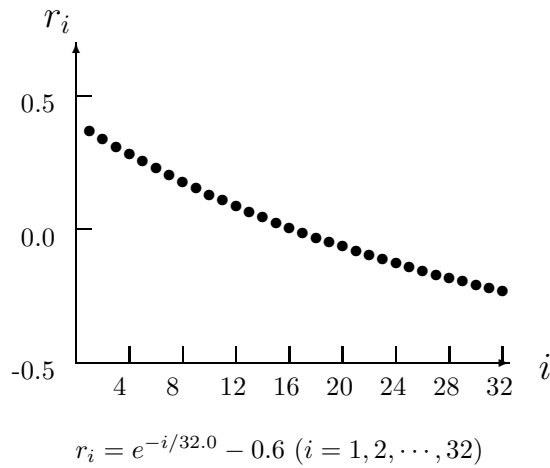


図 2-1 フーリエ変換前のデータ

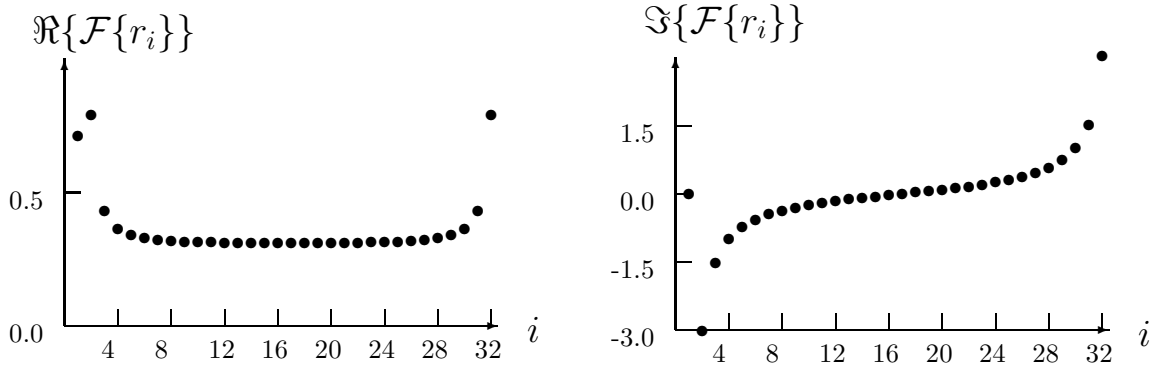
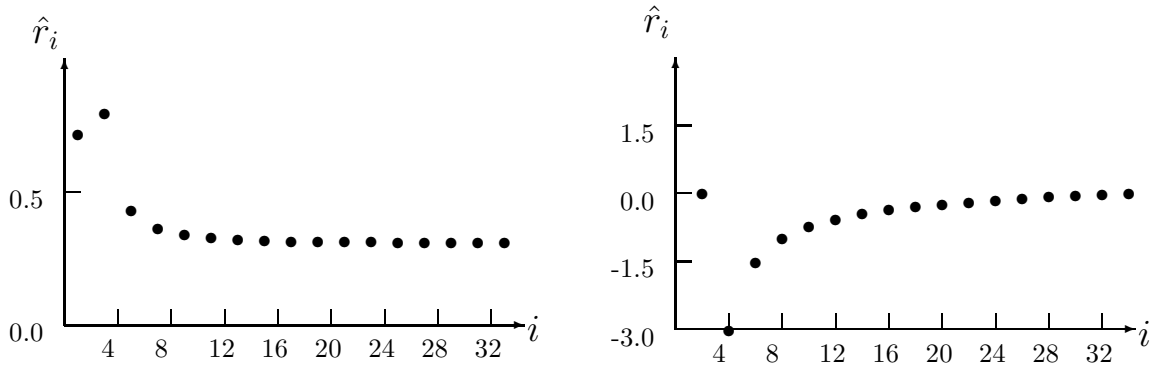


図 2-2 フーリエ変換後のデータ (複素フーリエ変換の場合)



$$\hat{r}_{2i-1} = \Re\{\mathcal{F}\{r_i\}\}, \quad \hat{r}_{2i} = \Im\{\mathcal{F}\{r_i\}\}, \quad (i = 1, 2, \dots, 17)$$

図 2-3 フーリエ変換後のデータ (実フーリエ変換の場合)

なお,  $n$  が偶数のとき,  $(C_0, C_{\frac{n}{2}})$  を新しい複素数  $C_0$  と定義して計算に必要な領域を節約する場合があるので注意されたい. また, 多重 1 次元複素フーリエ変換の場合と同様多重 1 次元実フーリエ変換ではデータを配列に一定の間隔をおいて格納する.

#### 2.1.2.4 多次元フーリエ変換

##### (1) 2 次元複素フーリエ変換

任意の整数  $k_x, k_y$  に対して  $\hat{c}_{k_x, k_y} = \hat{c}_{k_x+n_x, k_y+n_y}$  を満たす複素多重周期データ  $\hat{c}_{k_x, k_y}$  の 1 周期分  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1$ ) に対して, 複素フーリエ順変換  $C_{j_x, j_y}$  は次式で定義される.

$$C_{j_x, j_y} = \frac{1}{\alpha} \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)}$$

$$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

$\alpha$  は任意の定数であり通常 1 や  $n_x n_y$  が選ばれる. このとき変換後の複素データ  $C_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1$ ) も任意の整数  $j_x, j_y$  に対して  $\hat{C}_{j_x, j_y} = \hat{C}_{j_x+n_x, j_y+n_y}$  を満たす複素多重周期データ  $\hat{C}_{j_x, j_y}$  の 1 周

期分に相当する. 対応する逆変換は

$$c_{k_x, k_y} = \frac{1}{n_x n_y} \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} (\alpha C_{j_x, j_y}) e^{2\pi\sqrt{-1}(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y})}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

である. なお, 本書のサブルーチンでは通常フーリエ変換の定義から定数倍を除いた  $\alpha C_{j_x, j_y}$ ,  $n_x n_y c_{k_x, k_y}$  を求める.

## (2) 2次元実フーリエ変換

2次元フーリエ順変換を行うデータが実数の場合,

$$\alpha C_{n_x-j_x, n_y-j_y}^* = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y}^* e^{2\pi\sqrt{-1}\{\frac{(n_x-j_x)k_x}{n_x} + \frac{(n_y-j_y)k_y}{n_y}\}}$$

$$= \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\{\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\}}$$

$$= \alpha C_{j_x, j_y}$$

を満たす. ただし,  $z^*$  は複素数  $z$  の共役複素数を表す. 特に,  $C_{0,0}$  は実数で,  $n_x$  と  $n_y$  が偶数のときは,  $C_{\frac{n_x}{2}, \frac{n_y}{2}}$  も実数となる.

同様に

$$\alpha C_{n_x-j_x, j_y}^* = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y}^* e^{2\pi\sqrt{-1}\{\frac{(n_x-j_x)k_x}{n_x} + \frac{j_y k_y}{n_y}\}}$$

$$= \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\{\frac{j_x k_x}{n_x} + \frac{(n_y-j_y)k_y}{n_y}\}}$$

$$= \alpha C_{j_x, n_y-j_y}$$

したがって, フーリエ変換は一般の複素データの場合の半分のデータ ( $c_{k_x, k_y}$  についてはその実部のみ,  $C_{j_x, j_y}$  については  $j_x$  または  $j_y$  どちらかに対してその半周期分) で計算を実行できる.

## (3) 3次元複素フーリエ変換

任意の整数  $k_x, k_y, k_z$  に対して  $\hat{c}_{k_x, k_y, k_z} = \hat{c}_{k_x+n_x, k_y+n_y, k_z+n_z}$  を満たす複素多重周期データ  $\hat{c}_{k_x, k_y, k_z}$  の1周期分  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1$ ) に対して, 複素フーリエ順変換  $C_{j_x, j_y, j_z}$  は次式で定義される.

$$C_{j_x, j_y, j_z} = \frac{1}{\alpha} \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z})}$$

$$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

$\alpha$  は任意の定数であり通常1や  $n_x n_y n_z$  が選ばれる. このとき変換後の複素データ  $C_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1$ ) も任意の整数  $j_x, j_y, j_z$  に対して  $\hat{C}_{j_x, j_y, j_z} = \hat{C}_{j_x+n_x, j_y+n_y, j_z+n_z}$  を満たす複素多重周期データ  $\hat{C}_{j_x, j_y, j_z}$  の1周期分に相当する. 対応する逆変換は

$$c_{k_x, k_y, k_z} = \frac{1}{n_x n_y n_z} \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} \sum_{j_z=0}^{n_z-1} (\alpha C_{j_x, j_y, j_z}) e^{2\pi\sqrt{-1}(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z})}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

である. なお, 本書のサブルーチンでは通常フーリエ変換の定義から定数倍を除いた  $\alpha C_{j_x, j_y, j_z}$ ,  $n_x n_y n_z c_{k_x, k_y, k_z}$  を求める.

(4) 3次元実フーリエ変換

3次元フーリエ順変換を行うデータが実数の場合,

$$\begin{aligned} \alpha C_{n_x-j_x, n_y-j_y, n_z-j_z}^* &= \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} C_{k_x, k_y, k_z}^* e^{2\pi\sqrt{-1}\left\{\frac{(n_x-j_x)k_x}{n_x} + \frac{(n_y-j_y)k_y}{n_y} + \frac{(n_z-j_z)k_z}{n_z}\right\}} \\ &= \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} C_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left\{\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right\}} \\ &= \alpha C_{j_x, j_y, j_z} \end{aligned}$$

を満たす。ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。特に、 $C_{0,0,0}$  は実数で、 $n_x, n_y, n_z$  すべてが偶数のときは、 $C_{\frac{n_x}{2}, \frac{n_y}{2}, \frac{n_z}{2}}$  も実数となる。

同様に

$$C_{n_x-j_x, j_y, j_z}^* = C_{j_x, n_y-j_y, n_z-j_z}$$

等の関係が成立する。すなわち、 $j_x, j_y, j_z$  のすべてについて次の対応関係を用いて置き換えを行ったものは置き換え前のものと互いに複素共役の関係にある。

$$\begin{aligned} j_x &\leftrightarrow n_x - j_x \\ j_y &\leftrightarrow n_y - j_y \\ j_z &\leftrightarrow n_z - j_z \end{aligned}$$

したがって、フーリエ変換は一般の複素データの場合の半分のデータ ( $C_{k_x, k_y, k_z}$  についてはその実部のみ、 $C_{j_x, j_y, j_z}$  については  $j_x$  または  $j_y$  または  $j_z$  のどれかに対してその半周期分) で計算を実行できる。

2.1.2.5 高速フーリエ変換

(1) 高速フーリエ変換のアルゴリズム

ここでは高速フーリエ変換の任意基数の Temperton のアルゴリズムを説明する。

1次元フーリエ変換を次式で定義する。

$$C_k = \sum_{j=1}^N c_j W^{(j-1)(k-1)}$$

$$k = 1, \dots, N (N: \text{入力データ数}), W = e^{-2\pi i/N}, i = \sqrt{-1} \text{ (1/N は省略する)}.$$

ここで、フーリエ変換を行列形式で考える。変換式は以下のように表せる。

$$X = W_n C, [W_n](j, k) = \omega^{(j-1)(k-1)}, \omega = \exp(-2\pi i/N)$$

いま、データ数  $N = n_1 n_2 n_3 n_4 \dots n_k$  ならば

$$W_n = T_K T_{K-1} T_{K-2} \dots T_2 T_1$$

と表せる。

$$l_1 = 1, l_{i+1} = n_i l_i, m_i = N/l_{i+1} \text{ (} i = 1, 2, \dots, k \text{)}$$

とすると

$$T_i = (P_{m_i}^{n_i} D_{m_i}^{n_i} \times I_{l_i}) (W_{n_i} \times I_{N/n_i})$$

ゆえに

$$\begin{aligned} X &= W_n C \\ &= (P_1^{n_k} D_1^{n_k} \times I_{l_k})(W_{n_k} \times I_{N/n_k}) \cdots (P_{N/n_1}^{n_1} D_{N/n_1}^{n_1} \times I_1)(W_{n_1} \times I_{N/n_1})C \end{aligned}$$

例えば,  $N = 24$  の場合,

$$N = 2 \times 3 \times 4$$

$$T_1 = (P_{12}^2 D_{12}^2 \times I_1)(W_2 \times I_{12})$$

$$T_2 = (P_4^3 D_4^3 \times I_2)(W_3 \times I_8)$$

$$T_3 = (P_1^4 D_1^4 \times I_6)(W_4 \times I_6)$$

$$\begin{aligned} X &= W_{24} C \\ &= T_3 T_2 T_1 C \\ &= (P_1^4 D_1^4 \times I_6)(W_4 \times I_6)(P_4^3 D_4^3 \times I_2)(W_3 \times I_8)(P_{12}^2 D_{12}^2 \times I_1)(W_2 \times I_{12})C \end{aligned}$$

(注)

(a)  $\times$  はクロネッカー積を表し, たとえば,

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

に対して,

$$A \times B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a \begin{bmatrix} e & f \\ g & h \end{bmatrix} & b \begin{bmatrix} e & f \\ g & h \end{bmatrix} \\ c \begin{bmatrix} e & f \\ g & h \end{bmatrix} & d \begin{bmatrix} e & f \\ g & h \end{bmatrix} \end{bmatrix}$$

という演算で定義される.

(b)  $P_{m_i}^{n_i}$  はオーダ  $n_i, m_i$  の交換行列

(c)  $D_{m_i}^{n_i}$  はオーダ  $n_i, m_i$  の対角行列

(d)  $I_r$  はオーダ  $r$  の単位行列

この計算は, 次のような流れになっている. ( $C$ :入力データ,  $A$ :出力データとする.)

```

LA = 1
  I = 1, 基数の個数
  IFAC ← 基数 (2, 3, 4, 5, 6, 7, 8, または任意)
  基数 IFAC の FFT 計算
  M = N/IFAC
  I = 1
  J = 1
  JUMP = (IFAC - 1) × LA
  K = 0, M - LA, LA
    L = 1, LA
      A(J) = Ω(K) × (W(IFAC) × C(I))
      I = I + 1
      J = J + 1
    J = J + JUMP
(注) 1. W(IFAC): オーダ IFAC の DFT 配列
      2. Ω(K): diag(trigs(1), trigs(K + 1), trigs(2 × K + 1), ...,
                  trigs((IFAC - 1) × K + 1))
      3. trigs(K + 1) = exp(2iKπ/N), 0 ≤ K ≤ N - 1, i = √-1
  LA = LA × IFAC, A と C を変換
  
```

(2) 多重 1 次元複素フーリエ変換について

長さ  $N$  からなる  $M$  組のデータ列のそれぞれに 1 次元複素フーリエ変換を実行する多重 1 次元複素フーリエ変換では、 $M$  組のデータ列は各々独立であるため、 $M$  回の 1 次元複素フーリエ変換の実行のためのループと通常の 1 次元複素フーリエ変換のループを入れ替えることが可能である。これにより、通常の 1 次元複素高速フーリエ変換において生じる種々の問題（入力データ、回転因子のメモリ参照、定義時のバンクコンフリクト、ループ長の減少など）を容易に避けることができ、高速性能が得られる。

(3) 実フーリエ変換について

実フーリエ変換は、変換の対象となる実入力データ  $\{r_k\}$  を虚部が零である複素入力データとみなし、それに対して複素フーリエ変換を行えば得られる。この場合、複素フーリエ変換の結果は、次のような共役関係 (conjugate symmetry) がある。

$$C_{N-j+2} = C_j^* \quad j = 2, \dots, N$$

ここで、 $*$  は、複素共役を表わす。また、実フーリエ変換の結果の  $\{a_j\}$   $\{b_j\}$  と、複素フーリエ変換による結果の  $\{C_j\}$  との関連は、次式で示される。

$$\begin{aligned}
 a_0 &= 2C_1, \quad a_{N/2} = 2C_{N/2+1} \\
 a_j &= (C_{j+1} + C_{N-j+1}) \quad j = 1, \dots, \frac{N}{2} - 1 \\
 b_j &= i (C_{j+1} - C_{N-j+1}) \quad j = 1, \dots, \frac{N}{2} - 1
 \end{aligned}$$

従って、実フーリエ変換を複素フーリエ変換により行う場合には、 $C_j$  ( $j = 1, \dots, \frac{N}{2} + 1$ ) だけを求めれば  $\{a_j\}$  と  $\{b_j\}$  を求められることがわかる。

2.1.2.6 1 次元 (連続) 畳み込みと 1 次元 (連続) 相関

次式で定義される積分  $p(x)$  を  $f(x)$  と  $g(x)$  との畳み込み積分 (convolution integral) という。

$$p(x) = \int_{-\infty}^{\infty} f(\xi)g(x - \xi)d\xi$$



$f(x)$  と  $g(x)$  との畳み込み積分は  $(f \times g)(x)$  や  $f(x) \times g(x)$  と表される. なお, 畳み込み積分はしばしば単に「畳み込み」と呼ばれる.  $f(x)$  と  $g(x)$  のフーリエ変換をそれぞれ  $\mathcal{F}\{f(x)\}$ ,  $\mathcal{F}\{g(x)\}$  とすると  $(f \times g)(x)$  と  $\mathcal{F}\{f(x)\}\mathcal{F}\{g(x)\}$  はフーリエ変換対をなす. すなわち

$$\begin{aligned} & \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} f(\xi)g(x-\xi)d\xi \right\} e^{-\sqrt{-1}\eta x} dx \\ &= \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} g(x-\xi)e^{-\sqrt{-1}\eta(x-\xi)} dx \right\} f(\xi)e^{-\sqrt{-1}\eta\xi} d\xi \\ &= \left\{ \int_{-\infty}^{\infty} f(x)e^{-\sqrt{-1}\eta x} dx \right\} \left\{ \int_{-\infty}^{\infty} g(x)e^{-\sqrt{-1}\eta x} dx \right\} \end{aligned}$$

この関係は畳み込み定理 (convolution theorem) と呼ばれる. また,  $\mathcal{F}\{f(x)\}$  と  $\mathcal{F}\{g(x)\}$  の畳み込み積分と  $f(x)g(x)$  もフーリエ変換対をなす (周波数畳み込み定理 (frequency convolution theorem)). 通常, 畳み込み積分は, 畳み込む 2 つの関数のフーリエ変換を計算し, 2 つのフーリエ変換の積の逆フーリエ変換を行うことによって求める. 特に計算機を用いて計算する場合には, 効率がよい高速フーリエ変換アルゴリズムを利用できるのでこの方法は非常に有効である. なお, 畳み込み積分は次のように有限区間積分として定義される場合があるので注意が必要である.

$$\hat{p}(x) = \int_{-a}^a f(\xi)g(x-\xi)d\xi$$

特に Laplace 変換と対応づける場合には, 次のように定義する.

$$\tilde{p}(x) = \int_0^x x(\xi)h(x-\xi)d\xi$$

畳み込みには次のような性質がある.

- (1) 交換則 (commutative law) を満たす.

$$f(x) \times g(x) = g(x) \times f(x)$$

- (2) 結合則 (associative law) を満たす.

$$f(x) \times (g(x) \times h(x)) = (f(x) \times g(x)) \times h(x)$$

- (3) 分配則 (distributive law) を満たす.

$$f(x) \times (g(x) + h(x)) = f(x) \times g(x) + f(x) \times h(x)$$

理論ならびに実用的な応用面で重要なもう 1 つの積分として次式で定義される相関積分 (correlation integral) がある.

$$q(x) = \int_{-\infty}^{\infty} f(\xi)g(x+\xi)d\xi$$

$f(x)$  が実関数であり,  $f(x)$  と  $g(x)$  のフーリエ変換をそれぞれ  $\mathcal{F}\{f(x)\}$ ,  $\mathcal{F}\{g(x)\}$  とすると  $f(x)$  と  $g(x)$  の相関積分と  $\mathcal{F}\{f(x)\}^*\mathcal{F}\{g(x)\}$  はフーリエ変換対をなす (相関定理 (correlation theorem)). すなわち

$$\begin{aligned} & \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} f(\xi)g(x+\xi)d\xi \right\} e^{-\sqrt{-1}\eta x} dx \\ &= \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} g(x+\xi)e^{-\sqrt{-1}\eta(x+\xi)} dx \right\} f(\xi)e^{\sqrt{-1}\eta\xi} d\xi \\ &= \left\{ \int_{-\infty}^{\infty} f(x)^*e^{-\sqrt{-1}\eta x} dx \right\}^* \left\{ \int_{-\infty}^{\infty} g(x)e^{-\sqrt{-1}\eta x} dx \right\} \\ &= \left\{ \int_{-\infty}^{\infty} f(x)e^{-\sqrt{-1}\eta x} dx \right\}^* \left\{ \int_{-\infty}^{\infty} g(x)e^{-\sqrt{-1}\eta x} dx \right\} \end{aligned}$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。なお、相関積分は交換則を満たさないので、注意する必要がある。 $f(x)$  が偶関数であれば相関積分は畳み込み積分に一致する。

$$\begin{aligned} q(x) &= \int_{-\infty}^{\infty} f(\xi)g(x + \xi)d\xi \\ &= \int_{-\infty}^{\infty} f(-\eta)g(x - \eta)(-d\eta) \\ &= \int_{-\infty}^{\infty} f(\eta)g(x - \eta)d\eta \\ &= p(x) \end{aligned}$$

また、相関積分は  $f(x)$  と  $g(x)$  が同じ関数のとき特に自己相関関数 (autocorrelation function) と呼ばれ、異なるとき相互相関関数 (cross correlation function) と呼ばれる。

### 2.1.2.7 1次元離散畳み込みと1次元離散相関

連続フーリエ変換に対応して離散フーリエ変換を考えると同様連続畳み込みと連続相関のそれぞれに対応して離散畳み込みと離散相関を考えることができる。離散畳み込みは連続畳み込みを方形積分で近似したものとして次式で定義される。

$$p(k) = \Delta x \sum_{i=0}^{m-1} f(i)g(k-i) \quad (k = 0, \dots, m-1)$$

ここで  $\Delta x$  は標準化間隔である。また  $f(i)$ ,  $g(j)$ ,  $p(k)$  は整数値  $i, j, k$  に対してだけ値が定義される離散関数である。なお、計算機で計算することを前提としているので  $f(i)$  や  $g(j)$  に対応するもとの実数  $x$  の関数  $f(x)$  や  $g(x)$  は特定の有限区間でのみ非ゼロとなる関数または周期関数である必要がある。また、離散関数  $f(i)$ ,  $g(j)$  は任意の整数  $k$  に対して

$$f(i) = f(i + km), g(j) = g(j + km) \quad (i = 0, \dots, m-1)$$

を満たす周期  $m$  の周期関数とする。

なお、厳密には、上述の周期性を仮定しない離散畳み込みを線形畳み込み (linear convolution), 周期性を仮定する畳み込みを巡回畳み込み (circular convolution) と呼び区別する場合がある。以下特に断らない限り、離散畳み込みとして巡回畳み込みを考える。また、説明のために、ゼロ以外の値となり得る値のことを「有効値」と呼ぶことにする。いま、

$$f(i) = 0 \quad (i = n_1, \dots, m-1); \quad g(j) = 0 \quad (j = n_2, \dots, m-1)$$

すなわち、 $f(i)$  の1周期分のうち  $f(i)$  ( $i = 0, \dots, n_1 - 1$ ) の  $n_1$  個だけが有効値をとり、 $g(j)$  についても1周期分のうち  $g(j)$  ( $j = 0, \dots, n_2 - 1$ ) の  $n_2$  個だけが有効値をとるとすると、 $m \geq n_1 + n_2 - 1$  であれば  $p(k)$  は1周期分のうち  $p(k)$  ( $k = 0, \dots, n_1 + n_2 - 2$ ) の  $n_1 + n_2 - 1$  個が有効値をとる。したがって、 $m \geq n_1 + n_2 - 1$  ととれば、 $f(i)$  と  $g(j)$  に関して次の周期のデータとの重なりを起こさずに畳み込みを計算できる。すなわち、1周期分だけをみれば線形畳み込みと巡回畳み込みの結果は一致する。特に巡回畳み込みを考える場合には、離散フーリエ変換の場合と同様、その周期性により積分の方形近似は台形公式による近似と一致している。

たとえば、 $f(i)$  の有効値が  $\{f(0), f(1), f(2)\}$  の3個、 $g(j)$  の有効値が  $\{g(0), g(1)\}$  の2個とすると、対応する  $p(k)$  の有効値は

$$\begin{aligned} p(0) &= \Delta x ( f(0)g(0) ) \\ p(1) &= \Delta x ( f(0)g(1) + f(1)g(0) ) \\ p(2) &= \Delta x ( f(1)g(1) + f(2)g(0) ) \\ p(3) &= \Delta x ( f(2)g(1) ) \end{aligned}$$

の  $3 + 2 - 1 = 4$  個となる。ただし、連続畳み込みとの対応を考える場合には、 $m = n_1 + n_2 - 1$  とするよりも関数値  $p(n_1 + n_2 - 1) = 0$  を最後に追加して  $m = n_1 + n_2$  として考えた方がよい。このようにすれば、例えば、 $f(i)$  が区間  $[0, a]$  を  $n_1$  等分して標本化した標本値で  $g(j)$  が区間  $[0, b]$  を  $n_2$  等分して標本化した標本値であるとした場合に、 $p(k)$  が区間  $[0, a + b]$  を  $n_1 + n_2$  等分した場合の畳み込みの標本値に対応すると考えることができ、標本化間隔は  $f(i), g(j), p(k)$  で全て一致する。なお、 $f(i)$  と  $g(j)$  の有効値の数が不揃いな場合や、大きすぎる場合には、区分化 (sectioning) を行い、幾つかの区間に分けて離散畳み込みを求め加算するという手法がとられる。区分化には重複保持法 (overlap-save method) や重複加算法 (overlap-add method) があるが、本書のサブルーチンでは重複加算法を利用して畳み込みを計算する機能を提供している。

連続関数の場合と同様、離散関数の場合にもフーリエ変換と畳み込みの間には以下のような離散畳み込み定理が成立する。

$f(i)$  ( $i = 0, \dots, m$ ) と  $g(j)$  ( $j = 0, \dots, m$ ) の離散フーリエ変換をそれぞれ  $F(i)$  ( $i = 0, \dots, m$ ),  $G(j)$  ( $j = 0, \dots, m$ ) とすると、 $f(i)$  と  $g(j)$  との離散畳み込みと積  $F(j)G(j)$  ( $j = 0, \dots, m$ ) は (定数倍を除いて) フーリエ変換対をなす。すなわち、

$$\begin{aligned} p(k) &= \Delta x \sum_{i=0}^{m-1} f(i)g(k-i) \\ &= \Delta x \sum_{i=0}^{m-1} \left\{ \frac{1}{m} \sum_{j=0}^{m-1} (\alpha F_j) e^{2\pi\sqrt{-1}\frac{ji}{m}} \right\} \left\{ \frac{1}{m} \sum_{l=0}^{m-1} (\alpha G_l) e^{2\pi\sqrt{-1}\frac{l(k-i)}{m}} \right\} \\ &= \Delta x \frac{\alpha}{m^2} \sum_{j=0}^{m-1} \sum_{l=0}^{m-1} (\alpha F_j G_l) e^{2\pi\sqrt{-1}\frac{lk}{m}} \left( \sum_{i=0}^{m-1} e^{2\pi\sqrt{-1}\frac{(j-l)i}{m}} \right) \\ &= \Delta x \frac{\alpha}{m^2} \sum_{j=0}^{m-1} \sum_{l=0}^{m-1} (\alpha F_j G_l) e^{2\pi\sqrt{-1}\frac{lk}{m}} (m\delta_{j,l}) \\ &= \Delta x \frac{\alpha}{m} \sum_{j=0}^{m-1} (\alpha F_j G_j) e^{2\pi\sqrt{-1}\frac{jk}{m}} \end{aligned}$$

が成立する。なお、 $\delta_{i,j}$  はクロネッカーのデルタで以下のように定義される。

$$\delta_{i,j} = \begin{cases} 1 & (i = j) \\ 0 & (\text{その他}) \end{cases}$$

離散相関は連続相関を方形積分で近似したものとして次式で定義される。

$$q(k) = \Delta x \sum_{i=0}^{m-1} f(i)g(k+i) \quad (k = 0, \dots, m-1)$$

ここで  $\Delta x$  は標本化間隔である。また  $f(i), g(j), q(k)$  は整数値  $i, j, k$  に対してだけ値が定義される離散関数である。特に時系列を扱う場合、相関の定義として見掛け上異なる定義が採用されている場合があるので注意が必要である。例えば、標本数が  $n$  の 2 つの時系列  $x_i$  ( $i = 1, 2, \dots, n$ )  $y_i$  ( $i = 1, 2, \dots, n$ ) が与えられている場合、ラグ (lag)  $l$  ( $l = 0, 1, \dots, m-1; m \leq n$ ) の関数として、相互相関係数 (cross correlation coefficient)  $r_{xy}^{(l)}, r_{yx}^{(l)}$  は

$$\begin{aligned} r_{xy}^{(l)} &= \frac{c_{xy}^{(l)}}{\sqrt{u_x^{(l)}v_y^{(l)}}} \\ &= \frac{\sum_{i=1}^{n-l} (x_i - \mu_x^{(l)})(y_{i+l} - \nu_y^{(l)})}{\sqrt{\sum_{i=1}^{n-l} (x_i - \mu_x^{(l)})^2} \sqrt{\sum_{i=1}^{n-l} (y_{i+l} - \nu_y^{(l)})^2}} \end{aligned}$$

$$\begin{aligned}
 r_{yx}^{(l)} &= \frac{c_{yx}^{(l)}}{\sqrt{u_y^{(l)}v_x^{(l)}}} \\
 &= \frac{\sum_{i=1}^{n-l} (y_i - \mu_y^{(l)})(x_{i+l} - \nu_x^{(l)})}{\sqrt{\sum_{i=1}^{n-l} (y_i - \mu_y^{(l)})^2} \sqrt{\sum_{i=1}^{n-l} (x_{i+l} - \nu_x^{(l)})^2}}
 \end{aligned}$$

で定義される. ここで,  $\mu_x^{(l)}, \nu_x^{(l)}, \mu_y^{(l)}, \nu_y^{(l)}$  は  $x_i, y_i$  ( $i = 1, 2, \dots, n$ ) それぞれに対して, 前から  $n-l$  個のデータと後ろから  $n-l$  個のデータについての平均,

$$\mu_x^{(l)} = \frac{\sum_{i=1}^{n-l} x_i}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$\nu_x^{(l)} = \frac{\sum_{i=1}^{n-l} x_{i+l}}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$\mu_y^{(l)} = \frac{\sum_{i=1}^{n-l} y_i}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$\nu_y^{(l)} = \frac{\sum_{i=1}^{n-l} y_{i+l}}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

を表し,  $c_{xy}^{(l)}, c_{yx}^{(l)}$  はそれぞれ次式で定義される相互共分散 (cross covariance)

$$c_{xy}^{(l)} = \frac{\sum_{i=1}^{n-l} (x_i - \mu_x^{(l)})(y_{i+l} - \nu_y^{(l)})}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$c_{yx}^{(l)} = \frac{\sum_{i=1}^{n-l} (y_i - \mu_y^{(l)})(x_{i+l} - \nu_x^{(l)})}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$u_x^{(l)}, v_x^{(l)}, u_y^{(l)}, v_y^{(l)}$  はそれぞれ次式で定義される  $x_i, y_i$  ( $i = 1, 2, \dots, n$ ) それぞれに対して, 前から  $n-l$  個のデータと後ろから  $n-l$  個のデータについての分散 (variance) とする.

$$u_x^{(l)} = \frac{\sum_{i=1}^{n-l} (x_i - \mu_x^{(l)})^2}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$v_x^{(l)} = \frac{\sum_{i=1}^{n-l} (x_{i+l} - \nu_x^{(l)})^2}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$u_y^{(l)} = \frac{\sum_{i=1}^{n-l} (y_i - \mu_y^{(l)})^2}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

$$v_y^{(l)} = \frac{\sum_{i=1}^{n-l} (y_{i+l} - \nu_y^{(l)})^2}{(n-l)} \quad (l = 0, 1, \dots, m-1)$$

いま,

$$f(i) = \frac{x_{i+1} - \mu_x^{(l)}}{\sqrt{u_x^{(l)}}}$$

$$\hat{f}(i+l) = \frac{x_{i+l+1} - \nu_x^{(l)}}{\sqrt{v_x^{(l)}}}$$

$$g(i) = \frac{y_{i+1} - \mu_y^{(l)}}{\sqrt{u_y^{(l)}}}$$

$$\hat{g}(i+l) = \frac{y_{i+l+1} - \nu_y^{(l)}}{\sqrt{v_y^{(l)}}}$$

として  $x_i, y_i$  ( $i = 1, 2, \dots, n$ ) の前から  $n-l$  個のデータと後ろから  $n-l$  個のデータについての標準化した量  $f(i), g(i), \hat{f}(i+l), \hat{g}(i+l)$  を定義すれば

$$r_{xy}^{(l)} = \sum_{i=0}^{n-l-1} f(i)\hat{g}(i+l)$$

$$r_{yx}^{(l)} = \sum_{i=0}^{n-l-1} g(i)\hat{f}(i+l)$$

が成立する. したがって,  $r_{xy}^{(l)}$  と  $r_{yx}^{(l)}$  の定義は (定数倍を除いて) 離散相関  $q(k)$  の定義と一致する. なお, 今の場合,  $x_i$  と  $y_i$  とが同じ時系列であれば  $r_{xx}^{(l)}$  は自己相関係数 (autocorrelation coefficient),  $c_{xx}^{(l)}$  は自己共分散 (autocovariance coefficient) と呼ばれる.

また, 時系列の統計処理を考える場合には, 標本についての量と統計的な推定量とを区別するために以下の用語を用いる.

平均	→ 標本平均
分散	→ 標本分散
自己相関係数	→ 標本自己相関係数
自己共分散	→ 標本自己共分散
相互相関係数	→ 標本相互相関係数
相互共分散	→ 標本相互共分散

離散相関  $q(k)$  は, 計算機で計算することを前提としているので  $f(i)$  や  $g(j)$  に対応するもとの実数  $x$  の関数  $f(x)$  や  $g(x)$  は特定の有限区間でのみ非ゼロとなる関数または周期関数である必要がある. また, 離散関数  $f(i), g(j)$  は任意の整数  $k$  に対して

$$f(i) = f(i+km), g(i) = g(i+km) \quad (i = 0, \dots, m-1)$$

を満たす周期  $m$  の周期関数とする.

離散畳み込みの場合と同様,  $m \geq n_1 + n_2 - 1$  とすれば,  $f(i)$  と  $g(j)$  に関して次の周期のデータとの重なりを起こさずに相関を計算できる. たとえば,  $f(i)$  の有効値が  $\{f(0), f(1), f(2)\}$  の 3 個,  $g(j)$  の有効値が  $\{g(0), g(1)\}$  の 2 個とすると, 対応する  $q(k)$  の有効値は

$$q(-2)(= q(m-2)) = \Delta x \begin{pmatrix} f(2)g(0) \end{pmatrix}$$

$$q(-1)(= q(m-1)) = \Delta x \begin{pmatrix} f(1)g(0) + f(2)g(1) \end{pmatrix}$$

$$q(0) = \Delta x \begin{pmatrix} f(0)g(0) + f(1)g(1) \end{pmatrix}$$

$$q(1) = \Delta x \begin{pmatrix} f(0)g(1) \end{pmatrix}$$

の  $3+2-1 = 4$  個となる. ただし, 連続相関との対応を考える場合には,  $m = n_1 + n_2 - 1$  とするよりも関数値  $q(-n_1) = 0$  を最初に追加して  $m = n_1 + n_2$  とした方がよい. このようにすれば, 例えば,  $f(i)$  が区間  $[0, a]$  を  $n_1$  等分して標本化した標本値で  $g(j)$  が区間  $[0, b]$  を  $n_2$  等分して標本化した標本値であるとした場合に,  $q(k)$  が区間  $[-a, b]$  を  $n_1 + n_2$  等分した場合の相関の標本値に対応すると考えることができ, 標本化間隔は  $f(i), g(j), q(k)$  で全て

一致する。なお、 $f(i)$  と  $g(j)$  の有効値の数が不揃いな場合や、大きすぎる場合には、離散畳み込みの場合と同様重複加算法 (overlap-add method) を利用して区分化を行って計算することができる。連続関数の場合と同様、離散関数の場合にもフーリエ変換と相関との間には以下のような離散相関定理が成立する。

$f(i)$  が実離散関数であるとし、 $f(i)$  と  $g(j)$  の離散フーリエ変換をそれぞれ  $F(i)$  ( $i = 0, \dots, m$ ),  $G(j)$  ( $j = 0, \dots, m$ ) とすると、 $f(i)$  と  $g(j)$  との離散相関と積  $F(j)^*G(j)$  ( $j = 0, \dots, m$ ) はフーリエ変換対をなす。すなわち、

$$\begin{aligned}
 q(k) &= \Delta x \sum_{i=0}^{m-1} f(i)g(k+i) \\
 &= \Delta x \sum_{i=0}^{m-1} \left\{ \frac{1}{m} \sum_{j=0}^{m-1} (\alpha F(j))^* e^{-2\pi\sqrt{-1}\frac{ji}{m}} \right\}^* + \left\{ \frac{1}{m} \sum_{l=0}^{m-1} (\alpha G(l)) e^{2\pi\sqrt{-1}\frac{l(k+i)}{m}} \right\} \\
 &= \Delta x \sum_{i=0}^{m-1} \left\{ \frac{1}{m} \sum_{j=0}^{m-1} (\alpha F(j))^* e^{-2\pi\sqrt{-1}\frac{ji}{m}} \right\} + \left\{ \frac{1}{m} \sum_{l=0}^{m-1} (\alpha G(l)) e^{2\pi\sqrt{-1}\frac{l(k+i)}{m}} \right\} \\
 &= \Delta x \frac{\alpha}{m^2} \sum_{j=0}^{m-1} \sum_{l=0}^{m-1} (\alpha F(j)^* G(l)) e^{2\pi\sqrt{-1}\frac{lk}{m}} \left( \sum_{i=0}^{m-1} e^{2\pi\sqrt{-1}\frac{(l-j)i}{m}} \right) \\
 &= \Delta x \frac{\alpha}{m^2} \sum_{j=0}^{m-1} \sum_{l=0}^{m-1} (\alpha F(j)^* G(l)) e^{2\pi\sqrt{-1}\frac{lk}{m}} (m\delta_{j,l}) \\
 &= \Delta x \frac{\alpha}{m} \sum_{j=0}^{m-1} (\alpha F(j)^* G(j)) e^{2\pi\sqrt{-1}\frac{jk}{m}}
 \end{aligned}$$

が成立する。なお、 $\delta_{i,j}$  はクロネッカーのデルタで以下のように定義される。

$$\delta_{i,j} = \begin{cases} 1 & (i = j) \\ 0 & (\text{その他}) \end{cases}$$

$q(k)$  の有効値は  $q(k)$  ( $k = 0, \dots, n_2 - 1$ ) および  $q(-k) = q(m - k)$  ( $k = 1, \dots, n_1 - 1$ ) で与えられるが、このままでは区分化等を行う場合に不便であるので  $n_1 - 1$  だけシフトした次式で定義される  $\hat{q}(k)$  を計算する方が良い。

$$\hat{q}(k) = q(k - (n_1 - 1)) \quad (k = 0, \dots, n_1 + n_2 - 2)$$

$\hat{q}(k)$  を計算する場合、 $q(k)$  をシフトする代わりに  $g(k)$  をシフトする。いま

$$\begin{aligned}
 \hat{q}(k) &= q(k - (n_1 - 1)) \\
 &= \Delta x \frac{\alpha}{m} \sum_{j=0}^{m-1} (\alpha F(j)^* G(j)) e^{2\pi\sqrt{-1}\frac{j(k-(n_1-1))}{m}} \\
 &= \Delta x \frac{\alpha}{m} \sum_{j=0}^{m-1} (\alpha F(j)^* G(j)) e^{-2\pi\sqrt{-1}\frac{j(n_1-1)}{m}} e^{2\pi\sqrt{-1}\frac{jk}{m}} \\
 &= \Delta x \frac{\alpha}{m} \sum_{j=0}^{m-1} (\alpha F(j)^* \hat{G}(j)) e^{2\pi\sqrt{-1}\frac{jk}{m}}
 \end{aligned}$$

であり、

$$\begin{aligned}
 \hat{G}(j) &= G(j) e^{-2\pi\sqrt{-1}\frac{j(n_1-1)}{m}} \\
 &= \frac{1}{\alpha} \sum_{k=0}^{m-1} g(k) e^{-2\pi\sqrt{-1}\frac{j(k+(n_1-1))}{m}} \\
 &= \frac{1}{\alpha} \sum_{k=0}^{m-1} g(k - (n_1 - 1)) e^{-2\pi\sqrt{-1}\frac{jk}{m}}
 \end{aligned}$$

である。したがって、 $g(j)$  の方をあらかじめ以下のように定義される  $\hat{g}(j)$  となるようにシフトしておけば、直接  $\hat{q}(k)$  を得ることができる。

$$\hat{g}(j + n_1 - 1) = g(j) \quad (j = 0, \dots, n_2 - 1)$$

### 2.1.2.8 多次元 (連続) 畳み込みと多次元 (連続) 相関

畳み込み積分と相関積分は多次元の場合に拡張することができる。例えば、3次元の場合には  $f(x, y, z)$  と  $g(x, y, z)$  との畳み込みと相関は3重積分として以下のように定義される。

畳み込み:

$$p(x, y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta, \zeta) g(x - \xi, y - \eta, z - \zeta) d\xi d\eta d\zeta$$

相関:

$$q(x, y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta, \zeta) g(x + \xi, y + \eta, z + \zeta) d\xi d\eta d\zeta$$

また1次元の場合と同様、次のような畳み込み定理ならびに相関定理が成立する。

$$\mathcal{F}\{p(x, y, z)\} = \mathcal{F}\{f(x, y, z)\} \mathcal{F}\{g(x, y, z)\}$$

$$\mathcal{F}\{q(x, y, z)\} = \mathcal{F}\{f(x, y, z)\}^* \mathcal{F}\{g(x, y, z)\} \quad (f(x, y, z) \in \mathbf{R})$$

ここで  $\mathcal{F}\{f\}$  は  $f$  のフーリエ変換を表す。なお、相関積分において  $f(x, y, z)$  が複素関数の場合は、 $\mathcal{F}\{f(x, y, z)\}^*$  の代わりに  $f(x, y, z)$  の負の周波数のフーリエ変換を用いれば良い。

### 2.1.2.9 パワー・スペクトル

$f(x)$  フーリエ積分  $F(\xi)$

$$F(\xi) = a \int_{-\infty}^{\infty} f(x) e^{-\sqrt{-1}\xi x} dx$$

に対して  $P(\xi) = |F(\xi)|^2$  で定義される量を  $f(x)$  のパワー・スペクトル (密度関数) と呼ぶ。通常、パワー・スペクトルはパーセバルの定理 (Parseval's Theorem)

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \frac{1}{2\pi a} \int_{-\infty}^{\infty} P(\xi) d\xi$$

か成り立つように正規化する。連続関数  $f(x)$  のパワー・スペクトルに対応する量として  $n$  を周期とする離散関数  $c(k)$  ( $k = 0, 1, \dots, n-1$ ) の場合には生のピリオドグラム (raw periodogram)  $p(j)$  を考える。ピリオドグラム  $p(j)$  は  $c(i)$  の離散フーリエ変換  $C(j)$

$$C(j) = \frac{1}{\alpha} \sum_{k=0}^{n-1} c(k) e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1)$$

を用いて

$$p(j) = \beta |C(j)|^2$$

と定義される。 $\beta$  は単位系の選び方等で決まる適当な定数である。本書のサブルーチンでは通常  $\alpha = 1$ ;  $\beta = \frac{1}{n^2}$  が選ばれる。このときパーセバルの定理による、時間 (または空間) 領域の対応する全パワーは  $\frac{\sum_{k=0}^{n-1} \{c(k)\}^2}{n}$  になる。なお、 $c(k)$  が時系列の場合には、離散フーリエ変換 (フーリエ・スペクトルともいう) の場合と同様両側パワー・スペクトル (two-sided power spectrum) と片側パワー・スペクトル (one-sided power spectrum) を考えることができる。両側パワー・スペクトルでは連続フーリエ変換で、通常、関数の変域を原点に対して左右対称に取ることに対応して

$$\{\tilde{p}(j)\}_{j=-(n-m-2), \dots, -1, 0, 1, \dots, m} = \{p(m+1), p(m+2), \dots, p(n-1), p(0), p(1), \dots, p(m)\}$$

(ただし,  $m = \lfloor \frac{n}{2} \rfloor$  であり,  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す) として半周期分のデータを 1 周期ずらして考える. このとき  $p(0)$  がゼロ周波数に対応する要素である. 各ピリオドグラム  $\tilde{p}(j)$  に対応する周波数は  $\frac{j}{nT}$  ( $j = -(n-m-2), \dots, -1, 0, 1, \dots, m$ ) である. ただし,  $T$  は標本化間隔. 片側パワー・スペクトルでは負の周波数を避けるために,

$$\{\tilde{p}(j)\}_{j=0,1,\dots,m} = \begin{cases} \{p(0), 2p(1), \dots, 2p(m-1), p(m)\} & n:\text{偶数} \\ \{p(0), 2p(1), \dots, 2p(m)\} & n:\text{奇数} \end{cases}$$

として考える. 各ピリオドグラム  $\tilde{p}(j)$  に対応する周波数は  $\frac{j}{nT}$  ( $j = 0, 1, \dots, m$ ) である. なお, 離散フーリエ変換の周波数領域における標本点の周波数の間隔  $\frac{1}{nT}$  を分解度 (resolution) と呼ぶこともある.

離散フーリエ変換はフーリエ級数に対する方形近似 (台形公式による近似でもある) であるので, 近似精度を上げるためにはデータ数  $n$  を大きくとる必要がある. 一方, 前にも述べたように, 周期関数のフーリエ級数の値と周期関数をその 1 周期で打ち切った関数の連続フーリエ変換とは定数倍を除いて一致するので, ピリオドグラムはデータ数  $n$  が十分大きければこのような連続関数に対するパワー・スペクトルのよい近似を与えると期待できる. ただし, パワー・スペクトルの推定に用いる系列が反映するもとの関数は, 通常, 周期関数ではなく, また周期関数であってもその 1 周期でちょうど打ち切られているという状態にはない. 生のピリオドグラムはその定義から自己相関関数の離散フーリエ変換近似とみなせる.

図 2-4 に離散データ  $u_i = \cos(0.62\pi i) + \cos(0.14\pi i)$  ( $i = 0, 1, \dots, n-1$ ;  $n = 50$ ) についての生のピリオドグラムと自己相関関数の離散フーリエ変換との計算結果を示す. ここで, 自己相関関数を計算する場合の周期は  $2n$  とし,  $u_{n+1} = \dots = u_{2n-1} = 0$  としている. 図には標本化間隔  $T = \Delta t = 0.5[\text{sec}]$  とした場合の分解度  $\Delta f$  の値も参考に示している. この場合, 周波数  $0.14[\text{Hz}]$  と周波数  $0.62[\text{Hz}]$  に相当する部分にパワーが集中していることが分かり, 期待通りの結果を与える. なお, 自己相関関数の離散フーリエ変換の方がピリオドグラムの場合よりも分解度が小さいのでより望ましい形状をしているが, 対応する計算点数は 2 倍になっていることに注意する必要がある. また, 自己相関関数の離散フーリエ変換は実数となることに注意する (より一般的な相互相関関数の離散フーリエ変換は通常複素数である).



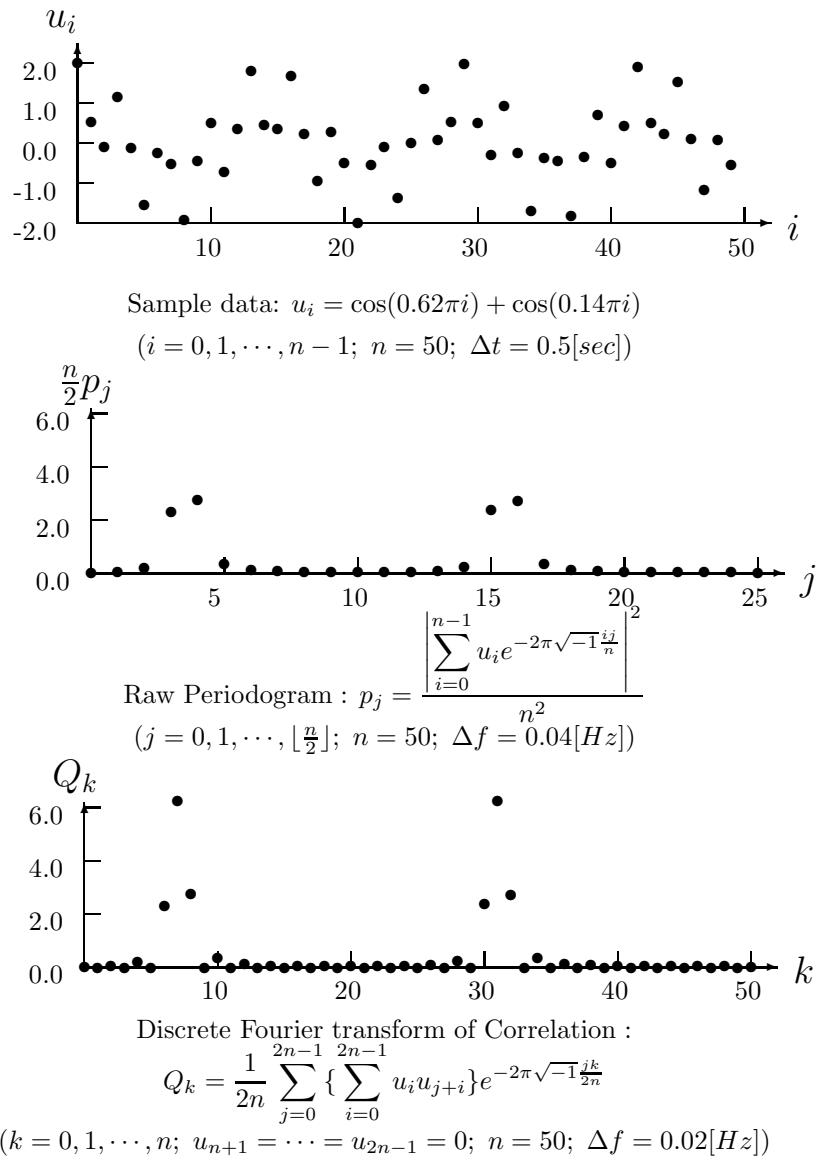


図 2-4 ピリオドグラムと自己相関関数のフーリエ変換

有効データ数  $n$  の離散関数の自己相関関数の有効データ長は  $2n - 1$  であるので、一般の関数のパワー・スペクトルを生みのピリオドグラムで近似することは、1つの周期が以下のように与えられる方形打ち切り関数  $w(k)$  で関数を打ち切ったことに相当する。

$$w(k) = \begin{cases} 1 & k = 0, 1, \dots, n - 1 \\ 0 & \text{それ以外} \end{cases}$$

方形関数のフーリエ変換は周波数を  $f$  とした場合、 $\frac{\sin f}{f}$  型の関数形をしており、中心周波数の周りに小さくないサイドローブを持っている。したがって、たとえば、周期関数を 1 周期の整数倍でない幅で単純に打ち切って標本化した場合、周波数領域では、生みのピリオドグラムはパワー・スペクトルを求めたい周期関数のフーリエ変換と  $\frac{\sin f}{f}$  型関数との畳み込みとなるので、漏れ (leakage) と呼ばれる余分な周波数成分が発生する。このような漏れを抑止するためには、単純な打ち切りを行わずに周波数領域でのサイドローブが小さい打ち切り関数を用いる。これにともない、ピリオドグラムの定義を修正した以下のような修正ピリオドグラム (modified periodogram)  $\hat{p}$  が通常用いられる。

$$\hat{p}(j) = \beta |\hat{C}(j)|^2$$

ここで  $\hat{C}(j)$  はもとの系列  $c(k)$  に打ち切り関数  $w(k)$  を乗じた値の離散フーリエ変換であり次式で定義される。

$$\hat{C}(j) = \frac{1}{\alpha} \sum_{k=0}^{n-1} w(k)c(k)e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1)$$

$w(k)$  は窓関数 (data window) と呼ばれる。修正ピリオドグラムは、 $w(k)$  のフーリエ変換  $W(j)$  を用いて周波数領域の畳み込みとして定義される場合もある。この場合、 $W(j)$  はスペクトル・ウィンドウ (spectral window) と呼ばれることもある。これに対応して  $w(k)$  はラグ・ウィンドウ (lag window) と呼ばれる。窓関数は本来、上述のように打ち切りによる周波数領域でのサイドローブ抑止のため提案されたが、数学的には、窓関数の効果は周波数領域における平滑化式と同じ形式となるので、窓関数を適当に選べば、パワー・スペクトルの平滑化を行うこともできる。また、時系列のスペクトル解析を行う場合よく、平均値 0 の時系列データを考えるが、平均値はフーリエ変換のゼロ周波数成分に対応するので、変換後ゼロ周波数成分をカットすれば、同様な効果が期待できる。ただし、修正ピリオドグラムを用いる場合には、窓関数を乗じることにより平均値が変動するので、前もって平均値を 0 としておく意味合いはない。なお、打ち切り関数  $w(k)$  を乗じることによってピリオドグラムの定義に定数倍の差が生じる。本来は、時間 (または空間) 領域での全パワーを計算し、これが周波数領域の全パワーと一致するように補正すること (パーセバルの定理による) が望ましいが、このような補正を行うための計算コストは小さくない。また、時間 (または空間) 領域での全パワーの推定が困難な場合もある。打ち切り関数によるパワーを補正するため次式で定義される  $\tilde{p}(j)$  を用いる場合がある。

$$\tilde{p}(j) = \frac{\hat{p}(j)}{n \sum_{k=0}^{n-1} \{w(k)\}^2}$$

この場合、用いる打ち切り関数によっては、2乗和を解析的に計算できるので、パワーの補正に要するコストはそれほど大きくない。対応する系列の全パワーは、方形打ち切りを用いた場合の一般化として  $\frac{\sum_{k=0}^{n-1} \{c(k)\}^2}{n}$  とする。

なお、スペクトル・ウィンドウ  $W(j)$  のサイドローブを抑止すればする程  $W(j)$  のスペクトル幅が増加するので得られるパワー・スペクトルの推定波形もぼやけたものになる。したがって、パワー・スペクトルを推定する場合、目的に応じて、すなわち、スペクトル幅を問題としているの中心周波数を問題としているのか等に応じて、適切な打ち切り関数を選択する必要がある。代表的な打ち切り関数としては以下のものがある。

$$w_j = \begin{cases} \begin{cases} \sin^2(\pi u_j) & \text{(Hanning ウィンドウ)} \\ 1 - |2u_j - 1| & \text{(Bartlett ウィンドウ)} \\ 1 - (2u_j - 1)^2 & \text{(Welch ウィンドウ)} \end{cases} \\ \begin{cases} \begin{cases} 16u_j^3 & 0 \leq u_j < \frac{1}{4} \\ 1 - 6u_j(u_j - 1)^2 & \frac{1}{4} \leq u_j \leq \frac{1}{2} \\ 1 - 6u_j(u_{n-j+1} - 1)^2 & \frac{1}{2} \leq u_j \leq \frac{3}{4} \\ 16u_{n-j+1}^3 & \frac{3}{4} \leq u_j < 1 \end{cases} \end{cases} \end{cases} \quad \text{(Parzen ウィンドウ)}$$

ただし、 $u_j = \frac{j}{n}$ 。このように選ぶと、 $w_0 = 0$  となるので、 $c_0$  に対応する成分が無効となるため、窓関数を多少変形して定義する場合もある。なお、窓関数は  $|t| \leq 1$  でのみ非ゼロとなる時間領域関数としてそれぞれ次のように表される。

$$w(t) = \begin{cases} \begin{cases} \frac{1 + \cos \pi t}{2} = \cos^2 \frac{\pi t}{2} & \text{(Hanning ウィンドウ)} \\ 1 - |t| & \text{(Bartlett ウィンドウ)} \\ 1 - t^2 & \text{(Welch ウィンドウ)} \end{cases} \\ \begin{cases} \begin{cases} 1 - 6t^2 + 6|t|^3 & |t| \leq \frac{1}{2} \\ 2(1 - |t|)^3 & \frac{1}{2} \leq |t| \leq 1 \end{cases} \end{cases} \end{cases} \quad \text{(Parzen ウィンドウ)}$$

図 2-5 にこれらの窓関数 ( $w_i$ ) とその離散フーリエ変換 ( $W_j$ ) の大きさ ( $|W_j|$ ) のグラフを示す。

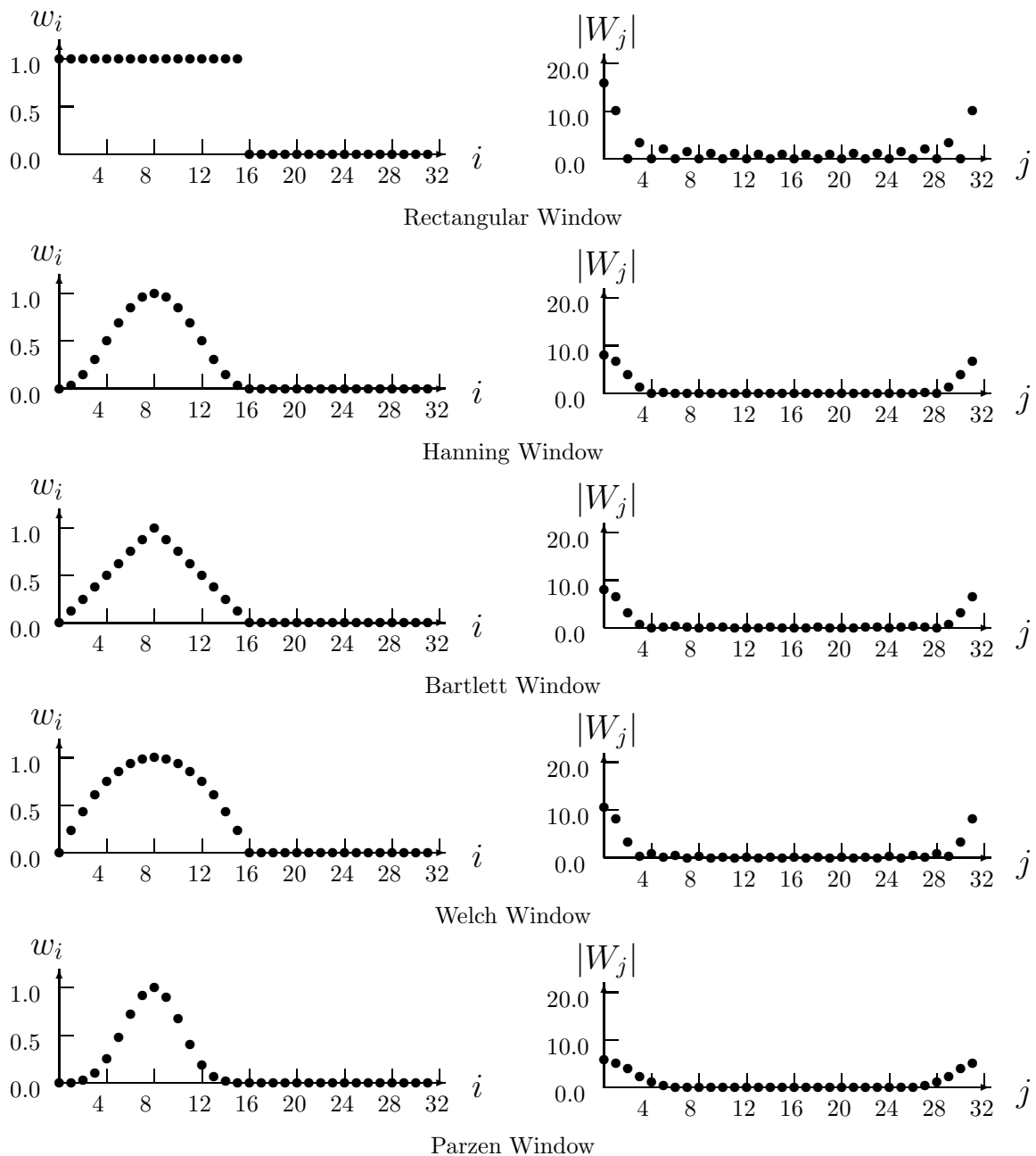


図 2-5 窓関数とその離散フーリエ変換の絶対値

離散フーリエ変換の分解度  $\frac{1}{nT}$  を上げるには標本のデータ数  $n$  を増やすか標本化間隔  $T$  を増やせば良いが、標本化間隔と分解度を一定に保った状態でパワー・スペクトルの推定値の精度を上げるために、データ数  $n$  の標本を  $m$  組として  $m$  組それぞれについて修正ピリオドグラムを求めてその平均をとるという手法がよく取られる。この場合、 $m$  組の標本データをもとの系列からオーバーラップして取るというような手法も提案されている。詳細は参考文献等を参照されたい。また、パワー・スペクトルを求める場合、フーリエ変換の周波数推移に関する性質すなわち時間(または空間)領域で  $e^{2\pi\sqrt{-1}f_0t}$  を掛けることは周波数領域では周波数を  $f_0$  だけシフトすることに対応し、関数の形状は変わらないという性質を利用して、パワー・スペクトルの中心周波数をあらかじめシフトして計算することで計算に必要なデータ点数を削減するという手法も良く用いられる。なお、このような操作は変調 (modulation) として知られている。

2.1.2.10 ラプラス変換

(1) ラプラス変換

関数  $f(t)$  が与えられたとき、ラプラス変換は複素数  $s$  をパラメータとして

$$F(s) = \int_0^{\infty} f(t)e^{-st} dt \tag{2.1}$$

で定義される。このとき、逆変換はプロムウィッチ積分として知られる

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} F(s)e^{st} ds \tag{2.2}$$

ただし、 $\gamma > \gamma_0$ ,  $\gamma_0$ :収束座標,  $i = \sqrt{-1}$

で与えられる。 $f(t)$  を原関数,  $F(s)$  を像関数といい、

$$\text{正変換 } F(s) = \mathcal{L}\{f(t)\} \tag{2.3}$$

$$\text{逆変換 } f(t) = \mathcal{L}^{-1}\{F(s)\} \tag{2.4}$$

と書く。

(a) 像関数  $F(s)$  が以下の条件を満たす場合

(I)  $\Re(s) > 0$  で正則

(II)  $\Re(s) > 0$  で  $\lim_{s \rightarrow \infty} F(s) = 0$

(III)  $\Re(s) > 0$  で  $F(s^*) = F^*(s)$

( $\Re(s)$ :  $s$  の実数部; \*:共役複素記号)

(i) 指数関数の近似

式 (2.2) は指数関数  $e^s$  を  $\Re(s) > 0$  にのみ極をもつ有理関数で近似できれば、この極の周りの積分に帰着できる。そこで、指数関数の近似法として次の関数式を考える。

$$E_{ec}(s, a) = \frac{e^a}{2 \cosh(a-s)} \quad (a > 0) \tag{2.5}$$

この式は、次のように 2 通りに書き換えられる。

$$E_{ec}(s, a) = \frac{e^a}{2} \sum_{n=-\infty}^{\infty} \frac{(-1)^n}{s - \{a + i(n-0.5)\pi\}} \tag{2.6}$$

$$E_{ec}(s, a) = e^s - e^{-2a}e^{3s} + e^{-4a}e^{5s} - \dots \tag{2.7}$$

式 (2.7) から、 $a \gg \Re(s)$  のとき、 $E_{ec}(s, a)$  は指数関数のよい近似であることがわかり、式 (2.6) から、 $E_{ec}(s, a)$  の極はすべて 1 位で、 $s = a$  という直線上に等間隔でならび、留数は  $\frac{(-1)^n e^a}{2}$  であることがわかる。

プロムウィッチ積分中の  $e^{st}$  の代わりに、 $E_{ec}(st)$  を代入した関数

$$f_{ec}(t, a) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} F(s)E_{ec}(st, a)ds \quad (0 < \gamma < a) \tag{2.8}$$

を定義し、式 (2.6), (2.7) を使って書き直すと

$$f_{ec}(t, a) = f(t) - e^{-2a}f(3t) + e^{-4a}f(5t) - \dots \tag{2.9}$$

$$f_{ec}(t, a) = \frac{e^a}{t} \sum_{n=1}^{\infty} (-1)^n \Im \left\{ F \left( \frac{a}{t} + i \frac{(n-0.5)\pi}{t} \right) \right\} \tag{2.10}$$

( $\Im(s)$ :  $s$  の虚数部)

(ii) オイラー変換

式 (2.10) のような無限級数の和を数値的に計算する場合は、適当な項数  $N$  で打ち切らなくてはならない。ここでは、Euler 変換を利用して効率的に計算を行う。Euler 変換は、

$$\sum_{n=0}^{\infty} a_n = \sum_{p=0}^{\infty} 2^{-(p+1)} \Delta^p a_0 \quad (2.11)$$

(ただし、 $\Delta a_0 = a_0 + a_1, \Delta^2 a_0 = \Delta a_0 + \Delta a_1, \dots, \Delta^n a_0 = \Delta^{n-1} a_0 + \Delta^{n-1} a_1$ )

により、左辺の級数を右辺の級数に変換する方法である。 $\Delta^n a_k$  は、第  $n$  段差であり、次式で定義されている。

$$\Delta^n a_k = a_k - \binom{n}{1} a_{k+1} + \binom{n}{2} a_{k+2} - \dots \pm \binom{n}{n} a_{k+n} \quad (2.12)$$

オイラー変換によって変換された級数は、次の条件が成り立つ時、元の級数よりも速く収束することが知られている。

(A)  $\sum_{n=0}^{\infty} a_n$  が交代級数で、数列  $\{|a_n|\}$  が完全に単調に 0 に収束する。

(B)  $\frac{1}{2} < \left| \frac{a_{n+1}}{a_n} \right| \leq 1$

オイラー変換の効果は、 $\left| \frac{a_{n+1}}{a_n} \right|$  が 1 に近いほど大きいので、式 (2.11) の初めの  $k$  項は普通に計算し、 $k+1$  項以降をオイラー変換すると

$$\sum_{n=0}^{p-1} 2^{-(n+1)} \Delta^n a_0 = 2^{-p} \sum_{q=1}^p A_{pq} a_{q-1} \quad (2.13)$$

$$\text{ただし, } A_{pp} = 1, A_{p,q-1} = A_{pq} + \binom{p+1}{q}$$

となる。よって  $f(t)$  の近似値は次式から計算する。

$$f_{ec}^{kp} = \frac{e^a}{t} \left( \sum_{n=0}^{k-1} F_n + 2^{-p} \sum_{q=0}^{p-1} A_{pq} F_{k+q} \right) \quad (2.14)$$

$$F_n = (-1)^{n+1} \Im \left\{ F \left( \frac{a}{t} + i \frac{(n+0.5)\pi}{t} \right) \right\}$$

また、実際の計算では、式 (2.11) の右辺を  $p$  項で打ち切る。この時の打ち切り誤差は、

$$R_p = \frac{1}{2^p} [\Delta^p a_0 + \Delta^p a_1 + \Delta^p a_1 + \dots] \quad (2.15)$$

で与えられるが、さらに次の条件

- $a_n = f(n)$  と書いて、 $f(x)$  の  $p$  階微係数  $f^{(n)}(x)$  が  $x$  の正值に対して定符号で  $x$  の増加とともに単調に減少する。

が成り立つと、

$$|R_p(0)| < \frac{1}{2^p} |\Delta^p v_0| \quad (2.16)$$

となることが証明されている。

[注意事項]  $F(s)$  が  $e^{-sx}$  の形の因数を持つ場合、 $F_n$  が条件 (1(a)iiA) を満たさないため、精度が落ちることがある。このとき像関数は、次の性質を持っている。

- $t < t_0$  において  $f(t) = 0$
- $t = t_0$  のとき  $f(t)$  またはその導関数が不連続となる

$t < t_0$  において  $f(t) = 0$  とわかっている場合は、 $t$  軸を  $t_0$  だけずらして

$$g(t') = f(t' + t_0) \quad (2.17)$$

の像関数

$$G(s) = \exp(t_0 s) \cdot F(s) \quad (2.18)$$

を扱おうと良い。

(b)  $\Re(s) > 0$  に特異点を持つ場合

$F(s)$  の収束座標  $\alpha$  がわかっているとき,

$$G(s) = F(s + b), b > \alpha \quad (2.19)$$

とすると,  $G(s)$  は  $\Re(s) > 0$  で正則である. さらに,  $F(t), G(t)$  の原関数をそれぞれ  $f(t), g(t)$  とすると

$$f(t) = e^{bt}g(t) \quad (2.20)$$

の関係がある. したがって,  $\Re(s) > 0$  の場合は, 以下の式を使って計算する.

$$\begin{aligned} f_{ec}^{kp}(t, a) &= e^{bt} \left( \frac{e^a}{t} \right) \sum F_n \\ F_n &= (-1)^n \Im \left( F \left[ \frac{a}{t} + b + i \frac{(n-0.5\pi)}{t} \right] \right) \end{aligned} \quad (2.21)$$

[収束座標の判定]

- $F(s)$  が有理関数の場合

実係数多項式  $Q(s), P(s)$  を用いて

$$F(s) = \frac{Q(s)}{P(s)} = \frac{b_1 s^m + b_2 s^{m-1} + \dots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \dots + a_{n+1}} \quad (2.22)$$

と表せる. 分母多項式  $P(s) = 0$  の根を  $s_1, s_2, \dots, s_n$  とすると収束座標は, 根の実数部の最大値

$$\alpha = \max \Re[s_k] \quad (2.23)$$

で与えられる. つまり,  $P(s) = 0$  の全根の実数部が負であれば, 収束座標  $\alpha < 0$  となり,  $\Re(s) > 0$  で正則である. そこで, 次の定理を使って収束座標の符号を判定する. 多項式  $P(s)$  の全根が負であるための必要十分条件は, 偶数項の和と奇数項の和の比を連分数に展開し,

$$\frac{a_1 s^n + a_3 s^{n-2} + \dots}{a_2 s^{n-1} + a_4 s^{n-3} + \dots} = h_1 s + \frac{1}{h_2 s + \frac{1}{h_3 s + \frac{1}{h_4 s + \dots}}} \quad (2.24)$$

としたとき, 係数  $h_1, h_2, \dots$ , がすべて正値となることである. この定理を満たす多項式は Hurwitz 多項式と呼ばれている. 連分数の係数  $h_1, h_2, \dots$ , は, Euclid の互除法を用いて計算する.  $F(s)$  が  $\Re(s) > 0$  に特異点を持つ場合, 上記の判定法を適当な正数  $b$  に対して多項式  $P(s + b)$  を作り, Hurwitz 多項式か否かと判定する.  $b$  を増加させながら繰り返し判定を行って  $F(s + b)$  が正則となるような  $b$  の収束座標を求める.

- $F(s)$  が一般関数の場合

$F(s)$  が非有理関数の場合,  $\Re(s) > 0$  で  $F(s)$  が正則かどうかを判定する有効な方法はないので収束座標は利用者が指定しなければならない.

(c)  $F(s^*) \neq F^*(s)$  の場合

$x$  を実数として

$$F_1(x) = 0.5[F(x) + F^*(x)] \quad (2.25)$$

$$F_2(x) = 0.5i[F(x) - F^*(x)] \quad (2.26)$$

とし, 次に  $x$  を  $s$  に書き換えて  $F_1(s)$  と  $F_2(s)$  は条件 (1(a)i)~(1(a)iii) を満足する. そこで, これを逆変換して  $f_1(t), f_2(t)$  を求めると

$$f(t) = f_1(t) - if_2(t) \quad (2.27)$$

が  $\mathcal{L}^{-1}F(s)$  となる.

(d) パラメータの値の決定

(i) 級数和の打ち切り項数  $k + p$

オイラー変換が有効で、打ち切り誤差の評価式 (2.16) が使えるためには、少なくとも  $|F_n|$  が単調減少しなければならない。一般に、 $|F_n| = |\Im(F[\frac{a}{t} + i\frac{(n+0.5)\pi}{t}])|$  は、 $n$  とともに複雑に変化する。 $F(s)$  の特異点の虚数部の最大値を  $\omega_m$  とするとき、 $\frac{(n+0.5)\pi}{t} > \omega_m$  ならば  $|F_n|$  は単調に減少する。したがって、式 (2.16) での普通和の項数  $k$  は、 $k > 0.5\frac{\omega_m}{\pi}t$  のように  $t$  に比例して増加させなければならない。実際には、 $a$  の値にも関係するので

$$k = k_1 + k_2 t \tag{2.28}$$

として、打ち切り誤差 (2.16) が希望の値となるように  $k_1, k_2$  を決める。 $t_1 \leq t \leq t_2$  の範囲で  $f(t)$  を求めたい場合、 $k_1, k_2$  を決める一つの方法は、まず、 $t = t_1$  として打ち切り誤差が無視できる  $k(t_1)$  を決め、次に、 $t = t_2$  として同様に打ち切り誤差が無視できる  $k(t_2)$  を決める。

$$\begin{cases} k_1 + k_2 t_1 = k(t_1) \\ k_1 + k_2 t_2 = k(t_2) \end{cases} \tag{2.29}$$

より  $k_1, k_2$  を決める。

オイラー変換の次数  $p$  の値は、計算精度とだいたい比例する傾向にあるので要求する計算精度が  $10^{-d}$  ならば  $p = d$  を与えるとよい。(参考文献 (10) 参照)

(ii) 指数関数を  $E(s, a)$  で近似するときのパラメータ  $a$

$$|f(t) - f_{ec}(t, a)| = |e^{-2a} f(3t) - e^{-4a} f(5t) + \dots| \tag{2.30}$$

をうるので、 $f_{ec}(t, a)$  と共に  $f_{ec}(3t, a), f_{ec}(5t, a)$  を計算すると

$$|f(t) - f_{ec}(t, a)| \simeq |e^{-2a} f(3t) - e^{-4a} f(5t) + \dots| \tag{2.31}$$

と評価できる。実際には、 $f(t)$  と  $f(3t)$  を同じオーダーであるとして、 $e^{-2a}$  を相対誤差と考えると下記のように相対誤差の指数部はほぼ  $a$  の値に等しい。

a	3	4	5	6
$e^{-2a}$	$2.4 \times 10^{-3}$	$3.4 \times 10^{-4}$	$4.5 \times 10^{-5}$	$6.2 \times 10^{-6}$

2.1.2.11 ウェーブレット変換

(1) Haar 関数

ウェーブレット変換対象となる入力データの定義域を  $[0, a]$  とする。Haar 関数  $H_{00}(x), H_{01}(x)$  を、

$$H_{00}(x) = 1/\sqrt{a}, 0 \leq x \leq a$$

$$H_{01}(x) = \begin{cases} 1/\sqrt{a} & \text{if } x \leq a/2 \\ -1/\sqrt{a} & \text{if } x > a/2 \end{cases}$$

とする。 $H_{01}(x)$  に対して  $H_{mn}(x)$  を次のように作る。区間  $[0, a]$  を  $2^m$  の等しい長さの区間に分割する。分割された小区間を小さい方から 1 からかぞえてその番号を  $n$  とする。この小区間を  $[b1, b2]$  とすれば、

$$b1 = \frac{a}{2^m} \times (n - 1)$$

$$b2 = \frac{a}{2^m} \times n$$

である。小区間  $[b1, b2]$  において、Haar 関数 を、

$$H_{mn}(x) = \begin{cases} \sqrt{2^m/a} & \text{if } x \leq (b1 + b2)/2 \\ -\sqrt{2^m/a} & \text{if } x > (b1 + b2)/2 \end{cases}$$

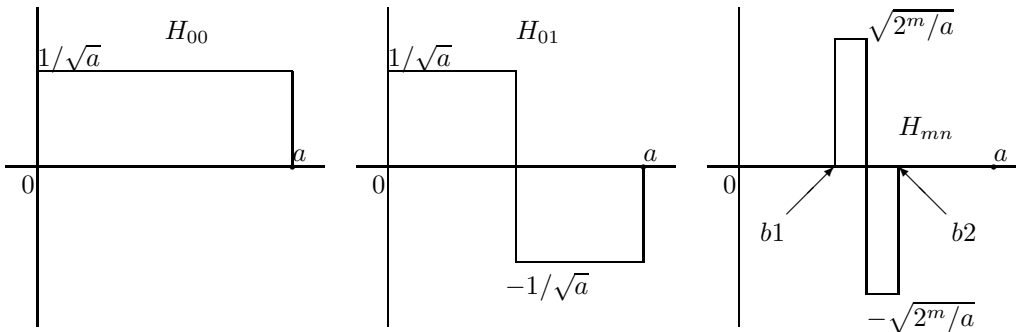
とする。関数の値は区間  $[0, a]$  で規格化されるように選んでいる。すなわち、

$$\int_0^a H_{mn}(x) \times H_{mn}(x) dx = 1$$

また、異なる  $m, m', n, n'$  の間では、

$$\int_0^a H_{mn}(x) \times H_{m'n'} dx = 0$$

になっていることに注意する。すなわち、これらの  $H_{mn}$  は正規直交系をなしている。また、以上の作り方からわかるように、 $n$  として指定できる数は  $m$  に対して  $n = 1, \dots, 2^m$  である。



## (2) Haar 関数によるウェーブレット変換

Haar 関数によってウェーブレット変換を行うということは、与えられた入力データと各 Haar 関数の積の積分を入力データの存在範囲にわたって行うということである。入力データを仮に連続関数  $y = f(x)$  とすれば、ウェーブレット変換は

$$C_{mn} = \int_0^a f(x) \times H_{mn}(x) dx$$

である。この計算を遂行するには Haar 関数  $H_{mn}$  の (1) 値  $y_H$ , (2) 立ち上がり点の位置  $x_0$ , (3) 正負が逆転する点の位置  $x_1$ , (4) 値が 0 に戻る位置  $x_2$ , の情報が必要である。そのため、インデックス  $(m, n)$  を与えると、これらの Haar 関数  $H_{mn}(x)$  の情報を作るために、本ライブラリではサブルーチン 2.18.1  $\left\{ \begin{array}{l} \text{DFWTH1} \\ \text{RFWTH1} \end{array} \right\}$  を用意している。入力データが連続関数の場合、これらのサブルーチンの結果を用いて数値積分でウェーブレット変換が可能である。

$$C_{mn} = \int_{x_0}^{x_1} f(x) \times y_H dx - \int_{x_1}^{x_2} f(x) \times y_H dx$$

である。入力データが離散の場合には、ウェーブレット変換は次のようにして計算を行う。離散の入力データの値の範囲を「隣接するデータの位置との半分の範囲内で一定」と仮定する。すなわち、データ  $(x_i, y_i)$  が与えられたとき、このデータに対して  $[\frac{1}{2}(x_{i-1} + x_i), \frac{1}{2}(x_i + x_{i+1})]$  において、入力データは  $y_i$  を持つと仮定する。このようにして区分的に連続になった入力データに対して上記の積分を行う。もしもサンプリングが等間隔  $dx$  で行われ、サンプリング数が  $2^k$  ( $k$ : 自然数) である場合には、 $H_{(k-1)n}$  の  $[b1, b2]$  に対して、 $x = b1 + (b2 - b1)/4$ ,  $x = b2 - (b2 - b1)/4$  のどちらかに入力データがあるようにできる。このような入力データに対しては Haar 関数は完全系であり、 $H_{mn}, m = 0, 1, \dots, (k-1), n = 1, 2, \dots, 2^{k-1}$  の線形結合で入力データを完全に表すことができる。たとえば  $2 = 2^1$  個のデータ  $(x1, y1), (x2, y2)$  があるとき、

$$C_{00} = \int_{-(x2-x1)/2}^{x2+(x2-x1)/2} H_{00}(x + (x2 - x1)/2) \times y dx$$



は2つのデータの平均値である。ただし、Haar関数が両端で変換データの存在範囲の外にでるので、積分範囲を両側で  $(b_2 - b_1)/2$  だけ外に広げる。さらに

$$C_{01} = \int_{-(x_2-x_1)/2}^{x_2+(x_2-x_1)/2} H_{01}(x + (x_2 - x_1)/2) \times y dx$$

は2つのデータの、平均値からのずれを表す。このずれは正負の差はあるものの、絶対値は同じで、したがって  $H_{01}$  で完全に表現することができる。したがって、

$$y = C_{00} \times H_{00}(x) + C_{01} \times H_{01}$$

で入力データそのものを表現することができる。

サンプリングが等間隔  $dx$  で行われ、サンプリング数が  $2^k$  ( $k$ : 自然数) である場合には、積分の計算で個々の入力データの存在間隔を考慮せずに済むので、計算が簡単になる。この積分計算を簡単に済ませるために、入力データ数と同じ数  $na = 2^k$  個の配列に区間  $[0, 1]$  で  $H_{mn}$  の正、負、0を示す配列 LR を出力するため、本ライブラリでは、サブルーチン 2.18.4  $\left\{ \begin{array}{l} \text{DFWTH2} \\ \text{RFWTH2} \end{array} \right\}$  を用意している。入力データの存在範囲を  $[b_1, b_2]$  としたとき、上の計算の積分範囲は  $[b_1 - dx/2, b_2 + dx/2]$  となるが、 $a = (b_2 - b_1) + dx$  として、

$$C_{mn} = \sqrt{\frac{2^m}{a}} \sum \text{LR}(i) \times y_i$$

である。

### (3) Haar関数による逆ウェーブレット変換

Haar関数のウェーブレット変換に対する逆変換は  $C_{mn}$  から元のデータを再構築する作業である。元のデータを  $f(x)$  とすると、 $f(x)$  の再構成は

$$f'(x) = \sum_{mn} C_{mn} \times H_{mn}(x)$$

である。連続値またはサンプリングの間隔が等間隔でない場合には  $f'(x)$  は  $f(x)$  には一致しないことは、この和が Haar関数が  $m$  の最大値のときの、値を持つ範囲の  $1/2$  以下の周波数の値を作らないことからわかる。サンプリング間隔が等間隔でサンプリング数が  $2^k$  ( $k$ : 自然数) の場合には、 $(m$  の最大値)  $= k$  としてやれば元のデータを再現することができる。

### (4) メキシカンハット関数

連続系ウェーブレット変換に用いられるメキシカンハット関数  $\varphi_{MH}(x)$  は以下の式で与えられる。

$$\varphi_{MH}(x) = (1 - 2x^2)e^{-x^2}$$

この関数は  $[-\infty, +\infty]$  で値を持つ。周波数に対応するパラメータを  $a$  とし、 $b$  だけ  $x$  軸方向にシフトするとして、ウェーブレット変換の基底を、

$$\phi_{MH}(x; a, b) = \frac{1}{\sqrt{C}} \varphi_{MH}\left(\frac{x-b}{a}\right)$$

とする。ただし、 $C$  は

$$\int_{-\infty}^{+\infty} \phi_{MH}(x; a, b)^2 dx = 1$$

となるような規格化定数とする.

$$\int_{-\infty}^{+\infty} e^{-x^2} dx = \sqrt{\pi}$$

$$\int_{-\infty}^{+\infty} x^2 e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

$$\int_{-\infty}^{+\infty} x^4 e^{-x^2} dx = \frac{3 \times \sqrt{\pi}}{4}$$

から, 規格化定数は

$$C = a \left( 1 - \frac{a^2}{2} + \frac{3a^4}{4} \right) \sqrt{\frac{\pi}{2}}$$

とした. 任意の関数  $f(x)$  に対して, この関数でのウェーブレット変換は

$$\int_{-\infty}^{+\infty} \phi_{MH}(x; a, b) f(x) dx (W_{\phi_{MH}} f)(b, a) = \int_{-\infty}^{+\infty} \phi_{MH}(x; a, b) f(x) dx$$

である.

(5) フレンチハット関数

フレンチハット関数  $\varphi_{FH}(x)$  は

$$\varphi_{FH}(x) = \begin{cases} 1 & \text{if } |x| \leq 1 \\ -\frac{1}{2} & \text{if } 1 < |x| \leq 3 \\ 0 & \text{otherwise} \end{cases}$$

で定義される. メキシカンハット関数と同様に, 周波数に対応するパラメータを  $a$  とし,  $b$  だけ  $x$  軸方向にシフトするとして, ウェーブレット変換の基底を,

$$\phi_{FH}(x; a, b) = \frac{1}{\sqrt{(C)}} \varphi_{FH}\left(\frac{x-b}{a}\right)$$

とする. ただし,  $C$  は

$$\int_{-\infty}^{+\infty} \phi_{FH}(x; a, b)^2 dx = 1$$

となるような規格化定数とする.

$$C = 3a$$

となる. 任意の関数  $f(x)$  に対して, この関数でのウェーブレット変換は

$$(W_{\phi_{FH}} f)(b, a) = \int_{-\infty}^{+\infty} \phi_{FH}(x; a, b) f(x) dx$$

である.

## 2.1.3 参考文献

- (1) Brigham, E. Oran, "The Fast Fourier Transform", Prentice-Hall Inc. , (1974).
- (2) Cochran, W. T. et al. , IEEE Trans. Audio and Electroacoustics, Vol. 15. pp. 45-55 (1967).
- (3) Gentleman, W. M. and Sande, G. , AFIPS Conf. Proc. , Fall Joint Comput. Conf. , Vol. 29, pp. 563-578 (1966).
- (4) Glassman, J. A. , IEEE Trans. Comput. , Vol. 19, pp. 105-116 (1970).
- (5) Swarztrauber, P. N. , SIAM Rev. , Vol. 19, pp. 490-501 (1977).
- (6) 牧之内三郎, 鳥居達生, "数値解析", オーム社 (1975).
- (7) Temperton, C. , "Implementation of a Self-Sorting In-Place Prime-Factor FFT Algorithm", J. Comp. Phys. , 58, 283 (1985).
- (8) Temperton, C. , "Self-Sorting Mixed-Radix Fast Fourier Transforms", J. Comp. Phys. , 52, 1 (1983).
- (9) Temperton, C. , "Fast Mixed-Radix Real Fourier Transforms", J. Comp. Phys. , 52, 340 (1983).
- (10) 細野敏夫, "BASIC による高速ラプラス変換", 共立出版 (1984).
- (11) Hosono, T. , "Numerical inversion of Laplace transform and some applications to wave optics", Radio Science, vol. 16, pp. 1015 (1981).
- (12) Welch, P. D. , "The Use of the FFT for Estimation of Power Spectra: A Method Based on Averaging Over Short, Modified Periodograms", IEEE Trans. on Audio and Electroacoustics, Vol. AU-15, No. 2, pp. 70-73 (1967).
- (13) Rader, C. M. , "An Improved Algorithm for High Speed Autocorrelation with Applications to Spectral Estimation", IEEE Trans. on Audio and Electroacoustics, Vol. AU-18, No. 4, pp. 439-442 (1970).
- (14) Childers, D. G. (Ed.), "Modern Spectrum Analysis", IEEE Press (1978).
- (15) Pease, M. C. , "An Adaption of the Fast Fourier Transform for Parallel Processing", J. Assn. Comput. Mach. , 15, 252 (1968). ; Stockham, T. G. , "High Speed Convolution and Correlation", AFIPS Conf. Proc. , 28, 229 (1966).
- (16) Swarztrauber, P. N. , "Vectorizing the FFTs", Parallel Computations, 51 (1982).
- (17) Singleton, R. C. , "An Algorithm for Computing the Mixed Radix Fast Fourier Transform", IEEE Trans. Audio and Electroacoust. , AU-17, 93 (1969). ; Singleton, R. C. , "ALGOL Procedures for the Fast Fourier Transform", Commun. ACM, 11, 773 (1968).

## 2.2 1次元複素フーリエ変換 (実数引数型)

### 2.2.1 [非推奨]DFC1FB, RFC1FB

#### 1次元複素フーリエ変換 (初期化を含む変換)

(1) 機能

順変換

複素数データ  $c_k (k = 0, \dots, n - 1)$  に対して、複素フーリエ順変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n - 1)$$

逆変換

複素数データ  $c_k (k = 0, \dots, n - 1)$  に対して、複素フーリエ逆変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n - 1)$$

(2) 使用法

倍精度サブルーチン:

CALL DFC1FB (N, CR, CI, LD, ISW, IFAX, TRIGS, WK, IERR)

単精度サブルーチン:

CALL RFC1FB (N, CR, CI, LD, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	入力データ数 $n$ (注意事項 (a) 参照)
2	CR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD	入 力	入力データ $c_k$ の実部 (注意事項 (b) 参照)
				出 力	出力データ $d_j$ の実部 (注意事項 (b), (c) 参照)
3	CI	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD	入 力	入力データ $c_k$ の虚部 (注意事項 (b) 参照)
				出 力	出力データ $d_j$ の虚部 (注意事項 (b), (c) 参照)
4	LD	I	1	入 力	配列 CR, CI の大きさ
5	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含んだ順変換 ISW = -1:初期化を含んだ逆変換
6	IFAX	I	20	出 力	基数分け情報 (注意事項 (d) 参照)
7	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	出 力	三角関数テーブル (注意事項 (d) 参照)
8	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $N > 0$
- (b)  $N \leq LD$
- (c)  $ISW \in \{0, 1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の内容がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) データ数  $N$  の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $N = 289 (= 17^2)$  とするよりも  $N = 300 (= 2^2 \times 3 \times 5^2)$  や  $320 (= 2^6 \times 5)$ ,  $384 (= 2^7 \times 3)$  などとした方が通常は効率が良い。

- (b) 複素数データ  $c_k$  ( $k = 0, \dots, n-1$ ) の実部と虚部をそれぞれ  $\Re\{c_k\}$ ,  $\Im\{c_k\}$  とすると、 $c_k$  と配列 CR, CI の各要素は以下の様に対応する。

$$\begin{aligned} \Re\{c_0\} &\leftrightarrow \text{CR}(1), \quad \Im\{c_0\} &\leftrightarrow \text{CI}(1) \\ \Re\{c_1\} &\leftrightarrow \text{CR}(2), \quad \Im\{c_1\} &\leftrightarrow \text{CI}(2) \\ \dots &\dots \dots, \quad \dots &\dots \dots \\ \Re\{c_{n-1}\} &\leftrightarrow \text{CR}(n), \quad \Im\{c_{n-1}\} &\leftrightarrow \text{CI}(n) \end{aligned}$$

複素数データ  $d_j$  ( $j = 0, \dots, n-1$ ) についても同様である。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_k$  ( $k = 0, \dots, n-1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_k$  ( $k = 0, \dots, n-1$ ) とすると

$$\hat{c}_k = nc_k \quad (k = 0, \dots, n-1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数  $N$  の変換を繰り返し行う場合、一度このサブルーチン呼びその後は初期化後の変換 2.2.2  $\left\{ \begin{array}{l} \text{DFC1BF} \\ \text{RFC1BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.2.2  $\left\{ \begin{array}{l} \text{DFC1BF} \\ \text{RFC1BF} \end{array} \right\}$  の入力としなければならない。

なお、 $ISW=0$  として初期化だけを行う場合には、配列 CR, CI に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔

を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.2.2 (7) 使用例参照。

## 2.2.2 [非推奨]DFC1BF, RFC1BF

## 1次元複素フーリエ変換 (初期化後の変換)

## (1) 機能

## 順変換

複素数データ  $c_k (k = 0, \dots, n-1)$  に対して、複素フーリエ順変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1)$$

## 逆変換

複素数データ  $c_k (k = 0, \dots, n-1)$  に対して、複素フーリエ逆変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1)$$

## (2) 使用法

## 倍精度サブルーチン:

CALL DFC1BF (N, CR, CI, LD, ISW, IFAX, TRIGS, WK, IERR)

## 単精度サブルーチン:

CALL RFC1BF (N, CR, CI, LD, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	入力データ数 $n$ (注意事項 (a) 参照)
2	CR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD	入 力	入力データ $c_k$ の実部 (注意事項 (b) 参照)
				出 力	出力データ $d_j$ の実部 (注意事項 (b), (c) 参照)
3	CI	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD	入 力	入力データ $c_k$ の虚部 (注意事項 (b) 参照)
				出 力	出力データ $d_j$ の虚部 (注意事項 (b), (c) 参照)
4	LD	I	1	入 力	配列 CR, CI の大きさ
5	ISW	I	1	入 力	処理スイッチ ISW= 1:初期化後の順変換 ISW=-1:初期化後の逆変換
6	IFAX	I	20	入 力	基数分け情報 (注意事項 (a) 参照)
7	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	入 力	三角関数テーブル (注意事項 (a) 参照)
8	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	ワーク	作業領域
9	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$
- (b)  $N \leq LD$
- (c)  $ISW \in \{1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の内容がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

(a) このサブルーチンは、同じデータ数  $N$  の変換を繰り返し行う場合に初期化を含む変換 2.2.1  $\left\{ \begin{array}{l} \text{DFC1FB} \\ \text{RFC1FB} \end{array} \right\}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

(b) 複素数データ  $c_k$  ( $k = 0, \dots, n-1$ ) の実部と虚部をそれぞれ  $\Re\{c_k\}$ ,  $\Im\{c_k\}$  とすると、 $c_k$  と配列 CR, CI の各要素は以下の様に対応する。

$$\begin{array}{llll} \Re\{c_0\} & \leftrightarrow & \text{CR}(1), & \Im\{c_0\} \leftrightarrow \text{CI}(1) \\ \Re\{c_1\} & \leftrightarrow & \text{CR}(2), & \Im\{c_1\} \leftrightarrow \text{CI}(2) \\ \dots & \dots & \dots & \dots \\ \Re\{c_{n-1}\} & \leftrightarrow & \text{CR}(n), & \Im\{c_{n-1}\} \leftrightarrow \text{CI}(n) \end{array}$$

複素数データ  $d_j$  ( $j = 0, \dots, n-1$ ) についても同様である。

(c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_k$  ( $k = 0, \dots, n-1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_k$  ( $k = 0, \dots, n-1$ ) とすると

$$\hat{c}_k = nc_k \quad (k = 0, \dots, n-1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

(d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

(e) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。

(f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。



## (7) 使用例

## (a) 問題

CR (1) = 3.000	CI (1) = 0.000
CR (2) = 2.786	CI (2) = 0.725
CR (3) = 2.300	CI (3) = 1.173
CR (4) = 1.792	CI (4) = 1.327
CR (5) = 1.381	CI (5) = 1.302
CR (6) = 1.080	CI (6) = 1.197
CR (7) = 0.865	CI (7) = 1.065
CR (8) = 0.711	CI (8) = 0.930
CR (9) = 0.600	CI (9) = 0.800
CR (10) = 0.519	CI (10) = 0.679
CR (11) = 0.459	CI (11) = 0.566
CR (12) = 0.415	CI (12) = 0.461
CR (13) = 0.383	CI (13) = 0.361
CR (14) = 0.360	CI (14) = 0.267
CR (15) = 0.345	CI (15) = 0.176
CR (16) = 0.336	CI (16) = 0.087

上記の数列を入力データとして、1次元複素フーリエ順・逆変換を行う。

## (b) 入力データ

配列 CR, CI, N=16, LD=16, ISW=1(順変換) および ISW=-1(逆変換)

## (c) 主プログラム

```

PROGRAM BFC1BF
! *** EXAMPLE OF DFC1FB AND DFC1BF ***
IMPLICIT REAL(8) (A-H,O-Z)
PARAMETER (LD = 16)
DIMENSION CR(LD),CI(LD),TRIGS(2*LD),WK(2*LD)
DIMENSION IFAX(20)
!**** INPUT ****
READ(5,*) N
READ(5,*) (CR(I),CI(I),I=1,N)
WRITE(6,1000) LD,N
WRITE(6,2000)
WRITE(6,2100) (I,CR(I),I,CI(I),I=1,N)
!**** OUTPUT ****
WRITE(6,1100)
!**** FORWARD TRANSFORM ****
ISW=1
CALL DFC1FB(N,CR,CI,LD,ISW,IFAX,TRIGS,WK,IERR)
!**** NORMALIZATION ****
DO 20 I = 1,N
  CR(I) = CR(I)/DBLE(N)
  CI(I) = CI(I)/DBLE(N)
20 CONTINUE
WRITE(6,1200)
WRITE(6,1400) IERR
WRITE(6,2000)
WRITE(6,2100) (I,CR(I),I,CI(I),I=1,N)
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL DFC1BF(N,CR,CI,LD,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1300)
WRITE(6,1400) IERR
WRITE(6,2000)
WRITE(6,2100) (I,CR(I),I,CI(I),I=1,N)
STOP
!**** FORMAT ****
1000 FORMAT(1X,'*** DFC1FB AND DFC1BF ***',/,/,&
  1X,' ** INPUT **',/,/,&
  1X,' LD = ',I2,/,/,&
  1X,' INPUT DATA',/,/,&
  1X,' N = ',I2,/)
1100 FORMAT(/,/,1X,' ** OUTPUT **')
1200 FORMAT(/,5X,'( FORWARD TRANSFORM )',/)
1300 FORMAT(/,5X,'( BACKWARD TRANSFORM )',/)
1400 FORMAT(5X,'IERR = ',I5,/,/,&
  5X,'SOLUTION',/)
2000 FORMAT(7X,'REAL PART',16X,'IMAGINARY PART')
2100 FORMAT(7X,'CR(',I2,') = ',F9.5,6X,'CI(',I2,') = ',F9.5)
END

```

(d) 出力結果

\*\*\* DFC1FB AND DFC1BF \*\*\*

\*\* INPUT \*\*

LD = 16

INPUT DATA

N = 16

REAL PART	IMAGINARY PART
CR( 1) = 3.00000	CI( 1) = 0.00000
CR( 2) = 2.78600	CI( 2) = 0.72500
CR( 3) = 2.30000	CI( 3) = 1.17300
CR( 4) = 1.79200	CI( 4) = 1.32700
CR( 5) = 1.38100	CI( 5) = 1.30200
CR( 6) = 1.08000	CI( 6) = 1.19700
CR( 7) = 0.86500	CI( 7) = 1.06500
CR( 8) = 0.71100	CI( 8) = 0.93000
CR( 9) = 0.60000	CI( 9) = 0.80000
CR(10) = 0.51900	CI(10) = 0.67900
CR(11) = 0.45900	CI(11) = 0.56600
CR(12) = 0.41500	CI(12) = 0.46100
CR(13) = 0.38300	CI(13) = 0.36100
CR(14) = 0.36000	CI(14) = 0.26700
CR(15) = 0.34500	CI(15) = 0.17600
CR(16) = 0.33600	CI(16) = 0.08700

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

SOLUTION

REAL PART	IMAGINARY PART
CR( 1) = 1.08325	CI( 1) = 0.69475
CR( 2) = 0.58324	CI( 2) = -0.46101
CR( 3) = 0.20845	CI( 3) = -0.32116
CR( 4) = 0.11461	CI( 4) = -0.19727
CR( 5) = 0.09112	CI( 5) = -0.12550
CR( 6) = 0.08538	CI( 6) = -0.08260
CR( 7) = 0.08389	CI( 7) = -0.05409
CR( 8) = 0.08346	CI( 8) = -0.03247
CR( 9) = 0.08338	CI( 9) = -0.01438
CR(10) = 0.08338	CI(10) = 0.00265
CR(11) = 0.08330	CI(11) = 0.01966
CR(12) = 0.08323	CI(12) = 0.03826
CR(13) = 0.08325	CI(13) = 0.06088
CR(14) = 0.08326	CI(14) = 0.09146
CR(15) = 0.08336	CI(15) = 0.13984
CR(16) = 0.08345	CI(16) = 0.24098

( BACKWARD TRANSFORM )

IERR = 0

SOLUTION

REAL PART	IMAGINARY PART
CR( 1) = 3.00000	CI( 1) = 0.00000
CR( 2) = 2.78600	CI( 2) = 0.72500
CR( 3) = 2.30000	CI( 3) = 1.17300
CR( 4) = 1.79200	CI( 4) = 1.32700
CR( 5) = 1.38100	CI( 5) = 1.30200
CR( 6) = 1.08000	CI( 6) = 1.19700
CR( 7) = 0.86500	CI( 7) = 1.06500
CR( 8) = 0.71100	CI( 8) = 0.93000
CR( 9) = 0.60000	CI( 9) = 0.80000
CR(10) = 0.51900	CI(10) = 0.67900
CR(11) = 0.45900	CI(11) = 0.56600
CR(12) = 0.41500	CI(12) = 0.46100
CR(13) = 0.38300	CI(13) = 0.36100
CR(14) = 0.36000	CI(14) = 0.26700
CR(15) = 0.34500	CI(15) = 0.17600
CR(16) = 0.33600	CI(16) = 0.08700

## 2.3 1次元複素フーリエ変換 (複素引数型)

### 2.3.1 [非推奨]ZFC1FB, CFC1FB

#### 1次元複素フーリエ変換 (初期化を含む変換)

(1) 機能

順変換

複素数データ  $c_k (k = 0, \dots, n - 1)$  に対して、複素フーリエ順変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n - 1)$$

逆変換

複素数データ  $c_k (k = 0, \dots, n - 1)$  に対して、複素フーリエ逆変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n - 1)$$

(2) 使用法

倍精度サブルーチン:

CALL ZFC1FB (N, C, LD, ISW, IFAX, TRIGS, WK, IERR)

単精度サブルーチン:

CALL CFC1FB (N, C, LD, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	入力データ数 $n$ (注意事項 (a) 参照)
2	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LD	入 力	入力データ $c_k$ (注意事項 (b) 参照)
				出 力	出力データ $d_j$ (注意事項 (b), (c) 参照)
3	LD	I	1	入 力	配列 C の大きさ
4	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含んだ順変換 ISW = -1:初期化を含んだ逆変換
5	IFAX	I	20	出 力	基数分け情報 (注意事項 (d) 参照)
6	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	出 力	三角関数テーブル (注意事項 (d) 参照)
7	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$
- (b)  $N \leq LD$
- (c)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の内容がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) データ数  $N$  の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $N = 289 (=17^2)$  とするよりも  $N = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い。

- (b) 複素数データ  $c_k (k = 0, \dots, n-1)$  と配列  $C$  の各要素は以下の様に対応する。

$$\begin{aligned} c_0 &\leftrightarrow C(1) \\ c_1 &\leftrightarrow C(2) \\ \dots &\dots \dots \\ c_{n-1} &\leftrightarrow C(n) \end{aligned}$$

複素数データ  $d_j (j = 0, \dots, n-1)$  についても同様である。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_k (k = 0, \dots, n-1)$  に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_k (k = 0, \dots, n-1)$  とすると

$$\hat{c}_k = nc_k \quad (k = 0, \dots, n-1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数  $N$  の変換を繰り返し行う場合、一度このサブルーチンを呼びその後は初期化後の変換 2.3.2  $\left\{ \begin{array}{l} \text{ZFC1BF} \\ \text{CFC1BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.3.2  $\left\{ \begin{array}{l} \text{ZFC1BF} \\ \text{CFC1BF} \end{array} \right\}$  の入力としなければならない。

なお、 $ISW=0$  として初期化だけを行う場合には、配列  $C$  に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔

を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.3.2 (7) 使用例参照。

### 2.3.2 [非推奨]ZFC1BF, CFC1BF 1次元複素フーリエ変換 (初期化後の変換)

(1) 機能

順変換

複素数データ  $c_k (k = 0, \dots, n - 1)$  に対して、複素フーリエ順変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n - 1)$$

逆変換

複素数データ  $c_k (k = 0, \dots, n - 1)$  に対して、複素フーリエ逆変換 (任意基数) を行う。

$$d_j = \sum_{k=0}^{n-1} c_k e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n - 1)$$

(2) 使用法

倍精度サブルーチン:

CALL ZFC1BF (N, C, LD, ISW, IFAX, TRIGS, WK, IERR)

単精度サブルーチン:

CALL CFC1BF (N, C, LD, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	入力データ数 $n$ (注意事項 (a) 参照)
2	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LD	入 力	入力データ $c_k$ (注意事項 (b) 参照)
				出 力	出力データ $d_j$ (注意事項 (b), (c) 参照)
3	LD	I	1	入 力	配列 C の大きさ
4	ISW	I	1	入 力	処理スイッチ ISW= 1:初期化後の順変換 ISW=-1:初期化後の逆変換
5	IFAX	I	20	入 力	基数分け情報 (注意事項 (a) 参照)
6	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	入 力	三角関数テーブル (注意事項 (a) 参照)
7	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	N	ワーク	作業領域
8	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $N > 0$
- (b)  $N \leq LD$
- (c)  $ISW \in \{1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の内容がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンは、同じデータ数  $N$  の変換を繰り返し行う場合に初期化を含む変換

2.3.1  $\begin{cases} \text{ZFC1FB} \\ \text{CFC1FB} \end{cases}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

- (b) 複素数データ  $c_k (k = 0, \dots, n-1)$  と配列 C の各要素は以下の様に対応する。

$$c_0 \leftrightarrow C(1)$$

$$c_1 \leftrightarrow C(2)$$

...

$$c_{n-1} \leftrightarrow C(n)$$

複素数データ  $d_j (j = 0, \dots, n-1)$  についても同様である。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_k (k = 0, \dots, n-1)$  に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_k (k = 0, \dots, n-1)$  とすると

$$\hat{c}_k = nc_k \quad (k = 0, \dots, n-1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (e) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。

- (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

## (a) 問題

$$\begin{aligned}
C(1) &= 3.000 + \sqrt{-1} \times 0.000 \\
C(2) &= 2.786 + \sqrt{-1} \times 0.725 \\
C(3) &= 2.300 + \sqrt{-1} \times 1.173 \\
C(4) &= 1.792 + \sqrt{-1} \times 1.327 \\
C(5) &= 1.381 + \sqrt{-1} \times 1.302 \\
C(6) &= 1.080 + \sqrt{-1} \times 1.197 \\
C(7) &= 0.865 + \sqrt{-1} \times 1.065 \\
C(8) &= 0.711 + \sqrt{-1} \times 0.930 \\
C(9) &= 0.600 + \sqrt{-1} \times 0.800 \\
C(10) &= 0.519 + \sqrt{-1} \times 0.679 \\
C(11) &= 0.459 + \sqrt{-1} \times 0.566 \\
C(12) &= 0.415 + \sqrt{-1} \times 0.461 \\
C(13) &= 0.383 + \sqrt{-1} \times 0.361 \\
C(14) &= 0.360 + \sqrt{-1} \times 0.267 \\
C(15) &= 0.345 + \sqrt{-1} \times 0.176 \\
C(16) &= 0.336 + \sqrt{-1} \times 0.087
\end{aligned}$$

上記の数列を入力データとして、1次元複素フーリエ順・逆変換を行う。

## (b) 入力データ

配列 C, N=16, LD=16, ISW=1(順変換) および ISW=-1(逆変換)

## (c) 主プログラム

```

PROGRAM AFC1BF
! *** EXAMPLE OF ZFC1FB AND ZFC1BF ***
IMPLICIT REAL(8) (A-H,O-Z)
PARAMETER (LD = 16)
DIMENSION IFAX(20)
COMPLEX(8) C,WK
DIMENSION C(LD),TRIGS(2*LD),WK(LD)
!**** INPUT ****
READ(5,*) N
READ(5,*) (C(I),I=1,N)
WRITE(6,1000) LD,N
WRITE(6,2000)
WRITE(6,2100) (I,DBLE(C(I)),I,DIMAG(C(I)),I=1,N)
!**** OUTPUT ****
WRITE(6,1100)
!**** FORWARD TRANSFORM ****
ISW=1
CALL ZFC1FB(N,C,LD,ISW,IFAX,TRIGS,WK,IERR)
!**** NORMALIZATION ****
DO 20 I = 1,N
  C(I) = C(I)/DBLE(N)
20 CONTINUE
WRITE(6,1200)
WRITE(6,1400) IERR
WRITE(6,2000)
WRITE(6,2100) (I,DBLE(C(I)),I,DIMAG(C(I)),I=1,N)
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL ZFC1FB(N,C,LD,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1300)
WRITE(6,1400) IERR
WRITE(6,2000)
WRITE(6,2100) (I,DBLE(C(I)),I,DIMAG(C(I)),I=1,N)
STOP
!**** FORMAT ****
1000 FORMAT(1X,'*** ZFC1FB AND ZFC1BF ***',/,/,&
  1X,' ** INPUT **',/,/,&
  1X,' LD = ',I2,/,/,&
  1X,' INPUT DATA',/,/,&
  1X,' N = ',I2,/)
1100 FORMAT(/,/,1X,' ** OUTPUT **')
1200 FORMAT(/,5X,'( FORWARD TRANSFORM )',/)
1300 FORMAT(/,5X,'( BACKWARD TRANSFORM )',/)
1400 FORMAT(5X,'IERR = ',I5,/,/,&
  5X,'SOLUTION',/)
2000 FORMAT(7X,'REAL PART',16X,'IMAGINARY PART')
2100 FORMAT(7X,' C(',I2,') = ',F9.5,6X,' C(',I2,') = ',F9.5)
END

```



## (d) 出力結果

\*\*\* ZFC1FB AND ZFC1BF \*\*\*

\*\* INPUT \*\*

LD = 16

INPUT DATA

N = 16

REAL PART	IMAGINARY PART
C( 1) = 3.00000	C( 1) = 0.00000
C( 2) = 2.78600	C( 2) = 0.72500
C( 3) = 2.30000	C( 3) = 1.17300
C( 4) = 1.79200	C( 4) = 1.32700
C( 5) = 1.38100	C( 5) = 1.30200
C( 6) = 1.08000	C( 6) = 1.19700
C( 7) = 0.86500	C( 7) = 1.06500
C( 8) = 0.71100	C( 8) = 0.93000
C( 9) = 0.60000	C( 9) = 0.80000
C(10) = 0.51900	C(10) = 0.67900
C(11) = 0.45900	C(11) = 0.56600
C(12) = 0.41500	C(12) = 0.46100
C(13) = 0.38300	C(13) = 0.36100
C(14) = 0.36000	C(14) = 0.26700
C(15) = 0.34500	C(15) = 0.17600
C(16) = 0.33600	C(16) = 0.08700

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

SOLUTION

REAL PART	IMAGINARY PART
C( 1) = 1.08325	C( 1) = 0.69475
C( 2) = 0.58324	C( 2) = -0.46101
C( 3) = 0.20845	C( 3) = -0.32116
C( 4) = 0.11461	C( 4) = -0.19727
C( 5) = 0.09112	C( 5) = -0.12550
C( 6) = 0.08538	C( 6) = -0.08260
C( 7) = 0.08389	C( 7) = -0.05409
C( 8) = 0.08346	C( 8) = -0.03247
C( 9) = 0.08338	C( 9) = -0.01438
C(10) = 0.08338	C(10) = 0.00265
C(11) = 0.08330	C(11) = 0.01966
C(12) = 0.08323	C(12) = 0.03826
C(13) = 0.08325	C(13) = 0.06088
C(14) = 0.08326	C(14) = 0.09146
C(15) = 0.08336	C(15) = 0.13984
C(16) = 0.08345	C(16) = 0.24098

( BACKWARD TRANSFORM )

IERR = 0

SOLUTION

REAL PART	IMAGINARY PART
C( 1) = 3.00000	C( 1) = 0.00000
C( 2) = 2.78600	C( 2) = 0.72500
C( 3) = 2.30000	C( 3) = 1.17300
C( 4) = 1.79200	C( 4) = 1.32700
C( 5) = 1.38100	C( 5) = 1.30200
C( 6) = 1.08000	C( 6) = 1.19700
C( 7) = 0.86500	C( 7) = 1.06500
C( 8) = 0.71100	C( 8) = 0.93000
C( 9) = 0.60000	C( 9) = 0.80000
C(10) = 0.51900	C(10) = 0.67900
C(11) = 0.45900	C(11) = 0.56600
C(12) = 0.41500	C(12) = 0.46100
C(13) = 0.38300	C(13) = 0.36100
C(14) = 0.36000	C(14) = 0.26700
C(15) = 0.34500	C(15) = 0.17600
C(16) = 0.33600	C(16) = 0.08700

## 2.4 1次元実フーリエ変換

### 2.4.1 [非推奨]DFR1FB, RFR1FB

#### 1次元実フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

実数データ  $r_k (k = 0, \dots, n-1)$  に対して, フーリエ順変換 (任意基数) の半周期分を求める.

$$c_j = \sum_{k=0}^{n-1} r_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, \lfloor \frac{n}{2} \rfloor)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す. なお, 残りの半周期分は以下の関係から得られる.

$$c_{n-j}^* = c_j$$

ただし,  $z^*$  は複素数  $z$  の共役複素数を表す.

###### 逆変換

$c_{n-j}^* = c_j$  を満たす  $n$  個の複素数データ  $c_j (j = 0, \dots, n-1)$  についてその半周期分  $c_j (j = 0, \dots, \lfloor \frac{n}{2} \rfloor)$  を与えて以下のように定義されるフーリエ逆変換 (任意基数) を求める.

$$\begin{aligned} r_k &= \sum_{j=0}^{n-1} c_j e^{2\pi\sqrt{-1}\frac{jk}{n}} \\ &= c_0 + (-1)^k \hat{c}_{\frac{n}{2}} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \Re\{c_j e^{2\pi\sqrt{-1}\frac{jk}{n}}\} \\ &= c_0 + (-1)^k \hat{c}_{\frac{n}{2}} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \left[ \Re\{c_j\} \cos(2\pi\frac{jk}{n}) - \Im\{c_j\} \sin(2\pi\frac{jk}{n}) \right] \\ &\quad (k = 0, \dots, n-1) \end{aligned}$$

ここで  $\lceil x \rceil$  は  $x$  以上の最小の整数を,  $\Re\{z\}$  と  $\Im\{z\}$  はそれぞれ複素数  $z$  の実部と虚部を表す. また,  $n$  が奇数のとき  $\hat{c}_{\frac{n}{2}} = 0$ ,  $n$  が偶数のとき  $\hat{c}_{\frac{n}{2}} = c_{\frac{n}{2}}$  である.

##### (2) 使用法

###### 倍精度サブルーチン:

CALL DFR1FB (N, R, LD, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL RFR1FB (N, R, LD, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	入力データ数 $n$ (注意事項 (a) 参照)
2	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD	入 力	入力データ $r_k$ (順変換), または $c_j$ (逆変換) (注意事項 (b) 参照)
				出 力	出力データ $c_j$ (順変換), または $r_k$ (逆変換) (注意事項 (b), (c) 参照)
3	LD	I	1	入 力	配列 R の大きさ
4	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含む順変換 ISW = -1:初期化を含む逆変換
5	IFAX	I	20	出 力	基数分け情報 (注意事項 (d) 参照)
6	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	出 力	三角関数テーブル (注意事項 (d) 参照)
7	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 N が奇数の時, 大きさ N+1 N が偶数の時, 大きさ N+2
8	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $N > 0$ 

(b) N が奇数の時 :

$$N + 1 \leq LD$$

N が偶数の時 :

$$N + 2 \leq LD$$

(c)  $ISW \in \{0, 1, -1\}$ 

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) データ数  $N$  の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $N = 289 (= 17^2)$  とするよりも  $N = 300 (= 2^2 \times 3 \times 5^2)$  や  $320 (= 2^6 \times 5)$ ,  $384 (= 2^7 \times 3)$  などとした方が通常は効率が良い。
- (b) 実数データ  $r_k (k = 0, \dots, n-1)$  と配列  $R$  の各要素は以下の様に対応する。

$$\begin{aligned} r_0 &\leftrightarrow R(1) \\ r_1 &\leftrightarrow R(2) \\ \dots &\dots \dots \\ r_{n-1} &\leftrightarrow R(n) \end{aligned}$$

なお、逆変換を行った場合、 $N (= n)$  が奇数のとき  $R(N+1) = 0$ ,  $N$  が偶数のとき  $R(N+1) = R(N+2) = 0$  となる。また、実数データ  $r_k (k = 0, \dots, n-1)$  を配列  $R$  に入力する場合、 $R(N+1)$  以降に対応する 0 を特に格納する必要はない。

複素数データ  $c_j (j = 0, \dots, \lfloor \frac{n}{2} \rfloor)$  の実部と虚部をそれぞれ  $\Re\{c_j\}$ ,  $\Im\{c_j\}$  とすると、 $c_j$  と配列  $R$  の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

$$\begin{aligned} \Re\{c_0\} &\leftrightarrow R(1) \\ \Im\{c_0\} &\leftrightarrow R(2) \\ \Re\{c_1\} &\leftrightarrow R(3) \\ \Im\{c_1\} &\leftrightarrow R(4) \\ \dots &\dots \dots \\ \Re\{c_{\lfloor \frac{n}{2} \rfloor}\} &\leftrightarrow R(m-1) \\ \Im\{c_{\lfloor \frac{n}{2} \rfloor}\} &\leftrightarrow R(m) \quad (m = N+1[N:\text{奇数}] \text{ または } N+2[N:\text{偶数}]) \end{aligned}$$

ただし、 $N$  が奇数のとき、 $m = N+1$ ,  $N$  が偶数のとき  $m = N+2$  とする。実フーリエ変換の性質より、 $N$  が奇数のとき  $\Im\{c_0\} = 0$ ,  $N$  が偶数のとき  $\Im\{c_0\} = \Im\{c_{\frac{n}{2}}\} = 0$  である。したがって、配列  $R$  の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_j (j = \lfloor \frac{n}{2} \rfloor + 1, \dots, n-1)$  の各要素は実フーリエ変換の対称性から以下の関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n-j} = c_j^*$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_k (k = 0, \dots, n-1)$  に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_k (k = 0, \dots, n-1)$  とすると

$$\hat{r}_k = nr_k \quad (k = 0, \dots, n-1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆に行っている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数  $N$  の変換を繰り返す場合、一度このサブルーチン呼びその後は初期化後の変換 2.4.2  $\left\{ \begin{array}{l} \text{DFR1BF} \\ \text{RFR1BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.4.2  $\left\{ \begin{array}{l} \text{DFR1BF} \\ \text{RFR1BF} \end{array} \right\}$  の入力としなければならない。

なお、ISW=0 として初期化だけを行う場合には、配列  $R$  に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標本数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.4.2 (7) 使用例参照。

## 2.4.2 [非推奨]DFR1BF, RFR1BF 1次元実フーリエ変換 (初期化後の変換)

### (1) 機能

#### 順変換

実数データ  $r_k (k = 0, \dots, n-1)$  に対して、フーリエ順変換 (任意基数) の半周期分を求める。

$$c_j = \sum_{k=0}^{n-1} r_k e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, \lfloor \frac{n}{2} \rfloor)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。なお、残りの半周期分は以下の関係から得られる。

$$c_{n-j}^* = c_j$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。

#### 逆変換

$c_{n-j}^* = c_j$  を満たす  $n$  個の複素数データ  $c_j (j = 0, \dots, n-1)$  についてその半周期分  $c_j (j = 0, \dots, \lfloor \frac{n}{2} \rfloor)$  を与えて以下のように定義されるフーリエ逆変換 (任意基数) を求める。

$$\begin{aligned} r_k &= \sum_{j=0}^{n-1} c_j e^{2\pi\sqrt{-1}\frac{jk}{n}} \\ &= c_0 + (-1)^k \hat{c}_{\frac{n}{2}} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \Re\{c_j e^{2\pi\sqrt{-1}\frac{jk}{n}}\} \\ &= c_0 + (-1)^k \hat{c}_{\frac{n}{2}} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \left[ \Re\{c_j\} \cos(2\pi\frac{jk}{n}) - \Im\{c_j\} \sin(2\pi\frac{jk}{n}) \right] \\ &\quad (k = 0, \dots, n-1) \end{aligned}$$

ここで  $\lceil x \rceil$  は  $x$  以上の最小の整数を、 $\Re\{z\}$  と  $\Im\{z\}$  はそれぞれ複素数  $z$  の実部と虚部を表す。また、 $n$  が奇数のとき  $\hat{c}_{\frac{n}{2}} = 0$ 、 $n$  が偶数のとき  $\hat{c}_{\frac{n}{2}} = c_{\frac{n}{2}}$  である。

### (2) 使用法

#### 倍精度サブルーチン:

CALL DFR1BF (N, R, LD, ISW, IFAX, TRIGS, WK, IERR)

#### 単精度サブルーチン:

CALL RFR1BF (N, R, LD, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	入力データ数 $n$ (注意事項 (a) 参照)
2	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD	入 力	入力データ $r_k$ (順変換), または $c_j$ (逆変換) (注意事項 (b) 参照)
				出 力	出力データ $c_j$ (順変換), または $r_k$ (逆変換) (注意事項 (b), (c) 参照)
3	LD	I	1	入 力	配列 R の大きさ
4	ISW	I	1	入 力	処理スイッチ ISW = 1:初期化後の順変換 ISW = -1:初期化後の逆変換
5	IFAX	I	20	入 力	基数分け情報 (注意事項 (a) 参照)
6	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	入 力	三角関数テーブル (注意事項 (a) 参照)
7	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 N が奇数の時大きさ, N+1 N が偶数の時大きさ, N+2
8	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $N > 0$ 

(b) N が奇数の時 :

$$N + 1 \leq LD$$

N が偶数の時 :

$$N + 2 \leq LD$$

(c)  $ISW \in \{1, -1\}$ 

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

(a) このサブルーチンは、同じデータ数  $N$  の変換を繰り返し行う場合に初期化を含む変換

2.4.1  $\begin{cases} \text{DFR1FB} \\ \text{RFR1FB} \end{cases}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

(b) 実数データ  $r_k (k = 0, \dots, n - 1)$  と配列 R の各要素は以下の様に対応する。

$$\begin{aligned} r_0 &\leftrightarrow R(1) \\ r_1 &\leftrightarrow R(2) \\ \dots &\dots \dots \\ r_{n-1} &\leftrightarrow R(n) \end{aligned}$$

なお、逆変換を行った場合、 $N(=n)$  が奇数のとき  $R(N+1) = 0$ 、 $N$  が偶数のとき  $R(N+1) = R(N+2) = 0$  となる。また、実数データ  $r_k (k = 0, \dots, n - 1)$  を配列 R に入力する場合、 $R(N+1)$  以降に対応する 0 を特に格納する必要はない。

複素数データ  $c_j (j = 0, \dots, \lfloor \frac{n}{2} \rfloor)$  の実部と虚部をそれぞれ  $\Re\{c_j\}$ ,  $\Im\{c_j\}$  とすると、 $c_j$  と配列 R の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

$$\begin{aligned} \Re\{c_0\} &\leftrightarrow R(1) \\ \Im\{c_0\} &\leftrightarrow R(2) \\ \Re\{c_1\} &\leftrightarrow R(3) \\ \Im\{c_1\} &\leftrightarrow R(4) \\ \dots &\dots \dots \\ \Re\{c_{\lfloor \frac{n}{2} \rfloor}\} &\leftrightarrow R(m-1) \\ \Im\{c_{\lfloor \frac{n}{2} \rfloor}\} &\leftrightarrow R(m) \quad (m = N+1[N:\text{奇数}] \text{ または } N+2[N:\text{偶数}]) \end{aligned}$$

ただし、 $N$  が奇数のとき、 $m=N+1$ 、 $N$  が偶数のとき  $m=N+2$  とする。実フーリエ変換の性質より、 $N$  が奇数のとき  $\Im\{c_0\} = 0$ 、 $N$  が偶数のとき  $\Im\{c_0\} = \Im\{c_{\frac{n}{2}}\} = 0$  である。したがって、配列 R の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_j (j = \lfloor \frac{n}{2} \rfloor + 1, \dots, n - 1)$  の各要素は実フーリエ変換の対称性から以下の関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n-j} = c_j^*$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。

(c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_k (k = 0, \dots, n - 1)$  に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_k (k = 0, \dots, n - 1)$  とすると

$$\hat{r}_k = nr_k \quad (k = 0, \dots, n - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

(d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi(t - iT)}$$



- (e) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

## (a) 問題

R (1) = 2.000  
 R (2) = 1.503  
 R (3) = 1.000  
 R (4) = 0.665  
 R (5) = 0.500  
 R (6) = 0.452  
 R (7) = 0.478  
 R (8) = 0.553  
 R (9) = 0.667  
 R (10) = 0.815  
 R (11) = 1.000  
 R (12) = 1.227  
 R (13) = 1.500  
 R (14) = 1.808  
 R (15) = 2.094  
 R (16) = 2.214

上記の数列を入力データとして、1次元実フーリエ順・逆変換を行う。

## (b) 入力データ

配列 R, N=16, LD=18, ISW=1(順変換) および ISW=-1(逆変換)

## (c) 主プログラム

```

PROGRAM BFR1BF
! *** EXAMPLE OF DFR1FB AND DFR1BF ***
IMPLICIT REAL(8) (A-H,O-Z)
PARAMETER (LD = 18)
DIMENSION R(LD), TRIGS(2*LD), WK(LD)
COMPLEX(8) C(*)
POINTER (CP, C)
DIMENSION IFAX(20)
CP=LOC(R)
!**** INPUT ****
READ(5,*) N
READ(5,*) (R(I), I=1,N)
WRITE(6,1000) LD,N
WRITE(6,2000)
WRITE(6,2100) (I,R(I), I=1,N)
!**** OUTPUT ****
WRITE(6,1100)
!**** FORWARD TRANSFORM ****
ISW=1
CALL DFR1FB(N,R,LD, ISW, IFAX, TRIGS, WK, IERR)
!**** NORMALIZATION ****
DO 20 I = 1, (N+2)/2
  C(I) = C(I)/DBLE(N)
20 CONTINUE
WRITE(6,1200)
WRITE(6,1400) IERR
WRITE(6,2200)
WRITE(6,2300) (I, DBLE(C(I)), I, DIMAG(C(I)), I=1, (N+2)/2)
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL DFR1BF(N,C,LD, ISW, IFAX, TRIGS, WK, IERR)
WRITE(6,1300)
WRITE(6,1400) IERR
WRITE(6,2000)
WRITE(6,2100) (I,R(I), I=1,N)
STOP
!**** FORMAT ****
1000 FORMAT(1X, '*** DFR1FB AND DFR1BF ***', /, /, &
1X, ' ** INPUT **', /, /, &
1X, ' LD = ', I2, /, /, &

```

```

          1X, ' INPUT DATA',/,/,&
          1X, ' N = ',I2,/)
1100 FORMAT(/,/,1X, ' ** OUTPUT **')
1200 FORMAT(/,5X, '( FORWARD TRANSFORM )',/)
1300 FORMAT(/,5X, '( BACKWARD TRANSFORM )',/)
1400 FORMAT(5X, 'IERR = ',I5,/,/,&
          5X, 'SOLUTION',/)
2000 FORMAT(7X, 'REAL PART')
2100 FORMAT(7X, ' R(',I2,') = ',F9.5)
2200 FORMAT(7X, 'REAL PART',16X, 'IMAGINARY PART')
2300 FORMAT(7X, ' C(',I2,') = ',F9.5,6X, ' C(',I2,') = ',F9.5)
          END

```

(d) 出力結果

\*\*\* DFR1FB AND DFR1BF \*\*\*

\*\* INPUT \*\*

LD = 18

INPUT DATA

N = 16

REAL PART

R( 1) = 2.00000  
R( 2) = 1.50300  
R( 3) = 1.00000  
R( 4) = 0.66500  
R( 5) = 0.50000  
R( 6) = 0.45200  
R( 7) = 0.47800  
R( 8) = 0.55300  
R( 9) = 0.66700  
R(10) = 0.81500  
R(11) = 1.00000  
R(12) = 1.22700  
R(13) = 1.50000  
R(14) = 1.80800  
R(15) = 2.09400  
R(16) = 2.21400

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

SOLUTION

REAL PART

C( 1) = 1.15475  
C( 2) = 0.30936  
C( 3) = 0.08292  
C( 4) = 0.02223  
C( 5) = 0.00594  
C( 6) = 0.00156  
C( 7) = 0.00045  
C( 8) = 0.00010  
C( 9) = 0.00013

IMAGINARY PART

C( 1) = 0.00000  
C( 2) = 0.26794  
C( 3) = 0.07186  
C( 4) = 0.01923  
C( 5) = 0.00506  
C( 6) = 0.00139  
C( 7) = 0.00036  
C( 8) = 0.00010  
C( 9) = 0.00000

( BACKWARD TRANSFORM )

IERR = 0

SOLUTION

REAL PART

R( 1) = 2.00000  
R( 2) = 1.50300  
R( 3) = 1.00000  
R( 4) = 0.66500  
R( 5) = 0.50000  
R( 6) = 0.45200  
R( 7) = 0.47800  
R( 8) = 0.55300  
R( 9) = 0.66700  
R(10) = 0.81500  
R(11) = 1.00000  
R(12) = 1.22700  
R(13) = 1.50000  
R(14) = 1.80800  
R(15) = 2.09400  
R(16) = 2.21400

---

## 2.5 多重 1 次元複素フーリエ変換 (実数引数型)

### 2.5.1 [非推奨]DFCMFB, RFCMFB

#### 多重 1 次元複素フーリエ変換 (初期化を含む変換)

(1) 機能

順変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して,  $m$  重 1 次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

逆変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して,  $m$  重 1 次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

(2) 使用法

倍精度サブルーチン:

CALL DFCMFB (N, M, CR, CI, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

単精度サブルーチン:

CALL RFCMFB (N, M, CR, CI, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	変換データ数 $n$ (注意事項 (a) 参照)
2	M	I	1	入 力	多重度 $m$
3	CR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	入 力	入力データ $c_{k,l}$ の実部 (注意事項 (b) 参照) 大きさ: $INCN \times (N - 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $d_{j,l}$ の実部 (注意事項 (b), (c) 参照)
4	CI	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	入 力	入力データ $c_{k,l}$ の虚部 (注意事項 (b) 参照) 大きさ: $INCN \times (N - 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $d_{j,l}$ の虚部 (注意事項 (b), (c) 参照)
5	INCN	I	1	入 力	変換データの格納間隔 (注意事項 (b) 参照)
6	INCM	I	1	入 力	変換データ間の格納間隔 (注意事項 (b) 参照)
7	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW=0: 初期化のみ ISW=1: 初期化を含む順変換 ISW=-1: 初期化を含む逆変換
8	IFAX	I	20	出 力	基数分け情報 (注意事項 (d) 参照).
9	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	出 力	三角関数テーブル (注意事項 (d) 参照)
10	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times M \times N$	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $N > 0$   
 $M > 0$
- (b)  $INCN > 0$   
 $INCM > 0$
- (c)  $INCN \geq M \times \text{gcm}(INCN, INCM)$  または  
 $INCM \geq N \times \text{gcm}(INCN, INCM)$   
 (ただし,  $\text{gcm}(i, j)$  は  $i, j$  の最大公約数を表す.)
- (d)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

(6) 注意事項

- (a) 変換データ数 N の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば,  $N = 289 (= 17^2)$  とするよりも  $N = 300 (= 2^2 \times 3 \times 5^2)$  や  $320 (= 2^6 \times 5)$ ,  $384 (= 2^7 \times 3)$  などとした方が通常は効率が良い.
- (b) 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) の実部と虚部をそれぞれ  $\Re\{c_{k,l}\}$ ,  $\Im\{c_{k,l}\}$  とすると,  $c_{k,l}$  と配列 CR, CI の各要素は以下の様に対応する.

$$\begin{aligned} \Re\{c_{k,l}\} &\leftrightarrow \text{CR}(1 + \text{INCN} * k + \text{INCM} * (l - 1)) \\ \Im\{c_{k,l}\} &\leftrightarrow \text{CI}(1 + \text{INCN} * k + \text{INCM} * (l - 1)) \end{aligned}$$

例えば,  $\text{INCN}=1, \text{INCM}=n$  とすると,

$$\Re\{c_{k,l}\} \leftrightarrow \text{CR}((k + 1) + n * (l - 1)), \quad \Im\{c_{k,l}\} \leftrightarrow \text{CI}((k + 1) + n * (l - 1))$$

となり, 添え字  $k$  について連続に詰めて格納することになり  $\text{INCN}=m, \text{INCM}=1$  とすると,

$$\Re\{c_{k,l}\} \leftrightarrow \text{CR}(l + m * k), \quad \Im\{c_{k,l}\} \leftrightarrow \text{CI}(l + m * k)$$

となり, 添え字  $l$  について連続に詰めて格納することになる. 複素数データ  $d_{j,l}$  ( $j = 0, \dots, n-1; l = 1, \dots, m$ ) についても同様である. なお, 配列 CR と CI のデータを格納しない領域の値はこのサブルーチンの呼びだして変更されない.

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合, 得られるデータは, 元のデータをデータ数倍した値になる. 例えば, 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) とすると

$$\hat{c}_{k,l} = n c_{k,l} \quad (k = 0, \dots, n-1; l = 1, \dots, m)$$

となる. したがって, 順変換または逆変換の結果のどちらかに対して正規化を行う必要がある. なお, 文献によっては, 順変換と逆変換の定義を本書と逆に行っている場合や正規化を行った結果を定義としている場合があるので注意されたい.

- (d) 同じ変換データ数 N の変換を繰り返し行う場合, 一度このサブルーチン呼びその後は初期化後の変換 2.5.2  $\left\{ \begin{array}{l} \text{DFCMFB} \\ \text{RFCMFB} \end{array} \right\}$  を利用すれば良い. このようにすれば, 初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため, 効率のよい処理ができる. ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.5.2  $\left\{ \begin{array}{l} \text{DFCMFB} \\ \text{RFCMFB} \end{array} \right\}$  の入力としなければならない.  
 なお,  $\text{ISW}=0$  として初期化だけを行う場合には, 配列 CR, CI に入力データを設定する必要がない.

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので, 連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔

を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.5.2 (7) 使用例参照。

## 2.5.2 [非推奨]DFCMBF, RFCMBF

## 多重 1 次元複素フーリエ変換 (初期化後の変換)

## (1) 機能

## 順変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1$ ;  $l = 1, \dots, m$ ) に対して,  $m$  重 1 次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

## 逆変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1$ ;  $l = 1, \dots, m$ ) に対して,  $m$  重 1 次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

## (2) 使用法

## 倍精度サブルーチン:

CALL DFCMBF (N, M, CR, CI, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

## 単精度サブルーチン:

CALL RFCMBF (N, M, CR, CI, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{array} \right\}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	変換データ数 $n$ (注意事項 (a) 参照)
2	M	I	1	入 力	多重度 $m$
3	CR	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	内容参照	入 力	入力データ $c_{k,l}$ の実部 (注意事項 (b) 参照) 大きさ: $INCN \times (N - 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $d_{j,l}$ の実部 (注意事項 (b), (c) 参照)
4	CI	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	内容参照	入 力	入力データ $c_{k,l}$ の虚部 (注意事項 (b) 参照) 大きさ: $INCN \times (N - 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $d_{j,l}$ の虚部 (注意事項 (b), (c) 参照)
5	INCN	I	1	入 力	変換データの格納間隔 (注意事項 (b) 参照)
6	INCM	I	1	入 力	変換データ間の格納間隔 (注意事項 (b) 参照)
7	ISW	I	1	入 力	処理スイッチ ISW=1: 初期化後の順変換 ISW=-1: 初期化後の逆変換
8	IFAX	I	20	入 力	基数分け情報 (注意事項 (a) 参照).
9	TRIGS	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times N$	入 力	三角関数テーブル (注意事項 (a) 参照)
10	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times M \times N$	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$   
 $M > 0$
- (b)  $INCN > 0$   
 $INCM > 0$
- (c)  $INCN \geq M \times \text{gcm}(INCN, INCM)$  または  
 $INCM \geq N \times \text{gcm}(INCN, INCM)$   
(ただし,  $\text{gcm}(i, j)$  は  $i, j$  の最大公約数を表す.)
- (d)  $ISW \in \{1, -1\}$



## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

## (6) 注意事項

(a) このサブルーチンは、同じ変換データ数 N の変換を繰り返し行う場合に初期化を含む変換

2.5.1  $\left\{ \begin{array}{l} \text{DFCMFB} \\ \text{RFCMBF} \end{array} \right\}$  を行った後で利用する. なお, この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある.

(b) 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) の実部と虚部をそれぞれ  $\Re\{c_{k,l}\}$ ,  $\Im\{c_{k,l}\}$  とすると,  $c_{k,l}$  と配列 CR, CI の各要素は以下の様に対応する.

$$\Re\{c_{k,l}\} \leftrightarrow \text{CR}(1 + \text{INCN} * k + \text{INCM} * (l - 1))$$

$$\Im\{c_{k,l}\} \leftrightarrow \text{CI}(1 + \text{INCN} * k + \text{INCM} * (l - 1))$$

例えば,  $\text{INCN}=1, \text{INCM}=n$  とすると,

$$\Re\{c_{k,l}\} \leftrightarrow \text{CR}((k+1) + n * (l-1)), \quad \Im\{c_{k,l}\} \leftrightarrow \text{CI}((k+1) + n * (l-1))$$

となり, 添え字  $k$  について連続に詰めて格納することになり  $\text{INCN}=m, \text{INCM}=1$  とすると,

$$\Re\{c_{k,l}\} \leftrightarrow \text{CR}(l + m * k), \quad \Im\{c_{k,l}\} \leftrightarrow \text{CI}(l + m * k)$$

となり, 添え字  $l$  について連続に詰めて格納することになる. 複素数データ  $d_{j,l}$  ( $j = 0, \dots, n-1; l = 1, \dots, m$ ) についても同様である. なお, 配列 CR と CI のデータを格納しない領域の値はこのサブルーチンの呼びだして変更されない.

(c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合, 得られるデータは, 元のデータをデータ数倍した値になる. 例えば, 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) とすると

$$\hat{c}_{k,l} = n c_{k,l} \quad (k = 0, \dots, n-1; l = 1, \dots, m)$$

となる. したがって, 順変換または逆変換の結果のどちらかに対して正規化を行う必要がある. なお, 文献によっては, 順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい.

(d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので, 連続フーリエ変換を標準化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある. なお, 標準化定理によれば, 周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合, 標準化間隔を  $T = \frac{1}{2f_c}$  ととれば, 以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる.

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

(e) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない.

- (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

(7) 使用例

(a) 問題

CR ( 1) = 1.000    CI ( 1) =4.000  
 CR ( 2) = 2.000    CI ( 2) =3.000  
 CR ( 3) = 3.000    CI ( 3) =2.000  
 CR ( 4) = 4.000    CI ( 4) =1.000  
 CR ( 5) = 4.000    CI ( 5) =1.000  
 CR ( 6) = 3.000    CI ( 6) =2.000  
 CR ( 7) = 2.000    CI ( 7) =3.000  
 CR ( 8) = 1.000    CI ( 8) =4.000  
 CR (10) = 1.000    CI (10) =2.000  
 CR (11) = 1.000    CI (11) =2.000  
 CR (12) = 2.000    CI (12) =1.000  
 CR (13) = 2.000    CI (13) =1.000  
 CR (14) = 2.000    CI (14) =1.000  
 CR (15) = 2.000    CI (15) =1.000  
 CR (16) = 1.000    CI (16) =2.000  
 CR (17) = 1.000    CI (17) =2.000  
 CR (19) = 1.000    CI (19) =2.000  
 CR (20) = 1.000    CI (20) =2.000  
 CR (21) = 1.000    CI (21) =2.000  
 CR (22) = 1.000    CI (22) =2.000  
 CR (23) = 2.000    CI (23) =1.000  
 CR (24) = 2.000    CI (24) =1.000  
 CR (25) = 2.000    CI (25) =1.000  
 CR (26) = 2.000    CI (26) =1.000  
 CR (28) = 1.000    CI (28) =1.000  
 CR (29) = 1.000    CI (29) =1.000  
 CR (30) = 1.000    CI (30) =1.000  
 CR (31) = 1.000    CI (31) =1.000  
 CR (32) = 1.000    CI (32) =1.000  
 CR (33) = 1.000    CI (33) =1.000  
 CR (34) = 1.000    CI (34) =1.000  
 CR (35) = 1.000    CI (35) =1.000

上記の数列を入力データとして、多重 1 次元複素フーリエ順・逆変換を行う。

(b) 入力データ

配列 CR, CI, N=8, M=4, INCN=1, INCM=9, ISW=1 (順変換) および ISW=-1 (逆変換)

(c) 主プログラム

```
PROGRAM BFCMBF
! *** EXAMPLE OF DFCMBF AND DFCMBF ***
IMPLICIT REAL(8) (A-H,O-Z)
PARAMETER (NN=9,MM=4)
DIMENSION CR(NN*MM),CI(NN*MM),TRIGS(2*NN),WK(2*NN*MM)
DIMENSION IFAX(20)
```

```

**** INPUT ****
  READ(5,*) N,M,INCN,INCM
  WRITE(6,1000) N,M,INCN,INCM
  DO 20 J=1,M
    DO 10 I=1,N
      READ(5,*) CR(1+(I-1)*INCN+(J-1)*INCM), &
                CI(1+(I-1)*INCN+(J-1)*INCM)
    10 CONTINUE
  20 CONTINUE
  WRITE(6,2000) 'REAL PART'
  WRITE(6,2010) ((CR(1+(I-1)*INCN+(J-1)*INCM), I=1, N), J=1, M)
  WRITE(6,2000) 'IMAGINARY PART'
  WRITE(6,2010) ((CI(1+(I-1)*INCN+(J-1)*INCM), I=1, N), J=1, M)
  30 CONTINUE
**** OUTPUT ****
  WRITE(6,1010)
**** FORWARD TRANSFORM ****
  ISW = 1
  CALL DFCMFB(N,M,CR,CI,INCN,INCM,ISW,IFAX,TRIGS,WK,IERR)
**** NORMALIZATION ****
  DO 50 J=1,M
    DO 40 I=1,N
      CR(1+(I-1)*INCN+(J-1)*INCM)=&
        CR(1+(I-1)*INCN+(J-1)*INCM)/DBLE(N)
      CI(1+(I-1)*INCN+(J-1)*INCM)=&
        CI(1+(I-1)*INCN+(J-1)*INCM)/DBLE(N)
    40 CONTINUE
  50 CONTINUE
  WRITE(6,1020) IERR
  WRITE(6,2000) 'REAL PART'
  WRITE(6,2010) ((CR(1+(I-1)*INCN+(J-1)*INCM), I=1, N), J=1, M)
  WRITE(6,2000) 'IMAGINARY PART'
  WRITE(6,2010) ((CI(1+(I-1)*INCN+(J-1)*INCM), I=1, N), J=1, M)
**** BACKWARD TRANSFORM ****
  ISW = -1
  CALL DFCMFB(N,M,CR,CI,INCN,INCM,ISW,IFAX,TRIGS,WK,IERR)
  WRITE(6,1030) IERR
  WRITE(6,2000) 'REAL PART'
  WRITE(6,2010) ((CR(1+(I-1)*INCN+(J-1)*INCM), I=1, N), J=1, M)
  WRITE(6,2000) 'IMAGINARY PART'
  WRITE(6,2010) ((CI(1+(I-1)*INCN+(J-1)*INCM), I=1, N), J=1, M)
  STOP
**** FORMAT ****
1000 FORMAT(1X,'*** DFCMFB AND DFCMBF ***',/,/,&
           1X,' ** INPUT **',/,/,&
           1X,'      N = ',I3,'      M = ',I3,/,&
           1X,'      INCN = ',I3,'      INCM = ',I3,/)
1010 FORMAT(1X,/,&
           1X,' ** OUTPUT **',/)
1020 FORMAT(1X,' ( FORWARD TRANSFORM )',/,/,&
           1X,'      IERR = ',I4,/)
1030 FORMAT(1X,' ( BACKWARD TRANSFORM )',/,/,&
           1X,'      IERR = ',I4,/)
2000 FORMAT(1X,4X,A)
2010 FORMAT(1X,4X,8F8.4,/)
      END

```

## (d) 出力結果

```

*** DFCMFB AND DFCMBF ***

** INPUT **

      N = 8      M = 4
      INCN = 1   INCM = 9

REAL PART
1.0000  2.0000  3.0000  4.0000  4.0000  3.0000  2.0000  1.0000
1.0000  1.0000  2.0000  2.0000  2.0000  2.0000  1.0000  1.0000
1.0000  1.0000  1.0000  1.0000  2.0000  2.0000  2.0000  2.0000
1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000

IMAGINARY PART
4.0000  3.0000  2.0000  1.0000  1.0000  2.0000  3.0000  4.0000
2.0000  2.0000  1.0000  1.0000  1.0000  1.0000  2.0000  2.0000
2.0000  2.0000  2.0000  2.0000  1.0000  1.0000  1.0000  1.0000
1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000

** OUTPUT **

( FORWARD TRANSFORM )

IERR = 0

REAL PART
2.5000 -1.0303  0.0000 -0.0732  0.0000  0.0303  0.0000 -0.4268
1.5000 -0.4268  0.0000  0.1768  0.0000 -0.0732  0.0000 -0.1768
1.5000  0.1768  0.0000 -0.0732  0.0000 -0.1768  0.0000 -0.4268
1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000

```

```
IMAGINARY PART
2.5000 0.4268 0.0000 -0.0303 0.0000 0.0732 0.0000 1.0303
1.5000 0.1768 0.0000 0.0732 0.0000 -0.1768 0.0000 0.4268
1.5000 0.4268 0.0000 0.1768 0.0000 0.0732 0.0000 -0.1768
1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
( BACKWARD TRANSFORM )
IERR = 0
REAL PART
1.0000 2.0000 3.0000 4.0000 4.0000 3.0000 2.0000 1.0000
1.0000 1.0000 2.0000 2.0000 2.0000 2.0000 1.0000 1.0000
1.0000 1.0000 1.0000 1.0000 2.0000 2.0000 2.0000 2.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
IMAGINARY PART
4.0000 3.0000 2.0000 1.0000 1.0000 2.0000 3.0000 4.0000
2.0000 2.0000 1.0000 1.0000 1.0000 1.0000 2.0000 2.0000
2.0000 2.0000 2.0000 2.0000 1.0000 1.0000 1.0000 1.0000
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
```

## 2.6 多重1次元複素フーリエ変換 (複素指数型)

### 2.6.1 [非推奨]ZFCMFB, CFCMFB

#### 多重1次元複素フーリエ変換 (初期化を含む変換)

(1) 機能

順変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して,  $m$  重1次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

逆変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して,  $m$  重1次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

(2) 使用法

倍精度サブルーチン:

CALL ZFCMFB (N, M, C, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

単精度サブルーチン:

CALL CFCMFB (N, M, C, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	変換データ数 $n$ (注意事項 (a) 参照)
2	M	I	1	入 力	多重度 $m$
3	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	内容参照	入 力	入力データ $c_{k,l}$ (注意事項 (b) 参照) 大きさ: $INCN \times (N - 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $d_{j,l}$ (注意事項 (b), (c) 参照)
4	INCN	I	1	入 力	変換データの格納間隔 (注意事項 (b) 参照)
5	INCM	I	1	入 力	変換データ間の格納間隔 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW=0: 初期化のみ ISW=1: 初期化を含む順変換 ISW=-1: 初期化を含む逆変換
7	IFAX	I	20	出 力	基数分け情報 (注意事項 (d) 参照).
8	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	出 力	三角関数テーブル (注意事項 (d) 参照)
9	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	$M \times N$	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$   
 $M > 0$
- (b)  $INCN > 0$   
 $INCM > 0$
- (c)  $INCN \geq M \times \text{gcm}(INCN, INCM)$  または  
 $INCM \geq N \times \text{gcm}(INCN, INCM)$   
(ただし,  $\text{gcm}(i, j)$  は  $i, j$  の最大公約数を表す.)
- (d)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

(6) 注意事項

- (a) 変換データ数  $N$  の値を調整できる場合には, 混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば,  $N = 289 (= 17^2)$  とするよりも  $N = 300 (= 2^2 \times 3 \times 5^2)$  や  $320 (= 2^6 \times 5)$ ,  $384 (= 2^7 \times 3)$  などとした方が通常は効率が良い.
- (b) 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) と配列  $C$  の各要素は以下の様に対応する.

$$c_{k,l} \leftrightarrow C(1 + INCN * k + INCM * (l - 1))$$

例えば,  $INCN=1, INCM=n$  とすると,

$$c_{k,l} \leftrightarrow C((k + 1) + n * (l - 1))$$

となり, 添え字  $k$  について連続に詰めて格納することになり  $INCN=m, INCM=1$  とすると,

$$c_{k,l} \leftrightarrow C(l + m * k)$$

となり, 添え字  $l$  について連続に詰めて格納することになる. 複素数データ  $d_{j,l}$  ( $j = 0, \dots, n-1; l = 1, \dots, m$ ) についても同様である. なお, 配列  $C$  のデータを格納しない領域の値はこのサブルーチンの呼び出しで変更されない.

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合, 得られるデータは, 元のデータをデータ数倍した値になる. 例えば, 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) とすると

$$\hat{c}_{k,l} = nc_{k,l} \quad (k = 0, \dots, n-1; l = 1, \dots, m)$$

となる. したがって, 順変換または逆変換の結果のどちらかに対して正規化を行う必要がある. なお, 文献によっては, 順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい.

- (d) 同じ変換データ数  $N$  の変換を繰り返し行う場合、一度このサブルーチン呼びその後は初期化後の変換 2.6.2  $\left\{ \begin{array}{l} \text{ZFCMBF} \\ \text{CFCMBF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.6.2  $\left\{ \begin{array}{l} \text{ZFCMBF} \\ \text{CFCMBF} \end{array} \right\}$  の入力としなければならない。
- なお、ISW=0 として初期化だけを行う場合には、配列 C に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.6.2 (7) 使用例参照。

## 2.6.2 [非推奨]ZFCMBF, CFCMBF 多重1次元複素フーリエ変換 (初期化後の変換)

### (1) 機能

#### 順変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して,  $m$  重1次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

#### 逆変換

複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して,  $m$  重1次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j,l} = \sum_{k=0}^{n-1} c_{k,l} e^{2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, n-1; l = 1, \dots, m)$$

### (2) 使用法

#### 倍精度サブルーチン:

CALL ZFCMBF (N, M, C, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

#### 単精度サブルーチン:

CALL CFCMBF (N, M, C, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

### (3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	変換データ数 $n$ (注意事項 (a) 参照)
2	M	I	1	入 力	多重度 $m$
3	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	内容参照	入 力	入力データ $c_{k,l}$ (注意事項 (b) 参照) 大きさ: $\text{INCN} \times (N - 1) + \text{INCM} \times (M - 1) + 1$
				出 力	出力データ $d_{j,l}$ (注意事項 (b), (c) 参照)
4	INCN	I	1	入 力	変換データの格納間隔 (注意事項 (b) 参照)
5	INCM	I	1	入 力	変換データ間の格納間隔 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ ISW=1: 初期化後の順変換 ISW=-1: 初期化後の逆変換
7	IFAX	I	20	入 力	基数分け情報 (注意事項 (a) 参照).
8	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times N$	入 力	三角関数テーブル (注意事項 (a) 参照)
9	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	$M \times N$	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ



## (4) 制限条件

- (a)  $N > 0$   
 $M > 0$
- (b)  $INCN > 0$   
 $INCM > 0$
- (c)  $INCN \geq M \times \text{gcm}(INCN, INCM)$  または  
 $INCM \geq N \times \text{gcm}(INCN, INCM)$   
(ただし,  $\text{gcm}(i, j)$  は  $i, j$  の最大公約数を表す.)
- (d)  $ISW \in \{1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$N = 1$ であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンは, 同じ変換データ数  $N$  の変換を繰り返し行う場合に初期化を含む変換  
2.6.1  $\left\{ \begin{array}{l} \text{ZFCMFB} \\ \text{CFCMFB} \end{array} \right\}$  を行った後で利用する. なお, この場合は配列 IFAX, TRIGS の内容はそのままこの  
サブルーチンの入力とする必要がある.

- (b) 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) と配列 C の各要素は以下の様に対応する.

$$c_{k,l} \leftrightarrow C(1 + INCN * k + INCM * (l - 1))$$

例えば,  $INCN=1, INCM=n$  とすると,

$$c_{k,l} \leftrightarrow C((k + 1) + n * (l - 1))$$

となり, 添え字  $k$  について連続に詰めて格納することになり  $INCN=m, INCM=1$  とすると,

$$c_{k,l} \leftrightarrow C(l + m * k)$$

となり, 添え字  $l$  について連続に詰めて格納することになる. 複素数データ  $d_{j,l}$  ( $j = 0, \dots, n-1; l = 1, \dots, m$ ) についても同様である. なお, 配列 C のデータを格納しない領域の値はこのサブルーチンの呼びだしで変更されない.

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合, 得られるデータは, 元のデータをデータ数倍した値になる. 例えば, 複素数データ  $c_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k,l}$  ( $k = 0, \dots, n-1; l = 1, \dots, m$ ) とすると

$$\hat{c}_{k,l} = n c_{k,l} \quad (k = 0, \dots, n-1; l = 1, \dots, m)$$

となる. したがって, 順変換または逆変換の結果のどちらかに対して正規化を行う必要がある. なお, 文献によっては, 順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい.

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標本数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (e) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。  
(f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

### (a) 問題

- C ( 1) = (1.000, 4.000)
- C ( 2) = (2.000, 3.000)
- C ( 3) = (3.000, 2.000)
- C ( 4) = (4.000, 1.000)
- C ( 5) = (4.000, 1.000)
- C ( 6) = (3.000, 2.000)
- C ( 7) = (2.000, 3.000)
- C ( 8) = (1.000, 4.000)
- C (10) = (1.000, 2.000)
- C (11) = (1.000, 2.000)
- C (12) = (2.000, 1.000)
- C (13) = (2.000, 1.000)
- C (14) = (2.000, 1.000)
- C (15) = (2.000, 1.000)
- C (16) = (1.000, 2.000)
- C (17) = (1.000, 2.000)
- C (19) = (1.000, 2.000)
- C (20) = (1.000, 2.000)
- C (21) = (1.000, 2.000)
- C (22) = (1.000, 2.000)
- C (23) = (2.000, 1.000)
- C (24) = (2.000, 1.000)
- C (25) = (2.000, 1.000)
- C (26) = (2.000, 1.000)
- C (28) = (1.000, 1.000)
- C (29) = (1.000, 1.000)
- C (30) = (1.000, 1.000)
- C (31) = (1.000, 1.000)
- C (32) = (1.000, 1.000)
- C (33) = (1.000, 1.000)
- C (34) = (1.000, 1.000)

C (35) = (1.000, 1.000)

上記の数列を入力データとして、多重1次元複素フーリエ順・逆変換を行う。

(b) 入力データ

配列 C, N=8, M=4, INCN=1, INCM=9, ISW=1 (順変換) および ISW=-1 (逆変換)

(c) 主プログラム

```

PROGRAM AFCMBF
! *** EXAMPLE OF ZFCMBF AND ZFCMBF ***
IMPLICIT REAL(8) (A-H,O-Z)
PARAMETER (NN=9,MM=4)
COMPLEX(8) C(NN*MM),WK(NN*MM)
DIMENSION IFAX(20)
DIMENSION TRIGS(2*NN)
!**** INPUT ****
READ(5,*) N,M,INCN,INCM
WRITE(6,1000) N,M,INCN,INCM
DO 20 J=1,M
  DO 10 I=1,N
    READ(5,*) C(1+(I-1)*INCN+(J-1)*INCM)
  10 CONTINUE
20 CONTINUE
WRITE(6,2000) 'REAL PART'
WRITE(6,2010) ((DBLE(C(1+(I-1)*INCN+(J-1)*INCM)),I=1,N),J=1,M)
WRITE(6,2000) 'IMAGINARY PART'
WRITE(6,2010) ((DIMAG(C(1+(I-1)*INCN+(J-1)*INCM)),I=1,N),J=1,M)
30 CONTINUE
!**** OUTPUT ****
WRITE(6,1010)
!**** FORWARD TRANSFORM ****
ISW = 1
CALL ZFCMBF(N,M,C,INCN,INCM,ISW,IFAX,TRIGS,WK,IERR)
!**** NORMALIZATION ****
DO 50 J=1,M
  DO 40 I=1,N
    C(1+(I-1)*INCN+(J-1)*INCM)=&
    C(1+(I-1)*INCN+(J-1)*INCM)/DBLE(N)
  40 CONTINUE
50 CONTINUE
WRITE(6,1020) IERR
WRITE(6,2000) 'REAL PART'
WRITE(6,2010) ((DBLE(C(1+(I-1)*INCN+(J-1)*INCM)),I=1,N),J=1,M)
WRITE(6,2000) 'IMAGINARY PART'
WRITE(6,2010) ((DIMAG(C(1+(I-1)*INCN+(J-1)*INCM)),I=1,N),J=1,M)
!**** BACKWARD TRANSFORM ****
ISW = -1
CALL ZFCMBF(N,M,C,INCN,INCM,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1030) IERR
WRITE(6,2000) 'REAL PART'
WRITE(6,2010) ((DBLE(C(1+(I-1)*INCN+(J-1)*INCM)),I=1,N),J=1,M)
WRITE(6,2000) 'IMAGINARY PART'
WRITE(6,2010) ((DIMAG(C(1+(I-1)*INCN+(J-1)*INCM)),I=1,N),J=1,M)
STOP
!**** FORMAT ****
1000 FORMAT(1X,'*** ZFCMBF AND ZFCMBF ***',/,/,&
1X,' ** INPUT **',/,/,&
1X,' N = ',I3,', M = ',I3,/,&
1X,' INCN = ',I3,', INCM = ',I3,/)
1010 FORMAT(1X,/,&
1X,' ** OUTPUT **',/)
1020 FORMAT(1X,' ( FORWARD TRANSFORM )',/,/,&
1X,' IERR = ',I4,/)
1030 FORMAT(1X,' ( BACKWARD TRANSFORM )',/,/,&
1X,' IERR = ',I4,/)
2000 FORMAT(1X,4X,A)
2010 FORMAT(1X,4X,8F8.4,/)
END

```

(d) 出力結果

\*\*\* ZFCMBF AND ZFCMBF \*\*\*

\*\* INPUT \*\*

N = 8 M = 4  
INCN = 1 INCM = 9

REAL PART

1.0000	2.0000	3.0000	4.0000	4.0000	3.0000	2.0000	1.0000
1.0000	1.0000	2.0000	2.0000	2.0000	2.0000	1.0000	1.0000
1.0000	1.0000	1.0000	1.0000	2.0000	2.0000	2.0000	2.0000
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

IMAGINARY PART

4.0000	3.0000	2.0000	1.0000	1.0000	2.0000	3.0000	4.0000
2.0000	2.0000	1.0000	1.0000	1.0000	1.0000	2.0000	2.0000
2.0000	2.0000	2.0000	2.0000	1.0000	1.0000	1.0000	1.0000
1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

REAL PART

2.5000 -1.0303 0.0000 -0.0732 0.0000 0.0303 0.0000 -0.4268

1.5000 -0.4268 0.0000 0.1768 0.0000 -0.0732 0.0000 -0.1768

1.5000 0.1768 0.0000 -0.0732 0.0000 -0.1768 0.0000 -0.4268

1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

IMAGINARY PART

2.5000 0.4268 0.0000 -0.0303 0.0000 0.0732 0.0000 1.0303

1.5000 0.1768 0.0000 0.0732 0.0000 -0.1768 0.0000 0.4268

1.5000 0.4268 0.0000 0.1768 0.0000 0.0732 0.0000 -0.1768

1.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

( BACKWARD TRANSFORM )

IERR = 0

REAL PART

1.0000 2.0000 3.0000 4.0000 4.0000 3.0000 2.0000 1.0000

1.0000 1.0000 2.0000 2.0000 2.0000 2.0000 1.0000 1.0000

1.0000 1.0000 1.0000 1.0000 2.0000 2.0000 2.0000 2.0000

1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000

IMAGINARY PART

4.0000 3.0000 2.0000 1.0000 1.0000 2.0000 3.0000 4.0000

2.0000 2.0000 1.0000 1.0000 1.0000 1.0000 2.0000 2.0000

2.0000 2.0000 2.0000 2.0000 1.0000 1.0000 1.0000 1.0000

1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000

## 2.7 多重1次元実フーリエ変換

### 2.7.1 [非推奨]DFRMFB, RFRMFB

#### 多重1次元実フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

実数データ  $r_{k,l}$  ( $k = 0, \dots, n-1$ ;  $l = 1, \dots, m$ ) に対して,  $m$  重1次元フーリエ順変換 (任意基数) の半周期分を求める.

$$c_{j,l} = \sum_{k=0}^{n-1} r_{k,l} e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, \lfloor \frac{n}{2} \rfloor; l = 1, \dots, m)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す. なお, 残りの半周期分は以下の関係から得られる.

$$c_{n-j,l}^* = c_{j,l}$$

ただし,  $z^*$  は複素数  $z$  の共役複素数を表す.

###### 逆変換

$c_{n-j,l}^* = c_{j,l}$  を満たす  $n$  個の複素数データの組み  $c_{j,l}$  ( $j = 0, \dots, n-1$ ;  $l = 1, \dots, m$ ) についてその半周期分  $c_{j,l}$  ( $j = 0, \dots, \lfloor \frac{n}{2} \rfloor$ ;  $l = 1, \dots, m$ ) を与えて以下のように定義される  $m$  重1次元フーリエ逆変換 (任意基数) を求める.

$$\begin{aligned} r_{k,l} &= \sum_{j=0}^{n-1} c_{j,l} e^{2\pi\sqrt{-1}\frac{jk}{n}} \\ &= c_{0,l} + (-1)^k \hat{c}_{\frac{n}{2},l} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \Re\{c_{j,l} e^{2\pi\sqrt{-1}\frac{jk}{n}}\} \\ &= c_{0,l} + (-1)^k \hat{c}_{\frac{n}{2},l} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \left[ \Re\{c_{j,l}\} \cos(2\pi\frac{jk}{n}) - \Im\{c_{j,l}\} \sin(2\pi\frac{jk}{n}) \right] \\ &\quad (k = 0, \dots, n-1; l = 1, \dots, m) \end{aligned}$$

ここで  $\lfloor x \rfloor$  は  $x$  以上の最小の整数を,  $\Re\{z\}$  と  $\Im\{z\}$  はそれぞれ複素数  $z$  の実部と虚部を表す. また,  $n$  が奇数のとき  $\hat{c}_{\frac{n}{2},l} = 0$ ,  $n$  が偶数のとき  $\hat{c}_{\frac{n}{2},l} = c_{\frac{n}{2},l}$  である.

##### (2) 使用法

###### 倍精度サブルーチン:

CALL DFRMFB (N, M, R, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL RFRMFB (N, M, R, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	変換データ数 $n$ (注意事項 (a) 参照)
2	M	I	1	入 力	多重度 $m$
3	R	$\begin{cases} D \\ R \end{cases}$	内容参照	入 力	入力データ $r_{k,l}$ (順変換), または $c_{j,l}$ (逆変換) (注意事項 (b) 参照) 大きさ: N が奇数の時, $INCN \times (N) + INCM \times (M - 1) + 1$ N が偶数の時, $INCN \times (N + 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $c_{j,l}$ (順変換), または $r_{k,l}$ (逆変換) (注意事項 (b)(c) 参照)
4	INCN	I	1	入 力	変換データの格納間隔 (注意事項 (b) 参照)
5	INCM	I	1	入 力	変換データ間の格納間隔 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含む順変換 ISW = -1:初期化を含む逆変換
7	IFAX	I	20	出 力	基数分け情報 (注意事項 (d) 参照)
8	TRIGS	$\begin{cases} D \\ R \end{cases}$	N	出 力	三角関数テーブル (注意事項 (d) 参照)
9	WK	$\begin{cases} D \\ R \end{cases}$	内容参照	ワーク	作業領域 大きさ: N が奇数の時, 大きさ $(N + 1) \times M$ N が偶数の時, 大きさ $(N + 2) \times M$
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$   
 $M > 0$
- (b)  $INCN > 0$   
 $INCM > 0$
- (c)  $INCN \geq M \times \text{gcm}(INCN, INCM)$  または  
N が奇数の時:  
 $INCM \geq (N + 1) \times \text{gcm}(INCN, INCM)$   
N が偶数の時:  
 $INCM \geq (N + 2) \times \text{gcm}(INCN, INCM)$   
(ただし,  $\text{gcm}(i, j)$  は  $i, j$  の最大公約数を表す.)
- (d)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

(6) 注意事項

- (a) データ数 N の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、N = 289(=17<sup>2</sup>) とするよりも N = 300(=2<sup>2</sup> × 3 × 5<sup>2</sup>) や 320(=2<sup>6</sup> × 5), 384(=2<sup>7</sup> × 3) などとした方が通常は効率が良い。
- (b) 実数データ  $r_{k,l}$  ( $k = 0, \dots, n - 1; l = 1, \dots, m$ ) と配列 R の各要素は以下の様に対応する。

$$r_{k,l} \leftrightarrow R(1 + \text{INCN} * k + \text{INCM} * (l - 1))$$

例えば、INCN=1, INCM=n とすると、

$$r_{k,l} \leftrightarrow R((k + 1) + n * (l - 1))$$

となり、添え字 k について連続に詰めて格納することになり INCN=m, INCM=1 とすると、

$$r_{k,l} \leftrightarrow R(l + m * k)$$

となり、添え字 l について連続に詰めて格納することになる。なお、逆変換を行った場合、N(=n) が奇数のとき  $R(1 + \text{INCN} * N + \text{INCM} * (l - 1)) = 0$ 、N が偶数のとき  $R(1 + \text{INCN} * N + \text{INCM} * (l - 1)) = R(1 + \text{INCN} * (N + 1) + \text{INCM} * (l - 1)) = 0$  となる。複素数データ  $c_{j,l}$  ( $j = 0, \dots, \lfloor \frac{n}{2} \rfloor; l = 1, \dots, m$ ) の実部と虚部をそれぞれ  $\Re\{c_{j,l}\}, \Im\{c_{j,l}\}$  とすると、 $c_{j,l}$  と配列 R の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は x を超えない最大の整数を表す。

$$\Re\{c_{j,l}\} \leftrightarrow R(1 + \text{INCN} * (2j) + \text{INCM} * (l - 1))$$

$$\Im\{c_{j,l}\} \leftrightarrow R(1 + \text{INCN} * (2j + 1) + \text{INCM} * (l - 1))$$

実フーリエ変換の性質より、N が奇数のとき  $\Im\{c_{0,l}\} = 0$ 、N が偶数のとき  $\Im\{c_{0,l}\} = \Im\{c_{\frac{n}{2},l}\} = 0$  である。したがって、配列 R の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_{j,l}$  ( $j = \lfloor \frac{n}{2} \rfloor + 1, \dots, n - 1; l = 1, \dots, m$ ) の各要素は実フーリエ変換の対称性から以下の関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n-j,l} = c_{j,l}^*$$

ただし、 $z^*$  は複素数 z の共役複素数を表す。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_{k,l}$  ( $k = 0, \dots, n - 1; l = 1, \dots, m$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_{k,l}$  ( $k = 0, \dots, n - 1; l = 1, \dots, m$ ) とすると

$$\hat{r}_{k,l} = nr_{k,l} \quad (k = 0, \dots, n - 1; l = 1, \dots, m)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じ変換データ数  $N$  の変換を繰り返し行う場合、一度このサブルーチン呼びその後は初期化後の変換 2.7.2  $\left\{ \begin{array}{l} \text{DFRMBF} \\ \text{RFRMBF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.7.2  $\left\{ \begin{array}{l} \text{DFRMBF} \\ \text{RFRMBF} \end{array} \right\}$  の入力としなければならない。
- なお、ISW=0 として初期化だけを行う場合には、配列 R に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (f) この機能は逐次版および OpenMP 不適用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.7.2 (7) 使用例参照。



## 2.7.2 [非推奨]DFRMBF, RFRMBF

## 多重 1 次元実フーリエ変換 (初期化後の変換)

## (1) 機能

## 順変換

実数データ  $r_{k,l}$  ( $k = 0, \dots, n-1$ ;  $l = 1, \dots, m$ ) に対して,  $m$  重 1 次元フーリエ順変換 (任意基数) の半周期分を求める.

$$c_{j,l} = \sum_{k=0}^{n-1} r_{k,l} e^{-2\pi\sqrt{-1}\frac{jk}{n}} \quad (j = 0, \dots, \lfloor \frac{n}{2} \rfloor; l = 1, \dots, m)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す. なお, 残りの半周期分は以下の関係から得られる.

$$c_{n-j,l}^* = c_{j,l}$$

ただし,  $z^*$  は複素数  $z$  の共役複素数を表す.

## 逆変換

$c_{n-j,l}^* = c_{j,l}$  を満たす  $n$  個の複素数データの組み  $c_{j,l}$  ( $j = 0, \dots, n-1$ ;  $l = 1, \dots, m$ ) についてその半周期分  $c_{j,l}$  ( $j = 0, \dots, \lfloor \frac{n}{2} \rfloor$ ;  $l = 1, \dots, m$ ) を与えて以下のように定義される  $m$  重 1 次元フーリエ逆変換 (任意基数) を求める.

$$\begin{aligned} r_{k,l} &= \sum_{j=0}^{n-1} c_{j,l} e^{2\pi\sqrt{-1}\frac{jk}{n}} \\ &= c_{0,l} + (-1)^k \hat{c}_{\frac{n}{2},l} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \Re\{c_{j,l} e^{2\pi\sqrt{-1}\frac{jk}{n}}\} \\ &= c_{0,l} + (-1)^k \hat{c}_{\frac{n}{2},l} + 2 \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor - 1} \left[ \Re\{c_{j,l}\} \cos(2\pi\frac{jk}{n}) - \Im\{c_{j,l}\} \sin(2\pi\frac{jk}{n}) \right] \\ &\quad (k = 0, \dots, n-1; l = 1, \dots, m) \end{aligned}$$

ここで  $\lfloor x \rfloor$  は  $x$  以上の最小の整数を,  $\Re\{z\}$  と  $\Im\{z\}$  はそれぞれ複素数  $z$  の実部と虚部を表す. また,  $n$  が奇数のとき  $\hat{c}_{\frac{n}{2},l} = 0$ ,  $n$  が偶数のとき  $\hat{c}_{\frac{n}{2},l} = c_{\frac{n}{2},l}$  である.

## (2) 使用法

## 倍精度サブルーチン:

CALL DFRMBF (N, M, R, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

## 単精度サブルーチン:

CALL RFRMBF (N, M, R, INCN, INCM, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	変換データ数 $n$ (注意事項 (a) 参照)
2	M	I	1	入 力	多重度 $m$
3	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	入 力	入力データ $r_{k,l}$ (順変換), または $c_{j,l}$ (逆変換) (注意事項 (b) 参照). 大きさ: N が奇数の時, $INCN \times (N) + INCM \times (M - 1) + 1$ N が偶数の時, $INCN \times (N + 1) + INCM \times (M - 1) + 1$
				出 力	出力データ $c_{j,l}$ (順変換), または $r_{k,l}$ (逆変換) (注意事項 (b)(c) 参照)
4	INCN	I	1	入 力	変換データの格納間隔 (注意事項 (b) 参照)
5	INCM	I	1	入 力	変換データ間の格納間隔 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ ISW = 1:初期化後の順変換 ISW = -1:初期化後の逆変換
7	IFAX	I	20	入 力	基数分け情報 (注意事項 (a) 参照)
8	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	入 力	三角関数テーブル (注意事項 (a) 参照)
9	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 大きさ: N が奇数の時, 大きさ $(N + 1) \times M$ N が偶数の時, 大きさ $(N + 2) \times M$
10	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$   
 $M > 0$
- (b)  $INCN > 0$   
 $INCM > 0$
- (c)  $INCN \geq M \times \text{gcm}(INCN, INCM)$  または  
N が奇数の時:  
 $INCM \geq (N + 1) \times \text{gcm}(INCN, INCM)$   
N が偶数の時:  
 $INCM \geq (N + 2) \times \text{gcm}(INCN, INCM)$   
(ただし,  $\text{gcm}(i, j)$  は  $i, j$  の最大公約数を表す.)
- (d)  $ISW \in \{1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	N = 1 であった.	入力時の情報がそのまま出力される.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンは、同じ変換データ数  $N$  の変換を繰り返し行う場合に初期化を含む変換  
2.7.1  $\left\{ \begin{array}{l} \text{DFRMBF} \\ \text{RFRMBF} \end{array} \right\}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

- (b) 実数データ  $r_{k,l} (k = 0, \dots, n-1; l = 1, \dots, m)$  と配列 R の各要素は以下の様に対応する。

$$r_{k,l} \leftrightarrow R(1 + \text{INCN} * k + \text{INCM} * (l - 1))$$

例えば、 $\text{INCN}=1, \text{INCM}=n$  とすると、

$$r_{k,l} \leftrightarrow R((k + 1) + n * (l - 1))$$

となり、添え字  $k$  について連続に詰めて格納することになり  $\text{INCN}=m, \text{INCM}=1$  とすると、

$$r_{k,l} \leftrightarrow R(l + m * k)$$

となり、添え字  $l$  について連続に詰めて格納することになる。なお、逆変換を行った場合、 $N(=n)$  が奇数のとき  $R(1 + \text{INCN} * N + \text{INCM} * (l - 1)) = 0$ 、 $N$  が偶数のとき  $R(1 + \text{INCN} * N + \text{INCM} * (l - 1)) = R(1 + \text{INCN} * (N + 1) + \text{INCM} * (l - 1)) = 0$  となる。複素数データ  $c_{j,l} (j = 0, \dots, \lfloor \frac{n}{2} \rfloor; l = 1, \dots, m)$  の実部と虚部をそれぞれ  $\Re\{c_{j,l}\}, \Im\{c_{j,l}\}$  とすると、 $c_{j,l}$  と配列 R の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

$$\Re\{c_{j,l}\} \leftrightarrow R(1 + \text{INCN} * (2j) + \text{INCM} * (l - 1))$$

$$\Im\{c_{j,l}\} \leftrightarrow R(1 + \text{INCN} * (2j + 1) + \text{INCM} * (l - 1))$$

実フーリエ変換の性質より、 $N$  が奇数のとき  $\Im\{c_{0,l}\} = 0$ 、 $N$  が偶数のとき  $\Im\{c_{0,l}\} = \Im\{c_{\frac{n}{2},l}\} = 0$  である。したがって、配列 R の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_{j,l} (j = \lfloor \frac{n}{2} \rfloor + 1, \dots, n-1; l = 1, \dots, m)$  の各要素は実フーリエ変換の対称性から以下の関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n-j,l} = c_{j,l}^*$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_{k,l} (k = 0, \dots, n-1; l = 1, \dots, m)$  に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_{k,l} (k = 0, \dots, n-1; l = 1, \dots, m)$  とすると

$$\hat{r}_{k,l} = nr_{k,l} \quad (k = 0, \dots, n-1; l = 1, \dots, m)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標本数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (e) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。  
 (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

### (a) 問題

R ( 1 ) = 1.000 R ( 2 ) = 2.000 R ( 3 ) = 3.000 R ( 4 ) = 4.000  
 R ( 5 ) = 5.000 R ( 6 ) = 6.000 R ( 7 ) = 7.000 R ( 8 ) = 8.000  
 R (13) = 1.000 R (14) = 1.000 R (15) = 2.000 R (16) = 2.000  
 R (17) = 3.000 R (18) = 3.000 R (19) = 4.000 R (20) = 4.000  
 R (25) = 1.000 R (26) = 1.000 R (27) = 1.000 R (28) = 1.000  
 R (29) = 2.000 R (30) = 2.000 R (31) = 2.000 R (32) = 2.000  
 R (37) = 1.000 R (38) = 1.000 R (39) = 1.000 R (40) = 1.000  
 R (41) = 1.000 R (42) = 1.000 R (43) = 1.000 R (44) = 1.000  
 上記の数列を入力データとして、多重 1 次元実フーリエ順・逆変換を行う。

### (b) 入力データ

配列 R, N=8, M=4, INCN=1, INCM=12, ISW=1 (順変換) および ISW=-1 (逆変換)

### (c) 主プログラム

```

PROGRAM BFRMBF
! *** EXAMPLE OF DFRMBF AND DFRMBF ***
IMPLICIT REAL(8) (A-H,O-Z)
PARAMETER (NN=12,MM=4)
DIMENSION R(NN*MM), TRIGS(NN), WK(NN*MM)
DIMENSION IFAX(20)
!**** INPUT ****
READ(5,*) N,M,INCN,INCM
WRITE(6,1000) N,M,INCN,INCM
DO 20 J=1,M
    DO 10 I=1,N
        READ(5,*) R(1+(I-1)*INCN+(J-1)*INCM)
    10 CONTINUE
    20 CONTINUE
WRITE(6,2000) 'REAL PART'
WRITE(6,2010) ((R(1+(I-1)*INCN+(J-1)*INCM), I=1,N), J=1,M)
30 CONTINUE
!**** OUTPUT ****
WRITE(6,1010)
!**** FORWARD TRANSFORM ****
ISW = 1
CALL DFRMBF(N,M,R,INCN,INCM,ISW,IFAX,TRIGS,WK,IERR)
!**** NORMALIZATION ****
DO 50 J=1,M
    DO 40 I=1,N+2
        R(1+(I-1)*INCN+(J-1)*INCM)=&
            R(1+(I-1)*INCN+(J-1)*INCM)/DBLE(N)
    40 CONTINUE
    50 CONTINUE
WRITE(6,1020) IERR
WRITE(6,2000) 'REAL PART'
WRITE(6,2020) ((R(1+(2*I-2)*INCN+(J-1)*INCM), &
    I=1,(N+2)/2), J=1,M)
WRITE(6,2000) 'IMAGINARY PART'
WRITE(6,2020) ((R(1+(2*I-1)*INCN+(J-1)*INCM), &
    I=1,(N+2)/2), J=1,M)
!**** BACKWARD TRANSFORM ****
ISW = -1
CALL DFRMBF(N,M,R,INCN,INCM,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1030) IERR
WRITE(6,2000) 'REAL PART'
WRITE(6,2010) ((R(1+(I-1)*INCN+(J-1)*INCM), I=1,N), J=1,M)
STOP
    
```

```

!**** FORMAT ****
1000 FORMAT(1X,'*** DFRMBF AND DFRMBF ***',/,/,&
1X,' ** INPUT **',/,/,&
1X,'      N = ',I3,',      M = ',I3,/,&
1X,'      INCN = ',I3,',      INCM = ',I3,/)
1010 FORMAT(1X,/,&
1X,' ** OUTPUT **',/)
1020 FORMAT(1X,' ( FORWARD TRANSFORM )',/,/,&
1X,'      IERR = ',I4,/)
1030 FORMAT(1X,' ( BACKWARD TRANSFORM )',/,/,&
1X,'      IERR = ',I4,/)
2000 FORMAT(1X,4X,A)
2010 FORMAT(1X,4X,8F8.4,/)
2020 FORMAT(1X,4X,5F8.4,/)
      END

```

## (d) 出力結果

```

*** DFRMBF AND DFRMBF ***
** INPUT **
      N = 8      M = 4
      INCN = 1   INCM = 12
REAL PART
  1.0000  2.0000  3.0000  4.0000  5.0000  6.0000  7.0000  8.0000
  1.0000  1.0000  2.0000  2.0000  3.0000  3.0000  4.0000  4.0000
  1.0000  1.0000  1.0000  1.0000  2.0000  2.0000  2.0000  2.0000
  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000
** OUTPUT **
( FORWARD TRANSFORM )
      IERR = 0
REAL PART
  4.5000 -0.5000 -0.5000 -0.5000 -0.5000
  2.5000 -0.2500 -0.2500 -0.2500  0.0000
  1.5000 -0.1250  0.0000 -0.1250  0.0000
  1.0000  0.0000  0.0000  0.0000  0.0000
IMAGINARY PART
  0.0000  1.2071  0.5000  0.2071  0.0000
  0.0000  0.6036  0.2500  0.1036  0.0000
  0.0000  0.3018  0.0000  0.0518  0.0000
  0.0000 -0.0000  0.0000  0.0000  0.0000
( BACKWARD TRANSFORM )
      IERR = 0
REAL PART
  1.0000  2.0000  3.0000  4.0000  5.0000  6.0000  7.0000  8.0000
  1.0000  1.0000  2.0000  2.0000  3.0000  3.0000  4.0000  4.0000
  1.0000  1.0000  1.0000  1.0000  2.0000  2.0000  2.0000  2.0000
  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000  1.0000

```

---

## 2.8 2次元複素フーリエ変換 (実数引数型)

### 2.8.1 [非推奨]DFC2FB, RFC2FB

#### 2次元複素フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して, 2次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

###### 逆変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して, 2次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

##### (2) 使用法

###### 倍精度サブルーチン:

CALL DFC2FB (NX, NY, CR, CI, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL RFC2FB (NX, NY, CR, CI, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	CR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY	入 力	入力データ $c_{k_x, k_y}$ の実部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y}$ の実部 (注意事項 (b), (c) 参照)
4	CI	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY	入 力	入力データ $c_{k_x, k_y}$ の虚部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y}$ の虚部 (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 CR, CI の整合寸法 (注意事項 (b) 参照)
6	LY	I	1	入 力	配列 CR, CI の第 2 寸法 (注意事項 (b) 参照)
7	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW= 0:初期化のみ ISW= 1:初期化を含む順変換 ISW=-1:初期化を含む逆変換
8	IFAX	I	40	出 力	基数分け情報 (注意事項 (d) 参照)
9	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (NX + NY)$	出 力	三角関数テーブル (注意事項 (d) 参照)
10	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times LX \times LY$	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 (b)  $NX \leq LX$   
 $NY \leq LY$   
 (c)  $ISW \in \{0, 1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) データ数 NX や NY の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $NX = 289 (=17^2)$  とするよりも  $NX = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い。
- (b) 複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{k_x, k_y}\}$ ,  $\Im\{c_{k_x, k_y}\}$  とすると、 $c_{k_x, k_y}$  と配列 CR, CI の各要素は以下の様に対応する。

$$\begin{aligned}\Re\{c_{k_x, k_y}\} &\leftrightarrow \text{CR}(k_x + 1, k_y + 1) \\ \Im\{c_{k_x, k_y}\} &\leftrightarrow \text{CI}(k_x + 1, k_y + 1)\end{aligned}$$

複素数データ  $d_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) についても同様である。

なお、主記憶のバンク競合を避けるために配列 CR, CI の整合寸法 LX, LY は奇数に設定するのが望ましい。通常、たとえば NX が偶数の時は  $LX = NX + 1$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) とすると

$$\hat{c}_{k_x, k_y} = n_x n_y c_{k_x, k_y} \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数 (NX, NY) の変換を繰り返し行う場合、一度このサブルーチン呼びその後は初期化後の変換 2.8.2  $\left\{ \begin{array}{l} \text{DFC2BF} \\ \text{RFC2BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.8.2  $\left\{ \begin{array}{l} \text{DFC2BF} \\ \text{RFC2BF} \end{array} \right\}$  の入力としなければならない。
- なお、ISW=0 として初期化だけを行う場合には、配列 CR, CI に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (f) この機能は逐次版および OpenMP 不適用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

(7) 使用例

2.8.2 (7) 使用例参照。



## 2.8.2 [非推奨]DFC2BF, RFC2BF

## 2次元複素フーリエ変換 (初期化後の変換)

## (1) 機能

## 順変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して、2次元複素フーリエ順変換 (任意基数) を行う。

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

## 逆変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して、2次元複素フーリエ逆変換 (任意基数) を行う。

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

## (2) 使用法

## 倍精度サブルーチン:

CALL DFC2BF (NX, NY, CR, CI, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

## 単精度サブルーチン:

CALL RFC2BF (NX, NY, CR, CI, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{array} \right\}$   
 R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	CR	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX, LY	入 力	入力データ $c_{k_x, k_y}$ の実部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y}$ の実部 (注意事項 (b), (c) 参照)
4	CI	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX, LY	入 力	入力データ $c_{k_x, k_y}$ の虚部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y}$ の虚部 (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 CR, CI の整合寸法 (注意事項 (b) 参照)
6	LY	I	1	入 力	配列 CR, CI の第2寸法 (注意事項 (b) 参照)
7	ISW	I	1	入 力	処理スイッチ ISW= 1:初期化後の順変換 ISW=-1:初期化後の逆変換
8	IFAX	I	40	入 力	基数分け情報 (注意事項 (a) 参照)
9	TRIGS	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times (NX + NY)$	入 力	三角関数テーブル (注意事項 (a) 参照)
10	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times LX \times LY$	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$
- (c)  $ISW \in \{1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) このサブルーチンは、同じデータ数 (NX, NY) の変換を繰り返す行う場合に初期化を含む変換 2.8.1  $\left\{ \begin{matrix} \text{DFC2FB} \\ \text{RFC2FB} \end{matrix} \right\}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。
- (b) 複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{k_x, k_y}\}$ ,  $\Im\{c_{k_x, k_y}\}$  とすると、 $c_{k_x, k_y}$  と配列 CR, CI の各要素は以下の様に対応する。

$$\begin{aligned} \Re\{c_{k_x, k_y}\} &\leftrightarrow \text{CR}(k_x + 1, k_y + 1) \\ \Im\{c_{k_x, k_y}\} &\leftrightarrow \text{CI}(k_x + 1, k_y + 1) \end{aligned}$$

複素数データ  $d_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) についても同様である。

なお、主記憶のバンク競合を避けるために配列 CR, CI の整合寸法 LX, LY は奇数に設定するのが望ましい。通常、たとえば NX が偶数の時は  $LX = NX + 1$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) とすると

$$\hat{c}_{k_x, k_y} = n_x n_y c_{k_x, k_y} \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi(t - iT)}$$

- (e) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

## (a) 問題

$$c_{k_x, k_y} = (k_x + 1) + (k_y + 1) + \sqrt{-1} \frac{(k_x + 1)(k_y + 1)}{n_x n_y}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

を入力データとして、2次元複素フーリエ順・逆変換を行う。

## (b) 入力データ

配列 CR, CI, NX=5, NY=4, LX=5, LY=5, ISW=1(順変換) および ISW=-1(逆変換)

## (c) 主プログラム

```

PROGRAM BFC2BF
! *** EXAMPLE OF DFC2FB AND DFC2BF ***
PARAMETER (NX=5, NY=4, LX=5, LY=5)
REAL(8) CR(LX, LY), CI(LX, LY)
REAL(8) TRIGS(2*(NX+NY)), WK(2*LX*LY)
INTEGER IFAX(40)
!**** INPUT ****
DO 20 J=1, NY
  DO 10 I=1, NX
    CR(I, J) = DBLE(I+J)
    CI(I, J) = DBLE(I*J)/(NX*NY)
  10 CONTINUE
20 CONTINUE
WRITE(6, 1000)
WRITE(6, 1010) NX, NY, LX, LY
WRITE(6, 1020)
WRITE(6, 1030) ((CR(I, J), CI(I, J), J=1, NY), I=1, NX)
!**** OUTPUT ****
WRITE(6, 1040)
!**** INITIALIZATION + FORWARD TRANSFORM ****
ISW= 1
CALL DFC2FB(NX, NY, CR, CI, LX, LY, ISW, IFAX, TRIGS, WK, IERR)
!**** NORMALIZATION ****
DO 40 J=1, NY
  DO 30 I=1, NX
    CR(I, J)=CR(I, J)/DBLE(NX*NY)
    CI(I, J)=CI(I, J)/DBLE(NX*NY)
  30 CONTINUE
40 CONTINUE
WRITE(6, 1050) IERR
WRITE(6, 1020)
WRITE(6, 1030) ((CR(I, J), CI(I, J), J=1, NY), I=1, NX)
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL DFC2BF(NX, NY, CR, CI, LX, LY, ISW, IFAX, TRIGS, WK, IERR)
WRITE(6, 1060) IERR
WRITE(6, 1020)
WRITE(6, 1030) ((CR(I, J), CI(I, J), J=1, NY), I=1, NX)
STOP
!**** FORMAT ****
1000 FORMAT(1X, '*** DFC2FB AND DFC2BF ***', /, /, &
1X, ' ** INPUT **', /)
1010 FORMAT(1X, '  NX =', I3, '  NY =', I3, /, &
1X, '  LX =', I3, '  LY =', I3, /)
1020 FORMAT(1X, ' ( CR(IX, IY) , CI(IX, IY) )')
1030 FORMAT(4(1X, ' (', F6.3, ', ', F6.3, ')')')
1040 FORMAT(/, 1X, ' ** OUTPUT **')
1050 FORMAT(/, 1X, ' ( FORWARD TRANSFORM )', /, &
/, 1X, ' IERR =', I4, /)
1060 FORMAT(/, 1X, ' ( BACKWARD TRANSFORM )', /, &
/, 1X, ' IERR =', I4, /)
END

```

## (d) 出力結果

```

*** DFC2FB AND DFC2BF ***
** INPUT **
  NX = 5  NY = 4
  LX = 5  LY = 5
( CR(IX, IY) , CI(IX, IY) )
( 2.000, 0.050) ( 3.000, 0.100) ( 4.000, 0.150) ( 5.000, 0.200)
( 3.000, 0.100) ( 4.000, 0.200) ( 5.000, 0.300) ( 6.000, 0.400)
( 4.000, 0.150) ( 5.000, 0.300) ( 6.000, 0.450) ( 7.000, 0.600)
( 5.000, 0.200) ( 6.000, 0.400) ( 7.000, 0.600) ( 8.000, 0.800)
( 6.000, 0.250) ( 7.000, 0.500) ( 8.000, 0.750) ( 9.000, 1.000)

```

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

( CR(IX,IY) , CI(IX,IY) )

( 5.500, 0.375)	( -0.575, 0.425)	( -0.500, -0.075)	( -0.425, -0.575)
( -0.586, 0.626)	( 0.030, -0.005)	( 0.017, 0.013)	( 0.005, 0.030)
( -0.520, 0.100)	( 0.017, 0.008)	( 0.004, 0.012)	( -0.008, 0.017)
( -0.480, -0.225)	( 0.008, 0.017)	( -0.004, 0.012)	( -0.017, 0.008)
( -0.414, -0.751)	( -0.005, 0.030)	( -0.017, 0.013)	( -0.030, -0.005)

( BACKWARD TRANSFORM )

IERR = 0

( CR(IX,IY) , CI(IX,IY) )

( 2.000, 0.050)	( 3.000, 0.100)	( 4.000, 0.150)	( 5.000, 0.200)
( 3.000, 0.100)	( 4.000, 0.200)	( 5.000, 0.300)	( 6.000, 0.400)
( 4.000, 0.150)	( 5.000, 0.300)	( 6.000, 0.450)	( 7.000, 0.600)
( 5.000, 0.200)	( 6.000, 0.400)	( 7.000, 0.600)	( 8.000, 0.800)
( 6.000, 0.250)	( 7.000, 0.500)	( 8.000, 0.750)	( 9.000, 1.000)

## 2.9 2次元複素フーリエ変換 (複素指数型)

### 2.9.1 [非推奨]ZFC2FB, CFC2FB

#### 2次元複素フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して, 2次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

###### 逆変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して, 2次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

##### (2) 使用法

###### 倍精度サブルーチン:

CALL ZFC2FB (NX, NY, C, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL CFC2FB (NX, NY, C, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

##### (3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX, LY	入力	入力データ $c_{k_x, k_y}$ (注意事項 (b) 参照)
				出力	出力データ $d_{j_x, j_y}$ (注意事項 (b), (c) 参照)
4	LX	I	1	入力	配列 C の整合寸法 (注意事項 (b) 参照)
5	LY	I	1	入力	配列 C の第 2 寸法 (注意事項 (b) 参照)
6	ISW	I	1	入力	処理スイッチ (注意事項 (d) 参照) ISW= 0:初期化のみ ISW= 1:初期化を含む順変換 ISW=-1:初期化を含む逆変換
7	IFAX	I	40	出力	基数分け情報 (注意事項 (d) 参照)
8	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (NX + NY)$	出力	三角関数テーブル (注意事項 (d) 参照)
9	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX $\times$ LY	ワーク	作業領域
10	IERR	I	1	出力	エラーインディケータ

(4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$
- (c)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) データ数  $NX$  や  $NY$  の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $NX = 289 (=17^2)$  とするよりも  $NX = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い。

- (b) 複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) と配列  $C$  の各要素は以下の様に対応する。

$$c_{k_x, k_y} \leftrightarrow C(k_x + 1, k_y + 1)$$

複素数データ  $d_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) についても同様である。

なお、主記憶のバンク競合を避けるために配列  $C$  の整合寸法  $LX$ ,  $LY$  は奇数に設定するのが望ましい。通常、たとえば  $NX$  が偶数の時は  $LX = NX + 1$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) とすると

$$\hat{c}_{k_x, k_y} = n_x n_y c_{k_x, k_y} \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数 ( $NX$ ,  $NY$ ) の変換を繰り返し行う場合、一度このサブルーチン呼びその後は初期化後の変換 2.9.2  $\left\{ \begin{matrix} \text{ZFC2BF} \\ \text{CFC2BF} \end{matrix} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.9.2  $\left\{ \begin{matrix} \text{ZFC2BF} \\ \text{CFC2BF} \end{matrix} \right\}$  の入力としなければならない。

なお、 $ISW=0$  として初期化だけを行う場合には、配列  $C$  に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や

標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.9.2 (7) 使用例参照。

## 2.9.2 [非推奨]ZFC2BF, CFC2BF 2次元複素フーリエ変換 (初期化後の変換)

### (1) 機能

#### 順変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して、2次元複素フーリエ順変換 (任意基数) を行う。

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

#### 逆変換

2次元複素データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して、2次元複素フーリエ逆変換 (任意基数) を行う。

$$d_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} c_{k_x, k_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1)$$

### (2) 使用法

#### 倍精度サブルーチン:

CALL ZFC2BF (NX, NY, C, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

#### 単精度サブルーチン:

CALL CFC2BF (NX, NY, C, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

### (3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX, LY	入 力	入力データ $c_{k_x, k_y}$ (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y}$ (注意事項 (b), (c) 参照)
4	LX	I	1	入 力	配列 C の整合寸法 (注意事項 (b) 参照)
5	LY	I	1	入 力	配列 C の第 2 寸法 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ ISW= 1:初期化後の順変換 ISW=-1:初期化後の逆変換
7	IFAX	I	40	入 力	基数分け情報 (注意事項 (a) 参照)
8	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (NX + NY)$	入 力	三角関数テーブル (注意事項 (a) 参照)
9	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	$LX \times LY$	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ



## (4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$
- (c)  $ISW \in \{1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンは、同じデータ数 ( $NX, NY$ ) の変換を繰り返し行う場合に初期化を含む変換 2.9.1  $\left\{ \begin{array}{l} \text{ZFC2FB} \\ \text{CFC2FB} \end{array} \right\}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

- (b) 複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1$ ) と配列 C の各要素は以下の様に対応する。

$$c_{k_x, k_y} \leftrightarrow C(k_x + 1, k_y + 1)$$

複素数データ  $d_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1$ ) についても同様である。

なお、主記憶のバンク競合を避けるために配列 C の整合寸法 LX, LY は奇数に設定するのが望ましい。通常、たとえば NX が偶数の時は  $LX = NX + 1$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1$ ) とすると

$$\hat{c}_{k_x, k_y} = n_x n_y c_{k_x, k_y} \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (e) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。
- (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

(7) 使用例

(a) 問題

$$c_{k_x, k_y} = (k_x + 1) + (k_y + 1) + \sqrt{-1} \frac{(k_x + 1)(k_y + 1)}{n_x n_y}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

を入力データとして、2次元複素フーリエ順・逆変換を行う。

(b) 入力データ

配列 C, NX=5, NY=4, LX=5, LY=5, ISW=1(順変換) および ISW=-1(逆変換)

(c) 主プログラム

```

PROGRAM AFC2BF
! *** EXAMPLE OF ZFC2FB AND ZFC2BF ***
PARAMETER (NX=5, NY=4, LX=5, LY=5)
COMPLEX(8) C(LX,LY), WK(LX*LY)
REAL(8) TRIGS(2*(NX+NY))
INTEGER IFAX(40)
!**** INPUT ****
DO 20 J=1, NY
  DO 10 I=1, NX
    C(I, J) = CMPLX(DBLE(I+J), DBLE(I*J)/DBLE(NX*NY), KIND=8)
  10 CONTINUE
20 CONTINUE
  WRITE(6,1000)
  WRITE(6,1010) NX, NY, LX, LY
  WRITE(6,1020)
  WRITE(6,1030) ((C(I, J), J=1, NY), I=1, NX)
!**** OUTPUT ****
  WRITE(6,1040)
!**** INITIALIZATION + FORWARD TRANSFORM ****
  ISW = 1
  CALL ZFC2FB(NX, NY, C, LX, LY, ISW, IFAX, TRIGS, WK, IERR)
!**** NORMALIZATION ****
  DO 40 J=1, NY
    DO 30 I=1, NX
      C(I, J) = C(I, J) / DBLE(NX*NY)
    30 CONTINUE
  40 CONTINUE
  WRITE(6,1050) IERR
  WRITE(6,1020)
  WRITE(6,1030) ((C(I, J), J=1, NY), I=1, NX)
!**** BACKWARD TRANSFORM ****
  ISW = -1
  CALL ZFC2BF(NX, NY, C, LX, LY, ISW, IFAX, TRIGS, WK, IERR)
  WRITE(6,1060) IERR
  WRITE(6,1020)
  WRITE(6,1030) ((C(I, J), J=1, NY), I=1, NX)
  STOP
!
!**** FORMAT ****
!
1000 FORMAT(1X, '*** ZFC2FB AND ZFC2BF ***', /, /, &
1X, ' ** INPUT **', /)
1010 FORMAT(1X, '  NX =', I3, '  NY =', I3, /, &
1X, '  LX =', I3, '  LY =', I3, /)
1020 FORMAT(1X, ' C(IX, IY)', /)
1030 FORMAT(4(1X, ' (', F6.3, ', ', F6.3, ')', /)
1040 FORMAT(/, 1X, ' ** OUTPUT **')
1050 FORMAT(/, 1X, ' ( FORWARD TRANSFORM )', /, &
/, 1X, ' IERR =', I4, /)
1060 FORMAT(/, 1X, ' ( BACKWARD TRANSFORM )', /, &
/, 1X, ' IERR =', I4, /)
END

```

(d) 出力結果

```

*** ZFC2FB AND ZFC2BF ***
** INPUT **
  NX = 5  NY = 4
  LX = 5  LY = 5
C(IX, IY)
( 2.000, 0.050)  ( 3.000, 0.100)  ( 4.000, 0.150)  ( 5.000, 0.200)
( 3.000, 0.100)  ( 4.000, 0.200)  ( 5.000, 0.300)  ( 6.000, 0.400)
( 4.000, 0.150)  ( 5.000, 0.300)  ( 6.000, 0.450)  ( 7.000, 0.600)
( 5.000, 0.200)  ( 6.000, 0.400)  ( 7.000, 0.600)  ( 8.000, 0.800)
( 6.000, 0.250)  ( 7.000, 0.500)  ( 8.000, 0.750)  ( 9.000, 1.000)
** OUTPUT **
( FORWARD TRANSFORM )
  IERR = 0
C(IX, IY)
( 5.500, 0.375)  (-0.575, 0.425)  (-0.500, -0.075)  (-0.425, -0.575)
(-0.586, 0.626)  ( 0.030, -0.005)  ( 0.017, 0.013)  ( 0.005, 0.030)

```

```
(-0.520, 0.100) ( 0.017, 0.008) ( 0.004, 0.012) (-0.008, 0.017)
(-0.480,-0.225) ( 0.008, 0.017) (-0.004, 0.012) (-0.017, 0.008)
(-0.414,-0.751) (-0.005, 0.030) (-0.017, 0.013) (-0.030,-0.005)
```

( BACKWARD TRANSFORM )

IERR = 0

```
C(IX,IY)
( 2.000, 0.050) ( 3.000, 0.100) ( 4.000, 0.150) ( 5.000, 0.200)
( 3.000, 0.100) ( 4.000, 0.200) ( 5.000, 0.300) ( 6.000, 0.400)
( 4.000, 0.150) ( 5.000, 0.300) ( 6.000, 0.450) ( 7.000, 0.600)
( 5.000, 0.200) ( 6.000, 0.400) ( 7.000, 0.600) ( 8.000, 0.800)
( 6.000, 0.250) ( 7.000, 0.500) ( 8.000, 0.750) ( 9.000, 1.000)
```

## 2.10 2次元実フーリエ変換

### 2.10.1 [非推奨]DFR2FB, RFR2FB

#### 2次元実フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

2次元実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して, 2次元フーリエ順変換 (任意基数) の  $j_x$  についての半周期分を求める.

$$c_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} r_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor; j_y = 0, \dots, n_y - 1)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す. なお, 残りの半周期分は以下の関係から得られる.

$$\begin{aligned} c_{n_x - j_x, n_y - j_y}^* &= c_{j_x, j_y} \\ c_{n_x - j_x, j_y}^* &= c_{j_x, n_y - j_y} \end{aligned}$$

ただし,  $z^*$  は複素数  $z$  の共役複素数を表す.

###### 逆変換

$c_{n_x - j_x, n_y - j_y}^* = c_{j_x, j_y}$ ,  $c_{n_x - j_x, j_y}^* = c_{j_x, n_y - j_y}$  を満たす  $n_x n_y$  個の複素数データ  $c_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) について  $j_x$  についての半周期分  $c_{j_x, j_y}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ) を与えて以下のよう定義される2次元フーリエ逆変換 (任意基数) を求める.

$$\begin{aligned} r_{k_x, k_y} &= \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} c_{j_x, j_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \\ &= \sum_{j_y=0}^{n_y-1} \{c_{0, j_y} + (-1)^{k_x} \hat{c}_{\frac{n_x}{2}, j_y}\} e^{2\pi\sqrt{-1}\frac{j_y k_y}{n_y}} + 2 \sum_{j_y=0}^{n_y-1} \sum_{j_x=1}^{\lfloor \frac{n_x}{2} \rfloor - 1} \Re\{c_{j_x, j_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)}\} \\ &\quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1) \end{aligned}$$

ここで  $\lceil x \rceil$  は  $x$  以上の最小の整数を,  $\Re\{z\}$  は複素数  $z$  の実部を表す. また,  $n_x$  が奇数のとき  $\hat{c}_{\frac{n_x}{2}, j_y} = 0$ ,  $n_x$  が偶数のとき  $\hat{c}_{\frac{n_x}{2}, j_y} = c_{\frac{n_x}{2}, j_y}$  である.

##### (2) 使用法

###### 倍精度サブルーチン:

CALL DFR2FB (NX, NY, R, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL RFR2FB (NX, NY, R, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	R	$\begin{cases} D \\ R \end{cases}$	LX, LY	入 力	入力データ $r_{k_x, k_y}$ (順変換), または $c_{j_x, j_y}$ (逆変換) (注意事項 (b) 参照)
				出 力	出力データ $c_{j_x, j_y}$ (順変換), または $r_{k_x, k_y}$ (逆変換) (注意事項 (b), (c) 参照)
4	LX	I	1	入 力	配列 R の整合寸法 (注意事項 (b) 参照)
5	LY	I	1	入 力	配列 R の第 2 寸法 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含む順変換 ISW = -1:初期化を含む逆変換
7	IFAX	I	40	出 力	基数分け情報 (注意事項 (d) 参照)
8	TRIGS	$\begin{cases} D \\ R \end{cases}$	$NX + 2 \times NY$	出 力	三角関数テーブル (注意事項 (d) 参照)
9	WK	$\begin{cases} D \\ R \end{cases}$	$LX \times LY$	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$
- (b) NX が奇数の時 :  
 $NX + 1 \leq LX$   
 NX が偶数の時 :  
 $NX + 2 \leq LX$
- (c)  $NY \leq LY$
- (d)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) または (c) を満足しなかった.	
3020	制限条件 (d) を満足しなかった.	

(6) 注意事項

- (a) データ数  $NX$  や  $NY$  の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $NX = 289 (=17^2)$  とするよりも  $NX = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い。
- (b) 実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) と配列  $R$  の各要素は以下の様に対応する。

$$r_{k_x, k_y} \leftrightarrow R(k_x + 1, k_y + 1)$$

なお、逆変換を行った場合、 $NX (=n_x)$  が奇数のとき  $R(NX + 1, k_y + 1) = 0$ 、 $NX$  が偶数のとき  $R(NX + 1, k_y + 1) = R(NX + 2, k_y + 1) = 0$  となる。また、実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) を配列  $R$  に入力する場合、上述の対応する 0 を特に格納する必要はない。

複素数データ  $c_{j_x, j_y}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{j_x, j_y}\}$ ,  $\Im\{c_{j_x, j_y}\}$  とすると、 $c_{j_x, j_y}$  と配列  $R$  の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

$$\Re\{c_{j_x, j_y}\} \leftrightarrow R(2 * j_x + 1, j_y + 1)$$

$$\Im\{c_{j_x, j_y}\} \leftrightarrow R(2 * j_x + 2, j_y + 1)$$

実フーリエ変換の性質より、 $\Im\{c_{0,0}\} = 0$  であり、 $NX$  と  $NY$  が共に偶数であれば、 $\Im\{c_{\frac{n_x}{2}, \frac{n_y}{2}}\} = 0$  である。したがって、配列  $R$  の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_{j_x, j_y}$  ( $j_x = \lfloor \frac{n_x}{2} \rfloor + 1, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) の各要素は実フーリエ変換の対称性から以下の関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n_x - j_x, n_y - j_y}^* = c_{j_x, j_y}$$

$$c_{n_x - j_x, j_y}^* = c_{j_x, n_y - j_y}$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。なお、主記憶のバンク競合を避けるために配列  $R$  の整合寸法について  $LX/2$ ,  $LY$  が奇数になるように設定するのが望ましい。通常、たとえば  $NX$  が (4 の倍数)+2 の時は  $LX = NX + 4$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) とすると

$$\hat{r}_{k_x, k_y} = n_x n_y r_{k_x, k_y} \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆に行っている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数 ( $NX, NY$ ) の変換を繰り返し行う場合、一度このサブルーチン呼びその後は初期化後の変換 2.10.2  $\left\{ \begin{array}{l} \text{DFR2BF} \\ \text{RFR2BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われないため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内

容をそのまま 2.10.2  $\left\{ \begin{array}{l} \text{DFR2BF} \\ \text{RFR2BF} \end{array} \right\}$  の入力としなければならない。

なお, ISW=0 として初期化だけを行う場合には, 配列 R に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$ ) を周期とする周期関数となっていることを前提としているので, 連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお, 標本化定理によれば, 周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合, 標本化間隔を  $T = \frac{1}{2f_c}$  ととれば, 以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので, そちらを使用されたい。

## (7) 使用例

2.10.2 (7) 使用例参照。

## 2.10.2 [非推奨]DFR2BF, RFR2BF 2次元実フーリエ変換 (初期化後の変換)

### (1) 機能

#### 順変換

2次元実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して、2次元フーリエ順変換 (任意基数) の  $j_x$  についての半周期分を求める。

$$c_{j_x, j_y} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} r_{k_x, k_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \quad (j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor; j_y = 0, \dots, n_y - 1)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。なお、残りの半周期分は以下の関係から得られる。

$$\begin{aligned} c_{n_x - j_x, n_y - j_y}^* &= c_{j_x, j_y} \\ c_{n_x - j_x, j_y}^* &= c_{j_x, n_y - j_y} \end{aligned}$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。

#### 逆変換

$c_{n_x - j_x, n_y - j_y}^* = c_{j_x, j_y}$ ,  $c_{n_x - j_x, j_y}^* = c_{j_x, n_y - j_y}$  を満たす  $n_x n_y$  個の複素数データ  $c_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) について  $j_x$  についての半周期分  $c_{j_x, j_y}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ) を与えて以下のよう定義される2次元フーリエ逆変換 (任意基数) を求める。

$$\begin{aligned} r_{k_x, k_y} &= \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} c_{j_x, j_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \\ &= \sum_{j_y=0}^{n_y-1} \{c_{0, j_y} + (-1)^{k_x} \hat{c}_{\frac{n_x}{2}, j_y}\} e^{2\pi\sqrt{-1} \frac{j_y k_y}{n_y}} + 2 \sum_{j_y=0}^{n_y-1} \sum_{j_x=1}^{\lfloor \frac{n_x}{2} \rfloor - 1} \Re\{c_{j_x, j_y} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)}\} \\ &\quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1) \end{aligned}$$

ここで  $\lceil x \rceil$  は  $x$  以上の最小の整数を、 $\Re\{z\}$  は複素数  $z$  の実部を表す。また、 $n_x$  が奇数のとき  $\hat{c}_{\frac{n_x}{2}, j_y} = 0$ ,  $n_x$  が偶数のとき  $\hat{c}_{\frac{n_x}{2}, j_y} = c_{\frac{n_x}{2}, j_y}$  である。

### (2) 使用法

#### 倍精度サブルーチン:

CALL DFR2BF (NX, NY, R, LX, LY, ISW, IFAX, TRIGS, WK, IERR)

#### 単精度サブルーチン:

CALL RFR2BF (NX, NY, R, LX, LY, ISW, IFAX, TRIGS, WK, IERR)



## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY	入 力	入力データ $r_{k_x, k_y}$ (順変換), または $c_{j_x, j_y}$ (逆変換) (注意事項 (b) 参照)
				出 力	出力データ $c_{j_x, j_y}$ (順変換), または $r_{k_x, k_y}$ (逆変換) (注意事項 (b), (c) 参照)
4	LX	I	1	入 力	配列 R の整合寸法 (注意事項 (b) 参照)
5	LY	I	1	入 力	配列 R の第 2 寸法 (注意事項 (b) 参照)
6	ISW	I	1	入 力	処理スイッチ ISW = 1:初期化後の順変換 ISW = -1:初期化後の逆変換
7	IFAX	I	40	入 力	基数分け情報 (注意事項 (a) 参照)
8	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$NX + 2 \times NY$	入 力	三角関数テーブル (注意事項 (a) 参照)
9	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$LX \times LY$	ワーク	作業領域
10	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $NX > 1$  $NY > 1$ 

(b) NX が奇数の時 :

 $NX + 1 \leq LX$ 

NX が偶数の時 :

 $NX + 2 \leq LX$ (c)  $NY \leq LY$ (d)  $ISW \in \{1, -1\}$ 

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) または (c) を満足しなかった.	
3020	制限条件 (d) を満足しなかった.	

(6) 注意事項

- (a) このサブルーチンは、同じデータ数 (NX, NY) の変換を繰り返し行う場合に初期化を含む変換 2.10.1  $\begin{cases} \text{DFR2FB} \\ \text{RFR2FB} \end{cases}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

- (b) 実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) と配列 R の各要素は以下の様に対応する。

$$r_{k_x, k_y} \leftrightarrow R(k_x + 1, k_y + 1)$$

なお、逆変換を行った場合、NX(=n<sub>x</sub>) が奇数のとき R(NX + 1, k<sub>y</sub> + 1) = 0, NX が偶数のとき R(NX + 1, k<sub>y</sub> + 1) = R(NX + 2, k<sub>y</sub> + 1) = 0 となる。また、実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) を配列 R に入力する場合、上述の対応する 0 を特に格納する必要はない。

複素数データ  $c_{j_x, j_y}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{j_x, j_y}\}$ ,  $\Im\{c_{j_x, j_y}\}$  とすると、 $c_{j_x, j_y}$  と配列 R の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

$$\Re\{c_{j_x, j_y}\} \leftrightarrow R(2 * j_x + 1, j_y + 1)$$

$$\Im\{c_{j_x, j_y}\} \leftrightarrow R(2 * j_x + 2, j_y + 1)$$

実フーリエ変換の性質より、 $\Im\{c_{0,0}\} = 0$  であり、NX と NY が共に偶数であれば、 $\Im\{c_{\frac{n_x}{2}, \frac{n_y}{2}}\} = 0$  である。したがって、配列 R の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_{j_x, j_y}$  ( $j_x = \lfloor \frac{n_x}{2} \rfloor + 1, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) の各要素は実フーリエ変換の対称性から以下の関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n_x - j_x, n_y - j_y}^* = c_{j_x, j_y}$$

$$c_{n_x - j_x, j_y}^* = c_{j_x, n_y - j_y}$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。なお、主記憶のバンク競合を避けるために配列 R の整合寸法について LX/2, LY が奇数になるように設定するのが望ましい。通常、たとえば NX が (4 の倍数)+2 の時は LX=NX+4 とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_{k_x, k_y}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ) とすると

$$\hat{r}_{k_x, k_y} = n_x n_y r_{k_x, k_y} \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (e) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。  
(f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

## (a) 問題

$$r_{k_x, k_y} = \frac{n_x + n_y}{(k_x + 1) + (k_y + 1)}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1)$$

を入力データとして、2次元実フーリエ順・逆変換を行う。

## (b) 入力データ

配列 R, NX=6, NY=4, LX=10, LY=5, ISW=1(順変換) および ISW=-1(逆変換)

## (c) 主プログラム

```

PROGRAM BFR2BF
! *** EXAMPLE OF DFR2FB AND DFR2BF ***
PARAMETER (NX=6, NY=4, LX=10, LY=5)
REAL(8) R(LX, LY), WK(LX*LY)
REAL(8) TRIGS(NX+2*NY)
COMPLEX(8) C(LX/2, LY)
INTEGER IFAX(40)
POINTER (CP, C)
CP=LOC(R)
!**** INPUT ****
DO 20 J=1, NY
  DO 10 I=1, NX
    R(I, J) = DBLE(NX+NY)/DBLE(I+J)
  10 CONTINUE
  20 CONTINUE
WRITE(6, 1000)
WRITE(6, 1010) NX, NY, LX, LY
WRITE(6, 1020) 'R(I, J)'
WRITE(6, 1030) ((R(I, J), J=1, NY), I=1, NX)
!**** OUTPUT ****
WRITE(6, 1040)
!**** INITIALIZATION + FORWARD TRANSFORM ****
ISW= 1
CALL DFR2FB(NX, NY, R, LX, LY, ISW, IFAX, TRIGS, WK, IERR)
!**** NORMALIZATION ****
DO 40 J=1, NY
  DO 30 I=1, (NX+2)/2
    C(I, J)=C(I, J)/DBLE(NX*NY)
  30 CONTINUE
  40 CONTINUE
WRITE(6, 1050) IERR
WRITE(6, 1020) 'C(I, J)'
DO 50 I=1, (NX+2)/2
  WRITE(6, 1060) (C(I, J), J=1, NY)
  50 CONTINUE
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL DFR2BF(NX, NY, R, LX, LY, ISW, IFAX, TRIGS, WK, IERR)
WRITE(6, 1070) IERR
WRITE(6, 1020) 'R(I, J)'
WRITE(6, 1030) ((R(I, J), J=1, NY), I=1, NX+2)
STOP
!**** FORMAT ****
1000 FORMAT(1X, '*** DFR2FB AND DFR2BF ***', /, /, &
1X, ' ** INPUT **', /)
1010 FORMAT(1X, '   NX =', I3, 4X, ' NY =', I3, /, &
1X, '   LX =', I3, 4X, ' LY =', I3, /)
1020 FORMAT(1X, 4X, A)
1030 FORMAT(4(4X, F7.4))
1040 FORMAT(/, 1X, ' ** OUTPUT **')
1050 FORMAT(/, 1X, ' ( FORWARD TRANSFORM )', /, /, &
4X, ' IERR =', I5, /)
1060 FORMAT(3X, 4(' (', F6.3, ', ', F6.3, ')'))
1070 FORMAT(/, 1X, ' ( BACKWARD TRANSFORM )', /, /, &
4X, ' IERR =', I5, /)
END

```

## (d) 出力結果

```

*** DFR2FB AND DFR2BF ***
** INPUT **
NX = 6   NY = 4
LX = 10  LY = 5

R(I, J)
5.0000   3.3333   2.5000   2.0000
3.3333   2.5000   2.0000   1.6667
2.5000   2.0000   1.6667   1.4286
2.0000   1.6667   1.4286   1.2500
1.6667   1.4286   1.2500   1.1111
1.4286   1.2500   1.1111   1.0000

** OUTPUT **
( FORWARD TRANSFORM )

```

IERR = 0

C(I,J)  
( 1.938, 0.000) ( 0.249,-0.155) ( 0.219, 0.000) ( 0.249, 0.155)  
( 0.296,-0.247) ( 0.058,-0.094) ( 0.076,-0.045) ( 0.119,-0.009)  
( 0.229,-0.093) ( 0.056,-0.053) ( 0.058,-0.019) ( 0.079, 0.010)  
( 0.219, 0.000) ( 0.064,-0.030) ( 0.055, 0.000) ( 0.064, 0.030)

( BACKWARD TRANSFORM )

IERR = 0

R(I,J)  
5.0000 3.3333 2.5000 2.0000  
3.3333 2.5000 2.0000 1.6667  
2.5000 2.0000 1.6667 1.4286  
2.0000 1.6667 1.4286 1.2500  
1.6667 1.4286 1.2500 1.1111  
1.4286 1.2500 1.1111 1.0000  
0.0000 0.0000 0.0000 0.0000  
0.0000 0.0000 0.0000 0.0000

---

## 2.11 3次元複素フーリエ変換 (実数引数型)

### 2.11.1 [非推奨]DFC3FB, RFC3FB

#### 3次元複素フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$

###### 逆変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$

##### (2) 使用法

###### 倍精度サブルーチン:

CALL DFC3FB (NX, NY, NZ, CR, CI, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL RFC3FB (NX, NY, NZ, CR, CI, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{array} \right\}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	NZ	I	1	入 力	3次元目のデータ数 $n_z$ (注意事項 (a) 参照)
4	CR	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX, LY, LZ	入 力	入力データ $c_{k_x, k_y, k_z}$ の実部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y, j_z}$ の実部 (注意事項 (b), (c) 参照)
5	CI	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX, LY, LZ	入 力	入力データ $c_{k_x, k_y, k_z}$ の虚部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y, j_z}$ の虚部 (注意事項 (b), (c) 参照)
6	LX	I	1	入 力	配列 CR, CI の整合寸法 (注意事項 (b) 参照)
7	LY	I	1	入 力	配列 CR, CI の第 2 寸法 (注意事項 (b) 参照)
8	LZ	I	1	入 力	配列 CR, CI の第 3 寸法 (注意事項 (b) 参照)
9	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW= 0:初期化のみ ISW= 1:初期化を含む順変換 ISW=-1:初期化を含む逆変換
10	IFAX	I	60	出 力	基数分け情報 (注意事項 (d) 参照)
11	TRIGS	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times (NX + NY + NZ)$	出 力	三角関数テーブル (注意事項 (d) 参照)
12	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times LX \times LY \times LZ$	ワーク	作業領域
13	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 $NZ > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$   
 $NZ \leq LZ$
- (c)  $ISW \in \{0, 1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) データ数  $N_x, N_y$  や  $N_z$  の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば,  $N_x = 289 (=17^2)$  とするよりも  $N_x = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い.
- (b) 複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{k_x, k_y, k_z}\}$ ,  $\Im\{c_{k_x, k_y, k_z}\}$  とすると,  $c_{k_x, k_y, k_z}$  と配列 CR, CI の各要素は以下の様に対応する.

$$\Re\{c_{k_x, k_y, k_z}\} \leftrightarrow \text{CR}(k_x + 1, k_y + 1, k_z + 1)$$

$$\Im\{c_{k_x, k_y, k_z}\} \leftrightarrow \text{CI}(k_x + 1, k_y + 1, k_z + 1)$$

複素数データ  $d_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) についても同様である. なお, 主記憶のバンク競合を避けるために配列 CR, CI の整合寸法  $L_x, L_y, L_z$  は奇数に設定するのが望ましい. また, 高速化のために配列 CR, CI 内のデータ設定領域以外の要素に対しても演算を実行する. 通常, たとえば  $N_x$  が偶数の時は  $L_x = N_x + 1$  とする.

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合, 得られるデータは, 元のデータをデータ数倍した値になる. 例えば, 複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) とすると

$$\hat{c}_{k_x, k_y, k_z} = n_x n_y n_z c_{k_x, k_y, k_z}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

となる. したがって, 順変換または逆変換の結果のどちらかに対して正規化を行う必要がある. なお, 文献によっては, 順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい.

- (d) 同じデータ数 ( $N_x, N_y, N_z$ ) の変換を繰り返し行う場合, 一度このサブルーチン呼びその後は初期化後の変換 2.11.2  $\left\{ \begin{array}{l} \text{DFC3BF} \\ \text{RFC3BF} \end{array} \right\}$  を利用すれば良い. このようにすれば, 初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため, 効率のよい処理ができる. ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.11.2  $\left\{ \begin{array}{l} \text{DFC3BF} \\ \text{RFC3BF} \end{array} \right\}$  の入力としなければならない.

なお, ISW=0 として初期化だけを行う場合には, 配列 CR, CI に入力データを設定する必要がない.

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$  または  $n_z$ ) を周期とする周期関数となっていることを前提としているので, 連続フーリエ変換を標準化して近似する場合にはこのことに注意して標準数や標準化間隔を設定する必要がある. なお, 標準化定理によれば, 周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合, 標準化間隔を  $T = \frac{1}{2f_c}$  ととれば, 以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる.

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (f) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

(7) 使用例

2.11.2 (7) 使用例参照.



## 2.11.2 [非推奨]DFC3BF, RFC3BF

## 3次元複素フーリエ変換 (初期化後の変換)

## (1) 機能

## 順変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

## 逆変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

## (2) 使用法

## 倍精度サブルーチン:

CALL DFC3BF (NX, NY, NZ, CR, CI, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

## 単精度サブルーチン:

CALL RFC3BF (NX, NY, NZ, CR, CI, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	NZ	I	1	入 力	3次元目のデータ数 $n_z$ (注意事項 (a) 参照)
4	CR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY, LZ	入 力	入力データ $c_{k_x, k_y, k_z}$ の実部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y, j_z}$ の実部 (注意事項 (b), (c) 参照)
5	CI	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY, LZ	入 力	入力データ $c_{k_x, k_y, k_z}$ の虚部 (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y, j_z}$ の虚部 (注意事項 (b), (c) 参照)
6	LX	I	1	入 力	配列 CR, CI の整合寸法 (注意事項 (b) 参照)
7	LY	I	1	入 力	配列 CR, CI の第2寸法 (注意事項 (b) 参照)
8	LZ	I	1	入 力	配列 CR, CI の第3寸法 (注意事項 (b) 参照)
9	ISW	I	1	入 力	処理スイッチ ISW=1:初期化後の順変換 ISW=-1:初期化後の逆変換
10	IFAX	I	60	入 力	基数分け情報 (注意事項 (d) 参照)
11	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (NX + NY + NZ)$	入 力	三角関数テーブル (注意事項 (d) 参照)
12	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times LX \times LY \times LZ$	ワーク	作業領域
13	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 $NZ > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$   
 $NZ \leq LZ$
- (c)  $ISW \in \{1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンは、同じデータ数 (NX, NY, NZ) の変換を繰り返し行う場合に初期化を含む変換 2.11.1  $\begin{cases} \text{DFC3FB} \\ \text{RFC3FB} \end{cases}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。
- (b) 複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{k_x, k_y, k_z}\}$ ,  $\Im\{c_{k_x, k_y, k_z}\}$  とすると、 $c_{k_x, k_y, k_z}$  と配列 CR, CI の各要素は以下の様に対応する。

$$\begin{aligned} \Re\{c_{k_x, k_y, k_z}\} &\leftrightarrow \text{CR}(k_x + 1, k_y + 1, k_z + 1) \\ \Im\{c_{k_x, k_y, k_z}\} &\leftrightarrow \text{CI}(k_x + 1, k_y + 1, k_z + 1) \end{aligned}$$

複素数データ  $d_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) についても同様である。なお、主記憶のバンク競合を避けるために配列 CR, CI の整合寸法 LX, LY, LZ は奇数に設定するのが望ましい。また、高速化のために配列 CR, CI 内のデータ設定領域以外の要素に対しても演算を実行する。通常、たとえば NX が偶数の時は LX=NX+1 とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) とすると

$$\begin{aligned} \hat{c}_{k_x, k_y, k_z} &= n_x n_y n_z c_{k_x, k_y, k_z} \\ & \quad (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1) \end{aligned}$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$  または  $n_z$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標準数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (e) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。
- (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

- (a) 問題

$$c_{k_x, k_y, k_z} = \frac{n_x + n_y + n_z}{(k_x + 1) + (k_y + 1) + (k_z + 1)} + \sqrt{-1} \frac{(k_x + 1)(k_y + 1)(k_z + 1)}{n_x n_y n_z}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

を入力データとして、3次元複素フーリエ順・逆変換を行う。

- (b) 入力データ

配列 CR, CI, NX=5, NY=4, NZ=3, LX=5, LY=5, LZ=3, ISW=1(順変換) および ISW=-1(逆変換)

(c) 主プログラム

```

PROGRAM BFC3BF
! *** EXAMPLE OF DFC3FB AND DFC3BF ***
PARAMETER (NX=5,NY=4,NZ=3,LX=5,LY=5,LZ=3)
REAL(8) CR(LX,LY,LZ),CI(LX,LY,LZ)
REAL(8) TRIGS(2*(NX+NY+NZ)),WK(2*LX*LY*LZ)
INTEGER IFAX(60)
!**** INPUT ****
DO 30 K=1,NZ
  DO 20 J=1,NY
    DO 10 I=1,NX
      CR(I,J,K)=DBLE(NX+NY+NZ)/DBLE(I+J+K)
      CI(I,J,K)=DBLE(I*J*K)/DBLE(NX*NY*NZ)
10    CONTINUE
20    CONTINUE
30    CONTINUE
WRITE(6,1000)
WRITE(6,1010) NX,NY,NZ,LX,LY,LZ
DO 100 K=1,NZ
  WRITE(6,1020) K,K,((CR(I,J,K),CI(I,J,K),J=1,NY),I=1,NX)
100  CONTINUE
!**** OUTPUT ****
WRITE(6,1030)
!**** INITIALIZATION + FORWARD TRANSFORM ****
ISW= 1
CALL DFC3BF(NX,NY,NZ,CR,CI,LX,LY,LZ,ISW,IFAX,TRIGS,WK,IERR)
!**** NORMALIZATION ****
DO 60 K=1,NZ
  DO 50 J=1,NY
    DO 40 I=1,NX
      CR(I,J,K)=CR(I,J,K)/DBLE(NX*NY*NZ)
      CI(I,J,K)=CI(I,J,K)/DBLE(NX*NY*NZ)
40    CONTINUE
50    CONTINUE
60    CONTINUE
WRITE(6,1040) IERR
DO 200 K=1,NZ
  WRITE(6,1020) K,K,((CR(I,J,K),CI(I,J,K),J=1,NY),I=1,NX)
200  CONTINUE
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL DFC3BF(NX,NY,NZ,CR,CI,LX,LY,LZ,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1050) IERR
DO 300 K=1,NZ
  WRITE(6,1020) K,K,((CR(I,J,K),CI(I,J,K),J=1,NY),I=1,NX)
300  CONTINUE
STOP
!**** FORMAT ****
1000 FORMAT(1X,'*** DFC3FB AND DFC3BF ***',/,/,&
1X,' ** INPUT **',/)
1010 FORMAT(1X,' NX =',I3,' NY =',I3,' NZ =',I3,/,&
1X,' LY =',I3,' LY =',I3,' LZ =',I3)
1020 FORMAT(/,4X,' ( CR(IX,IY,',I2,', CI(IX,IY,',I2,', )',&
/,4(4X,' (',F6.3,',',F6.3,',')')
1030 FORMAT(/,1X,'** OUTPUT **')
1040 FORMAT(/,1X,' ( FORWARD TRANSFORM )',/,/,6X,'IERR =',I6)
1050 FORMAT(/,1X,' ( BACKWARD TRANSFORM )',/,/,6X,'IERR =',I6)
END

```

## (d) 出力結果

\*\*\* DFC3FB AND DFC3BF \*\*\*

\*\* INPUT \*\*

NX = 5 NY = 4 NZ = 3  
LY = 5 LZ = 3

( CR(IX,IY, 1) , CI(IX,IY, 1) )			
( 4.000, 0.017 ) ( 3.000, 0.033 )	( 2.400, 0.050 )	( 2.000, 0.067 )	
( 3.000, 0.033 ) ( 2.400, 0.067 )	( 2.000, 0.100 )	( 1.714, 0.133 )	
( 2.400, 0.050 ) ( 2.000, 0.100 )	( 1.714, 0.150 )	( 1.500, 0.200 )	
( 2.000, 0.067 ) ( 1.714, 0.133 )	( 1.500, 0.200 )	( 1.333, 0.267 )	
( 1.714, 0.083 ) ( 1.500, 0.167 )	( 1.333, 0.250 )	( 1.200, 0.333 )	

( CR(IX,IY, 2) , CI(IX,IY, 2) )			
( 3.000, 0.033 ) ( 2.400, 0.067 )	( 2.000, 0.100 )	( 1.714, 0.133 )	
( 2.400, 0.067 ) ( 2.000, 0.133 )	( 1.714, 0.200 )	( 1.500, 0.267 )	
( 2.000, 0.100 ) ( 1.714, 0.200 )	( 1.500, 0.300 )	( 1.333, 0.400 )	
( 1.714, 0.133 ) ( 1.500, 0.267 )	( 1.333, 0.400 )	( 1.200, 0.533 )	
( 1.500, 0.167 ) ( 1.333, 0.333 )	( 1.200, 0.500 )	( 1.091, 0.667 )	

( CR(IX,IY, 3) , CI(IX,IY, 3) )			
( 2.400, 0.050 ) ( 2.000, 0.100 )	( 1.714, 0.150 )	( 1.500, 0.200 )	
( 2.000, 0.100 ) ( 1.714, 0.200 )	( 1.500, 0.300 )	( 1.333, 0.400 )	
( 1.714, 0.150 ) ( 1.500, 0.300 )	( 1.333, 0.450 )	( 1.200, 0.600 )	
( 1.500, 0.200 ) ( 1.333, 0.400 )	( 1.200, 0.600 )	( 1.091, 0.800 )	
( 1.333, 0.250 ) ( 1.200, 0.500 )	( 1.091, 0.750 )	( 1.000, 1.000 )	

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

( CR(IX,IY, 1) , CI(IX,IY, 1) )			
( 1.737, 0.250 ) ( 0.102, -0.160 )	( 0.137, -0.050 )	( 0.202, 0.060 )	
( 0.108, -0.189 ) ( 0.038, -0.047 )	( 0.041, -0.012 )	( 0.052, 0.016 )	
( 0.125, -0.078 ) ( 0.034, -0.017 )	( 0.026, 0.003 )	( 0.025, 0.021 )	
( 0.152, -0.005 ) ( 0.037, 0.001 )	( 0.021, 0.014 )	( 0.012, 0.028 )	
( 0.223, 0.106 ) ( 0.046, 0.024 )	( 0.018, 0.029 )	( -0.002, 0.041 )	

( CR(IX,IY, 2) , CI(IX,IY, 2) )			
( 0.106, -0.127 ) ( 0.041, -0.022 )	( 0.031, 0.003 )	( 0.030, 0.025 )	
( 0.042, -0.032 ) ( -0.002, -0.009 )	( 0.002, -0.008 )	( 0.009, -0.010 )	
( 0.032, -0.007 ) ( 0.001, -0.007 )	( 0.004, -0.005 )	( 0.009, -0.004 )	
( 0.030, 0.008 ) ( 0.005, -0.007 )	( 0.007, -0.003 )	( 0.011, 0.000 )	
( 0.032, 0.029 ) ( 0.011, -0.009 )	( 0.012, -0.002 )	( 0.016, 0.006 )	

( CR(IX,IY, 3) , CI(IX,IY, 3) )			
( 0.178, 0.002 ) ( 0.040, 0.014 )	( 0.017, 0.022 )	( 0.001, 0.033 )	
( 0.048, 0.009 ) ( 0.005, -0.010 )	( 0.008, -0.006 )	( 0.015, -0.002 )	
( 0.024, 0.016 ) ( 0.007, -0.006 )	( 0.008, -0.002 )	( 0.011, 0.003 )	
( 0.013, 0.024 ) ( 0.010, -0.003 )	( 0.008, 0.002 )	( 0.008, 0.008 )	
( 0.001, 0.036 ) ( 0.016, 0.001 )	( 0.011, 0.007 )	( 0.007, 0.016 )	

( BACKWARD TRANSFORM )

IERR = 0

( CR(IX,IY, 1) , CI(IX,IY, 1) )			
( 4.000, 0.017 ) ( 3.000, 0.033 )	( 2.400, 0.050 )	( 2.000, 0.067 )	
( 3.000, 0.033 ) ( 2.400, 0.067 )	( 2.000, 0.100 )	( 1.714, 0.133 )	
( 2.400, 0.050 ) ( 2.000, 0.100 )	( 1.714, 0.150 )	( 1.500, 0.200 )	
( 2.000, 0.067 ) ( 1.714, 0.133 )	( 1.500, 0.200 )	( 1.333, 0.267 )	
( 1.714, 0.083 ) ( 1.500, 0.167 )	( 1.333, 0.250 )	( 1.200, 0.333 )	

( CR(IX,IY, 2) , CI(IX,IY, 2) )			
( 3.000, 0.033 ) ( 2.400, 0.067 )	( 2.000, 0.100 )	( 1.714, 0.133 )	
( 2.400, 0.067 ) ( 2.000, 0.133 )	( 1.714, 0.200 )	( 1.500, 0.267 )	
( 2.000, 0.100 ) ( 1.714, 0.200 )	( 1.500, 0.300 )	( 1.333, 0.400 )	
( 1.714, 0.133 ) ( 1.500, 0.267 )	( 1.333, 0.400 )	( 1.200, 0.533 )	
( 1.500, 0.167 ) ( 1.333, 0.333 )	( 1.200, 0.500 )	( 1.091, 0.667 )	

( CR(IX,IY, 3) , CI(IX,IY, 3) )			
( 2.400, 0.050 ) ( 2.000, 0.100 )	( 1.714, 0.150 )	( 1.500, 0.200 )	
( 2.000, 0.100 ) ( 1.714, 0.200 )	( 1.500, 0.300 )	( 1.333, 0.400 )	
( 1.714, 0.150 ) ( 1.500, 0.300 )	( 1.333, 0.450 )	( 1.200, 0.600 )	
( 1.500, 0.200 ) ( 1.333, 0.400 )	( 1.200, 0.600 )	( 1.091, 0.800 )	
( 1.333, 0.250 ) ( 1.200, 0.500 )	( 1.091, 0.750 )	( 1.000, 1.000 )	

---

## 2.12 3次元複素フーリエ変換 (複索引数型)

### 2.12.1 [非推奨]ZFC3FB, CFC3FB

#### 3次元複素フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$

###### 逆変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$

##### (2) 使用法

###### 倍精度サブルーチン:

CALL ZFC3FB (NX, NY, NZ, C, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL CFC3FB (NX, NY, NZ, C, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	NZ	I	1	入 力	3次元目のデータ数 $n_z$ (注意事項 (a) 参照)
4	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX, LY, LZ	入 力	入力データ $c_{k_x, k_y, k_z}$ (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y, j_z}$ (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 C の整合寸法 (注意事項 (b) 参照)
6	LY	I	1	入 力	配列 C の第 2 寸法 (注意事項 (b) 参照)
7	LZ	I	1	入 力	配列 C の第 3 寸法 (注意事項 (b) 参照)
8	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含む順変換 ISW = -1:初期化を含む逆変換
9	IFAX	I	60	出 力	基数分け情報 (注意事項 (d) 参照)
10	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (NX + NY + NZ)$	出 力	三角関数テーブル (注意事項 (d) 参照)
11	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX $\times$ LY $\times$ LZ	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 $NZ > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$   
 $NZ \leq LZ$
- (c)  $ISW \in \{0, 1, -1\}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) データ数 NX, NY や NZ の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $NX = 289 (=17^2)$  とするよりも  $NX = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い。
- (b) 複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) と配列 C の各要素は以下の様に対応する。

$$c_{k_x, k_y, k_z} \leftrightarrow C(k_x + 1, k_y + 1, k_z + 1)$$

複素数データ  $d_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) についても同様である。なお、主記憶のバンク競合を避けるために配列 C の整合寸法 LX, LY, LZ は奇数に設定するのが望ましい。また、高速化のために配列 C 内のデータ設定領域以外の要素に対しても演算を実行する。通常、たとえば NX が偶数の時は  $LX = NX + 1$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) とすると

$$\hat{c}_{k_x, k_y, k_z} = n_x n_y n_z c_{k_x, k_y, k_z} \\ (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数 (NX, NY, NZ) の変換を繰り返し行う場合、一度このサブルーチンを呼びその後は初期化後の変換 2.12.2  $\left\{ \begin{array}{l} \text{ZFC3BF} \\ \text{CFC3BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.12.2  $\left\{ \begin{array}{l} \text{ZFC3BF} \\ \text{CFC3BF} \end{array} \right\}$  の入力としなければならない。
- なお、ISW=0 として初期化だけを行う場合には、配列 C に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$  または  $n_z$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標本化して近似する場合にはこのことに注意して標本数や標本化間隔を設定する必要がある。なお、標本化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標本化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (f) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない。
- (g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

(7) 使用例

2.12.2 (7) 使用例参照。



## 2.12.2 [非推奨]ZFC3BF, CFC3BF

## 3次元複素フーリエ変換 (初期化後の変換)

## (1) 機能

## 順変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ順変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

## 逆変換

3次元複素データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元複素フーリエ逆変換 (任意基数) を行う.

$$d_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} c_{k_x, k_y, k_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$(j_x = 0, \dots, n_x - 1; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

## (2) 使用法

## 倍精度サブルーチン:

CALL ZFC3BF (NX, NY, NZ, C, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

## 単精度サブルーチン:

CALL CFC3BF (NX, NY, NZ, C, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	NZ	I	1	入 力	3次元目のデータ数 $n_z$ (注意事項 (a) 参照)
4	C	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX, LY, LZ	入 力	入力データ $c_{k_x, k_y, k_z}$ (注意事項 (b) 参照)
				出 力	出力データ $d_{j_x, j_y, j_z}$ (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 C の整合寸法 (注意事項 (b) 参照)
6	LY	I	1	入 力	配列 C の第 2 寸法 (注意事項 (b) 参照)
7	LZ	I	1	入 力	配列 C の第 3 寸法 (注意事項 (b) 参照)
8	ISW	I	1	入 力	処理スイッチ ISW = 1:初期化後の順変換 ISW = -1:初期化後の逆変換
9	IFAX	I	60	入 力	基数分け情報 (注意事項 (a) 参照)
10	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (NX + NY + NZ)$	入 力	三角関数テーブル (注意事項 (a) 参照)
11	WK	$\begin{Bmatrix} Z \\ C \end{Bmatrix}$	LX $\times$ LY $\times$ LZ	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 $NZ > 1$
- (b)  $NX \leq LX$   
 $NY \leq LY$   
 $NZ \leq LZ$
- (c)  $ISW \in \{1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンは、同じデータ数 (NX, NY, NZ) の変換を繰り返し行う場合に初期化を含む変換 2.12.1  $\begin{cases} \text{ZFC3FB} \\ \text{CFC3FB} \end{cases}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。

- (b) 複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) と配列 C の各要素は以下の様に対応する。

$$c_{k_x, k_y, k_z} \leftrightarrow C(k_x + 1, k_y + 1, k_z + 1)$$

複素数データ  $d_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) についても同様である。なお、主記憶のバンク競合を避けるために配列 C の整合寸法 LX, LY, LZ は奇数に設定するのが望ましい。また、高速化のために配列 C 内のデータ設定領域以外の要素に対しても演算を実行する。通常、たとえば NX が偶数の時は  $LX = NX + 1$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、複素数データ  $c_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{c}_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) とすると

$$\hat{c}_{k_x, k_y, k_z} = n_x n_y n_z c_{k_x, k_y, k_z} \\ (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$  または  $n_z$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標本数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c (t - iT)}{\pi (t - iT)}$$

- (e) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。  
 (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

(7) 使用例

(a) 問題

$$C_{k_x, k_y, k_z} = \frac{n_x + n_y + n_z}{(k_x + 1) + (k_y + 1) + (k_z + 1)} + \sqrt{-1} \frac{(k_x + 1)(k_y + 1)(k_z + 1)}{n_x n_y n_z}$$

$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$

を入力データとして、3次元複素フーリエ順・逆変換を行う。

(b) 入力データ

配列 C, NX=5, NY=4, NZ=3, LX=5, LY=5, LZ=3, ISW=1 (順変換) および ISW=-1 (逆変換)

(c) 主プログラム

```

PROGRAM AFC3BF
! *** EXAMPLE OF ZFC3FB AND ZFC3BF ***
PARAMETER (NX=5,NY=4,NZ=3,LX=5,LY=5,LZ=3)
COMPLEX(8) C(LX,LY,LZ),WK(LX*LY*LZ)
REAL(8) TRIGS(2*(NX+NY+NZ))
INTEGER IFAX(60)
!**** INPUT ****
DO 30 K=1,NZ
  DO 20 J=1,NY
    DO 10 I=1,NX
      C(I,J,K)=CMPLX( DBLE(NX+NY+NZ)/DBLE(I+J+K), &
        DBLE(I*J*K)/DBLE(NX*NY*NZ), &
        KIND=8 )
    10 CONTINUE
  20 CONTINUE
  30 CONTINUE
  WRITE(6,1000)
  WRITE(6,1010) NX,NY,NZ,LX,LY,LZ
  DO 100 K=1,NZ
    WRITE(6,1020) K,((C(I,J,K),J=1,NY),I=1,NX)
  100 CONTINUE
!**** OUTPUT ****
WRITE(6,1030)
!**** INITIALIZATION + FORWARD TRANSFORM ****
ISW= 1
CALL ZFC3FB(NX,NY,NZ,C,LX,LY,LZ,ISW,IFAX,TRIGS,WK,IERR)
!**** NORMALIZATION ****
DO 60 K=1,NZ
  DO 50 J=1,NY
    DO 40 I=1,NX
      C(I,J,K)=C(I,J,K)/DBLE(NX*NY*NZ)
    40 CONTINUE
  50 CONTINUE
  60 CONTINUE
  WRITE(6,1040) IERR
  DO 200 K=1,NZ
    WRITE(6,1020) K,((C(I,J,K),J=1,NY),I=1,NX)
  200 CONTINUE
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL ZFC3BF(NX,NY,NZ,C,LX,LY,LZ,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1050) IERR
DO 300 K=1,NZ
  WRITE(6,1020) K,((C(I,J,K),J=1,NY),I=1,NX)
  300 CONTINUE
STOP
!**** FORMAT ****
1000 FORMAT(1X,'*** ZFC3FB AND ZFC3BF ***',/,/, &
  1X,' ** INPUT **',/)
1010 FORMAT(1X,' NX =',I3,' NY =',I3,' NZ =',I3,/, &
  1X,' LX =',I3,' LY =',I3,' LZ =',I3)
1020 FORMAT(/,1X,' C(IX,IY,',I2,',) ', &
  /,4(1X,' (',F6.3,',',F6.3,',)')')
1030 FORMAT(/,1X,' ** OUTPUT **')
1040 FORMAT(/,1X,' ( FORWARD TRANSFORM )',/,/,4X,' IERR =',I6)
1050 FORMAT(/,1X,' ( BACKWARD TRANSFORM )',/,/,4X,' IERR =',I6)
END

```

(d) 出力結果

```

*** ZFC3FB AND ZFC3BF ***
** INPUT **
  NX = 5  NY = 4  NZ = 3
  LX = 5  LY = 5  LZ = 3

C(IX,IY, 1)
( 4.000, 0.017) ( 3.000, 0.033) ( 2.400, 0.050) ( 2.000, 0.067)
( 3.000, 0.033) ( 2.400, 0.067) ( 2.000, 0.100) ( 1.714, 0.133)
( 2.400, 0.050) ( 2.000, 0.100) ( 1.714, 0.150) ( 1.500, 0.200)
( 2.000, 0.067) ( 1.714, 0.133) ( 1.500, 0.200) ( 1.333, 0.267)
( 1.714, 0.083) ( 1.500, 0.167) ( 1.333, 0.250) ( 1.200, 0.333)

C(IX,IY, 2)
( 3.000, 0.033) ( 2.400, 0.067) ( 2.000, 0.100) ( 1.714, 0.133)
( 2.400, 0.067) ( 2.000, 0.133) ( 1.714, 0.200) ( 1.500, 0.267)
( 2.000, 0.100) ( 1.714, 0.200) ( 1.500, 0.300) ( 1.333, 0.400)

```

( 1.714, 0.133)	( 1.500, 0.267)	( 1.333, 0.400)	( 1.200, 0.533)
( 1.500, 0.167)	( 1.333, 0.333)	( 1.200, 0.500)	( 1.091, 0.667)

C(IX,IY, 3)			
( 2.400, 0.050)	( 2.000, 0.100)	( 1.714, 0.150)	( 1.500, 0.200)
( 2.000, 0.100)	( 1.714, 0.200)	( 1.500, 0.300)	( 1.333, 0.400)
( 1.714, 0.150)	( 1.500, 0.300)	( 1.333, 0.450)	( 1.200, 0.600)
( 1.500, 0.200)	( 1.333, 0.400)	( 1.200, 0.600)	( 1.091, 0.800)
( 1.333, 0.250)	( 1.200, 0.500)	( 1.091, 0.750)	( 1.000, 1.000)

\*\* OUTPUT \*\*

( FORWARD TRANSFORM )

IERR = 0

C(IX,IY, 1)			
( 1.737, 0.250)	( 0.102,-0.160)	( 0.137,-0.050)	( 0.202, 0.060)
( 0.108,-0.189)	( 0.038,-0.047)	( 0.041,-0.012)	( 0.052, 0.016)
( 0.125,-0.078)	( 0.034,-0.017)	( 0.026, 0.003)	( 0.025, 0.021)
( 0.152,-0.005)	( 0.037, 0.001)	( 0.021, 0.014)	( 0.012, 0.028)
( 0.223, 0.106)	( 0.046, 0.024)	( 0.018, 0.029)	(-0.002, 0.041)

C(IX,IY, 2)			
( 0.106,-0.127)	( 0.041,-0.022)	( 0.031, 0.003)	( 0.030, 0.025)
( 0.042,-0.032)	(-0.002,-0.009)	( 0.002,-0.008)	( 0.009,-0.010)
( 0.032,-0.007)	( 0.001,-0.007)	( 0.004,-0.005)	( 0.009,-0.004)
( 0.030, 0.008)	( 0.005,-0.007)	( 0.007,-0.003)	( 0.011, 0.000)
( 0.032, 0.029)	( 0.011,-0.009)	( 0.012,-0.002)	( 0.016, 0.006)

C(IX,IY, 3)			
( 0.178, 0.002)	( 0.040, 0.014)	( 0.017, 0.022)	( 0.001, 0.033)
( 0.048, 0.009)	( 0.005,-0.010)	( 0.008,-0.006)	( 0.015,-0.002)
( 0.024, 0.016)	( 0.007,-0.006)	( 0.008,-0.002)	( 0.011, 0.003)
( 0.013, 0.024)	( 0.010,-0.003)	( 0.008, 0.002)	( 0.008, 0.008)
( 0.001, 0.036)	( 0.016, 0.001)	( 0.011, 0.007)	( 0.007, 0.016)

( BACKWARD TRANSFORM )

IERR = 0

C(IX,IY, 1)			
( 4.000, 0.017)	( 3.000, 0.033)	( 2.400, 0.050)	( 2.000, 0.067)
( 3.000, 0.033)	( 2.400, 0.067)	( 2.000, 0.100)	( 1.714, 0.133)
( 2.400, 0.050)	( 2.000, 0.100)	( 1.714, 0.150)	( 1.500, 0.200)
( 2.000, 0.067)	( 1.714, 0.133)	( 1.500, 0.200)	( 1.333, 0.267)
( 1.714, 0.083)	( 1.500, 0.167)	( 1.333, 0.250)	( 1.200, 0.333)

C(IX,IY, 2)			
( 3.000, 0.033)	( 2.400, 0.067)	( 2.000, 0.100)	( 1.714, 0.133)
( 2.400, 0.067)	( 2.000, 0.133)	( 1.714, 0.200)	( 1.500, 0.267)
( 2.000, 0.100)	( 1.714, 0.200)	( 1.500, 0.300)	( 1.333, 0.400)
( 1.714, 0.133)	( 1.500, 0.267)	( 1.333, 0.400)	( 1.200, 0.533)
( 1.500, 0.167)	( 1.333, 0.333)	( 1.200, 0.500)	( 1.091, 0.667)

C(IX,IY, 3)			
( 2.400, 0.050)	( 2.000, 0.100)	( 1.714, 0.150)	( 1.500, 0.200)
( 2.000, 0.100)	( 1.714, 0.200)	( 1.500, 0.300)	( 1.333, 0.400)
( 1.714, 0.150)	( 1.500, 0.300)	( 1.333, 0.450)	( 1.200, 0.600)
( 1.500, 0.200)	( 1.333, 0.400)	( 1.200, 0.600)	( 1.091, 0.800)
( 1.333, 0.250)	( 1.200, 0.500)	( 1.091, 0.750)	( 1.000, 1.000)

## 2.13 3次元実フーリエ変換

### 2.13.1 [非推奨]DFR3FB, RFR3FB

#### 3次元実フーリエ変換 (初期化を含む変換)

##### (1) 機能

###### 順変換

3次元実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元フーリエ順変換 (任意基数) の  $j_x$  についての半周期分を求める.

$$c_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} r_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$(j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す. なお, 残りの半周期分は以下のような関係から得られる.

$$c_{n_x-j_x, n_y-j_y, n_z-j_z}^* = c_{j_x, j_y, j_z}$$

$$c_{n_x-j_x, j_y, j_z}^* = c_{j_x, n_y-j_y, n_z-j_z}$$

$$c_{n_x-j_x, n_y-j_y, j_z}^* = c_{j_x, j_y, n_z-j_z}$$

ただし,  $z^*$  は複素数  $z$  の共役複素数を表す.

###### 逆変換

$c_{n_x-j_x, n_y-j_y, n_z-j_z}^* = c_{j_x, j_y, j_z}$ ,  $c_{n_x-j_x, j_y, j_z}^* = c_{j_x, n_y-j_y, n_z-j_z}$ ,  $c_{n_x-j_x, n_y-j_y, j_z}^* = c_{j_x, j_y, n_z-j_z}$  等の関係を満たす 3次元実フーリエ変換後の  $n_x n_y n_z$  個の複素数データ  $c_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) について  $j_x$  についての半周期分  $c_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) を与えて以下のように定義される 3次元フーリエ逆変換 (任意基数) を求める.

$$r_{k_x, k_y, k_z} = \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} \sum_{j_z=0}^{n_z-1} c_{j_x, j_y, j_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$= \sum_{j_z=0}^{n_z-1} \sum_{j_y=0}^{n_y-1} \{c_{0, j_y, j_z} + (-1)^{k_x} \hat{c}_{\frac{n_x}{2}, j_y, j_z}\} e^{2\pi\sqrt{-1}\left(\frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$+ 2 \sum_{j_z=0}^{n_z-1} \sum_{j_y=0}^{n_y-1} \sum_{j_x=1}^{\lfloor \frac{n_x}{2} \rfloor - 1} \Re\{c_{j_x, j_y, j_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}\}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

ここで  $\lfloor x \rfloor$  は  $x$  以上の最小の整数を,  $\Re\{z\}$  は複素数  $z$  の実部を表す. また,  $n_x$  が奇数のとき  $\hat{c}_{\frac{n_x}{2}, j_y, j_z} = 0$ ,  $n_x$  が偶数のとき  $\hat{c}_{\frac{n_x}{2}, j_y, j_z} = c_{\frac{n_x}{2}, j_y, j_z}$  である.

##### (2) 使用法

###### 倍精度サブルーチン:

CALL DFR3FB (NX, NY, NZ, R, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

###### 単精度サブルーチン:

CALL RFR3FB (NX, NY, NZ, R, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	NZ	I	1	入 力	3次元目のデータ数 $n_z$ (注意事項 (a) 参照)
4	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY, LZ	入 力	入力データ $r_{k_x, k_y, k_z}$ (順変換), または $c_{j_x, j_y, j_z}$ (逆変換) (注意事項 (b) 参照)
				出 力	出力データ $c_{j_x, j_y, j_z}$ (順変換), または $r_{k_x, k_y, k_z}$ (逆変換) (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 R の整合寸法 (注意事項 (b) 参照)
6	LY	I	1	入 力	配列 R の第 2 寸法 (注意事項 (b) 参照)
7	LZ	I	1	入 力	配列 R の第 3 寸法 (注意事項 (b) 参照)
8	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW = 0:初期化のみ ISW = 1:初期化を含む順変換 ISW = -1:初期化を含む逆変換
9	IFAX	I	60	出 力	基数分け情報 (注意事項 (d) 参照)
10	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$NX + 2 \times (NY + NZ)$	出 力	三角関数テーブル (注意事項 (d) 参照)
11	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$LX \times LY \times LZ$	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 $NZ > 1$
- (b) NX が奇数の時 :  
 $NX + 1 \leq LX$   
 NX が偶数の時 :  
 $NX + 2 \leq LX$
- (c)  $NY \leq LY$
- (d)  $NZ \leq LZ$
- (e) LX は偶数である.
- (f)  $ISW \in \{0, 1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b), (c) または (d) を満足しなかつた.	
3020	制限条件 (e) を満足しなかった.	
3030	制限条件 (f) を満足しなかった.	

(6) 注意事項

- (a) データ数 NX, NY や NZ の値を調整できる場合には、混合基数 FFT アルゴリズムが有効に働く数 (2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば,  $NX = 289 (=17^2)$  とするよりも  $NX = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い.
- (b) 実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) と配列 R の各要素は以下の様に対応する.

$$r_{k_x, k_y, k_z} \leftrightarrow R(k_x + 1, k_y + 1, k_z + 1)$$

なお、逆変換を行った場合、 $NX (=n_x)$  が奇数のとき  $R(NX + 1, k_y + 1, k_z + 1) = 0$ ,  $NX$  が偶数のとき  $R(NX + 1, k_y + 1, k_z + 1) = R(NX + 2, k_y + 1, k_z + 1) = 0$  となる. また、実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) を配列 R に入力する場合、上述の対応する 0 を特に格納する必要はない.

複素数データ  $c_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{j_x, j_y, j_z}\}$ ,  $\Im\{c_{j_x, j_y, j_z}\}$  とすると、 $c_{j_x, j_y, j_z}$  と配列 R の各要素は以下の様に対応する. ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す.

$$\begin{aligned} \Re\{c_{j_x, j_y, j_z}\} &\leftrightarrow R(2 * j_x + 1, j_y + 1, j_z + 1) \\ \Im\{c_{j_x, j_y, j_z}\} &\leftrightarrow R(2 * j_x + 2, j_y + 1, j_z + 1) \end{aligned}$$

実フーリエ変換の性質より、 $\Im\{c_{0,0,0}\} = 0$  であり、NX, NY と NZ が共に偶数であれば、 $\Im\{c_{\frac{n_x}{2}, \frac{n_y}{2}, \frac{n_z}{2}}\} = 0$  である. したがって、配列 R の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う. なお、 $c_{j_x, j_y, j_z}$  ( $j_x = \lfloor \frac{n_x}{2} \rfloor + 1, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) の各要素は実フーリエ変換の対称性から以下のような関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない.

$$\begin{aligned} c_{n_x - j_x, n_y - j_y, n_z - j_z}^* &= c_{j_x, j_y, j_z} \\ c_{n_x - j_x, j_y, j_z}^* &= c_{j_x, n_y - j_y, n_z - j_z} \\ c_{n_x - j_x, n_y - j_y, j_z}^* &= c_{j_x, j_y, n_z - j_z} \end{aligned}$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す. なお、主記憶のバンク競合を避けるために配列 R の整合寸法について LX/2, LY, LZ が奇数になるように設定するのが望ましい. また、高速化のために配列 R 内のデータ設定領域以外の要素に対しても演算を実行する. 通常、たとえば NX が (4 の倍数)+2 の時は  $LX = NX + 4$  とする.

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる. 例えば、実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ )



に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_{k_x, k_y, k_z}(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$  とすると

$$\hat{r}_{k_x, k_y, k_z} = n_x n_y n_z r_{k_x, k_y, k_z} \\ (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 同じデータ数 (NX, NY, NZ) の変換を繰り返し行う場合、一度このサブルーチンと呼びその後は初期化後の変換 2.13.2  $\left\{ \begin{array}{l} \text{DFR3BF} \\ \text{RFR3BF} \end{array} \right\}$  を利用すれば良い。このようにすれば、初期化 (基数分けや三角関数テーブルの作成) が一度だけしか行われなため、効率のよい処理ができる。ただしこの場合は配列 IFAX, TRIGS の内容をそのまま 2.13.2  $\left\{ \begin{array}{l} \text{DFR3BF} \\ \text{RFR3BF} \end{array} \right\}$  の入力としなければならない。  
なお、ISW=0 として初期化だけを行う場合には、配列 R に入力データを設定する必要がない。

- (e) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$  または  $n_z$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標本数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標本値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (f) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。  
(g) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インタフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

2.13.2 (7) 使用例参照。

## 2.13.2 [非推奨]DFR3BF, RFR3BF 3次元実フーリエ変換 (初期化後の変換)

### (1) 機能

#### 順変換

3次元実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対して, 3次元フーリエ順変換 (任意基数) の  $j_x$  についての半周期分を求める。

$$c_{j_x, j_y, j_z} = \sum_{k_x=0}^{n_x-1} \sum_{k_y=0}^{n_y-1} \sum_{k_z=0}^{n_z-1} r_{k_x, k_y, k_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$(j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor; j_y = 0, \dots, n_y - 1; j_z = 0, \dots, n_z - 1)$$

ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。なお, 残りの半周期分は以下のような関係から得られる。

$$c_{n_x - j_x, n_y - j_y, n_z - j_z}^* = c_{j_x, j_y, j_z}$$

$$c_{n_x - j_x, j_y, j_z}^* = c_{j_x, n_y - j_y, n_z - j_z}$$

$$c_{n_x - j_x, n_y - j_y, j_z}^* = c_{j_x, j_y, n_z - j_z}$$

ただし,  $z^*$  は複素数  $z$  の共役複素数を表す。

#### 逆変換

$c_{n_x - j_x, n_y - j_y, n_z - j_z}^* = c_{j_x, j_y, j_z}$ ,  $c_{n_x - j_x, j_y, j_z}^* = c_{j_x, n_y - j_y, n_z - j_z}$ ,  $c_{n_x - j_x, n_y - j_y, j_z}^* = c_{j_x, j_y, n_z - j_z}$  等の関係を満たす 3次元実フーリエ変換後の  $n_x n_y n_z$  個の複素数データ  $c_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) について  $j_x$  についての半周期分  $c_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) を与えて以下のように定義される 3次元フーリエ逆変換 (任意基数) を求める。

$$r_{k_x, k_y, k_z} = \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} \sum_{j_z=0}^{n_z-1} c_{j_x, j_y, j_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$= \sum_{j_z=0}^{n_z-1} \sum_{j_y=0}^{n_y-1} \{c_{0, j_y, j_z} + (-1)^{k_x} \hat{c}_{\frac{n_x}{2}, j_y, j_z}\} e^{2\pi\sqrt{-1}\left(\frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}$$

$$+ 2 \sum_{j_z=0}^{n_z-1} \sum_{j_y=0}^{n_y-1} \sum_{j_x=1}^{\lfloor \frac{n_x}{2} \rfloor - 1} \Re\{c_{j_x, j_y, j_z} e^{2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)}\}$$

$$(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

ここで  $\lfloor x \rfloor$  は  $x$  以上の最小の整数を,  $\Re\{z\}$  は複素数  $z$  の実部を表す。また,  $n_x$  が奇数のとき  $\hat{c}_{\frac{n_x}{2}, j_y, j_z} = 0$ ,  $n_x$  が偶数のとき  $\hat{c}_{\frac{n_x}{2}, j_y, j_z} = c_{\frac{n_x}{2}, j_y, j_z}$  である。

### (2) 使用法

#### 倍精度サブルーチン:

CALL DFR3BF (NX, NY, NZ, R, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

#### 単精度サブルーチン:

CALL RFR3BF (NX, NY, NZ, R, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	1次元目のデータ数 $n_x$ (注意事項 (a) 参照)
2	NY	I	1	入 力	2次元目のデータ数 $n_y$ (注意事項 (a) 参照)
3	NZ	I	1	入 力	3次元目のデータ数 $n_z$ (注意事項 (a) 参照)
4	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY, LZ	入 力	入力データ $r_{k_x, k_y, k_z}$ (順変換), または $c_{j_x, j_y, j_z}$ (逆変換) (注意事項 (b) 参照)
				出 力	出力データ $c_{j_x, j_y, j_z}$ (順変換), または $r_{k_x, k_y, k_z}$ (逆変換) (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 R の整合寸法 (注意事項 (b) 参照)
6	LY	I	1	入 力	配列 R の第 2 寸法 (注意事項 (b) 参照)
7	LZ	I	1	入 力	配列 R の第 3 寸法 (注意事項 (b) 参照)
8	ISW	I	1	入 力	処理スイッチ ISW= 1:初期化後の順変換 ISW=-1:初期化後の逆変換
9	IFAX	I	60	入 力	基数分け情報 (注意事項 (a) 参照)
10	TRIGS	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$NX + 2 \times (NY + NZ)$	入 力	三角関数テーブル (注意事項 (a) 参照)
11	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$LX \times LY \times LZ$	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $NX > 1$   
 $NY > 1$   
 $NZ > 1$
- (b) NX が奇数の時 :  
 $NX + 1 \leq LX$   
 NX が偶数の時 :  
 $NX + 2 \leq LX$
- (c)  $NY \leq LY$
- (d)  $NZ \leq LZ$
- (e) LX は偶数である.
- (f)  $ISW \in \{1, -1\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b), (c) または (d) を満足しなかつた.	
3020	制限条件 (e) を満足しなかった.	
3030	制限条件 (f) を満足しなかった.	

(6) 注意事項

- (a) このサブルーチンは、同じデータ数 (NX, NY, NZ) の変換を繰り返し行う場合に初期化を含む変換 2.13.1  $\begin{Bmatrix} \text{DFR3FB} \\ \text{RFR3FB} \end{Bmatrix}$  を行った後で利用する。なお、この場合は配列 IFAX, TRIGS の内容はそのままこのサブルーチンの入力とする必要がある。
- (b) 実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) と配列 R の各要素は以下の様に対応する。

$$r_{k_x, k_y, k_z} \leftrightarrow R(k_x + 1, k_y + 1, k_z + 1)$$

なお、逆変換を行った場合、NX(=n<sub>x</sub>) が奇数のとき  $R(NX + 1, k_y + 1, k_z + 1) = 0$ 、NX が偶数のとき  $R(NX + 1, k_y + 1, k_z + 1) = R(NX + 2, k_y + 1, k_z + 1) = 0$  となる。また、実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) を配列 R に入力する場合、上述の対応する 0 を特に格納する必要はない。

複素数データ  $c_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) の実部と虚部をそれぞれ  $\Re\{c_{j_x, j_y, j_z}\}$ ,  $\Im\{c_{j_x, j_y, j_z}\}$  とすると、 $c_{j_x, j_y, j_z}$  と配列 R の各要素は以下の様に対応する。ここで  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

$$\Re\{c_{j_x, j_y, j_z}\} \leftrightarrow R(2 * j_x + 1, j_y + 1, j_z + 1)$$

$$\Im\{c_{j_x, j_y, j_z}\} \leftrightarrow R(2 * j_x + 2, j_y + 1, j_z + 1)$$

実フーリエ変換の性質より、 $\Im\{c_{0,0,0}\} = 0$  であり、NX, NY と NZ が共に偶数であれば、 $\Im\{c_{\frac{n_x}{2}, \frac{n_y}{2}, \frac{n_z}{2}}\} = 0$  である。したがって、配列 R の対応する要素に 0 以外の値が設定されていても 0 とみなして処理を行う。なお、 $c_{j_x, j_y, j_z}$  ( $j_x = \lfloor \frac{n_x}{2} \rfloor + 1, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) の各要素は実フーリエ変換の対称性から以下のような関係より得られるので逆変換の場合、入力として与える必要は無く、また順変換の場合、出力は行わない。

$$c_{n_x - j_x, n_y - j_y, n_z - j_z}^* = c_{j_x, j_y, j_z}$$

$$c_{n_x - j_x, j_y, j_z}^* = c_{j_x, n_y - j_y, n_z - j_z}$$

$$c_{n_x - j_x, n_y - j_y, j_z}^* = c_{j_x, j_y, n_z - j_z}$$

ただし、 $z^*$  は複素数  $z$  の共役複素数を表す。なお、主記憶のバンク競合を避けるために配列 R の整合寸法について LX/2, LY, LZ が奇数になるように設定するのが望ましい。また、高速化のために配列 R 内のデータ設定領域以外の要素に対しても演算を実行する。通常、たとえば NX が (4 の倍数)+2 の時は  $LX = NX + 4$  とする。

- (c) このサブルーチンを使用して順変換に引き続き逆変換を行った場合、得られるデータは、元のデータをデータ数倍した値になる。例えば、実数データ  $r_{k_x, k_y, k_z}$  ( $k_x = 0, \dots, n_x - 1$ ;  $k_y = 0, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ )

に対して順変換を行い引き続き逆変換を行ったデータを  $\hat{r}_{k_x, k_y, k_z}(k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$  とすると

$$\hat{r}_{k_x, k_y, k_z} = n_x n_y n_z r_{k_x, k_y, k_z} \\ (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

となる。したがって、順変換または逆変換の結果のどちらかに対して正規化を行う必要がある。なお、文献によっては、順変換と逆変換の定義を本書と逆にしている場合や正規化を行った結果を定義としている場合があるので注意されたい。

- (d) 離散フーリエ変換は変換前後のデータ列がデータ数 ( $n_x$  または  $n_y$  または  $n_z$ ) を周期とする周期関数となっていることを前提としているので、連続フーリエ変換を標準化して近似する場合にはこのことに注意して標準数や標準化間隔を設定する必要がある。なお、標準化定理によれば、周波数  $f_c$  で帯域制限された時間関数  $h(t)$  の場合、標準化間隔を  $T = \frac{1}{2f_c}$  ととれば、以下の様に標準値列  $\{h(iT)\}$  だけの知識から  $h(t)$  を復元できる。

$$h(t) = T \sum_{i=-\infty}^{\infty} h(iT) \frac{\sin 2\pi f_c(t - iT)}{\pi(t - iT)}$$

- (e) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。  
 (f) 非推奨 この機能は将来廃止予定である。より高速な実装が ASL 統合インターフェースで提供されているので、そちらを使用されたい。

## (7) 使用例

### (a) 問題

$$r_{k_x, k_y, k_z} = \frac{(k_x + 1)(k_y + 1)(k_z + 1)}{n_x n_y n_z} \\ (k_x = 0, \dots, n_x - 1; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

を入力データとして、3次元実フーリエ順・逆変換を行う。

### (b) 入力データ

配列 R, NX=6, NY=4, NZ=3, LX=10, LY=5, LZ=3, ISW=1(順変換) および ISW=-1(逆変換)

### (c) 主プログラム

```

PROGRAM BFR3BF
! *** EXAMPLE OF DFR3BF AND DFR3BF ***
PARAMETER (NX=6, NY=4, NZ=3, LX=10, LY=5, LZ=3)
REAL(8) R(LX, LY, LZ)
REAL(8) TRIGS(NX+2*(NY+NZ)), WK(LX*LY*LZ)
COMPLEX(8) C(LX/2, LY, LZ)
INTEGER IFAX(60)
POINTER (CP, C)
CP=LOC(R)
!**** INPUT ****
DO 30 K=1, NZ
  DO 20 J=1, NY
    DO 10 I=1, NX
      R(I, J, K)=DBLE(I*J*K)/DBLE(NX*NY*NZ)
    10 CONTINUE
  20 CONTINUE
  30 CONTINUE
WRITE(6, 1000)
WRITE(6, 1010) NX, NY, NZ, LX, LY, LZ
DO 100 K=1, NZ
  WRITE(6, 1020) K, ((R(I, J, K), J=1, NY), I=1, NX)
100 CONTINUE
!**** OUTPUT ****
WRITE(6, 1030)
!**** INITIALIZATION + FORWARD TRANSFORM ****
ISW= 1
CALL DFR3FB(NX, NY, NZ, R, LX, LY, LZ, ISW, IFAX, TRIGS, WK, IERR)
!**** NORMALIZATION ****
DO 60 K=1, NZ
  DO 50 J=1, NY
    DO 40 I=1, (NX+2)/2
      C(I, J, K)=C(I, J, K)/DBLE(NX*NY*NZ)
    40 CONTINUE
  50 CONTINUE
  60 CONTINUE

```

```

WRITE(6,1040) IERR
DO 220 K=1,NZ
  WRITE(6,1050) K
  DO 210 I=1,(NX+2)/2
    WRITE(6,1060) (C(I,J,K),J=1,NY)
  210 CONTINUE
220 CONTINUE
!**** BACKWARD TRANSFORM ****
ISW=-1
CALL DFR3BF(NX,NY,NZ,R,LX,LY,LZ,ISW,IFAX,TRIGS,WK,IERR)
WRITE(6,1070) IERR
DO 300 K=1,NZ
  WRITE(6,1020) K,((R(I,J,K),J=1,NY),I=1,NX+2)
300 CONTINUE
STOP
!**** FORMAT ****
1000 FORMAT(1X,'*** DFR3FB AND DFR3BF ***',/,/,&
1X,' ** INPUT **',/)
1010 FORMAT(1X,' NX =',I3,' NY =',I3,' NZ =',I3,/,&
1X,' LX =',I3,' LY =',I3,' LZ =',I3)
1020 FORMAT(/,3X,'R(IX,IY,',I2,',)',&
/,4(4X,F7.4))
1030 FORMAT(/,1X,' ** OUTPUT **')
1040 FORMAT(/,1X,' ( FORWARD TRANSFORM )',/,/,4X,'IERR =',I6)
1050 FORMAT(/,3X,'C(IX,IY,',I2,',)',)
1060 FORMAT(3X,4('(',F6.3,',',F6.3,')'))
1070 FORMAT(/,1X,' ( BACKWARD TRANSFORM )',/,/,4X,'IERR =',I6)
END

```

(d) 出力結果

```

*** DFR3FB AND DFR3BF ***

** INPUT **

NX = 6  NY = 4  NZ = 3
LX = 10 LY = 5  LZ = 3

R(IX,IY, 1)
0.0139    0.0278    0.0417    0.0556
0.0278    0.0556    0.0833    0.1111
0.0417    0.0833    0.1250    0.1667
0.0556    0.1111    0.1667    0.2222
0.0694    0.1389    0.2083    0.2778
0.0833    0.1667    0.2500    0.3333

R(IX,IY, 2)
0.0278    0.0556    0.0833    0.1111
0.0556    0.1111    0.1667    0.2222
0.0833    0.1667    0.2500    0.3333
0.1111    0.2222    0.3333    0.4444
0.1389    0.2778    0.4167    0.5556
0.1667    0.3333    0.5000    0.6667

R(IX,IY, 3)
0.0417    0.0833    0.1250    0.1667
0.0833    0.1667    0.2500    0.3333
0.1250    0.2500    0.3750    0.5000
0.1667    0.3333    0.5000    0.6667
0.2083    0.4167    0.6250    0.8333
0.2500    0.5000    0.7500    1.0000

** OUTPUT **

( FORWARD TRANSFORM )

IERR =      0

C(IX,IY, 1)
( 0.243, 0.000) (-0.049, 0.049) (-0.049, 0.000) (-0.049,-0.049)
(-0.035, 0.060) (-0.005,-0.019) ( 0.007,-0.012) ( 0.019,-0.005)
(-0.035, 0.020) ( 0.003,-0.011) ( 0.007,-0.004) ( 0.011, 0.003)
(-0.035, 0.000) ( 0.007,-0.007) ( 0.007, 0.000) ( 0.007, 0.007)

C(IX,IY, 2)
(-0.061, 0.035) ( 0.005,-0.019) ( 0.012,-0.007) ( 0.019, 0.005)
( 0.000,-0.020) ( 0.004, 0.004) (-0.000, 0.004) (-0.004, 0.004)
( 0.006,-0.010) ( 0.001, 0.003) (-0.001, 0.002) (-0.003, 0.001)
( 0.009,-0.005) (-0.001, 0.003) (-0.002, 0.001) (-0.003,-0.001)

C(IX,IY, 3)
(-0.061,-0.035) ( 0.019,-0.005) ( 0.012, 0.007) ( 0.005, 0.019)
( 0.017,-0.010) (-0.001, 0.005) (-0.003, 0.002) (-0.005,-0.001)
( 0.012, 0.000) (-0.002, 0.002) (-0.002, 0.000) (-0.002,-0.002)
( 0.009, 0.005) (-0.003, 0.001) (-0.002,-0.001) (-0.001,-0.003)

( BACKWARD TRANSFORM )

IERR =      0

R(IX,IY, 1)
0.0139    0.0278    0.0417    0.0556
0.0278    0.0556    0.0833    0.1111
0.0417    0.0833    0.1250    0.1667
0.0556    0.1111    0.1667    0.2222
0.0694    0.1389    0.2083    0.2778
0.0833    0.1667    0.2500    0.3333
0.0000    0.0000    0.0000    0.0000
0.0000    0.0000    0.0000    0.0000

```

```
R(IX,IY, 2)
0.0278  0.0556  0.0833  0.1111
0.0556  0.1111  0.1667  0.2222
0.0833  0.1667  0.2500  0.3333
0.1111  0.2222  0.3333  0.4444
0.1389  0.2778  0.4167  0.5556
0.1667  0.3333  0.5000  0.6667
0.0000  0.0000  0.0000  0.0000
0.0000  0.0000  0.0000  0.0000

R(IX,IY, 3)
0.0417  0.0833  0.1250  0.1667
0.0833  0.1667  0.2500  0.3333
0.1250  0.2500  0.3750  0.5000
0.1667  0.3333  0.5000  0.6667
0.2083  0.4167  0.6250  0.8333
0.2500  0.5000  0.7500  1.0000
0.0000  0.0000  0.0000  0.0000
0.0000  0.0000  0.0000  0.0000
```

---

## 2.14 畳み込み

### 2.14.1 DFCN1D, RFCN1D

#### 1次元畳み込み

(1) 機能

任意の整数  $k$  に対して

$$f(i) = f(i + km), g(i) = g(i + km) \quad (i = 0, \dots, m - 1)$$

を満たす 2 組の周期  $m$  の離散関数  $f(i), g(j)$  ただし,

$$f(i) = 0 \quad (i = n_1, \dots, m - 1); \quad g(j) = 0 \quad (j = n_2, \dots, m - 1)$$

が与えられているとき次式で定義される離散畳み込み  $p(k)$  ( $k = 0, \dots, m - 1$ ) を計算する.

$$p(k) = \sum_{i=0}^{m-1} f(i)g(k-i) = \sum_{i=0}^{m-1} g(i)f(k-i) \quad (k = 0, \dots, m - 1)$$

ただし,  $m = \min(n_1 + n_2 - 1, M)$  であり,  $M$  は  $M \geq \max(n_1, n_2)$  を満たす任意の整数である. なお,  $p(k)$  の実フーリエ変換を求めることもできる.

(2) 使用法

倍精度サブルーチン:

CALL DFCN1D (N1, N2, R1, LD1, R2, LD2, M, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFCN1D (N1, N2, R1, LD1, R2, LD2, M, ISW, IWK, WK, IERR)



## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N1	I	1	入 力	離散関数 $f(i)$ の有効データ数 $n_1$
2	N2	I	1	入 力	離散関数 $g(j)$ の有効データ数 $n_2$
3	R1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD1	入 力	離散関数 $f(i)$ の値 (注意事項 (a), (b) 参照)
				出 力	ISW $\geq 1$ のとき離散関数 $f(i)$ の実フーリエ変換結果 (周期 $M$ )
4	LD1	I	1	入 力	配列 R1 の大きさ
5	R2	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD2	入 力	離散関数 $g(j)$ の値 (注意事項 (a), (b) 参照)
				出 力	離散関数 $p(k)$ の値またはその実フーリエ変換 (注意事項 (a), (c) 参照)
6	LD2	I	1	入 力	配列 R2 の大きさ
7	M	I	1	入 力	離散関数 $f(i), g(j), p(k)$ の周期 $m$ に対応するパラメータ $M$ (注意事項 (d) 参照)
8	ISW	I	1	入 力	処理スイッチ (注意事項 (a), (e) 参照) ISW= 0:定義により畳み込みを計算する ISW= 1:FFT 法により畳み込みを計算する ISW= 2:畳み込みの実フーリエ変換を計算する ISW= 3:区分化 FFT 法により畳み込みを計算する
9	IWK	I	内容参照	ワーク	作業領域 大きさ: 0 (ISW= 0 のとき) 20 (ISW $\geq 1$ のとき)
10	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 大きさ: N2 (ISW= 0 のとき) $2 \times M + 1$ (ISW= 1 または 2, M が奇数のとき) $2 \times M + 2$ (ISW= 1 または 2, M が偶数のとき) $2 \times M + N1$ (ISW= 3, M が奇数のとき) $2 \times M + N1 + 1$ (ISW= 3, M が偶数のとき)
11	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, 1, 2, 3\}$

(b)  $N1 > 1$

(c)  $N2 > 1$

(d)  $M \geq \max(N1, N2)$

(e)  $ISW = 0$  の時 :

$LD1 \geq N1$

 $ISW > 0$  で  $M$  が奇数の時 :

$LD1 \geq M + 1$

 $ISW > 0$  で  $M$  が偶数の時 :

$LD1 \geq M + 2$

(f)  $ISW = 0$  の時 :

$LD2 \geq M$

 $ISW > 0$  で  $M$  が奇数の時 :

$LD2 \geq M + 1$

 $ISW > 0$  で  $M$  が偶数の時 :

$LD2 \geq M + 2$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$M < N1 + N2 - 1$ であった.	畳み込みの計算で重なりが発生する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3050	制限条件 (f) を満足しなかった.	

## (6) 注意事項

- (a) 畳み込みを計算する関数値の一方の有効データ数が他方に比較して非常に大きい場合、区分化によりデータ数の多い方を(必要であれば最後に0を付加して)等分し、本サブルーチンを繰り返し適用して離散畳み込みを計算した方が効率が良く、必要な記憶容量も小さくて済む。

たとえば、2つの系列  $\{u_1, u_2, \dots, u_k\}$  (有効データ数  $k$ )、 $\{v_1, \dots, v_{pq}\}$  (有効データ数  $pq$  ( $pq \gg k$ )) の離散畳み込みを計算したい場合、まず  $ISW=1$ ,  $N1=k$ ,  $N2=q$ ,  $M \geq N1 + N2 - 1$ ,  $R1=\{u_1, u_2, \dots, u_k\}$   $R2=\{v_1, v_2, \dots, v_q\}$  として本サブルーチンを適用する。この結果、計算したい畳み込みの最初の  $q$  個の値が配列  $R2$  の先頭から  $q$  個の要素として得られる。

次に、 $ISW=3$ ,  $R2=\{v_{q+1}, \dots, v_{2q}\}$  と変更し、ほかの引数の内容はそのままとして本サブルーチンを適用する。この結果、計算したい畳み込みの次の  $q$  個の値が配列  $R2$  の先頭から  $q$  個の要素として得られる。以下同様にして  $R2$  に設定する値を順次ずらして計算する。なお、最後の繰り返しすなわち  $R2=\{v_{(p-1)q+1}, \dots, v_{pq}\}$  と設定して計算した畳み込みは計算したい畳み込みの最後の  $2q - 1$  個の要素を与える(ただし、系列  $\{v_j\}$  が有限波形でない場合は最後から  $q - 1$  個は不定)。

- (b) 配列  $R1, R2$  にはそれぞれ離散関数  $f(i)$  と離散関数  $g(j)$  の値を次のように格納する。ただし、 $ISW = 3$  とする場合には、 $R2$  にのみ値を格納し、 $R1$  の内容はそのまま利用する(注意事項 (a) 参照)。

$$\begin{aligned} f(0) &\rightarrow R1(1) \\ f(1) &\rightarrow R1(2) \\ \dots &\dots \dots \\ f(n_1 - 1) &\rightarrow R1(N1) \\ \\ g(0) &\rightarrow R2(1) \\ g(1) &\rightarrow R2(2) \\ \dots &\dots \dots \\ g(n_2 - 1) &\rightarrow R2(N2) \end{aligned}$$

なお、配列  $R1, R2$  の  $R1(N1+1)$  並びに  $R2(N2+1)$  以降の要素には値を入力する必要が無い。また、特に、 $ISW = 3$  とする場合には、 $R2(N2+1)$  以降の要素は計算に利用するので変更してはならない。

- (c) 離散畳み込み  $p(k)$  の値は配列  $R2$  に以下のように得られる。

$$\begin{aligned} p(0) &\rightarrow R2(1) \\ p(1) &\rightarrow R2(2) \\ \dots &\dots \dots \\ p(M - 1) &\rightarrow R2(M) \end{aligned}$$

なお、 $M$  が奇数のとき  $R2(M+1)$  が  $M$  が偶数のとき  $R2(M+1)$  と  $R2(M+2)$  がそれぞれ  $0.0$  となる。また、区分化を行う場合には、通常、最初の  $N2$  個のデータが畳み込みとして意味を持つ(注意事項 (a) 参照)。  $ISW=2$  として、離散畳み込み  $p(k)$  の実フーリエ変換  $P(j)$ :

$$P(j) = \frac{1}{M} \sum_{k=0}^{M-1} p(k) e^{-2\pi\sqrt{-1} \frac{jk}{M}} \quad (j = 0, \dots, \lfloor \frac{M}{2} \rfloor)$$

( $\lfloor x \rfloor$  は  $x$  を超えない最大の整数) を求める場合には,

$$\begin{aligned} \Re\{P(0)\} &\leftrightarrow R2(1) \\ \Im\{P(0)\} &\leftrightarrow R2(2) \\ \Re\{P(1)\} &\leftrightarrow R2(3) \\ \Im\{P(1)\} &\leftrightarrow R2(4) \\ \dots &\dots \dots \\ \Re\{P(\lfloor \frac{M}{2} \rfloor)\} &\leftrightarrow R2(l-1) \\ \Im\{P(\lfloor \frac{M}{2} \rfloor)\} &\leftrightarrow R2(l) \quad (l = M+1[M:\text{奇数}] \text{ または } M+2[M:\text{偶数}]) \end{aligned}$$

と対応する。なお、この場合、得られるフーリエ変換は正規化されていることに注意する必要がある。フーリエ変換の残りの半周期分は実フーリエ変換の対称性

$$P(M-j) = P(j)^*$$

(ただし、 $z^*$  は複素数  $z$  の共役複素数) から得られる。

- (d)  $M \geq N1 + N2 - 1$  とすれば、次の周期の畳み込みとの重なりを起こさずに畳み込みを計算できる。  $M > N1 + N2 - 1$  の場合  $N1 + N2$  以降の要素には誤差の範囲で 0.0 と一致する値が格納される。  $ISW=0$  のときは、 $M = N1 + N2 - 1$  とするのがよい。  $ISW \geq 1$  とする場合、 $M$  の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $N1=N2=145$  の場合、 $ISW=0$  のときは、 $M = 289(=17^2)$  とした方が良いが、 $ISW \geq 1$  の場合には  $M = 300(=2^2 \times 3 \times 5^2)$  や  $320(=2^6 \times 5)$ ,  $384(=2^7 \times 3)$  などとした方が通常は効率が良い。
- (e) 通常は  $ISW=1$  と設定して FFT 畳み込みを計算した方が効率良く計算を行える。ただし、作業領域を節約したい場合やパラメータ  $M$  の選び方に制限がある場合などは  $ISW=0$  として計算する。
- (f) 非ゼロ部分の開始位置が原点から離れている離散関数の畳み込みを計算したい場合には、まず開始位置が原点に来るようにシフトして計算した後、計算結果を再度シフトして最終結果を得た方が効率が良い。例えば、離散関数  $f(i)$ ,  $g(j)$  の非ゼロ部分がそれぞれ区間  $[i_0, i_0 + n_1 - 1]$ ,  $[j_0, j_0 + n_2 - 1]$  のとき

$$\hat{f}(i) = f(i - i_0), \quad \hat{g}(j) = g(j - j_0)$$

として  $\hat{f}(i)$ ,  $\hat{g}(j)$  についてこのサブルーチンを適用し、得られた結果を  $\hat{p}(k)$  とすれば、もとの  $f(i)$ ,  $g(j)$  の畳み込み  $p(k)$  は

$$p(k) = \hat{p}(k + (i_0 + j_0))$$

となる。すなわち、離散畳み込みを計算する前に  $f(i)$ ,  $g(j)$  をそれぞれ負の方向に  $i_0$ ,  $j_0$  だけシフトしたとすれば、このサブルーチンの適用後畳み込みの計算値を正の方向に  $i_0 + j_0$  だけシフトすれば所望の結果が得られる。

- (g) このサブルーチンで計算する離散畳み込みに標準化間隔を乗じたものは帯域制限された関数の連続畳み込み積分を方形近似 (台形公式による近似でもある) したものになる。したがって、近似精度を上げるためには、標準化間隔を小さくとり、標本データ数を大きくとる必要がある。なお、連続畳み込みと対応をとる場合には、 $p(n_1 + n_2 - 1) = 0$  として  $p(k)$  ( $k = 0, 1, \dots, n_1 + n_2 - 1$ ) の  $n_1 + n_2$  個のデータを考えた方が、対応をとりやすい。このとき通常は座標 0 の要素は  $p(0)$  に対応させる。ただし、
- $ISW=0$  の場合、 $LD1=N1$ ,  $LD2=M$ ,  $NWK=N2$
- $ISW=1$  または 2 の場合、 $M$  が奇数ならば  $LD1=LD2=M+1$ ,  $NWK = 2 \times M + 1$
- $M$  が偶数ならば  $LD1=LD2=M+2$ ,  $NWK = 2 \times M + 2$
- $ISW=3$  の場合、 $M$  が奇数ならば  $LD1=LD2=M+1$ ,  $NWK = 2 \times M + N1$
- $M$  が偶数ならば  $LD1=LD2=M+2$ ,  $NWK = 2 \times M + N1 + 1$
- である。

(h) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。

## (7) 使用例

### (a) 問題

次式で定義される 2 つの有限波形を標準化間隔  $\Delta x$  で離散化し、離散畳み込みを計算する。

$$f(x) = \begin{cases} x & 0 \leq x \leq a \\ 0 & \text{それ以外} \end{cases}$$

$$g(x) = \begin{cases} b - x & 0 \leq x \leq b \\ 0 & \text{それ以外} \end{cases}$$

### 備考

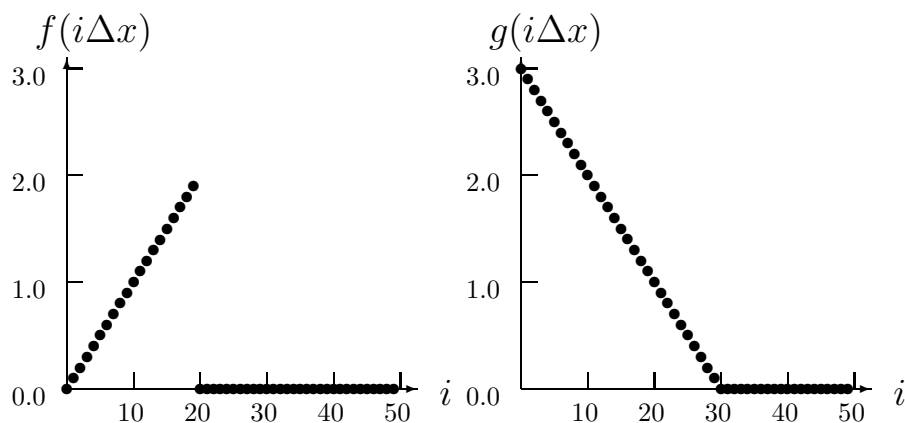
$f(x)$  と  $g(x)$  の連続畳み込み  $p(x) = (f \times g)(x)$  は

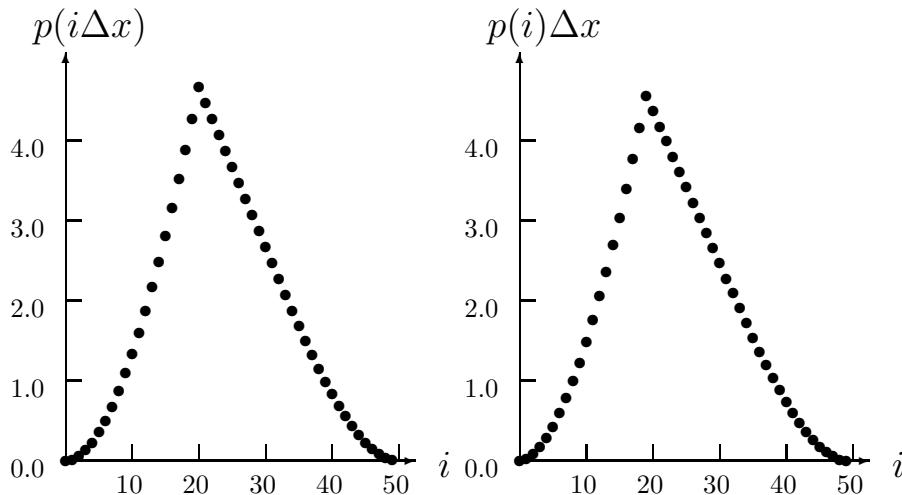
$$p(x) = \int_{-\infty}^{\infty} f(\xi)g(x - \xi)d\xi = \begin{cases} G(0, x, x) & 0 \leq x \leq a \\ G(0, a, x) & a \leq x \leq b \\ G(x - b, a, x) & b \leq x \leq a + b \\ 0 & \text{それ以外} \end{cases}$$

である。ここで、 $G(\alpha, \beta, x)$  は

$$G(\alpha, \beta, x) = \left[ \frac{\xi^2}{6}(3(b-x) + 2\xi) \right]_{\xi=\alpha}^{\beta} \\ = \frac{\xi^2}{6}(3(b-x) + 2\xi) \Big|_{\xi=\beta} - \frac{\xi^2}{6}(3(b-x) + 2\xi) \Big|_{\xi=\alpha}$$

$a = 2, b = 3$  とした場合、 $f(x), g(x), p(x) = (f \times g)(x)$  を  $\Delta x = 0.1$  とし標準化した値  $f(i\Delta x), g(i\Delta x), p(i\Delta x)$  はそれぞれ以下のようなグラフとなる。参考のため本サブルーチンで計算した離散畳み込みに  $\Delta x$  を乗じた値  $p(i)\Delta x$  も示してある。標本数が少ないわりには連続畳み込みの値と良く一致する。





なお、以下の使用例では参考までに連続畳み込みの値も計算している。

(b) 入力データ

標本化データ

$$R1(i) = f((i-1)\Delta x) \quad (i = 1, 2, \dots, N1)$$

$$R2(j) = g((j-1)\Delta x) \quad (j = 1, 2, \dots, N2)$$

ただし、 $\Delta x = 0.1$

$$N1 = \frac{a}{\Delta x}, \quad N2 = \frac{b}{\Delta x}, \quad M, \text{ ISW}$$

(c) 主プログラム

```

PROGRAM BFCN1D
! *** EXAMPLE OF DFCN1D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER I
INTEGER N1,N2,LD1,LD2,M,ISW,IERR,IWK(20)
INTEGER MO
PARAMETER (MO = 100)
PARAMETER (LD1 = MO+2)
PARAMETER (LD2 = MO+2)
REAL(8) R1(LD1),R2(LD2),WK(2*MO+2)
REAL(8) CR(LD2),T,DT
REAL(8) A,B,F
!
ISW=1
DT=0.1D0
A=2.0D0
B=3.0D0
N1=(A+0.5D0*DT)/DT
N2=(B+0.5D0*DT)/DT
M=50
WRITE (6,1000) ISW,N1,N2,M
DO 100 I=1,N1
  T=DBLE(I-1)*DT
  R1(I)=T
100 CONTINUE
DO 200 I=1,N2
  T=DBLE(I-1)*DT
  R2(I)=B-T
200 CONTINUE
!***** ASSUME N2.GT.N1
WRITE (6,1100) (I-1,R1(I),R2(I),I=1,N1),(I-1,R2(I),I=N1+1,N2)
CALL DFCN1D(N1,N2,R1,LD1,R2,LD2,M,ISW,IWK,WK,IERR)
WRITE (6,1300)
WRITE (6,1400) IERR
DO 500 I=1,N1
  T=DBLE(I-1)*DT
  CR(I)=F(T,T,B)
500 CONTINUE
DO 300 I=N1+1,N2
  T=DBLE(I-1)*DT
  CR(I)=F(A,T,B)
300 CONTINUE
DO 400 I=N2+1,N2+N1
  T=DBLE(I-1)*DT
  CR(I)=F(A,T,B)-F(T-B,T,B)
400 CONTINUE
WRITE (6,1200) (I-1,R2(I),R2(I)*DT,CR(I),I=1,N1+N2)
1000 FORMAT(' ',/,/,&
  ' *** DFCN1D ***',/,&
  2X,'** INPUT **',/,&
  6X,'ISW =',I3,/,&
  6X,'N1 =',I3,/,&
  6X,'N2 =',I3,/,&
  6X,'M =',I3)

```

```

1100 FORMAT(12X,'DATA(R1,R2)',/,&
           7X,'I-1',4X,'R1(I)',4X,'R2(I)',/,&
           20(8X,I2,2F9.4,/),10(8X,I2,9X,F9.4,/))
1300 FORMAT(2X,'** OUTPUT **')
1400 FORMAT(6X,'IERR =',I5)
1200 FORMAT(17X,'CONVOLUTION',/,&
           7X,'I-1',4X,'R2(I)',3X,'R2(I)*DT',2X,'CR(I)',/,&
           50(8X,I2,3F9.4,/))

      END
      REAL(8) FUNCTION F(TAU,T,B)
      REAL(8) TAU,T,B
      F=TAU*TAU*(0.5D0*(B-T)+TAU/3.0D0)
      RETURN
      END

```

## (d) 出力結果

```

*** DFCN1D ***
** INPUT **
ISW = 1
N1 = 20
N2 = 30
M = 50

      DATA(R1,R2)
      I-1  R1(I)  R2(I)
      0  0.0000  3.0000
      1  0.1000  2.9000
      2  0.2000  2.8000
      3  0.3000  2.7000
      4  0.4000  2.6000
      5  0.5000  2.5000
      6  0.6000  2.4000
      7  0.7000  2.3000
      8  0.8000  2.2000
      9  0.9000  2.1000
     10  1.0000  2.0000
     11  1.1000  1.9000
     12  1.2000  1.8000
     13  1.3000  1.7000
     14  1.4000  1.6000
     15  1.5000  1.5000
     16  1.6000  1.4000
     17  1.7000  1.3000
     18  1.8000  1.2000
     19  1.9000  1.1000
     20          1.0000
     21          0.9000
     22          0.8000
     23          0.7000
     24          0.6000
     25          0.5000
     26          0.4000
     27          0.3000
     28          0.2000
     29          0.1000

** OUTPUT **
IERR = 0

      CONVOLUTION
      I-1  R2(I)  R2(I)*DT  CR(I)
      0  0.0000  0.0000  0.0000
      1  0.3000  0.0300  0.0148
      2  0.8900  0.0890  0.0587
      3  1.7600  0.1760  0.1305
      4  2.9000  0.2900  0.2293
      5  4.3000  0.4300  0.3542
      6  5.9500  0.5950  0.5040
      7  7.8400  0.7840  0.6778
      8  9.9600  0.9960  0.8747
      9  12.3000  1.2300  1.0935
     10  14.8500  1.4850  1.3333
     11  17.6000  1.7600  1.5932
     12  20.5400  2.0540  1.8720
     13  23.6600  2.3660  2.1688
     14  26.9500  2.6950  2.4827
     15  30.4000  3.0400  2.8125
     16  34.0000  3.4000  3.1573
     17  37.7400  3.7740  3.5162
     18  41.6100  4.1610  3.8880
     19  45.6000  4.5600  4.2718
     20  43.7000  4.3700  4.6667
     21  41.8000  4.1800  4.4667
     22  39.9000  3.9900  4.2667
     23  38.0000  3.8000  4.0667
     24  36.1000  3.6100  3.8667
     25  34.2000  3.4200  3.6667
     26  32.3000  3.2300  3.4667
     27  30.4000  3.0400  3.2667
     28  28.5000  2.8500  3.0667
     29  26.6000  2.6600  2.8667
     30  24.7000  2.4700  2.6667
     31  22.8000  2.2800  2.4668
     32  20.9100  2.0910  2.2680
     33  19.0400  1.9040  2.0712
     34  17.2000  1.7200  1.8773
     35  15.4000  1.5400  1.6875
     36  13.6500  1.3650  1.5027
     37  11.9600  1.1960  1.3238
     38  10.3400  1.0340  1.1520
     39   8.8000  0.8800  0.9882

```

40	7.3500	0.7350	0.8333
41	6.0000	0.6000	0.6885
42	4.7600	0.4760	0.5547
43	3.6400	0.3640	0.4328
44	2.6500	0.2650	0.3240
45	1.8000	0.1800	0.2292
46	1.1000	0.1100	0.1493
47	0.5600	0.0560	0.0855
48	0.1900	0.0190	0.0387
49	0.0000	0.0000	0.0098



## 2.14.2 DFCN2D, RFCN2D

## 2次元畳み込み

## (1) 機能

任意の整数  $L_x, L_y$  に対して

$$\begin{aligned} f(i_x, i_y) &= f(i_x + L_x m_x, i_y + L_y m_y), \\ g(j_x, j_y) &= g(j_x + L_x m_x, j_y + L_y m_y), \\ &(i_x, j_x = 0, \dots, m_x - 1; i_y, j_y = 0, \dots, m_y - 1) \end{aligned}$$

を満たす2組の周期  $(m_x, m_y)$  の多重周期離散関数  $f(i_x, i_y), g(j_x, j_y)$  についてこれらがそれぞれ基本周期内では  $(i_x, i_y) \in [0, n_x^{(f)} - 1] \times [0, n_y^{(f)} - 1], (j_x, j_y) \in [0, n_x^{(g)} - 1] \times [0, n_y^{(g)} - 1]$  でのみ非ゼロ値をとるとする。ここで,  $[0, a] \times [0, b]$  は平面座標  $(i, j)$  がはる平面の直積領域 (点  $(0, 0)$  と点  $(a, b)$  を対角頂点とする長方形で囲まれる領域) とする。このとき, 次式で定義される離散畳み込み  $p(k_x, k_y)$  を計算する。

$$\begin{aligned} p(k_x, k_y) &= \sum_{i_x=0}^{m_x-1} \sum_{i_y=0}^{m_y-1} f(i_x, i_y) g(k_x - i_x, k_y - i_y) \\ &= \sum_{j_x=0}^{m_x-1} \sum_{j_y=0}^{m_y-1} g(j_x, j_y) f(k_x - j_x, k_y - j_y) \\ &(k_x = 0, \dots, m_x - 1; k_y = 0, \dots, m_y - 1) \end{aligned}$$

ただし,  $m_x = \min(n_x^{(f)} + n_x^{(g)} - 1, M_x), m_y = \min(n_y^{(f)} + n_y^{(g)} - 1, M_y)$  であり,  $M_x, M_y$  はそれぞれ  $M_x \geq \max(n_x^{(f)}, n_x^{(g)}), M_y \geq \max(n_y^{(f)}, n_y^{(g)})$  を満たす任意の整数である。なお,  $p(k_x, k_y)$  の2次元実フーリエ変換を求めることもできる。

## (2) 使用法

倍精度サブルーチン:

CALL DFCN2D (NX1, NY1, NX2, NY2, R1, LX1, LY1, R2, LX2, LY2, MX, MY, ISW, IWK,  
WK, IERR)

単精度サブルーチン:

CALL RFCN2D (NX1, NY1, NX2, NY2, R1, LX1, LY1, R2, LX2, LY2, MX, MY, ISW, IWK,  
WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{array} \right\}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX1	I	1	入力	離散関数 $f(i_x, i_y)$ の $i_x$ 方向の有効データ数 $n_x^{(f)}$
2	NY1	I	1	入力	離散関数 $f(i_x, i_y)$ の $i_y$ 方向の有効データ数 $n_y^{(f)}$
3	NX2	I	1	入力	離散関数 $g(j_x, j_y)$ の $j_x$ 方向の有効データ数 $n_x^{(g)}$
4	NY2	I	1	入力	離散関数 $g(j_x, j_y)$ の $j_y$ 方向の有効データ数 $n_y^{(g)}$
5	R1	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX1, LY1	入力	離散関数 $f(i_x, i_y)$ の値 (注意事項 (a) 参照)
				出力	ISW $\geq 1$ のとき離散関数 $f(i_x, i_y)$ の 2 次元実フーリエ変換結果 (周期 ( $M_x, M_y$ ))
6	LX1	I	1	入力	配列 R1 の整合寸法
7	LY1	I	1	入力	配列 R1 の第 2 寸法
8	R2	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX2, LY2	入力	離散関数 $g(j_x, j_y)$ の値 (注意事項 (a) 参照)
				出力	離散関数 $p(k_x, k_y)$ の値またはその 2 次元実フーリエ変換 (注意事項 (b) 参照)
9	LX2	I	1	入力	配列 R2 の整合寸法
10	LY2	I	1	入力	配列 R2 の第 2 寸法
11	MX	I	1	入力	離散関数 $f(i_x, i_y), g(j_x, j_y), p(k_x, k_y)$ の周期 ( $m_x, m_y$ ) に対応するパラメータ $M_x$ (注意事項 (c) 参照)
12	MY	I	1	入力	離散関数 $f(i_x, i_y), g(j_x, j_y), p(k_x, k_y)$ の周期 ( $m_x, m_y$ ) に対応するパラメータ $M_y$ (注意事項 (c) 参照)
13	ISW	I	1	入力	処理スイッチ (注意事項 (d) 参照) ISW = 0: 定義により畳み込みを計算する ISW = 1: FFT 法により畳み込みを計算する ISW = 2: 畳み込みの実フーリエ変換を計算する
14	IWK	I	内容参照	ワーク	作業領域 大きさ: 0 (ISW = 0 のとき) 40 (ISW $\geq 1$ のとき)
15	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	内容参照	ワーク	作業領域 大きさ: NX2 $\times$ NY2 (ISW = 0, NX2: 奇数のとき) (NX2 + 1) $\times$ NY2 (ISW = 0, NX2: 偶数のとき) MX + 2 $\times$ MY + MAX(LX1 $\times$ LY1, LX2 $\times$ LY2) (ISW $\geq 1$ のとき)
16	IERR	I	1	出力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, 1, 2\}$

(b)  $NX1 > 1$   
 $NY1 > 1$

(c)  $NX2 > 1$   
 $NY2 > 1$

(d)  $MX \geq \max(NX1, NX2)$   
 $MY \geq \max(NY1, NY2)$

(e)  $ISW = 0$  の時 :

$LX1 \geq NX1$

$LY1 \geq NY1$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX1 \geq MX + 1$

$LY1 \geq MY$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX1 \geq MX + 2$

$LY1 \geq MY$

(f)  $ISW = 0$  の時 :

$LX2 \geq MX$

$LY2 \geq MY$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX2 \geq MX + 1$

$LY2 \geq MY$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX2 \geq MX + 2$

$LY2 \geq MY$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$MX < NX1 + NX2 - 1$ または $MY < NY1 + NY2 - 1$ であった.	畳み込みの計算で重なりが発生する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3050	制限条件 (f) を満足しなかった.	

## (6) 注意事項

- (a) 配列 R1, R2 の各要素と離散関数
- $f(i_x, i_y)$
- と離散関数
- $g(j_x, j_y)$
- の値は以下の様に対応する。

$$\begin{aligned} f(i_x, i_y) &\leftrightarrow R1(i_x + 1, i_y + 1) \\ g(j_x, j_y) &\leftrightarrow R2(j_x + 1, j_y + 1) \end{aligned}$$

ただし,  $i_x = 0, \dots, n_x^{(f)} - 1$ ;  $i_y = 0, \dots, n_y^{(f)} - 1$ ,  $j_x = 0, \dots, n_x^{(g)} - 1$ ;  $j_y = 0, \dots, n_y^{(g)} - 1$  であり, それ以外の要素には値を入力する必要が無い。なお, 主記憶のバンク競合を避けるために配列 R1, R2 の整合寸法について LX1/2, LY1, LX2/2, LY2 が奇数になるように設定するのが望ましい。通常, たとえば MX が 4 の倍数のときは LX1=MX+3 とする。

- (b) 離散畳み込み
- $p(k_x, k_y)$
- の値は配列 R2 の各要素と以下の様に対応する。

$$p(k_x, k_y) \leftrightarrow R2(k_x + 1, k_y + 1)$$

ただし,  $k_x = 0, \dots, M_x - 1$ ;  $k_y = 0, \dots, M_y - 1$  である。ISW=2 として, 離散畳み込み  $p(k_x, k_y)$  の 2 次元実フーリエ変換  $P(j_x, j_y)$ :

$$\begin{aligned} P(j_x, j_y) &= \frac{1}{M_x M_y} \sum_{k_x=0}^{M_x-1} \sum_{k_y=0}^{M_y-1} p(k_x, k_y) e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{M_x} + \frac{j_y k_y}{M_y}\right)} \\ &\quad (j_x = 0, \dots, \lfloor \frac{M_x}{2} \rfloor; j_y = 0, \dots, \lfloor \frac{M_y}{2} \rfloor) \end{aligned}$$

( $\lfloor x \rfloor$  は  $x$  を超えない最大の整数) を求める場合には,

$$\begin{aligned} \Re\{P(j_x, j_y)\} &\leftrightarrow R2(2 * j_x + 1, j_y + 1) \\ \Im\{P(j_x, j_y)\} &\leftrightarrow R2(2 * j_x + 2, j_y + 1) \end{aligned}$$

と対応する。なお, この場合, 得られるフーリエ変換は正規化されていることに注意する必要がある。フーリエ変換の残りの半周期分は実フーリエ変換の対称性

$$\begin{aligned} P(M_x - j_x, M_y - j_y)^* &= P(j_x, j_y) \\ P(M_x - j_x, j_y)^* &= P(j_x, M_y - j_y) \end{aligned}$$

(ただし,  $z^*$  は複素数  $z$  の共役複素数) から得られる。

- (c)
- $MX \geq NX1 + NX2 - 1$
- かつ
- $MY \geq NY1 + NY2 - 1$
- とすれば, 次の周期の畳み込みとの重なりを起こさずに畳み込みを計算できる。
- $MX > NX1 + NX2 - 1$
- または
- $MY > NY1 + NY2 - 1$
- の場合

$$p(k_x, k_y) \leftrightarrow R2(k_x + 1, k_y + 1)$$

$k_x = NX1 + NX2 - 1, \dots, MX - 1$ ;  $k_y = 0, \dots, MY - 1$  または  $k_x = 0, \dots, MX - 1$ ;  $k_y = NY1 + NY2 - 1, \dots, MY - 1$  に対応する要素には誤差の範囲で 0.0 と一致する値が格納される。ISW=0 のときは,  $MX = NX1 + NX2 - 1$ ,  $MY = NY1 + NY2 - 1$  とするのがよい。ISW  $\geq 1$  とする場合, MX, MY の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば,  $NX1=NX2=145$  の場合, ISW=0 のときは,  $MX = 289(=17^2)$  とした方がよいが, ISW  $\geq 1$  の場合には  $MX = 300(=2^2 \times 3 \times 5^2)$  や  $320(=2^6 \times 5)$ ,  $384(=2^7 \times 3)$  などとした方が通常は効率が良い。

- (d) 通常は ISW=1 と設定して FFT 畳み込みを計算した方が効率良く計算を行える。ただし, 作業領域を節約したい場合やパラメータ MX や MY の選び方に制限がある場合などは ISW=0 として計算する。

- (e) 非ゼロ部分の開始位置が原点から離れている離散関数の畳み込みを計算したい場合には, まず開始位置が原点に来るようにシフトして計算した後, 計算結果を再度シフトして最終結果を得た方が効率が良い。例えば, 離散関数
- $f(i_x, i_y)$
- ,
- $g(j_x, j_y)$
- の非ゼロ部分が
- $i_x, j_x$
- についてそれぞれ区間
- $[i_0, i_0 + n_x^{(f)} - 1]$
- ,
- $[j_0, j_0 + n_x^{(g)} - 1]$
- のとき

$$\hat{f}(i_x, i_y) = f(i_x - i_0, i_y), \quad \hat{g}(j_x, j_y) = g(j_x - j_0, j_y)$$

として  $\hat{f}(i_x, i_y)$ ,  $\hat{g}(j_x, j_y)$  についてこのサブルーチンを適用し, 得られた結果を  $\hat{p}(k_x, k_y)$  とすれば, もとの  $f(i_x, i_y)$ ,  $g(j_x, j_y)$  の畳み込み  $p(k_x, k_y)$  は

$$p(k_x, k_y) = \hat{p}(k_x + (i_0 + j_0), k_y)$$

となる. すなわち, 離散畳み込みを計算する前に  $f(i_x, i_y)$ ,  $g(j_x, j_y)$  をそれぞれ  $i_x, j_x$  の負の方向に  $i_0, j_0$  だけシフトしたとすれば, このサブルーチンの適用後畳み込みの計算値を  $k_x$  の正の方向に  $i_0 + j_0$  だけシフトすれば所望の結果が得られる.  $i_y, j_y, k_y$  についても同様である.

- (f) このサブルーチンで計算する離散畳み込みに標本化間隔の2乗を乗じたものは帯域制限された関数の連続畳み込み積分を方形近似(台形公式による近似でもある)したものになる. したがって, 近似精度を上げるためには, 標本化間隔を小さくとり, 標本データ数を大きくとる必要がある. なお, 連続畳み込みと対応をとる場合には,  $p(n_x^{(f)} + n_x^{(g)} - 1, k_y) = 0$ ,  $p(k_x, n_y^{(f)} + n_y^{(g)} - 1) = 0$  として  $p(k_x, k_y)$  ( $k_x = 0, 1, \dots, n_x^{(f)} + n_x^{(g)} - 1$ ;  $k_y = 0, 1, \dots, n_y^{(f)} + n_y^{(g)} - 1$ ) の  $(n_x^{(f)} + n_x^{(g)})(n_y^{(f)} + n_y^{(g)})$  個のデータを考えた方が, 対応をとりやすい. このとき通常は座標  $(0, 0)$  の要素は  $p(0, 0)$  に対応させる. ただし,

ISW=0の場合,

$$LX1 = NX1, LY1 = NY1, LX2 = MX, LY2 = MY,$$

$$NWK = NX2 \times NY2 (NX2:奇数のとき) \text{ または}$$

$$NWK = (NX2 + 1) \times NY2 (NX2:偶数のとき)$$

ISW  $\geq 1$ の場合,

$$LX1=LX2=MX+1 (MX \text{ が奇数のとき}) \text{ または}$$

$$LX1=LX2=MX+2 (MX \text{ が偶数のとき}),$$

$$LY1=LY2=MY, NWK = MX + (LX1 + 2) \times MY$$

である.

- (g) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない.

## (7) 使用例

## (a) 問題

次式で定義される2つの有限波形を標準化間隔 $\Delta$ で離散化し、離散畳み込みを計算する。

$$f(x, y) = \begin{cases} x & ((x, y) \in [0, x_f] \times [0, y_f]) \\ 0 & (\text{それ以外}) \end{cases}$$

$$g(x, y) = \begin{cases} x_g - x & ((x, y) \in [0, x_g] \times [0, y_g]) \\ 0 & (\text{それ以外}) \end{cases}$$

## (b) 入力データ

標準化データ

$$R1(i_x + 1, i_y + 1) = f(i_x \Delta, i_y \Delta) \quad (i_x = 0, 1, \dots, NX1 - 1; i_y = 0, 1, \dots, NY1 - 1)$$

$$R2(j_x + 1, j_y + 1) = g(j_x \Delta, j_y \Delta) \quad (j_x = 0, 1, \dots, NX2 - 1; j_y = 0, 1, \dots, NY2 - 1)$$

ただし、 $\Delta = 0.5$

$NX1, NY1, NX2, NY2, MX, MY, ISW$

## (c) 主プログラム

```

PROGRAM BFCN2D
! *** EXAMPLE OF DFCN2D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER I,J
INTEGER ISW,IERR,IWK(40)
INTEGER NX1,NX2,LX1,LX2,MX
INTEGER NY1,NY2,LY1,LY2,MY
INTEGER MO
PARAMETER (MO = 8)
PARAMETER (LX1 = MO+2)
PARAMETER (LY1 = MO)
PARAMETER (LX2 = MO+2)
PARAMETER (LY2 = MO)
REAL(8) R1(LX1,LY1),R2(LX2,LY2),WK(3*MO+LX2*MO)
REAL(8) T
REAL(8) XF,YF,XG,YG,DT
PARAMETER (DT = 0.5D0)
PARAMETER (XF = 2.0D0,YF=2.0D0)
PARAMETER (XG = 2.0D0,YG=2.0D0)
!
ISW=1
NX1=XF/DT
NY1=YF/DT
NX2=XG/DT
NY2=YG/DT
MX=MO
MY=MO
WRITE (6,1000) ISW,NX1,NY1,NX2,NY2,MX,MY
DO 100 J=1,NY1
DO 101 I=1,NX1
T=DBLE(I-1)*DT
R1(I,J)=T
101 CONTINUE
100 CONTINUE
DO 200 J=1,NY2
DO 201 I=1,NX2
T=DBLE(I-1)*DT
R2(I,J)=XG-T
201 CONTINUE
200 CONTINUE
WRITE (6,1100) (I,(R1(I,J),J=1,NY1),I=1,NX1)
WRITE (6,1150) (I,(R2(I,J),J=1,NY2),I=1,NX2)
CALL DFCN2D(NX1,NY1,NX2,NY2,R1,LX1,LY1,&
R2,LX2,LY2,MX,MY,ISW,IWK,WK,IERR)
WRITE (6,1300)
WRITE (6,1400) IERR
WRITE (6,1200)&
(I,(R2(I,J),J=1,MY),I=1,MX)
1000 FORMAT(' ',/,/,&
' *** DFCN2D ***',/,&
2X,'** INPUT **',/,&
6X,'ISW =',I3,/,&
6X,'(NX1,NY1) =(',I3,',',I3,')',/,&
6X,'(NX2,NY2) =(',I3,',',I3,')',/,&
6X,'(MX, MY) =(',I3,',',I3,')')
1100 FORMAT(12X,'DATA R1(I,J)',/,&
10X,'I/J 1 2 3 4',/,&
10X,'-----',/,&
6(8X,I3,4F9.4,/) )
1150 FORMAT(12X,'DATA R2(I,J)',/,&
10X,'I/J 1 2 3 4',/,&
10X,'-----',/,&
6(8X,I3,4F9.4,/) )
1300 FORMAT(2X,'** OUTPUT **')
1400 FORMAT(6X,'IERR =',I5)

```

```

1200 FORMAT(17X,'CONVOLUTION R2(I,J)',/,&
10X,'I/J 1 2 3 4 5',&
10X,'-----',&
8(8X,I3,8F7.2,/)
END

```

## (d) 出力結果

```

*** DFCN2D ***
** INPUT **
ISW = 1
(NX1,NY1) = ( 4, 4)
(NX2,NY2) = ( 4, 4)
(MX, MY) = ( 8, 8)
DATA R1(I,J)
I/J 1 2 3 4
-----
1 0.0000 0.0000 0.0000 0.0000
2 0.5000 0.5000 0.5000 0.5000
3 1.0000 1.0000 1.0000 1.0000
4 1.5000 1.5000 1.5000 1.5000
DATA R2(I,J)
I/J 1 2 3 4
-----
1 2.0000 2.0000 2.0000 2.0000
2 1.5000 1.5000 1.5000 1.5000
3 1.0000 1.0000 1.0000 1.0000
4 0.5000 0.5000 0.5000 0.5000
** OUTPUT **
IERR = 0
CONVOLUTION R2(I,J)
I/J 1 2 3 4 5 6 7 8
-----
1 0.00 0.00 0.00 0.00 0.00 0.00 -0.00 -0.00
2 1.00 2.00 3.00 4.00 3.00 2.00 1.00 -0.00
3 2.75 5.50 8.25 11.00 8.25 5.50 2.75 -0.00
4 5.00 10.00 15.00 20.00 15.00 10.00 5.00 -0.00
5 3.50 7.00 10.50 14.00 10.50 7.00 3.50 -0.00
6 2.00 4.00 6.00 8.00 6.00 4.00 2.00 -0.00
7 0.75 1.50 2.25 3.00 2.25 1.50 0.75 -0.00
8 -0.00 0.00 0.00 0.00 -0.00 -0.00 -0.00 -0.00

```

### 2.14.3 DFCN3D, RFCN3D 3次元畳み込み

#### (1) 機能

任意の整数  $L_x, L_y, L_z$  に対して

$$\begin{aligned} f(i_x, i_y, i_z) &= f(i_x + L_x m_x, i_y + L_y m_y, i_z + L_z m_z), \\ g(j_x, j_y, j_z) &= g(j_x + L_x m_x, j_y + L_y m_y, j_z + L_z m_z), \\ &(i_x, j_x = 0, \dots, m_x - 1; i_y, j_y = 0, \dots, m_y - 1; i_z, j_z = 0, \dots, m_z - 1) \end{aligned}$$

を満たす2組の周期  $(m_x, m_y, m_z)$  の多重周期離散関数  $f(i_x, i_y, i_z), g(j_x, j_y, j_z)$  についてこれらがそれぞれ基本周期内では  $(i_x, i_y, i_z) \in [0, n_x^{(f)} - 1] \times [0, n_y^{(f)} - 1] \times [0, n_z^{(f)} - 1], (j_x, j_y, j_z) \in [0, n_x^{(g)} - 1] \times [0, n_y^{(g)} - 1] \times [0, n_z^{(g)} - 1]$  でのみ非ゼロ値をとるとする。ここで、 $[0, a] \times [0, b] \times [0, c]$  は空間座標  $(i, j, k)$  がはる空間の直積領域 (点  $(0, 0, 0)$  と点  $(a, b, c)$  を対角頂点とする直方体で囲まれる領域) とする。このとき、次式で定義される離散畳み込み  $p(k_x, k_y, k_z)$  を計算する。

$$\begin{aligned} p(k_x, k_y, k_z) &= \sum_{i_x=0}^{m_x-1} \sum_{i_y=0}^{m_y-1} \sum_{i_z=0}^{m_z-1} f(i_x, i_y, i_z) g(k_x - i_x, k_y - i_y, k_z - i_z) \\ &= \sum_{j_x=0}^{m_x-1} \sum_{j_y=0}^{m_y-1} \sum_{j_z=0}^{m_z-1} g(j_x, j_y, j_z) f(k_x - j_x, k_y - j_y, k_z - j_z) \\ &(k_x = 0, \dots, m_x - 1; k_y = 0, \dots, m_y - 1; k_z = 0, \dots, m_z - 1) \end{aligned}$$

ただし、 $m_x = \min(n_x^{(f)} + n_x^{(g)} - 1, M_x), m_y = \min(n_y^{(f)} + n_y^{(g)} - 1, M_y), m_z = \min(n_z^{(f)} + n_z^{(g)} - 1, M_z)$  であり、 $M_x, M_y, M_z$  はそれぞれ  $M_x \geq \max(n_x^{(f)}, n_x^{(g)}), M_y \geq \max(n_y^{(f)}, n_y^{(g)}), M_z \geq \max(n_z^{(f)}, n_z^{(g)})$  を満たす任意の整数である。なお、 $p(k_x, k_y, k_z)$  の3次元実フーリエ変換を求めることもできる。

#### (2) 使用法

倍精度サブルーチン:

CALL DFCN3D (NX1, NY1, NZ1, NX2, NY2, NZ2, R1, LX1, LY1, LZ1, R2, LX2, LY2, LZ2,  
MX, MY, MZ, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFCN3D (NX1, NY1, NZ1, NX2, NY2, NZ2, R1, LX1, LY1, LZ1, R2, LX2, LY2, LZ2,  
MX, MY, MZ, ISW, IWK, WK, IERR)



## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX1	I	1	入力	離散関数 $f(i_x, i_y, i_z)$ の $i_x$ 方向の有効データ数 $n_x^{(f)}$
2	NY1	I	1	入力	離散関数 $f(i_x, i_y, i_z)$ の $i_y$ 方向の有効データ数 $n_y^{(f)}$
3	NZ1	I	1	入力	離散関数 $f(i_x, i_y, i_z)$ の $i_z$ 方向の有効データ数 $n_z^{(f)}$
4	NX2	I	1	入力	離散関数 $g(j_x, j_y, j_z)$ の $j_x$ 方向の有効データ数 $n_x^{(g)}$
5	NY2	I	1	入力	離散関数 $g(j_x, j_y, j_z)$ の $j_y$ 方向の有効データ数 $n_y^{(g)}$
6	NZ2	I	1	入力	離散関数 $g(j_x, j_y, j_z)$ の $j_z$ 方向の有効データ数 $n_z^{(g)}$
7	R1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX1, LY1, LZ1	入力	離散関数 $f(i_x, i_y, i_z)$ の値 (注意事項 (a) 参照)
				出力	ISW $\geq 1$ のとき離散関数 $f(i_x, i_y, i_z)$ の3次元実フーリエ変換結果 (周期 ( $M_x, M_y, M_z$ ))
8	LX1	I	1	入力	配列 R1 の整合寸法
9	LY1	I	1	入力	配列 R1 の第2寸法
10	LZ1	I	1	入力	配列 R1 の第3寸法
11	R2	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX2, LY2, LZ2	入力	離散関数 $g(j_x, j_y, j_z)$ の値 (注意事項 (a) 参照)
				出力	離散関数 $p(k_x, k_y, k_z)$ の値またはその3次元実フーリエ変換 (注意事項 (b) 参照)
12	LX2	I	1	入力	配列 R2 の整合寸法
13	LY2	I	1	入力	配列 R2 の第2寸法
14	LZ2	I	1	入力	配列 R2 の第3寸法
15	MX	I	1	入力	離散関数 $f(i_x, i_y, i_z), g(j_x, j_y, j_z), p(k_x, k_y, k_z)$ の周期 ( $m_x, m_y, m_z$ ) に対応するパラメータ $M_x$ (注意事項 (c) 参照)
16	MY	I	1	入力	離散関数 $f(i_x, i_y, i_z), g(j_x, j_y, j_z), p(k_x, k_y, k_z)$ の周期 ( $m_x, m_y, m_z$ ) に対応するパラメータ $M_y$ (注意事項 (c) 参照)

項番	引数名	型	大きさ	入出力	内 容
17	MZ	I	1	入 力	離散関数 $f(i_x, i_y, i_z), g(j_x, j_y, j_z), p(k_x, k_y, k_z)$ の周期 $(m_x, m_y, m_z)$ に対応するパラメータ $M_z$ (注意事項 (c) 参照)
18	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW= 0:定義により畳み込みを計算する ISW= 1:FFT 法により畳み込みを計算する ISW= 2:畳み込みの実フーリエ変換を計算する
19	IWK	I	内容参照	ワーク	作業領域 大きさ: 0 (ISW= 0 のとき) 60 (ISW $\geq$ 1 のとき)
20	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 大きさ: (NX2 + 1) $\times$ (NY2 + 1) $\times$ NZ2 (ISW= 0, NX2:偶数, NY2:偶数のとき) NX2 $\times$ (NY2+1) $\times$ NZ2 (ISW= 0, NX2:奇数, NY2:偶数のとき) (NX2+1) $\times$ NY2 $\times$ NZ2 (ISW= 0, NX2:偶数, NY2:奇数のとき) NX2 $\times$ NY2 $\times$ NZ2 (ISW= 0, NX2:奇数, NY2:奇数のとき) MX + 2 $\times$ (MY + MZ) + MAX(LX1 $\times$ LY1 $\times$ LZ1, LX2 $\times$ LY2 $\times$ LZ2) (ISW $\geq$ 1 のとき)
21	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, 1, 2\}$

(b)  $NX1 > 1$

$NY1 > 1$

$NZ1 > 1$

(c)  $NX2 > 1$

$NY2 > 1$

$NZ2 > 1$

(d)  $MX \geq \max(NX1, NX2)$

$MY \geq \max(NY1, NY2)$

$MZ \geq \max(NZ1, NZ2)$

(e)  $ISW = 0$  の時 :

$LX1 \geq NX1$

$LY1 \geq NY1$

$LZ1 \geq NZ1$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX1 \geq MX + 1$ ,  $LX1$  は偶数

$LY1 \geq MY$

$LZ1 \geq MZ$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX1 \geq MX + 2$ ,  $LX1$  は偶数

$LY1 \geq MY$

$LZ1 \geq MZ$

(f)  $ISW = 0$  の時 :

$LX2 \geq MX$

$LY2 \geq NY2$

$LZ2 \geq NZ2$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX2 \geq MX + 1$ ,  $LX2$  は偶数

$LY2 \geq MY$

$LZ2 \geq MZ$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX2 \geq MX + 2$ ,  $LX2$  は偶数

$LY2 \geq MY$

$LZ2 \geq MZ$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	MX < NX1 + NX2 - 1, MY < NY1 + NY2 - 1 または MZ < NZ1 + NZ2 - 1 であった.	畳み込みの計算で重なりが発生する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3050	制限条件 (f) を満足しなかった.	

## (6) 注意事項

- (a) 配列 R1, R2 の各要素と離散関数
- $f(i_x, i_y, i_z)$
- と離散関数
- $g(j_x, j_y, j_z)$
- の値は以下の様に対応する.

$$f(i_x, i_y, i_z) \leftrightarrow R1(i_x + 1, i_y + 1, i_z + 1)$$

$$g(j_x, j_y, j_z) \leftrightarrow R2(j_x + 1, j_y + 1, j_z + 1)$$

ただし,  $i_x = 0, \dots, n_x^{(f)} - 1$ ;  $i_y = 0, \dots, n_y^{(f)} - 1$ ;  $i_z = 0, \dots, n_z^{(f)} - 1$ ;  $j_x = 0, \dots, n_x^{(g)} - 1$ ;  $j_y = 0, \dots, n_y^{(g)} - 1$ ;  $j_z = 0, \dots, n_z^{(g)} - 1$  であり, それ以外の要素には値を入力する必要が無い. なお, 主記憶のバンク競合を避けるために配列 R1, R2 の整合寸法について LX1/2, LY1, LZ1, LX2/2, LY2, LZ2 が奇数になるように設定するのが望ましい. また, 高速化のために配列 R1, R2 内のデータ設定領域以外の要素に対しても演算を実行する. 通常, たとえば MX が (4 の倍数)+2 のときは LX1=MX+4 とする.

- (b) 離散畳み込み
- $p(k_x, k_y, k_z)$
- の値は配列 R2 の各要素と以下の様に対応する.

$$p(k_x, k_y, k_z) \leftrightarrow R2(k_x + 1, k_y + 1, k_z + 1)$$

ただし,  $k_x = 0, \dots, M_x - 1$ ;  $k_y = 0, \dots, M_y - 1$ ;  $k_z = 0, \dots, M_z - 1$  である. ISW=2 として, 離散畳み込み  $p(k_x, k_y, k_z)$  の 3 次元実フーリエ変換  $P(j_x, j_y, j_z)$ :

$$P(j_x, j_y, j_z) = \frac{1}{M_x M_y M_z} \sum_{k_x=0}^{M_x-1} \sum_{k_y=0}^{M_y-1} \sum_{k_z=0}^{M_z-1} p(k_x, k_y, k_z) e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{M_x} + \frac{j_y k_y}{M_y} + \frac{j_z k_z}{M_z}\right)}$$

$$(j_x = 0, \dots, \lfloor \frac{M_x}{2} \rfloor; j_y = 0, \dots, \lfloor \frac{M_y}{2} \rfloor; j_z = 0, \dots, \lfloor \frac{M_z}{2} \rfloor)$$

( $\lfloor x \rfloor$  は  $x$  を超えない最大の整数) を求める場合には,

$$\Re\{P(j_x, j_y, j_z)\} \leftrightarrow R2(2 * j_x + 1, j_y + 1, j_z + 1)$$

$$\Im\{P(j_x, j_y, j_z)\} \leftrightarrow R2(2 * j_x + 2, j_y + 1, j_z + 1)$$

と対応する. なお, この場合, 得られるフーリエ変換は正規化されていることに注意する必要がある. フーリエ変換の残りの半周期分は実フーリエ変換の対称性

$$P(M_x - j_x, M_y - j_y, M_z - j_z)^* = P(j_x, j_y, j_z)$$

$$P(M_x - j_x, j_y, j_z)^* = P(j_x, M_y - j_y, M_z - j_z)$$

$$P(M_x - j_x, M_y - j_y, j_z)^* = P(j_x, j_y, M_z - j_z)$$

(ただし,  $z^*$  は複素数  $z$  の共役複素数) から得られる.

- (c)  $MX \geq NX1 + NX2 - 1$  かつ  $MY \geq NY1 + NY2 - 1$  かつ  $MZ \geq NZ1 + NZ2 - 1$  とすれば、次の周期の畳み込みとの重なりを起こさずに畳み込みを計算できる。  $MX > NX1 + NX2 - 1$  または  $MY > NY1 + NY2 - 1$  または  $MZ > NZ1 + NZ2 - 1$  の場合

$$p(k_x, k_y, k_z) \leftrightarrow R2(k_x + 1, k_y + 1, k_z + 1)$$

$k_x = NX1 + NX2 - 1, \dots, MX - 1$ ;  $k_y = 0, \dots, MY - 1$ ;  $k_z = 0, \dots, MZ - 1$  または  $k_x = 0, \dots, MX - 1$ ;  $k_y = NY1 + NY2 - 1, \dots, MY - 1$ ;  $k_z = 0, \dots, MZ - 1$  または  $k_x = 0, \dots, MX - 1$ ;  $k_y = 0, \dots, MY - 1$ ;  $k_z = NZ1 + NZ2 - 1, \dots, MZ - 1$  に対応する要素には誤差の範囲で 0.0 と一致する値が格納される。ISW=0 のときは、 $MX = NX1 + NX2 - 1$ ,  $MY = NY1 + NY2 - 1$ ,  $MZ = NZ1 + NZ2 - 1$  とするのがよい。ISW  $\geq 1$  とする場合、MX, MY, MZ の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $NX1=NX2=145$  の場合、ISW=0 のときは、 $MX = 289(=17^2)$  とした方がよいが、ISW  $\geq 1$  の場合には  $MX = 300(=2^2 \times 3 \times 5^2)$  や  $320(=2^6 \times 5)$ ,  $384(=2^7 \times 3)$  などとした方が通常は効率が良い。

- (d) 通常は ISW=1 と設定して FFT 畳み込みを計算した方が効率良く計算を行える。ただし、作業領域を節約したい場合やパラメータ MX や MY, MZ の選び方に制限がある場合などは ISW=0 として計算する。
- (e) 非ゼロ部分の開始位置が原点から離れている離散関数の畳み込みを計算したい場合には、まず開始位置が原点に来るようにシフトして計算した後、計算結果を再度シフトして最終結果を得た方が効率が良い。例えば、離散関数  $f(i_x, i_y, i_z)$ ,  $g(j_x, j_y, j_z)$  の非ゼロ部分が  $i_x, j_x$  についてそれぞれ区間  $[i_0, i_0 + n_x^{(f)} - 1]$ ,  $[j_0, j_0 + n_x^{(g)} - 1]$  のとき

$$\hat{f}(i_x, i_y, i_z) = f(i_x - i_0, i_y, i_z), \quad \hat{g}(j_x, j_y, j_z) = g(j_x - j_0, j_y, j_z)$$

として  $\hat{f}(i_x, i_y, i_z)$ ,  $\hat{g}(j_x, j_y, j_z)$  についてこのサブルーチンを適用し、得られた結果を  $\hat{p}(k_x, k_y, k_z)$  とすれば、もとの  $f(i_x, i_y, i_z)$ ,  $g(j_x, j_y, j_z)$  の畳み込み  $p(k_x, k_y, k_z)$  は

$$p(k_x, k_y, k_z) = \hat{p}(k_x + (i_0 + j_0), k_y, k_z)$$

となる。すなわち、離散畳み込みを計算する前に  $f(i_x, i_y, i_z)$ ,  $g(j_x, j_y, j_z)$  をそれぞれ  $i_x, j_x$  の負の方向に  $i_0, j_0$  だけシフトしたとすれば、このサブルーチンの適用後畳み込みの計算値を  $k_x$  の正の方向に  $i_0 + j_0$  だけシフトすれば所望の結果が得られる。 $i_y, j_y, k_y$ ;  $i_z, j_z, k_z$  についても同様である。

- (f) このサブルーチンで計算する離散畳み込みに標準化間隔の 3 乗を乗じたものは帯域制限された関数の連続畳み込み積分を方形近似 (台形公式による近似でもある) したものになる。したがって、近似精度を上げるためには、標準化間隔を小さくとり、標本データ数を大きくとる必要がある。なお、連続畳み込みと対応をとる場合には、 $p(n_x^{(f)} + n_x^{(g)} - 1, k_y, k_z) = 0$ ,  $p(k_x, n_y^{(f)} + n_y^{(g)} - 1, k_z) = 0$ ,  $p(k_x, k_y, n_z^{(f)} + n_z^{(g)} - 1) = 0$ , とし

て  $p(k_x, k_y, k_z)$  ( $k_x = 0, 1, \dots, n_x^{(f)} + n_x^{(g)} - 1$ ;  $k_y = 0, 1, \dots, n_y^{(f)} + n_y^{(g)} - 1$ ;  $k_z = 0, 1, \dots, n_z^{(f)} + n_z^{(g)} - 1$ ) の  $(n_x^{(f)} + n_x^{(g)})(n_y^{(f)} + n_y^{(g)})(n_z^{(f)} + n_z^{(g)})$  個のデータを考えた方が、対応をとりやすい。このとき通常は座標 (0, 0, 0) の要素は  $p(0, 0, 0)$  に対応させる。ただし、

ISW=0 の場合、

$$LX1 = NX1, LY1 = NY1, LZ1 = NZ1, LX2 = MX, LY2 = MY, LZ2 = MZ,$$

$$NWK = (NX2 + 1) \times (NY2 + 1) \times NZ2 \text{ (NX2:偶数, NY2:偶数のとき) または}$$

$$NWK = NX2 \times (NY2 + 1) \times NZ2 \text{ (NX2:奇数, NY2:偶数のとき) または}$$

$$NWK = (NX2 + 1) \times NY2 \times NZ2 \text{ (NX2:偶数, NY2:奇数のとき) または}$$

$$NWK = NX2 \times NY2 \times NZ2 \text{ (NX2:奇数, NY2:奇数のとき)}$$

ISW  $\geq 1$  の場合、

$$LX1=LX2=MX+1 \text{ (MX が奇数のとき) または}$$

$$LX1=LX2=MX+2 \text{ (MX が偶数のとき),}$$

$$LY1=LY2=MY, LZ1=LZ2=MZ, NWK = MX + 2 \times (MY + MZ) + LX1 \times MY \times MZ \text{ である。}$$

(g) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。

## (7) 使用例

### (a) 問題

次式で定義される 2 つの有限波形を標準化間隔  $\Delta$  で離散化し、離散畳み込みを計算する。

$$f(x, y, z) = \begin{cases} x & ((x, y, z) \in [0, x_f] \times [0, y_f] \times [0, z_f]) \\ 0 & (\text{それ以外}) \end{cases}$$

$$g(x, y, z) = \begin{cases} x_g - x & ((x, y, z) \in [0, x_g] \times [0, y_g] \times [0, z_g]) \\ 0 & (\text{それ以外}) \end{cases}$$

### (b) 入力データ

標準化データ

$$R1(i_x + 1, i_y + 1, i_z + 1) = f(i_x \Delta, i_y \Delta, i_z \Delta)$$

$$(i_x = 0, 1, \dots, NX1 - 1; i_y = 0, 1, \dots, NY1 - 1; i_z = 0, 1, \dots, NZ1 - 1)$$

$$R2(j_x + 1, j_y + 1, j_z + 1) = g(j_x \Delta, j_y \Delta, j_z \Delta)$$

$$(j_x = 0, 1, \dots, NX2 - 1; j_y = 0, 1, \dots, NY2 - 1; j_z = 0, 1, \dots, NZ2 - 1)$$

ただし、 $\Delta = 0.5$

$NX1, NY1, NZ1, NX2, NY2, NZ2, MX, MY, MZ, ISW$

### (c) 主プログラム

```

PROGRAM BFCN3D
! *** EXAMPLE OF DFCN3D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER I, J, K
INTEGER ISW, IERR, IWK(60)
INTEGER NX1, NX2, LX1, LX2, MX
INTEGER NY1, NY2, LY1, LY2, MY
INTEGER NZ1, NZ2, LZ1, LZ2, MZ
INTEGER MO
PARAMETER (MO = 8)
PARAMETER (LX1 = (MO+2)/2*2)
PARAMETER (LY1 = MO)
PARAMETER (LZ1 = MO)
PARAMETER (LX2 = LX1)
PARAMETER (LY2 = LY1)
PARAMETER (LZ2 = LZ1)
REAL(8) R1(LX1, LY1, LZ1), R2(LX2, LY2, LZ2)
REAL(8) WK(5*MO+LX1*MO*MO)
REAL(8) T
REAL(8) XF, YF, ZF, XG, YG, ZG, DT
PARAMETER (DT = 0.5D0)
PARAMETER (XF = 2.0D0, YF=2.0D0, ZF=2.0D0)
PARAMETER (XG = 2.0D0, YG=2.0D0, ZG=2.0D0)
!
ISW=1
NX1=XF/DT
NY1=YF/DT
NZ1=ZF/DT
NX2=XG/DT
NY2=YG/DT
NZ2=ZG/DT
MX=MO
MY=MO
MZ=MO
WRITE (6,1000) ISW, NX1, NY1, NZ1, NX2, NY2, NZ2, MX, MY, MZ
DO 100 K=1, NZ1
DO 101 J=1, NY1
DO 102 I=1, NX1
T=DBLE(I-1)*DT
R1(I, J, K)=T
102 CONTINUE
101 CONTINUE
100 CONTINUE
DO 200 K=1, NZ2
DO 201 J=1, NY2
DO 202 I=1, NX2
T=DBLE(I-1)*DT
R2(I, J, K)=XG-T
202 CONTINUE
201 CONTINUE
200 CONTINUE
DO 300 K=1, NZ1
WRITE (6,1100) K, (I, (R1(I, J, K), J=1, NY1), I=1, NX1)
300 CONTINUE
DO 400 K=1, NZ2
WRITE (6,1150) K, (I, (R2(I, J, K), J=1, NY2), I=1, NX2)
400 CONTINUE

```

```

CALL DFCN3D(NX1,NY1,NZ1,NX2,NY2,NZ2,R1,LX1,LY1,LZ1,&
R2,LX2,LY2,LZ2,MX,MY,MZ,ISW,IWK,WK,IERR)
WRITE (6,1300)
WRITE (6,1400) IERR
DO 500 K=1,MZ
WRITE (6,1200) K,(I,(R2(I,J,K),J=1,MY),I=1,MX)
500 CONTINUE
1000 FORMAT(' ',/,/,&
' *** DFCN3D ***',/,&
2X,'** INPUT **',/,&
6X,'ISW =',I3,/,&
6X,'(NX1,NY1,NZ1) =(',I3,',',I3,',',I3,')',/,&
6X,'(NX2,NY2,NZ2) =(',I3,',',I3,',',I3,')',/,&
6X,'(MX,MY,MZ) =(',I3,',',I3,',',I3,')')
1100 FORMAT(12X,'DATA R1(I,J,',I3,')',/,&
10X,'I/J 1 2 3 4',/,&
10X,'-----',/,&
6(8X,I3,4F9.4,/) )
1150 FORMAT(12X,'DATA R2(I,J,',I3,')',/,&
10X,'I/J 1 2 3 4',/,&
10X,'-----',/,&
6(8X,I3,4F9.4,/) )
1300 FORMAT(2X,'** OUTPUT **')
1400 FORMAT(6X,'IERR =',I5)
1200 FORMAT(17X,'CONVOLUTION R2(I,J,',I3,')',/,&
10X,'I/J 1 2 3 4 5',&
, 6 7 8',/,&
10X,'-----',/,&
,-----',/,&
8(8X,I3,8F7.2,/) )
END

```

## (d) 出力結果

```

*** DFCN3D ***
** INPUT **
ISW = 1
(NX1,NY1,NZ1) = ( 4, 4, 4)
(NX2,NY2,NZ2) = ( 4, 4, 4)
(MX,MY,MZ) = ( 8, 8, 8)
DATA R1(I,J, 1)
I/J 1 2 3 4
-----
1 0.0000 0.0000 0.0000 0.0000
2 0.5000 0.5000 0.5000 0.5000
3 1.0000 1.0000 1.0000 1.0000
4 1.5000 1.5000 1.5000 1.5000

DATA R1(I,J, 2)
I/J 1 2 3 4
-----
1 0.0000 0.0000 0.0000 0.0000
2 0.5000 0.5000 0.5000 0.5000
3 1.0000 1.0000 1.0000 1.0000
4 1.5000 1.5000 1.5000 1.5000

DATA R1(I,J, 3)
I/J 1 2 3 4
-----
1 0.0000 0.0000 0.0000 0.0000
2 0.5000 0.5000 0.5000 0.5000
3 1.0000 1.0000 1.0000 1.0000
4 1.5000 1.5000 1.5000 1.5000

DATA R1(I,J, 4)
I/J 1 2 3 4
-----
1 0.0000 0.0000 0.0000 0.0000
2 0.5000 0.5000 0.5000 0.5000
3 1.0000 1.0000 1.0000 1.0000
4 1.5000 1.5000 1.5000 1.5000

DATA R2(I,J, 1)
I/J 1 2 3 4
-----
1 2.0000 2.0000 2.0000 2.0000
2 1.5000 1.5000 1.5000 1.5000
3 1.0000 1.0000 1.0000 1.0000
4 0.5000 0.5000 0.5000 0.5000

DATA R2(I,J, 2)
I/J 1 2 3 4
-----
1 2.0000 2.0000 2.0000 2.0000
2 1.5000 1.5000 1.5000 1.5000
3 1.0000 1.0000 1.0000 1.0000
4 0.5000 0.5000 0.5000 0.5000

DATA R2(I,J, 3)
I/J 1 2 3 4
-----
1 2.0000 2.0000 2.0000 2.0000
2 1.5000 1.5000 1.5000 1.5000
3 1.0000 1.0000 1.0000 1.0000
4 0.5000 0.5000 0.5000 0.5000

DATA R2(I,J, 4)
I/J 1 2 3 4
-----
1 2.0000 2.0000 2.0000 2.0000

```

```

2  1.5000  1.5000  1.5000  1.5000
3  1.0000  1.0000  1.0000  1.0000
4  0.5000  0.5000  0.5000  0.5000

```

```

** OUTPUT **
IERR = 0

```

```

CONVOLUTION R2(I,J, 1)
I/J  1  2  3  4  5  6  7  8
-----
1  0.00  0.00  0.00  0.00  0.00 -0.00 -0.00  0.00
2  1.00  2.00  3.00  4.00  3.00  2.00  1.00  0.00
3  2.75  5.50  8.25  11.00  8.25  5.50  2.75  0.00
4  5.00  10.00  15.00  20.00  15.00  10.00  5.00  0.00
5  3.50  7.00  10.50  14.00  10.50  7.00  3.50  0.00
6  2.00  4.00  6.00  8.00  6.00  4.00  2.00  0.00
7  0.75  1.50  2.25  3.00  2.25  1.50  0.75  -0.00
8  0.00  0.00  0.00  0.00  -0.00 -0.00 -0.00  0.00

```

```

CONVOLUTION R2(I,J, 2)
I/J  1  2  3  4  5  6  7  8
-----
1  0.00  0.00  0.00  0.00  0.00  0.00 -0.00 -0.00
2  2.00  4.00  6.00  8.00  6.00  4.00  2.00 -0.00
3  5.50  11.00  16.50  22.00  16.50  11.00  5.50 -0.00
4  10.00  20.00  30.00  40.00  30.00  20.00  10.00 -0.00
5  7.00  14.00  21.00  28.00  21.00  14.00  7.00 -0.00
6  4.00  8.00  12.00  16.00  12.00  8.00  4.00 -0.00
7  1.50  3.00  4.50  6.00  4.50  3.00  1.50 -0.00
8  0.00 -0.00 -0.00  0.00 -0.00 -0.00 -0.00 -0.00

```

```

CONVOLUTION R2(I,J, 3)
I/J  1  2  3  4  5  6  7  8
-----
1  0.00  0.00  0.00  0.00  0.00  0.00  0.00 -0.00
2  3.00  6.00  9.00  12.00  9.00  6.00  3.00 -0.00
3  8.25  16.50  24.75  33.00  24.75  16.50  8.25 -0.00
4  15.00  30.00  45.00  60.00  45.00  30.00  15.00 -0.00
5  10.50  21.00  31.50  42.00  31.50  21.00  10.50 -0.00
6  6.00  12.00  18.00  24.00  18.00  12.00  6.00 -0.00
7  2.25  4.50  6.75  9.00  6.75  4.50  2.25 -0.00
8  0.00 -0.00  0.00 -0.00  0.00  0.00 -0.00 -0.00

```

```

CONVOLUTION R2(I,J, 4)
I/J  1  2  3  4  5  6  7  8
-----
1  0.00  0.00  0.00  0.00  0.00  0.00 -0.00  0.00
2  4.00  8.00  12.00  16.00  12.00  8.00  4.00  0.00
3  11.00  22.00  33.00  44.00  33.00  22.00  11.00 -0.00
4  20.00  40.00  60.00  80.00  60.00  40.00  20.00 -0.00
5  14.00  28.00  42.00  56.00  42.00  28.00  14.00 -0.00
6  8.00  16.00  24.00  32.00  24.00  16.00  8.00 -0.00
7  3.00  6.00  9.00  12.00  9.00  6.00  3.00 -0.00
8  -0.00 -0.00  0.00 -0.00 -0.00  0.00 -0.00  0.00

```

```

CONVOLUTION R2(I,J, 5)
I/J  1  2  3  4  5  6  7  8
-----
1  0.00 -0.00  0.00  0.00  0.00  0.00 -0.00 -0.00
2  3.00  6.00  9.00  12.00  9.00  6.00  3.00 -0.00
3  8.25  16.50  24.75  33.00  24.75  16.50  8.25 -0.00
4  15.00  30.00  45.00  60.00  45.00  30.00  15.00 -0.00
5  10.50  21.00  31.50  42.00  31.50  21.00  10.50  0.00
6  6.00  12.00  18.00  24.00  18.00  12.00  6.00  0.00
7  2.25  4.50  6.75  9.00  6.75  4.50  2.25 -0.00
8  -0.00 -0.00  0.00  0.00  0.00 -0.00 -0.00 -0.00

```

```

CONVOLUTION R2(I,J, 6)
I/J  1  2  3  4  5  6  7  8
-----
1  0.00  0.00  0.00  0.00  0.00  0.00 -0.00  0.00
2  2.00  4.00  6.00  8.00  6.00  4.00  2.00  0.00
3  5.50  11.00  16.50  22.00  16.50  11.00  5.50 -0.00
4  10.00  20.00  30.00  40.00  30.00  20.00  10.00 -0.00
5  7.00  14.00  21.00  28.00  21.00  14.00  7.00 -0.00
6  4.00  8.00  12.00  16.00  12.00  8.00  4.00 -0.00
7  1.50  3.00  4.50  6.00  4.50  3.00  1.50 -0.00
8  0.00 -0.00  0.00 -0.00 -0.00 -0.00 -0.00  0.00

```

```

CONVOLUTION R2(I,J, 7)
I/J  1  2  3  4  5  6  7  8
-----
1  -0.00  0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
2  1.00  2.00  3.00  4.00  3.00  2.00  1.00 -0.00
3  2.75  5.50  8.25  11.00  8.25  5.50  2.75 -0.00
4  5.00  10.00  15.00  20.00  15.00  10.00  5.00 -0.00
5  3.50  7.00  10.50  14.00  10.50  7.00  3.50 -0.00
6  2.00  4.00  6.00  8.00  6.00  4.00  2.00 -0.00
7  0.75  1.50  2.25  3.00  2.25  1.50  0.75 -0.00
8  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00

```

```

CONVOLUTION R2(I,J, 8)
I/J  1  2  3  4  5  6  7  8
-----
1  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
2  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
3  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
4  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
5  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
6  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
7  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00
8  -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00 -0.00

```





---

## 2.15 相関

### 2.15.1 DFCR1D, RFCR1D

#### 1次元相関

(1) 機能

任意の整数  $k$  に対して

$$f(i) = f(i + km), g(i) = g(i + km) \quad (i = 0, \dots, m - 1)$$

を満たす 2 組の周期  $m$  の離散関数  $f(i), g(j)$  ただし,

$$f(i) = 0 \quad (i = n_1, \dots, m - 1); \quad g(j) = 0 \quad (j = n_2, \dots, m - 1)$$

が与えられているとき次式で定義される離散相関  $q(k)$  ( $k = 0, \dots, m - 1$ ):

$$q(k) = \sum_{i=0}^{m-1} f(i)g(k+i) \quad (k = 0, \dots, m-1)$$

を正の方向に  $n_1 - 1$  だけシフトした量  $\tilde{q}(k)$  ( $k = 0, \dots, m - 1$ )

$$\tilde{q}(k) = q(k - (n_1 - 1)) \quad (k = 0, \dots, m - 1)$$

を計算する。ただし,  $m = \min(n_1 + n_2 - 1, M)$  であり,  $M$  は  $M \geq \max(n_1, n_2)$  を満たす任意の整数である。なお, 離散相関  $q(k)$  の実フーリエ変換を求めることもできる。

(2) 使用法

倍精度サブルーチン:

CALL DFCR1D (N1, N2, R1, LD1, R2, LD2, M, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFCR1D (N1, N2, R1, LD1, R2, LD2, M, ISW, IWK, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N1	I	1	入 力	離散関数 $f(i)$ の有効データ数 $n_1$
2	N2	I	1	入 力	離散関数 $g(j)$ の有効データ数 $n_2$
3	R1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD1	入 力	離散関数 $f(i)$ の値 (注意事項 (a), (b) 参照)
				出 力	ISW $\geq 1$ のとき離散関数 $f(i)$ の実フーリエ変換結果 (周期 $M$ )
4	LD1	I	1	入 力	配列 R1 の大きさ
5	R2	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LD2	入 力	離散関数 $g(j)$ の値 (注意事項 (a), (b) 参照)
				出 力	離散関数 $\tilde{q}(k)$ の値またはその実フーリエ変換 (注意事項 (a), (c) 参照)
6	LD2	I	1	入 力	配列 R2 の大きさ
7	M	I	1	入 力	離散関数 $f(i), g(j), \tilde{q}(k)$ の周期 $m$ に対応するパラメータ $M$ (注意事項 (d) 参照)
8	ISW	I	1	入 力	処理スイッチ (注意事項 (a), (e) 参照) ISW= 0:定義により相関を計算する ISW= 1:FFT 法により相関を計算する ISW= 2:相関の実フーリエ変換を計算する ISW= 3:区分化 FFT 法により相関を計算する
9	IWK	I	内容参照	ワーク	作業領域 大きさ: 0 (ISW= 0 のとき) 20 (ISW $\geq 1$ のとき)
10	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 大きさ: N2 (ISW= 0 のとき) $2 \times M + 1$ (ISW= 1 または 2, M が奇数のとき) $2 \times M + 2$ (ISW= 1 または 2, M が偶数のとき) $2 \times M + N1$ (ISW= 3, M が奇数のとき) $2 \times M + N1 + 1$ (ISW= 3, M が偶数のとき)
11	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $ISW \in \{0, 1, 2, 3\}$
- (b)  $N1 > 1$
- (c)  $N2 > 1$
- (d)  $M \geq \max(N1, N2)$
- (e)  $ISW = 0$  の時 :
- $LD1 \geq N1$
- $ISW > 0$  で  $M$  が奇数の時 :
- $LD1 \geq M + 1$
- $ISW > 0$  で  $M$  が偶数の時 :
- $LD1 \geq M + 2$
- (f)  $ISW = 0$  の時 :
- $LD2 \geq M$
- $ISW > 0$  で  $M$  が奇数の時 :
- $LD2 \geq M + 1$
- $ISW > 0$  で  $M$  が偶数の時 :
- $LD2 \geq M + 2$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$M < N1 + N2 - 1$ であった.	相関の計算で重なりが発生する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3050	制限条件 (f) を満足しなかった.	

## (6) 注意事項

- (a) 相関を計算する関数値の一方の有効データ数が他方に比較して非常に大きい場合、区分化によりデータ数の多い方を (必要であれば最後に 0 を付加して) 等分し、本サブルーチンを繰り返し適用して離散相関を計算した方が効率が高く、必要な記憶容量も小さくて済む。
- たとえば、2 つの系列  $\{u_1, u_2, \dots, u_k\}$  (有効データ数  $k$ )、 $\{v_1, \dots, v_{pq}\}$  (有効データ数  $pq$  ( $pq \gg k$ )) の離散相関を計算したい場合、まず  $ISW=1$ ,  $N1=k$ ,  $N2=q$ ,  $M \geq N1 + N2 - 1$ ,  $R1=\{u_1, u_2, \dots, u_k\}$   $R2=\{v_1, v_2, \dots, v_q\}$  として本サブルーチンを適用する。この結果、計算したい相関の最初の  $q$  個の値が配列  $R2$  の先頭から  $q$  個の要素として得られる。
- 次に、 $ISW=3$ ,  $R2=\{v_{q+1}, \dots, v_{2q}\}$  と変更し、ほかの引数の内容はそのままとして本サブルーチンを適用する。この結果、計算したい相関の次の  $q$  個の値が配列  $R2$  の先頭から  $q$  個の要素として得られる。以下同様にして  $R2$  に設定する値を順次ずらして計算する。なお、最後の繰り返しすなわち  $R2=\{v_{(p-1)q+1}, \dots, v_{pq}\}$  と設定して計算した相関は計算したい相関の最後の  $2q - 1$  個の要素を与える (ただし、系列  $\{v_j\}$  が有限波形でない場合は最後から  $q - 1$  個は不定)。

- (b) 配列 R1, R2 にはそれぞれ離散関数  $f(i)$  と離散関数  $g(j)$  の値を次のように格納する。ただし, ISW = 3 とする場合には, R2 にのみ値を格納し, R1 の内容はそのまま利用する (注意事項 (a) 参照)。

$$\begin{aligned} f(0) &\rightarrow R1(1) \\ f(1) &\rightarrow R1(2) \\ \dots &\dots \dots \\ f(n_1 - 1) &\rightarrow R1(N1) \\ \\ g(0) &\rightarrow R2(1) \\ g(1) &\rightarrow R2(2) \\ \dots &\dots \dots \\ g(n_2 - 1) &\rightarrow R2(N2) \end{aligned}$$

なお, 配列 R1, R2 の R1 (N1+1) 並びに R2 (N2+1) 以降の要素には値を入力する必要が無い。また, 特に, ISW = 3 とする場合には, R2 (N2+1) 以降の要素は計算に利用するので変更してはならない。

- (c) 離散相関  $\tilde{q}(k)$  の値は配列 R2 に以下のように得られる。

$$\begin{aligned} \tilde{q}(0) &\rightarrow R2(1) \\ \tilde{q}(1) &\rightarrow R2(2) \\ \dots &\dots \dots \\ \tilde{q}(M - 1) &\rightarrow R2(M) \end{aligned}$$

なお, M が奇数のとき R2 (M+1) が M が偶数のとき R2 (M+1) と R2 (M+2) がそれぞれ 0.0 となる。また, 区分化を行う場合には, 通常, 最初の N2 個のデータが畳み込みとして意味を持つ (注意事項 (a) 参照)。ISW=2 として, 離散相関  $q(k)$  の実フーリエ変換  $Q(j)$ :

$$Q(j) = \frac{1}{M} \sum_{k=0}^{M-1} q(k) e^{-2\pi\sqrt{-1}\frac{jk}{M}} \quad (j = 0, \dots, \lfloor \frac{M}{2} \rfloor)$$

( $\lfloor x \rfloor$  は  $x$  を超えない最大の整数) を求める場合には,

$$\begin{aligned} \Re\{Q(0)\} &\leftrightarrow R2(1) \\ \Im\{Q(0)\} &\leftrightarrow R2(2) \\ \Re\{Q(1)\} &\leftrightarrow R2(3) \\ \Im\{Q(1)\} &\leftrightarrow R2(4) \\ \dots &\dots \dots \\ \Re\{Q(\lfloor \frac{M}{2} \rfloor)\} &\leftrightarrow R2(l - 1) \\ \Im\{Q(\lfloor \frac{M}{2} \rfloor)\} &\leftrightarrow R2(l) \quad (l = M+1[M:\text{奇数}] \text{ または } M+2[M:\text{偶数}]) \end{aligned}$$

と対応する。なお, この場合, 得られるフーリエ変換は正規化されていることに注意する必要がある。フーリエ変換の残りの半周期分は実フーリエ変換の対称性

$$Q(M - j) = Q(j)^*$$

(ただし,  $z^*$  は複素数  $z$  の共役複素数) から得られる。なお,  $Q(j)$  は相関を計算するもとの 2 つの関数のクロス・スペクトルの近似量と考えることができる。この場合,  $M = n_1 + n_2$  として考えた方がよい。特に, 相関を計算するもとの 2 つの関数が同じ関数であれば,  $Q(j)$  は生のフーリエ・ピリオドグラム (パワー・スペクトルの近似量) に対応し,  $Q(j)$  は実数となる。

- (d)  $M \geq N1+N2-1$  とすれば, 次の周期の相関との重なりを起こさずに相関を計算できる。  $M > N1+N2-1$  の場合  $N1+N2$  以降の要素には誤差の範囲で 0.0 と一致する値が格納される。ISW=0 のときは,  $M = N1+N2-1$  とするのがよい。ISW  $\geq 1$  とする場合, M の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば,  $N1=N2=145$  の

場合, ISW=0 のときは,  $M = 289 (=17^2)$  とした方が良いが,  $ISW \geq 1$  の場合には  $M = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い。

- (e) 通常は ISW=1 と設定して FFT 相関を計算した方が効率良く計算を行える。ただし, 作業領域を節約したい場合やパラメータ M の選び方に制限がある場合などは ISW=0 として計算する。
- (f) 非ゼロ部分の開始位置が原点から離れている離散関数の相関を計算したい場合には, まず開始位置が原点に来るようにシフトして計算した後, 計算結果を再度シフトして最終結果を得た方が効率が良い。例えば, 離散関数  $f(i)$ ,  $g(j)$  の非ゼロ部分がそれぞれ区間  $[i_0, i_0 + n_1 - 1]$ ,  $[j_0, j_0 + n_2 - 1]$  のとき

$$\hat{f}(i) = f(i - i_0), \quad \hat{g}(j) = g(j - j_0)$$

として  $\hat{f}(i)$ ,  $\hat{g}(j)$  についてこのサブルーチンを適用し, 得られた結果を  $\tilde{q}(k)$  とすれば, もとの  $f(i)$ ,  $g(j)$  の相関  $q(k)$  は

$$q(k) = \tilde{q}(k - (j_0 - i_0) + (n_1 - 1))$$

となる。したがって,  $i_0 = j_0 = 0$  の場合でも通常定義と整合する相関  $q(k)$  を考える場合にはこのサブルーチンの適用後, 負の方向に  $n_1 - 1$  だけシフトして考える必要があり, また, 離散相関を計算する前に  $f(i)$ ,  $g(j)$  をそれぞれ負の方向に  $i_0$ ,  $j_0$  だけシフトしたとすれば, 計算結果をさらに  $j_0 - i_0$  だけ正の方向にシフトする必要がある。

- (g) このサブルーチンで計算する離散相関に標準化間隔を乗じたものは帯域制限された関数の連続相関積分を方形近似 (台形公式による近似でもある) したものになる。したがって, 近似精度を上げるためには, 標準化間隔を小さくとり, 標準データ数を大きくとる必要がある。なお, 連続相関と対応をとる場合には,  $q(-n_1) = \tilde{q}(-1) = 0$  として  $q(k)$  ( $k = -n_1, \dots, -1, 0, 1, \dots, n_2 - 1$ ) の  $n_1 + n_2$  個のデータを考えた方が, 対応をとりやすい。もちろん,  $q(n_1 + n_2) = \tilde{q}(n_2) = 0$  として  $q(k)$  ( $k = -(n_1 - 1), \dots, -1, 0, 1, \dots, n_2$ ) を考えても同じである。このとき通常は座標 0 の要素は  $q(0)$  に対応させる。ただし,

ISW=0 の場合,

$$LD1=N1, LD2=M, NWK=N2$$

ISW=1 または 2 の場合,

$$M \text{ が奇数ならば } LD1=LD2=M+1, NWK = 2 \times M + 1$$

$$M \text{ が偶数ならば } LD1=LD2=M+2, NWK = 2 \times M + 2$$

ISW=3 の場合,

$$M \text{ が奇数ならば } LD1=LD2=M+1, NWK = 2 \times M + N1$$

$$M \text{ が偶数ならば } LD1=LD2=M+2, NWK = 2 \times M + N1 + 1$$

である。

- (h) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。

## (7) 使用例

### (a) 問題

次式で定義される 2 つの有限波形を標準化間隔  $\Delta x$  で離散化し, 離散相関を計算する。

$$f(x) = \begin{cases} x & 0 \leq x \leq a \\ 0 & \text{それ以外} \end{cases}$$

$$g(x) = \begin{cases} b - x & 0 \leq x \leq b \\ 0 & \text{それ以外} \end{cases}$$

## 備考

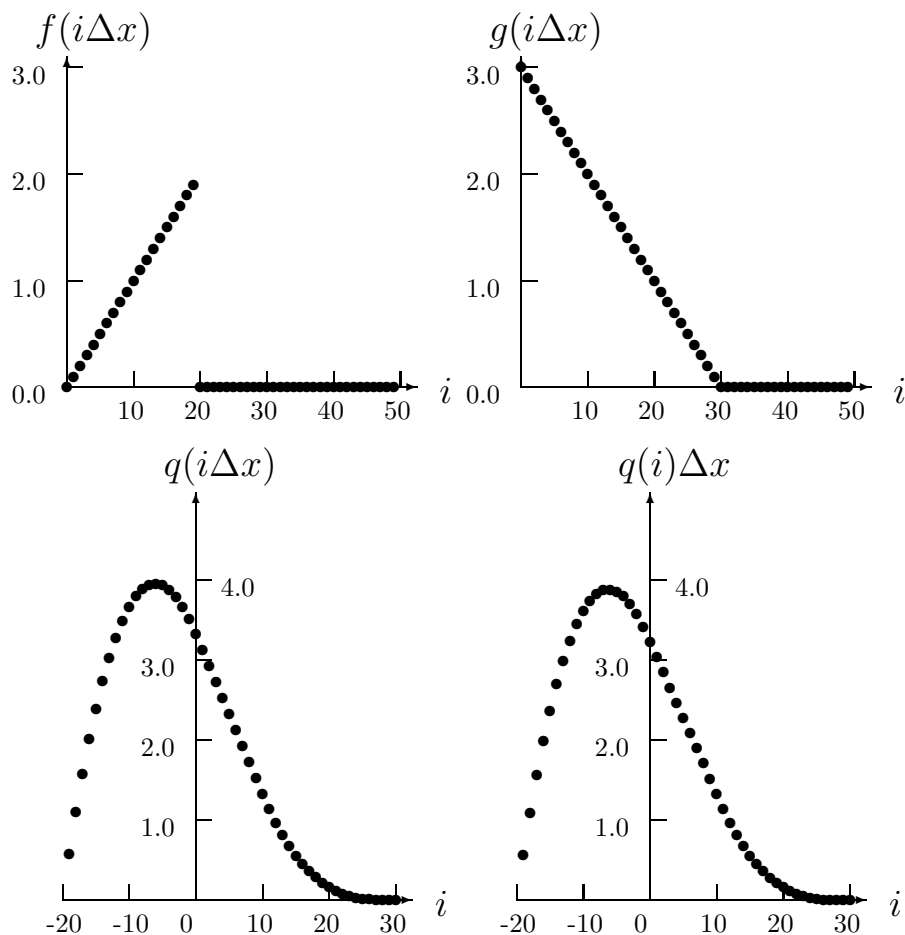
$f(x)$  と  $g(x)$  の連続相関  $q(x)$  は

$$q(x) = \int_{-\infty}^{\infty} f(\xi)g(x+\xi)d\xi = \begin{cases} G(-x, a, x) & -a \leq x \leq 0 \\ G(0, a, x) & 0 \leq x \leq b-a \\ G(0, b-x, x) & b-a \leq x \leq b \\ 0 & \text{それ以外} \end{cases}$$

である。ここで、 $G(\alpha, \beta, x)$  は

$$\begin{aligned} G(\alpha, \beta, x) &= \left[ \frac{\xi^2}{6}(3(b-x) - 2\xi) \right]_{\alpha}^{\beta} \\ &= \frac{\xi^2}{6}(3(b-x) - 2\xi) \Big|_{\xi=\beta} - \frac{\xi^2}{6}(3(b-x) - 2\xi) \Big|_{\xi=\alpha} \end{aligned}$$

$a = 2, b = 3$  とした場合、 $f(x), g(x), q(x)$  を  $\Delta x = 0.1$  として標本化した値  $f(i\Delta x), g(i\Delta x), q(i\Delta x)$  はそれぞれ以下のようなグラフとなる。参考のため本サブルーチンで計算した離散相関に  $\Delta x$  を乗じた値  $q(i)\Delta x$  も示してある。標本数が少ないわりには連続相関の値と良く一致する。



なお、以下の使用例では参考までに連続相関の値も計算している。

## (b) 入力データ

標本化データ

$$R1(i) = f((i-1)\Delta x) \quad (i = 1, 2, \dots, N1)$$

$$R2(j) = g((j-1)\Delta x) \quad (j = 1, 2, \dots, N2)$$

ただし、 $\Delta x = 0.1$

$$N1 = \frac{a}{\Delta x}, N2 = \frac{b}{\Delta x}, M, ISW$$

## (c) 主プログラム

```

PROGRAM BFCR1D
! *** EXAMPLE OF DFCR1D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER I
INTEGER N1,N2,LD1,LD2,M,ISW,IERR,IWK(20)
INTEGER MO
PARAMETER (MO = 100)
PARAMETER (LD1 = MO+2)
PARAMETER (LD2 = MO+2)
REAL(8) R1(LD1),R2(LD2),WK(2*M0+2)
REAL(8) CR(LD2),T,DT
REAL(8) A,B,F

!
ISW=1
DT=0.1D0
A=2.0D0
B=3.0D0
N1=(A+0.5D0*DT)/DT
N2=(B+0.5D0*DT)/DT
M=50
WRITE (6,1000) ISW,N1,N2,M
DO 100 I=1,N1
  T=DBLE(I-1)*DT
  R1(I)=T
100 CONTINUE
DO 200 I=1,N2
  T=DBLE(I-1)*DT
  R2(I)=B-T
200 CONTINUE
!***** ASSUME N2.GT.N1
WRITE (6,1100) (I-1,R1(I),R2(I),I=1,N1),&
  (I-1,R2(I),I=N1+1,N2)
CALL DFCR1D(N1,N2,R1,LD1,R2,LD2,M,ISW,IWK,WK,IERR)
WRITE (6,1300)
WRITE (6,1400) IERR
DO 500 I=1,N1
  T=DBLE(I-N1)*DT
  CR(I)=F(A,T,B)-F(-T,T,B)
500 CONTINUE
DO 300 I=N1+1,N2
  T=DBLE(I-N1)*DT
  CR(I)=F(A,T,B)
300 CONTINUE
DO 400 I=N2+1,N2+N1
  T=DBLE(I-N1)*DT
  CR(I)=F(B-T,T,B)
400 CONTINUE
WRITE (6,1200) (I-N1,R2(I),R2(I)*DT,CR(I),I=1,N1+N2)
1000 FORMAT(' ',/,/,&
  ' *** DFCR1D ***',/,&
  2X,'** INPUT **',/,&
  6X,'ISW =',I3,/,&
  6X,'N1 =',I3,/,&
  6X,'N2 =',I3,/,&
  6X,'M =',I3)
1100 FORMAT(12X,'DATA(R1,R2)',/,&
  7X,'I-1',4X,'R1(I)',4X,'R2(I)',/,&
  20(7X,I3,2F9.4,/,),10(7X,I3,9X,F9.4,/)
1300 FORMAT(2X,'** OUTPUT **')
1400 FORMAT(6X,'IERR =',I5)
1200 FORMAT(17X,'CORRELATION',/,&
  6X,'I-N1',4X,'R2(I)',3X,'R2(I)*DT',2X,'CR(I)',/,&
  50(7X,I3,1X,3F9.4,/)

END
REAL(8) FUNCTION F(TAU,T,B)
REAL(8) TAU,T,B
F=TAU*TAU*(0.5D0*(B-T)-TAU/3.0D0)
RETURN
END

```

## (d) 出力結果

```

*** DFCR1D ***
** INPUT **
ISW = 1
N1 = 20
N2 = 30
M = 50
DATA(R1,R2)
I-1 R1(I) R2(I)
0 0.0000 3.0000
1 0.1000 2.9000
2 0.2000 2.8000
3 0.3000 2.7000
4 0.4000 2.6000
5 0.5000 2.5000
6 0.6000 2.4000
7 0.7000 2.3000
8 0.8000 2.2000
9 0.9000 2.1000
10 1.0000 2.0000
11 1.1000 1.9000
12 1.2000 1.8000
13 1.3000 1.7000
14 1.4000 1.6000
15 1.5000 1.5000
16 1.6000 1.4000

```



```

17 1.7000 1.3000
18 1.8000 1.2000
19 1.9000 1.1000
20      1.0000
21      0.9000
22      0.8000
23      0.7000
24      0.6000
25      0.5000
26      0.4000
27      0.3000
28      0.2000
29      0.1000

** OUTPUT **
IERR = 0
CORRELATION
I-N1 R2(I) R2(I)*DT CR(I)
-19 5.7000 0.5700 0.5752
-18 10.9100 1.0910 1.1013
-17 15.6400 1.5640 1.5795
-16 19.9000 1.9900 2.0107
-15 23.7000 2.3700 2.3958
-14 27.0500 2.7050 2.7360
-13 29.9600 2.9960 3.0322
-12 32.4400 3.2440 3.2853
-11 34.5000 3.4500 3.4965
-10 36.1500 3.6150 3.6667
-9 37.4000 3.7400 3.7968
-8 38.2600 3.8260 3.8880
-7 38.7400 3.8740 3.9412
-6 38.8500 3.8850 3.9573
-5 38.6000 3.8600 3.9375
-4 38.0000 3.8000 3.8827
-3 37.0600 3.7060 3.7938
-2 35.7900 3.5790 3.6720
-1 34.2000 3.4200 3.5182
0 32.3000 3.2300 3.3333
1 30.4000 3.0400 3.1333
2 28.5000 2.8500 2.9333
3 26.6000 2.6600 2.7333
4 24.7000 2.4700 2.5333
5 22.8000 2.2800 2.3333
6 20.9000 2.0900 2.1333
7 19.0000 1.9000 1.9333
8 17.1000 1.7100 1.7333
9 15.2000 1.5200 1.5333
10 13.3000 1.3300 1.3333
11 11.4000 1.1400 1.1432
12 9.6900 0.9690 0.9720
13 8.1600 0.8160 0.8188
14 6.8000 0.6800 0.6827
15 5.6000 0.5600 0.5625
16 4.5500 0.4550 0.4573
17 3.6400 0.3640 0.3662
18 2.8600 0.2860 0.2880
19 2.2000 0.2200 0.2218
20 1.6500 0.1650 0.1667
21 1.2000 0.1200 0.1215
22 0.8400 0.0840 0.0853
23 0.5600 0.0560 0.0572
24 0.3500 0.0350 0.0360
25 0.2000 0.0200 0.0208
26 0.1000 0.0100 0.0107
27 0.0400 0.0040 0.0045
28 0.0100 0.0010 0.0013
29 0.0000 0.0000 0.0002
30 0.0000 0.0000 0.0000

```

## 2.15.2 DFCR2D, RFCR2D

## 2次元相関

## (1) 機能

任意の整数  $L_x, L_y$  に対して

$$\begin{aligned} f(i_x, i_y) &= f(i_x + L_x m_x, i_y + L_y m_y), \\ g(j_x, j_y) &= g(j_x + L_x m_x, j_y + L_y m_y), \\ &(i_x, j_x = 0, \dots, m_x - 1; i_y, j_y = 0, \dots, m_y - 1) \end{aligned}$$

を満たす2組の周期  $(m_x, m_y)$  の多重周期離散関数  $f(i_x, i_y), g(j_x, j_y)$  についてこれらがそれぞれ基本周期内では  $(i_x, i_y) \in [0, n_x^{(f)} - 1] \times [0, n_y^{(f)} - 1], (j_x, j_y) \in [0, n_x^{(g)} - 1] \times [0, n_y^{(g)} - 1]$  でのみ非ゼロ値をとるとする。ここで、 $[0, a] \times [0, b]$  は平面座標  $(i, j)$  がはる平面の直積領域 (点  $(0, 0)$  と点  $(a, b)$  を対角頂点とする長方形で囲まれる領域) とする。このとき、次式で定義される離散相関  $q(k_x, k_y)$ :

$$\begin{aligned} q(k_x, k_y) &= \sum_{i_x=0}^{m_x-1} \sum_{i_y=0}^{m_y-1} f(i_x, i_y) g(k_x + i_x, k_y + i_y) \\ &(k_x = 0, \dots, m_x - 1; k_y = 0, \dots, m_y - 1) \end{aligned}$$

を  $(k_x, k_y)$  についてそれぞれ正の方向に  $(n_x^{(f)} - 1, n_y^{(f)} - 1)$  だけシフトした量  $\tilde{q}(k_x, k_y)$ :

$$\begin{aligned} \tilde{q}(k_x, k_y) &= q(k_x - (n_x^{(f)} - 1), k_y - (n_y^{(f)} - 1)) \\ &(k_x = 0, \dots, m_x - 1; k_y = 0, \dots, m_y - 1) \end{aligned}$$

を計算する。ただし、 $m_x = \min(n_x^{(f)} + n_x^{(g)} - 1, M_x), m_y = \min(n_y^{(f)} + n_y^{(g)} - 1, M_y)$  であり、 $M_x, M_y$  はそれぞれ  $M_x \geq \max(n_x^{(f)}, n_x^{(g)}), M_y \geq \max(n_y^{(f)}, n_y^{(g)})$  を満たす任意の整数である。なお、離散相関  $q(k_x, k_y)$  の2次元実フーリエ変換を求めることもできる。

## (2) 使用法

倍精度サブルーチン:

CALL DFCR2D (NX1, NY1, NX2, NY2, R1, LX1, LY1, R2, LX2, LY2, MX, MY, ISW, IWK,  
WK, IERR)

単精度サブルーチン:

CALL RFCR2D (NX1, NY1, NX2, NY2, R1, LX1, LY1, R2, LX2, LY2, MX, MY, ISW, IWK,  
WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{array} \right\}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX1	I	1	入力	離散関数 $f(i_x, i_y)$ の $i_x$ 方向の有効データ数 $n_x^{(f)}$
2	NY1	I	1	入力	離散関数 $f(i_x, i_y)$ の $i_y$ 方向の有効データ数 $n_y^{(f)}$
3	NX2	I	1	入力	離散関数 $g(j_x, j_y)$ の $j_x$ 方向の有効データ数 $n_x^{(g)}$
4	NY2	I	1	入力	離散関数 $g(j_x, j_y)$ の $j_y$ 方向の有効データ数 $n_y^{(g)}$
5	R1	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX1, LY1	入力	離散関数 $f(i_x, i_y)$ の値 (注意事項 (a) 参照)
				出力	ISW $\geq 1$ のとき離散関数 $f(i_x, i_y)$ の 2 次元実フーリエ変換結果 (周期 ( $M_x, M_y$ ))
6	LX1	I	1	入力	配列 R1 の整合寸法
7	LY1	I	1	入力	配列 R1 の第 2 寸法
8	R2	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX2, LY2	入力	離散関数 $g(j_x, j_y)$ の値 (注意事項 (a) 参照)
				出力	離散関数 $\tilde{q}(k_x, k_y)$ の値または $q(k_x, k_y)$ の 2 次元実フーリエ変換 (注意事項 (b) 参照)
9	LX2	I	1	入力	配列 R2 の整合寸法
10	LY2	I	1	入力	配列 R2 の第 2 寸法
11	MX	I	1	入力	離散関数 $f(i_x, i_y), g(j_x, j_y), \tilde{q}(k_x, k_y)$ の周期 ( $m_x, m_y$ ) に対応するパラメータ $M_x$ (注意事項 (c) 参照)
12	MY	I	1	入力	離散関数 $f(i_x, i_y), g(j_x, j_y), \tilde{q}(k_x, k_y)$ の周期 ( $m_x, m_y$ ) に対応するパラメータ $M_y$ (注意事項 (c) 参照)
13	ISW	I	1	入力	処理スイッチ (注意事項 (d) 参照) ISW= 0:定義により相関を計算する ISW= 1:FFT 法により相関を計算する ISW= 2:相関の実フーリエ変換を計算する
14	IWK	I	内容参照	ワーク	作業領域 大きさ: 0 (ISW= 0 のとき) 40 (ISW $\geq 1$ のとき)
15	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	内容参照	ワーク	作業領域 大きさ: NX2 $\times$ NY2 (ISW= 0, NX2:奇数のとき) (NX2 + 1) $\times$ NY2 (ISW= 0, NX2:偶数のとき) MX + 2 $\times$ MY + MAX(LX1 $\times$ LY1, LX2 $\times$ LY2) (ISW $\geq 1$ のとき)
16	IERR	I	1	出力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, 1, 2\}$

(b)  $NX1 > 1$   
 $NY1 > 1$

(c)  $NX2 > 1$   
 $NY2 > 1$

(d)  $MX \geq \max(NX1, NX2)$   
 $MY \geq \max(NY1, NY2)$

(e)  $ISW = 0$  の時 :

$LX1 \geq NX1$

$LY1 \geq NY1$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX1 \geq MX + 1$

$LY1 \geq MY$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX1 \geq MX + 2$

$LY1 \geq MY$

(f)  $ISW = 0$  の時 :

$LX2 \geq MX$

$LY2 \geq MY$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX2 \geq MX + 1$

$LY2 \geq MY$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX2 \geq MX + 2$

$LY2 \geq MY$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$MX < NX1 + NX2 - 1$ または $MY < NY1 + NY2 - 1$ であった.	相関の計算で重なりが発生する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3050	制限条件 (f) を満足しなかった.	

## (6) 注意事項

- (a) 配列 R1, R2 の各要素と離散関数
- $f(i_x, i_y)$
- と離散関数
- $g(j_x, j_y)$
- の値は以下の様に対応する。

$$\begin{aligned} f(i_x, i_y) &\leftrightarrow R1(i_x + 1, i_y + 1) \\ g(j_x, j_y) &\leftrightarrow R2(j_x + 1, j_y + 1) \end{aligned}$$

ただし,  $i_x = 0, \dots, n_x^{(f)} - 1$ ;  $i_y = 0, \dots, n_y^{(f)} - 1$ ,  $j_x = 0, \dots, n_x^{(g)} - 1$ ;  $j_y = 0, \dots, n_y^{(g)} - 1$  であり, それ以外の要素には値を入力する必要が無い. なお, 主記憶のバンク競合を避けるために配列 R1, R2 の整合寸法について LX1/2, LY1, LX2/2, LY2 が奇数になるように設定するのが望ましい. 通常, たとえば MX が 4 の倍数のときは LX1=MX+3 とする.

- (b) 離散相関
- $\tilde{q}(k_x, k_y)$
- の値は配列 R2 の各要素と以下の様に対応する。

$$\tilde{q}(k_x, k_y) \leftrightarrow R2(k_x + 1, k_y + 1)$$

ただし,  $k_x = 0, \dots, M_x - 1$ ;  $k_y = 0, \dots, M_y - 1$  である. ISW=2 として, 離散相関  $q(k_x, k_y)$  の 2 次元実フーリエ変換  $Q(j_x, j_y)$ :

$$\begin{aligned} Q(j_x, j_y) &= \frac{1}{M_x M_y} \sum_{k_x=0}^{M_x-1} \sum_{k_y=0}^{M_y-1} q(k_x, k_y) e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{M_x} + \frac{j_y k_y}{M_y}\right)} \\ &\quad (j_x = 0, \dots, \lfloor \frac{M_x}{2} \rfloor; j_y = 0, \dots, \lfloor \frac{M_y}{2} \rfloor) \end{aligned}$$

( $\lfloor x \rfloor$  は  $x$  を超えない最大の整数) を求める場合には,

$$\begin{aligned} \Re\{Q(j_x, j_y)\} &\leftrightarrow R2(2 * j_x + 1, j_y + 1) \\ \Im\{Q(j_x, j_y)\} &\leftrightarrow R2(2 * j_x + 2, j_y + 1) \end{aligned}$$

と対応する. なお, この場合, 得られるフーリエ変換は正規化されていることに注意する必要がある. フーリエ変換の残りの半周期分は実フーリエ変換の対称性

$$\begin{aligned} Q(M_x - j_x, M_y - j_y)^* &= Q(j_x, j_y) \\ Q(M_x - j_x, j_y)^* &= Q(j_x, M_y - j_y) \end{aligned}$$

(ただし,  $z^*$  は複素数  $z$  の共役複素数) から得られる. なお,  $Q(j_x, j_y)$  は相関を計算するもとの 2 つの関数のクロス・スペクトルの近似量と考えることができる. この場合,  $M_x = n_x^{(f)} + n_x^{(g)}$ ,  $M_y = n_y^{(f)} + n_y^{(g)}$  として考えた方がよい. 特に, 相関を計算するもとの 2 つの関数が同じ関数であれば,  $Q(j_x, j_y)$  は生のフーリエ・ピリオドグラム (パワー・スペクトルの近似量) に対応し,  $Q(j_x, j_y)$  は実数となる.

- (c)
- $MX \geq NX1 + NX2 - 1$
- かつ
- $MY \geq NY1 + NY2 - 1$
- とすれば, 次の周期の相関との重なりを起こさずに相関を計算できる.
- $MX > NX1 + NX2 - 1$
- または
- $MY > NY1 + NY2 - 1$
- の場合

$$\tilde{q}(k_x, k_y) \leftrightarrow R2(k_x + 1, k_y + 1)$$

$k_x = NX1 + NX2 - 1, \dots, MX - 1$ ;  $k_y = 0, \dots, MY - 1$  または  $k_x = 0, \dots, MX - 1$ ;  $k_y = NY1 + NY2 - 1, \dots, MY - 1$  に対応する要素には誤差の範囲で 0.0 と一致する値が格納される. ISW=0 のときは,  $MX = NX1 + NX2 - 1$ ,  $MY = NY1 + NY2 - 1$  とするのがよい. ISW  $\geq 1$  とする場合, MX, MY の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば,  $NX1=NX2=145$  の場合, ISW=0 のときは,  $MX = 289 (=17^2)$  とした方がよいが, ISW  $\geq 1$  の場合には  $MX = 300 (=2^2 \times 3 \times 5^2)$  や  $320 (=2^6 \times 5)$ ,  $384 (=2^7 \times 3)$  などとした方が通常は効率が良い.

- (d) 通常は ISW=1 と設定して FFT 相関を計算した方が効率良く計算を行える. ただし, 作業領域を節約したい場合やパラメータ MX や MY の選び方に制限がある場合などは ISW=0 として計算する.

- (e) 非ゼロ部分の開始位置が原点から離れている離散関数の相関を計算したい場合には、まず開始位置が原点に来るようにシフトして計算した後、計算結果を再度シフトして最終結果を得た方が効率が良い。例えば、離散関数  $f(i_x, i_y), g(j_x, j_y)$  の非ゼロ部分が  $i_x, j_x$  についてそれぞれ区間  $[i_0, i_0 + n_x^{(f)} - 1], [j_0, j_0 + n_x^{(g)} - 1]$  のとき

$$\hat{f}(i_x, i_y) = f(i_x - i_0, i_y), \quad \hat{g}(j_x, j_y) = g(j_x - j_0, j_y)$$

として  $\hat{f}(i_x, i_y), \hat{g}(j_x, j_y)$  についてこのサブルーチンを適用し、得られた結果を  $\tilde{q}(k_x, k_y)$  とすれば、もとの  $f(i_x, i_y), g(j_x, j_y)$  の相関  $q(k_x, k_y)$  は

$$q(k_x, k_y) = \tilde{q}(k_x - (j_0 - i_0) + (n_x^{(f)} - 1), k_y)$$

となる。したがって、 $i_0 = j_0 = 0$  の場合でも通常定義と整合する相関  $q(k_x, k_y)$  を考える場合にはこのサブルーチンの適用後、 $k_x$  の負の方向に  $n_x^{(f)} - 1$  だけシフトして考える必要があり、また、離散相関を計算する前に  $f(i_x, i_y), g(j_x, j_y)$  をそれぞれ  $i_x, j_x$  の負の方向に  $i_0, j_0$  だけシフトしたとすれば、計算結果をさらに  $j_0 - i_0$  だけ  $k_x$  の正の方向にシフトする必要がある。 $i_y, j_y, k_y$  についても同様である。

- (f) このサブルーチンで計算する離散相関に標準化間隔の2乗を乗じたものは帯域制限された関数の連続相関積分を方形近似(台形公式による近似でもある)したものになる。したがって、近似精度を上げるためには、標準化間隔を小さくとり、標準データ数を大きくとる必要がある。なお、連続相関と対応をとる場合には、 $q(-n_x^{(f)}, k_y) = \tilde{q}(-1, k_y) = 0, q(k_x, -n_y^{(f)}) = \tilde{q}(k_x, -1) = 0$  として  $q(k_x, k_y)$  ( $k_x = -n_x^{(f)}, \dots, -1, 0, 1, \dots, n_x^{(g)} - 1; k_y = -n_y^{(f)}, \dots, -1, 0, 1, \dots, n_y^{(g)} - 1$ ) の  $(n_x^{(f)} + n_x^{(g)})(n_y^{(f)} + n_y^{(g)})$  個のデータを考えた方が、対応をとりやすい。もちろん、 $q(n_x^{(f)} + n_x^{(g)}, k_y) = \tilde{q}(n_x^{(g)}, k_y) = 0, q(k_x, n_y^{(f)} + n_y^{(g)}) = \tilde{q}(k_x, n_y^{(g)}) = 0$  として  $q(k_x, k_y)$  ( $k_x = -(n_x^{(f)} - 1), \dots, -1, 0, 1, \dots, n_x^{(g)}; k_y = -(n_y^{(f)} - 1), \dots, -1, 0, 1, \dots, n_y^{(g)}$ ) を考えても同じである。このとき通常は座標  $(0, 0)$  の要素は  $q(0, 0)$  に対応させる。ただし、

ISW=0 の場合、

$$LX1 = NX1, LY1 = NY1, LX2 = MX, LY2 = MY,$$

$$NWK = NX2 \times NY2 \text{ (NX2:奇数のとき) または}$$

$$NWK = (NX2 + 1) \times NY2 \text{ (NX2:偶数のとき)}$$

ISW ≥ 1 の場合、

$$LX1=LX2=MX+1 \text{ (MX が奇数のとき) または}$$

$$LX1=LX2=MX+2 \text{ (MX が偶数のとき),}$$

$$LY1=LY2=MY, NWK = MX + (LX1 + 2) \times MY$$

である。

- (g) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。

## (7) 使用例

### (a) 問題

次式で定義される2つの有限波形を標準化間隔  $\Delta$  で離散化し、離散相関を計算する。

$$f(x, y) = \begin{cases} x & ((x, y) \in [0, x_f] \times [0, y_f]) \\ 0 & \text{(それ以外)} \end{cases}$$

$$g(x, y) = \begin{cases} x_g - x & ((x, y) \in [0, x_g] \times [0, y_g]) \\ 0 & \text{(それ以外)} \end{cases}$$

## (b) 入力データ

## 標本化データ

$$R1(i_x + 1, i_y + 1) = f(i_x \Delta, i_y \Delta) \quad (i_x = 0, 1, \dots, NX1 - 1; i_y = 0, 1, \dots, NY1 - 1)$$

$$R2(j_x + 1, j_y + 1) = g(j_x \Delta, j_y \Delta) \quad (j_x = 0, 1, \dots, NX2 - 1; j_y = 0, 1, \dots, NY2 - 1)$$

ただし,  $\Delta = 0.5$

$NX1, NY1, NX2, NY2, MX, MY, ISW$

## (c) 主プログラム

```

PROGRAM BFCR2D
! *** EXAMPLE OF DFCR2D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER I,J
INTEGER ISW,IERR,IWK(40)
INTEGER NX1,NX2,LX1,LX2,MX
INTEGER NY1,NY2,LY1,LY2,MY
INTEGER MO
PARAMETER (MO = 8)
PARAMETER (LX1 = MO+2)
PARAMETER (LY1 = MO)
PARAMETER (LX2 = MO+2)
PARAMETER (LY2 = MO)
REAL(8) R1(LX1,LY1),R2(LX2,LY2),WK(3*MO+LX2*MO)
REAL(8) T
REAL(8) XF,YF,XG,YG,DT
PARAMETER (DT = 0.5D0)
PARAMETER (XF = 2.0D0,YF=2.0D0)
PARAMETER (XG = 2.0D0,YG=2.0D0)
!
ISW=1
NX1=XF/DT
NY1=YF/DT
NX2=XG/DT
NY2=YG/DT
MX=MO
MY=MO
WRITE (6,1000) ISW,NX1,NY1,NX2,NY2,MX,MY
DO 100 J=1,NY1
DO 101 I=1,NX1
T=DBLE(I-1)*DT
R1(I,J)=T
101 CONTINUE
100 CONTINUE
DO 200 J=1,NY2
DO 201 I=1,NX2
T=DBLE(I-1)*DT
R2(I,J)=XG-T
201 CONTINUE
200 CONTINUE
WRITE (6,1100) (I,(R1(I,J),J=1,NY1),I=1,NX1)
WRITE (6,1150) (I,(R2(I,J),J=1,NY2),I=1,NX2)
CALL DFCR2D(NX1,NY1,NX2,NY2,R1,LX1,LY1,&
R2,LX2,LY2,MX,MY,ISW,IWK,WK,IERR)
WRITE (6,1300)
WRITE (6,1400) IERR
WRITE (6,1200)&
(I,(R2(I,J),J=1,MY),I=1,MX)
1000 FORMAT(' ',/,/,&
' *** DFCR2D ***',/,&
2X,'** INPUT **',/,&
6X,' ISW =',I3,/,&
6X,' (NX1,NY1) =(',I3,',',I3,')',/,&
6X,' (NX2,NY2) =(',I3,',',I3,')',/,&
6X,' (MX, MY) =(',I3,',',I3,')')
1100 FORMAT(12X,'DATA R1(I,J)',/,&
10X,'I/J 1 2 3 4',/,&
10X,'-----',/,&
6(8X,I3,4F9.4,/) )
1150 FORMAT(12X,'DATA R2(I,J)',/,&
10X,'I/J 1 2 3 4',/,&
10X,'-----',/,&
6(8X,I3,4F9.4,/) )
1300 FORMAT(2X,'** OUTPUT **')
1400 FORMAT(6X,' IERR =',I5)
1200 FORMAT(17X,'CORRELATION R2(I,J)',/,&
10X,'I/J 1 2 3 4 5',&
', 6 7 8',/,&
10X,'-----',/,&
8(8X,I3,8F7.2,/) )
END

```

## (d) 出力結果

```

*** DFCR2D ***
** INPUT **
ISW = 1
(NX1,NY1) = ( 4, 4)
(NX2,NY2) = ( 4, 4)
(MX, MY) = ( 8, 8)
DATA R1(I,J)
I/J 1 2 3 4
-----

```

```

1  0.0000  0.0000  0.0000  0.0000
2  0.5000  0.5000  0.5000  0.5000
3  1.0000  1.0000  1.0000  1.0000
4  1.5000  1.5000  1.5000  1.5000

```

```

      DATA R2(I,J)
      I/J   1       2       3       4
-----
1  2.0000  2.0000  2.0000  2.0000
2  1.5000  1.5000  1.5000  1.5000
3  1.0000  1.0000  1.0000  1.0000
4  0.5000  0.5000  0.5000  0.5000

```

```

** OUTPUT **
IERR =      0

```

```

      CORRELATION R2(I,J)
      I/J   1       2       3       4       5       6       7       8
-----
1  3.00  6.00  9.00 12.00  9.00  6.00  3.00 -0.00
2  4.25  8.50 12.75 17.00 12.75  8.50  4.25 -0.00
3  4.00  8.00 12.00 16.00 12.00  8.00  4.00 -0.00
4  2.50  5.00  7.50 10.00  7.50  5.00  2.50 -0.00
5  1.00  2.00  3.00  4.00  3.00  2.00  1.00 -0.00
6  0.25  0.50  0.75  1.00  0.75  0.50  0.25  0.00
7  0.00  0.00  0.00 -0.00  0.00  0.00  0.00  0.00
8  0.00  0.00 -0.00 -0.00  0.00  0.00  0.00  0.00

```



## 2.15.3 DFCR3D, RFCR3D

## 3次元相関

## (1) 機能

任意の整数  $L_x, L_y, L_z$  に対して

$$\begin{aligned} f(i_x, i_y, i_z) &= f(i_x + L_x m_x, i_y + L_y m_y, i_z + L_z m_z), \\ g(j_x, j_y, j_z) &= g(j_x + L_x m_x, j_y + L_y m_y, j_z + L_z m_z), \\ &(i_x, j_x = 0, \dots, m_x - 1; i_y, j_y = 0, \dots, m_y - 1; i_z, j_z = 0, \dots, m_z - 1) \end{aligned}$$

を満たす2組の周期  $(m_x, m_y, m_z)$  の多重周期離散関数  $f(i_x, i_y, i_z), g(j_x, j_y, j_z)$  についてこれらがそれぞれ基本周期内では  $(i_x, i_y, i_z) \in [0, n_x^{(f)} - 1] \times [0, n_y^{(f)} - 1] \times [0, n_z^{(f)} - 1], (j_x, j_y, j_z) \in [0, n_x^{(g)} - 1] \times [0, n_y^{(g)} - 1] \times [0, n_z^{(g)} - 1]$  でのみ非ゼロ値をとるとする。ここで、 $[0, a] \times [0, b] \times [0, c]$  は空間座標  $(i, j, k)$  がはる空間の直積領域 (点  $(0, 0, 0)$  と点  $(a, b, c)$  を対角頂点とする直方体で囲まれる領域) とする。このとき、次式で定義される離散相関  $q(k_x, k_y, k_z)$ :

$$\begin{aligned} q(k_x, k_y, k_z) &= \sum_{i_x=0}^{m_x-1} \sum_{i_y=0}^{m_y-1} \sum_{i_z=0}^{m_z-1} f(i_x, i_y, i_z) g(k_x + i_x, k_y + i_y, k_z + i_z) \\ &(k_x = 0, \dots, m_x - 1; k_y = 0, \dots, m_y - 1; k_z = 0, \dots, m_z - 1) \end{aligned}$$

を  $(k_x, k_y, k_z)$  についてそれぞれ正の方向に  $(n_x^{(f)} - 1, n_y^{(f)} - 1, n_z^{(f)} - 1)$  だけシフトした量  $\tilde{q}(k_x, k_y, k_z)$ :

$$\begin{aligned} \tilde{q}(k_x, k_y, k_z) &= q(k_x - (n_x^{(f)} - 1), k_y - (n_y^{(f)} - 1), k_z - (n_z^{(f)} - 1)) \\ &(k_x = 0, \dots, m_x - 1; k_y = 0, \dots, m_y - 1; k_z = 0, \dots, m_z - 1) \end{aligned}$$

を計算する。ただし、 $m_x = \min(n_x^{(f)} + n_x^{(g)} - 1, M_x), m_y = \min(n_y^{(f)} + n_y^{(g)} - 1, M_y), m_z = \min(n_z^{(f)} + n_z^{(g)} - 1, M_z)$  であり、 $M_x, M_y, M_z$  はそれぞれ  $M_x \geq \max(n_x^{(f)}, n_x^{(g)}), M_y \geq \max(n_y^{(f)}, n_y^{(g)}), M_z \geq \max(n_z^{(f)}, n_z^{(g)})$  を満たす任意の整数である。なお、離散相関  $q(k_x, k_y, k_z)$  の3次元実フーリエ変換を求めることもできる。

## (2) 使用法

倍精度サブルーチン:

CALL DFCR3D (NX1, NY1, NZ1, NX2, NY2, NZ2, R1, LX1, LY1, LZ1, R2, LX2, LY2, LZ2,  
MX, MY, MZ, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFCR3D (NX1, NY1, NZ1, NX2, NY2, NZ2, R1, LX1, LY1, LZ1, R2, LX2, LY2, LZ2,  
MX, MY, MZ, ISW, IWK, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32\text{ビット整数版では INTEGER}(4) \\ 64\text{ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX1	I	1	入力	離散関数 $f(i_x, i_y, i_z)$ の $i_x$ 方向の有効データ数 $n_x^{(f)}$
2	NY1	I	1	入力	離散関数 $f(i_x, i_y, i_z)$ の $i_y$ 方向の有効データ数 $n_y^{(f)}$
3	NZ1	I	1	入力	離散関数 $f(i_x, i_y, i_z)$ の $i_z$ 方向の有効データ数 $n_z^{(f)}$
4	NX2	I	1	入力	離散関数 $g(j_x, j_y, j_z)$ の $j_x$ 方向の有効データ数 $n_x^{(g)}$
5	NY2	I	1	入力	離散関数 $g(j_x, j_y, j_z)$ の $j_y$ 方向の有効データ数 $n_y^{(g)}$
6	NZ2	I	1	入力	離散関数 $g(j_x, j_y, j_z)$ の $j_z$ 方向の有効データ数 $n_z^{(g)}$
7	R1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX1, LY1, LZ1	入力	離散関数 $f(i_x, i_y, i_z)$ の値 (注意事項 (a) 参照)
				出力	ISW $\geq 1$ のとき離散関数 $f(i_x, i_y, i_z)$ の3次元実フーリエ変換結果 (周期 $(M_x, M_y, M_z)$ )
8	LX1	I	1	入力	配列 R1 の整合寸法
9	LY1	I	1	入力	配列 R1 の第2寸法
10	LZ1	I	1	入力	配列 R1 の第3寸法
11	R2	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX2, LY2, LZ2	入力	離散関数 $g(j_x, j_y, j_z)$ の値 (注意事項 (a) 参照)
				出力	離散関数 $\tilde{q}(k_x, k_y, k_z)$ の値 または $q(k_x, k_y, k_z)$ の3次元実フーリエ変換 (注意事項 (b) 参照)
12	LX2	I	1	入力	配列 R2 の整合寸法
13	LY2	I	1	入力	配列 R2 の第2寸法
14	LZ2	I	1	入力	配列 R2 の第3寸法
15	MX	I	1	入力	離散関数 $f(i_x, i_y, i_z), g(j_x, j_y, j_z), \tilde{q}(k_x, k_y, k_z)$ の周期 $(m_x, m_y, m_z)$ に対応するパラメータ $M_x$ (注意事項 (c) 参照)
16	MY	I	1	入力	離散関数 $f(i_x, i_y, i_z), g(j_x, j_y, j_z), \tilde{q}(k_x, k_y, k_z)$ の周期 $(m_x, m_y, m_z)$ に対応するパラメータ $M_y$ (注意事項 (c) 参照)

項番	引数名	型	大きさ	入出力	内 容
17	MZ	I	1	入 力	離散関数 $f(i_x, i_y, i_z), g(j_x, j_y, j_z), \tilde{q}(k_x, k_y, k_z)$ の周期 $(m_x, m_y, m_z)$ に対応するパラメータ $M_z$ (注意事項 (c) 参照)
18	ISW	I	1	入 力	処理スイッチ (注意事項 (d) 参照) ISW= 0:定義により相関を計算する ISW= 1:FFT 法により相関を計算する ISW= 2:相関の実フーリエ変換を計算する
19	IWK	I	内容参照	ワーク	作業領域 大きさ: 0 (ISW= 0 のとき) 60 (ISW $\geq$ 1 のとき)
20	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 大きさ: $(NX2 + 1) \times (NY2 + 1) \times NZ2$ (ISW= 0, NX2:偶数, NY2:偶数のとき) $NX2 \times (NY2 + 1) \times NZ2$ (ISW= 0, NX2:奇数, NY2:偶数のとき) $(NX2 + 1) \times NY2 \times NZ2$ (ISW= 0, NX2:偶数, NY2:奇数のとき) $NX2 \times NY2 \times NZ2$ (ISW= 0, NX2:奇数, NY2:奇数のとき) $MX + 2 \times (MY + MZ) + \text{MAX}(LX1 \times LY1 \times LZ1, LX2 \times LY2 \times LZ2)$ (ISW $\geq$ 1 のとき)
21	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, 1, 2\}$

(b)  $NX1 > 1$

$NY1 > 1$

$NZ1 > 1$

(c)  $NX2 > 1$

$NY2 > 1$

$NZ2 > 1$

(d)  $MX \geq \max(NX1, NX2)$

$MY \geq \max(NY1, NY2)$

$MZ \geq \max(NZ1, NZ2)$

(e)  $ISW = 0$  の時 :

$LX1 \geq NX1$

$LY1 \geq NY1$

$LZ1 \geq NZ1$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX1 \geq MX + 1$ ,  $LX1$  は偶数

$LY1 \geq MY$

$LZ1 \geq MZ$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX1 \geq MX + 2$ ,  $LX1$  は偶数

$LY1 \geq MY$

$LZ1 \geq MZ$

(f)  $ISW = 0$  の時 :

$LX2 \geq MX$

$LY2 \geq NY2$

$LZ2 \geq NZ2$

 $ISW > 0$  で  $MX$  が奇数の時 :

$LX2 \geq MX + 1$ ,  $LX2$  は偶数

$LY2 \geq MY$

$LZ2 \geq MZ$

 $ISW > 0$  で  $MX$  が偶数の時 :

$LX2 \geq MX + 2$ ,  $LX2$  は偶数

$LY2 \geq MY$

$LZ2 \geq MZ$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	MX < NX1 + NX2 - 1, MY < NY1 + NY2 - 1 または MZ < NZ1 + NZ2 - 1 であった.	相関の計算で重なりが発生する.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	
3040	制限条件 (e) を満足しなかった.	
3050	制限条件 (f) を満足しなかった.	

## (6) 注意事項

(a) 配列 R1, R2 の各要素と離散関数  $f(i_x, i_y, i_z)$  と離散関数  $g(j_x, j_y, j_z)$  の値は以下の様に対応する.

$$f(i_x, i_y, i_z) \leftrightarrow R1(i_x + 1, i_y + 1, i_z + 1)$$

$$g(j_x, j_y, j_z) \leftrightarrow R2(j_x + 1, j_y + 1, j_z + 1)$$

ただし,  $i_x = 0, \dots, n_x^{(f)} - 1$ ;  $i_y = 0, \dots, n_y^{(f)} - 1$ ;  $i_z = 0, \dots, n_z^{(f)} - 1$ ;  $j_x = 0, \dots, n_x^{(g)} - 1$ ;  $j_y = 0, \dots, n_y^{(g)} - 1$ ;  $j_z = 0, \dots, n_z^{(g)} - 1$  であり, それ以外の要素には値を入力する必要が無い. なお, 主記憶のバンク競合を避けるために配列 R1, R2 の整合寸法について LX1/2, LY1, LZ1, LX2/2, LY2, LZ2 が奇数になるように設定するのが望ましい. また, 高速化のために配列 R1, R2 内のデータ設定領域以外の要素に対しても演算を実行する. 通常, たとえば MX が (4 の倍数)+2 のときは LX1=MX+4 とする.

(b) 離散相関  $\tilde{q}(k_x, k_y, k_z)$  の値は配列 R2 の各要素と以下の様に対応する.

$$\tilde{q}(k_x, k_y, k_z) \leftrightarrow R2(k_x + 1, k_y + 1, k_z + 1)$$

ただし,  $k_x = 0, \dots, M_x - 1$ ;  $k_y = 0, \dots, M_y - 1$ ;  $k_z = 0, \dots, M_z - 1$  である. ISW=2 として, 離散相関  $q(k_x, k_y, k_z)$  の 3次元実フーリエ変換  $Q(j_x, j_y, j_z)$ :

$$Q(j_x, j_y, j_z) = \frac{1}{M_x M_y M_z} \sum_{k_x=0}^{M_x-1} \sum_{k_y=0}^{M_y-1} \sum_{k_z=0}^{M_z-1} q(k_x, k_y, k_z) e^{-2\pi\sqrt{-1}(\frac{j_x k_x}{M_x} + \frac{j_y k_y}{M_y} + \frac{j_z k_z}{M_z})}$$

$$(j_x = 0, \dots, \lfloor \frac{M_x}{2} \rfloor; j_y = 0, \dots, \lfloor \frac{M_y}{2} \rfloor; j_z = 0, \dots, \lfloor \frac{M_z}{2} \rfloor)$$

( $\lfloor x \rfloor$  は  $x$  を超えない最大の整数) を求める場合には,

$$\Re\{Q(j_x, j_y, j_z)\} \leftrightarrow R2(2 * j_x + 1, j_y + 1, j_z + 1)$$

$$\Im\{Q(j_x, j_y, j_z)\} \leftrightarrow R2(2 * j_x + 2, j_y + 1, j_z + 1)$$

と対応する. なお, この場合, 得られるフーリエ変換は正規化されていることに注意する必要がある. フーリエ変換の残りの半周期分は実フーリエ変換の対称性

$$Q(M_x - j_x, M_y - j_y, M_z - j_z)^* = Q(j_x, j_y, j_z)$$

$$Q(M_x - j_x, j_y, j_z)^* = Q(j_x, M_y - j_y, M_z - j_z)$$

$$Q(M_x - j_x, M_y - j_y, j_z)^* = Q(j_x, j_y, M_z - j_z)$$

(ただし,  $z^*$  は複素数  $z$  の共役複素数) から得られる. なお,  $Q(j_x, j_y, j_z)$  は相関を計算するもとの 2 つの関数のクロス・スペクトルの近似量と考えることができる. この場合,  $M_x = n_x^{(f)} + n_x^{(g)}$ ,  $M_y = n_y^{(f)} + n_y^{(g)}$ ,

$M_z = n_z^{(f)} + n_z^{(g)}$  として考えた方がよい。特に、相関を計算するもとの2つの関数が同じ関数であれば、 $Q(j_x, j_y, j_z)$  は生のフーリエ・ピリオドグラム (パワー・スペクトルの近似量) に対応し、 $Q(j_x, j_y, j_z)$  は実数となる。

- (c)  $MX \geq NX1 + NX2 - 1$  かつ  $MY \geq NY1 + NY2 - 1$  かつ  $MZ \geq NZ1 + NZ2 - 1$  とすれば、次の周期の相関との重なりを起こさずに相関を計算できる。  $MX > NX1 + NX2 - 1$  または  $MY > NY1 + NY2 - 1$  または  $MZ > NZ1 + NZ2 - 1$  の場合

$$\tilde{q}(k_x, k_y) \leftrightarrow R2(k_x + 1, k_y + 1, k_z + 1)$$

$k_x = NX1 + NX2 - 1, \dots, MX - 1$ ;  $k_y = 0, \dots, MY - 1$ ;  $k_z = 0, \dots, MZ - 1$  または  $k_x = 0, \dots, MX - 1$ ;  $k_y = NY1 + NY2 - 1, \dots, MY - 1$ ;  $k_z = 0, \dots, MZ - 1$  または  $k_x = 0, \dots, MX - 1$ ;  $k_y = 0, \dots, MY - 1$ ;  $k_z = NZ1 + NZ2 - 1, \dots, MZ - 1$  に対応する要素には誤差の範囲で 0.0 と一致する値が格納される。ISW=0 のときは、 $MX = NX1 + NX2 - 1$ ,  $MY = NY1 + NY2 - 1$ ,  $MZ = NZ1 + NZ2 - 1$  とするのがよい。ISW  $\geq 1$  とする場合、MX, MY, MZ の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば、 $NX1=NX2=145$  の場合、ISW=0 のときは、 $MX = 289(=17^2)$  とした方がよいが、ISW  $\geq 1$  の場合には  $MX = 300(=2^2 \times 3 \times 5^2)$  や  $320(=2^6 \times 5)$ ,  $384(=2^7 \times 3)$  などとした方が通常は効率が良い。

- (d) 通常は ISW=1 と設定して FFT 相関を計算した方が効率良く計算を行える。ただし、作業領域を節約したい場合やパラメータ MX や MY, MZ の選び方に制限がある場合などは ISW=0 として計算する。

- (e) 非ゼロ部分の開始位置が原点から離れている離散関数の相関を計算したい場合には、まず開始位置が原点に来るようにシフトして計算した後、計算結果を再度シフトして最終結果を得た方が効率が良い。例えば、離散関数  $f(i_x, i_y, i_z)$ ,  $g(j_x, j_y, j_z)$  の非ゼロ部分が  $i_x, j_x$  についてそれぞれ区間  $[i_0, i_0 + n_x^{(f)} - 1]$ ,  $[j_0, j_0 + n_x^{(g)} - 1]$  のとき

$$\hat{f}(i_x, i_y, i_z) = f(i_x - i_0, i_y, i_z), \quad \hat{g}(j_x, j_y, j_z) = g(j_x - j_0, j_y, j_z)$$

として  $\hat{f}(i_x, i_y, i_z)$ ,  $\hat{g}(j_x, j_y, j_z)$  についてこのサブルーチンを適用し、得られた結果を  $\tilde{q}(k_x, k_y, k_z)$  とすれば、もとの  $f(i_x, i_y, i_z)$ ,  $g(j_x, j_y, j_z)$  の相関  $q(k_x, k_y, k_z)$  は

$$q(k_x, k_y, k_z) = \tilde{q}(k_x - (j_0 - i_0) + (n_x^{(f)} - 1), k_y, k_z)$$

となる。したがって、 $i_0 = j_0 = 0$  の場合でも通常定義と整合する相関  $q(k_x, k_y, k_z)$  を考える場合にはこのサブルーチンの適用後、 $k_x$  の負の方向に  $n_x^{(f)} - 1$  だけシフトして考える必要があり、また、離散相関を計算する前に  $f(i_x, i_y, i_z)$ ,  $g(j_x, j_y, j_z)$  をそれぞれ  $i_x, j_x$  の負の方向に  $i_0, j_0$  だけシフトしたとすれば、計算結果をさらに  $j_0 - i_0$  だけ  $k_x$  の正の方向にシフトする必要がある。 $i_y, j_y, k_y$ ;  $i_z, j_z, k_z$  についても同様である。

- (f) このサブルーチンで計算する離散相関に標本化間隔の3乗を乗じたものは帯域制限された関数の連続相関積分を方形近似 (台形公式による近似でもある) したことになる。したがって、近似精度を上げるためには、標本化間隔を小さくとり、標本データ数を大きくとる必要がある。なお、連続相関と対応をとる場合には、

$$q(-n_x^{(f)}, k_y, k_z) = \tilde{q}(-1, k_y, k_z) = 0,$$

$$q(k_x, -n_y^{(f)}, k_z) = \tilde{q}(k_x, -1, k_z) = 0,$$

$$q(k_x, k_y, -n_z^{(f)}) = \tilde{q}(k_x, k_y, -1) = 0$$

として  $q(k_x, k_y, k_z)$  ( $k_x = -n_x^{(f)}, \dots, -1, 0, 1, \dots, n_x^{(g)} - 1$ ;  $k_y = -n_y^{(f)}, \dots, -1, 0, 1, \dots, n_y^{(g)} - 1$ ;  $k_z = -n_z^{(f)}, \dots, -1, 0, 1, \dots, n_z^{(g)} - 1$ ) の  $(n_x^{(f)} + n_x^{(g)})(n_y^{(f)} + n_y^{(g)})(n_z^{(f)} + n_z^{(g)})$  個のデータを考えた方が、対応をとりやすい。もちろん、

$$q(n_x^{(f)} + n_x^{(g)}, k_y, k_z) = \tilde{q}(n_x^{(g)}, k_y, k_z) = 0,$$

$$q(k_x, n_y^{(f)} + n_y^{(g)}, k_z) = \tilde{q}(k_x, n_y^{(g)}, k_z) = 0,$$

$$q(k_x, k_y, n_z^{(f)} + n_z^{(g)}) = \tilde{q}(k_x, k_y, n_z^{(g)}) = 0,$$

として  $q(k_x, k_y, k_z)$  ( $k_x = -(n_x^{(f)} - 1), \dots, -1, 0, 1, \dots, n_x^{(g)}$ ;  $k_y = -(n_y^{(f)} - 1), \dots, -1, 0, 1, \dots, n_y^{(g)}$ ;  $k_z = -(n_z^{(f)} - 1), \dots, -1, 0, 1, \dots, n_z^{(g)}$ ) を考えても同じである. このとき通常は座標  $(0, 0, 0)$  の要素は  $q(0, 0, 0)$  に対応させる. ただし,

ISW=0 の場合,

$$LX1 = NX1, LY1 = NY1, LZ1 = NZ1, LX2 = MX, LY2 = MY, LZ2 = MZ,$$

$$NWK = (NX2 + 1) \times (NY2 + 1) \times NZ2 \text{ (NX2:偶数, NY2:偶数のとき) または}$$

$$NWK = NX2 \times (NY2 + 1) \times NZ2 \text{ (NX2:奇数, NY2:偶数のとき) または}$$

$$NWK = (NX2 + 1) \times NY2 \times NZ2 \text{ (NX2:偶数, NY2:奇数のとき) または}$$

$$NWK = NX2 \times NY2 \times NZ2 \text{ (NX2:奇数, NY2:奇数のとき)}$$

ISW ≥ 1 の場合,

$$LX1=LX2=MX+1 \text{ (MX が奇数のとき) または}$$

$$LX1=LX2=MX+2 \text{ (MX が偶数のとき),}$$

$$LY1=LY2=MY, LZ1=LZ2=MZ, NWK = MX + 2 \times (MY + MZ) + LX1 \times MY \times MZ$$

である.

(g) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない.

## (7) 使用例

### (a) 問題

次式で定義される 2 つの有限波形を標本化間隔  $\Delta$  で離散化し, 離散相関を計算する.

$$f(x, y, z) = \begin{cases} x & ((x, y, z) \in [0, x_f] \times [0, y_f] \times [0, z_f]) \\ 0 & \text{(それ以外)} \end{cases}$$

$$g(x, y, z) = \begin{cases} x_g - x & ((x, y, z) \in [0, x_g] \times [0, y_g] \times [0, z_g]) \\ 0 & \text{(それ以外)} \end{cases}$$

### (b) 入力データ

標本化データ

$$R1(i_x + 1, i_y + 1, i_z + 1) = f(i_x \Delta, i_y \Delta, i_z \Delta) \quad (i_x = 0, 1, \dots, NX1 - 1; i_y = 0, 1, \dots, NY1 - 1; i_z = 0, 1, \dots, NZ1 - 1)$$

$$R2(j_x + 1, j_y + 1, j_z + 1) = g(j_x \Delta, j_y \Delta, j_z \Delta) \quad (j_x = 0, 1, \dots, NX2 - 1; j_y = 0, 1, \dots, NY2 - 1; j_z = 0, 1, \dots, NZ2 - 1)$$

ただし,  $\Delta = 0.5$

$NX1, NY1, NZ1, NX2, NY2, NZ2, MX, MY, MZ, ISW$

### (c) 主プログラム

```
PROGRAM BFCR3D
! *** EXAMPLE OF DFCR3D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER I, J, K
INTEGER ISW, IERR, IWK(60)
INTEGER NX1, NX2, LX1, LX2, MX
INTEGER NY1, NY2, LY1, LY2, MY
INTEGER NZ1, NZ2, LZ1, LZ2, MZ
INTEGER MO
PARAMETER (MO = 8)
PARAMETER (LX1 = (MO+2)/2*2)
PARAMETER (LY1 = MO)
PARAMETER (LZ1 = MO)
PARAMETER (LX2 = LX1)
PARAMETER (LY2 = LY1)
```

```

PARAMETER (LZ2 = LZ1)
REAL(8) R1(LX1,LY1,LZ1),R2(LX2,LY2,LZ2)
REAL(8) WK(5*M0+LX1*M0*M0)
REAL(8) T
REAL(8) XF,YF,ZF,XG,YG,ZG,DT
PARAMETER (DT = 0.5D0)
PARAMETER (XF = 2.0D0,YF=2.0D0,ZF=2.0D0)
PARAMETER (XG = 2.0D0,YG=2.0D0,ZG=2.0D0)

ISW=1
NX1=XF/DT
NY1=YF/DT
NZ1=ZF/DT
NX2=XG/DT
NY2=YG/DT
NZ2=ZG/DT
MX=M0
MY=M0
MZ=M0
WRITE (6,1000) ISW,NX1,NY1,NZ1,NX2,NY2,NZ2,MX,MY,MZ
DO 100 K=1,NZ1
DO 101 J=1,NY1
DO 102 I=1,NX1
  T=DBLE(I-1)*DT
  R1(I,J,K)=T
102 CONTINUE
101 CONTINUE
100 CONTINUE
DO 200 K=1,NZ2
DO 201 J=1,NY2
DO 202 I=1,NX2
  T=DBLE(I-1)*DT
  R2(I,J,K)=XG-T
202 CONTINUE
201 CONTINUE
200 CONTINUE
DO 300 K=1,NZ1
WRITE (6,1100) K,(I,(R1(I,J,K),J=1,NY1),I=1,NX1)
300 CONTINUE
DO 400 K=1,NZ2
WRITE (6,1150) K,(I,(R2(I,J,K),J=1,NY2),I=1,NX2)
400 CONTINUE
CALL DFCR3D(NX1,NY1,NZ1,NX2,NY2,NZ2,R1,LX1,LY1,LZ1,&
  R2,LX2,LY2,LZ2,MX,MY,MZ,ISW,IWK,WK,IERR)
WRITE (6,1300)
WRITE (6,1400) IERR
DO 500 K=1,MZ
WRITE (6,1200) K,(I,(R2(I,J,K),J=1,NY2),I=1,NX2)
500 CONTINUE
1000 FORMAT(' ',/,/,&
  ' *** DFCR3D ***',/,&
  2X,'** INPUT **',/,&
  6X,'ISW =',I3,/,&
  6X,'(NX1,NY1,NZ1) =(',I3,',',I3,',',I3,')',/,&
  6X,'(NX2,NY2,NZ2) =(',I3,',',I3,',',I3,')',/,&
  6X,'(MX,MY,MZ) =(',I3,',',I3,',',I3,')')
1100 FORMAT(12X,'DATA R1(I,J,',I3,')',/,&
  10X,'I/J 1 2 3 4',/,&
  10X,'-----',/,&
  6(8X,I3,4F9.4,/) )
1150 FORMAT(12X,'DATA R2(I,J,',I3,')',/,&
  10X,'I/J 1 2 3 4',/,&
  10X,'-----',/,&
  6(8X,I3,4F9.4,/) )
1300 FORMAT(2X,'** OUTPUT **')
1400 FORMAT(6X,'IERR =',I5)
1200 FORMAT(17X,'CORRELATION R2(I,J,',I3,')',/,&
  10X,'I/J 1 2 3 4 5',&
  10X,' 6 7 8',/,&
  10X,'-----',/,&
  8(8X,I3,8F7.2,/) )
END

```

## (d) 出力結果

```

*** DFCR3D ***
** INPUT **
ISW = 1
(NX1,NY1,NZ1) =( 4, 4, 4)
(NX2,NY2,NZ2) =( 4, 4, 4)
(MX,MY,MZ) =( 8, 8, 8)
  DATA R1(I,J, 1)
  I/J 1 2 3 4
  -----
  1 0.0000 0.0000 0.0000 0.0000
  2 0.5000 0.5000 0.5000 0.5000
  3 1.0000 1.0000 1.0000 1.0000
  4 1.5000 1.5000 1.5000 1.5000

  DATA R1(I,J, 2)
  I/J 1 2 3 4
  -----
  1 0.0000 0.0000 0.0000 0.0000
  2 0.5000 0.5000 0.5000 0.5000
  3 1.0000 1.0000 1.0000 1.0000
  4 1.5000 1.5000 1.5000 1.5000

  DATA R1(I,J, 3)
  I/J 1 2 3 4

```



```

-----
1  0.0000  0.0000  0.0000  0.0000
2  0.5000  0.5000  0.5000  0.5000
3  1.0000  1.0000  1.0000  1.0000
4  1.5000  1.5000  1.5000  1.5000

```

```

DATA R1(I,J, 4)
I/J  1      2      3      4
-----
1  0.0000  0.0000  0.0000  0.0000
2  0.5000  0.5000  0.5000  0.5000
3  1.0000  1.0000  1.0000  1.0000
4  1.5000  1.5000  1.5000  1.5000

```

```

DATA R2(I,J, 1)
I/J  1      2      3      4
-----
1  2.0000  2.0000  2.0000  2.0000
2  1.5000  1.5000  1.5000  1.5000
3  1.0000  1.0000  1.0000  1.0000
4  0.5000  0.5000  0.5000  0.5000

```

```

DATA R2(I,J, 2)
I/J  1      2      3      4
-----
1  2.0000  2.0000  2.0000  2.0000
2  1.5000  1.5000  1.5000  1.5000
3  1.0000  1.0000  1.0000  1.0000
4  0.5000  0.5000  0.5000  0.5000

```

```

DATA R2(I,J, 3)
I/J  1      2      3      4
-----
1  2.0000  2.0000  2.0000  2.0000
2  1.5000  1.5000  1.5000  1.5000
3  1.0000  1.0000  1.0000  1.0000
4  0.5000  0.5000  0.5000  0.5000

```

```

DATA R2(I,J, 4)
I/J  1      2      3      4
-----
1  2.0000  2.0000  2.0000  2.0000
2  1.5000  1.5000  1.5000  1.5000
3  1.0000  1.0000  1.0000  1.0000
4  0.5000  0.5000  0.5000  0.5000

```

\*\* OUTPUT \*\*  
IERR = 0

```

CORRELATION R2(I,J, 1)
I/J  1      2      3      4      5      6      7      8
-----
1  3.00  6.00  9.00  12.00  9.00  6.00  3.00 -0.00
2  4.25  8.50  12.75  17.00  12.75  8.50  4.25 -0.00
3  4.00  8.00  12.00  16.00  12.00  8.00  4.00 -0.00
4  2.50  5.00  7.50  10.00  7.50  5.00  2.50 -0.00
5  1.00  2.00  3.00  4.00  3.00  2.00  1.00 -0.00
6  0.25  0.50  0.75  1.00  0.75  0.50  0.25  0.00
7  0.00  0.00  -0.00  -0.00  0.00  0.00  0.00  0.00
8  0.00  0.00  -0.00  -0.00  -0.00  0.00  0.00  0.00

```

```

CORRELATION R2(I,J, 2)
I/J  1      2      3      4      5      6      7      8
-----
1  6.00  12.00  18.00  24.00  18.00  12.00  6.00 -0.00
2  8.50  17.00  25.50  34.00  25.50  17.00  8.50 -0.00
3  8.00  16.00  24.00  32.00  24.00  16.00  8.00 -0.00
4  5.00  10.00  15.00  20.00  15.00  10.00  5.00 -0.00
5  2.00  4.00  6.00  8.00  6.00  4.00  2.00 -0.00
6  0.50  1.00  1.50  2.00  1.50  1.00  0.50 -0.00
7  0.00  0.00  0.00  -0.00  0.00  0.00  0.00  0.00
8  0.00  0.00  -0.00  -0.00  0.00  0.00  0.00  0.00

```

```

CORRELATION R2(I,J, 3)
I/J  1      2      3      4      5      6      7      8
-----
1  9.00  18.00  27.00  36.00  27.00  18.00  9.00 -0.00
2  12.75  25.50  38.25  51.00  38.25  25.50  12.75 -0.00
3  12.00  24.00  36.00  48.00  36.00  24.00  12.00 -0.00
4  7.50  15.00  22.50  30.00  22.50  15.00  7.50 -0.00
5  3.00  6.00  9.00  12.00  9.00  6.00  3.00 -0.00
6  0.75  1.50  2.25  3.00  2.25  1.50  0.75 -0.00
7  0.00  0.00  -0.00  -0.00  -0.00  -0.00  -0.00 -0.00
8  0.00  -0.00  0.00  -0.00  0.00  0.00  -0.00  0.00

```

```

CORRELATION R2(I,J, 4)
I/J  1      2      3      4      5      6      7      8
-----
1  12.00  24.00  36.00  48.00  36.00  24.00  12.00 -0.00
2  17.00  34.00  51.00  68.00  51.00  34.00  17.00 -0.00
3  16.00  32.00  48.00  64.00  48.00  32.00  16.00 -0.00
4  10.00  20.00  30.00  40.00  30.00  20.00  10.00 -0.00
5  4.00  8.00  12.00  16.00  12.00  8.00  4.00  0.00
6  1.00  2.00  3.00  4.00  3.00  2.00  1.00  0.00
7  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  0.00
8  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  -0.00  0.00

```

```

CORRELATION R2(I,J, 5)
I/J  1      2      3      4      5      6      7      8
-----
1  9.00  18.00  27.00  36.00  27.00  18.00  9.00 -0.00
2  12.75  25.50  38.25  51.00  38.25  25.50  12.75 -0.00

```

3	12.00	24.00	36.00	48.00	36.00	24.00	12.00	-0.00
4	7.50	15.00	22.50	30.00	22.50	15.00	7.50	-0.00
5	3.00	6.00	9.00	12.00	9.00	6.00	3.00	-0.00
6	0.75	1.50	2.25	3.00	2.25	1.50	0.75	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	-0.00	0.00	-0.00	0.00	0.00

CORRELATION R2(I,J, 6)								
I/J	1	2	3	4	5	6	7	8
1	6.00	12.00	18.00	24.00	18.00	12.00	6.00	-0.00
2	8.50	17.00	25.50	34.00	25.50	17.00	8.50	-0.00
3	8.00	16.00	24.00	32.00	24.00	16.00	8.00	-0.00
4	5.00	10.00	15.00	20.00	15.00	10.00	5.00	-0.00
5	2.00	4.00	6.00	8.00	6.00	4.00	2.00	-0.00
6	0.50	1.00	1.50	2.00	1.50	1.00	0.50	0.00
7	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	-0.00	-0.00	0.00	0.00	0.00	0.00

CORRELATION R2(I,J, 7)								
I/J	1	2	3	4	5	6	7	8
1	3.00	6.00	9.00	12.00	9.00	6.00	3.00	0.00
2	4.25	8.50	12.75	17.00	12.75	8.50	4.25	-0.00
3	4.00	8.00	12.00	16.00	12.00	8.00	4.00	-0.00
4	2.50	5.00	7.50	10.00	7.50	5.00	2.50	-0.00
5	1.00	2.00	3.00	4.00	3.00	2.00	1.00	0.00
6	0.25	0.50	0.75	1.00	0.75	0.50	0.25	0.00
7	0.00	-0.00	-0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	-0.00	-0.00	-0.00	0.00	0.00	0.00	0.00

CORRELATION R2(I,J, 8)								
I/J	1	2	3	4	5	6	7	8
1	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	0.00	0.00
2	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
3	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
4	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
5	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
6	-0.00	-0.00	0.00	0.00	0.00	0.00	-0.00	-0.00
7	-0.00	-0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	-0.00	0.00	-0.00	0.00	0.00	0.00	0.00	0.00

---

## 2.16 パワー・スペクトル解析

### 2.16.1 DFPS1D, RFPS1D

#### 1次元フーリエ・ピリオドグラム

##### (1) 機能

系列  $u_j$  ( $j = 0, \dots, n-1$ ) の (修正) フーリエ・ピリオドグラムを求める. フーリエ・ピリオドグラム  $p_k$  は次式で定義される.

$$p_k = \frac{\left| \sum_{j=0}^{n-1} w_j u_j e^{-2\pi\sqrt{-1}\frac{jk}{n}} \right|^2}{n\beta} \quad (k = 0, 1, \dots, \lfloor \frac{n}{2} \rfloor)$$

ただし,  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す.  $w_j$  は打ち切り関数 (窓関数) であり, 生のフーリエ・ピリオドグラムの場合には,  $w_j = 1$  ( $j = 0, \dots, n-1$ ),  $\beta = n$  とし, 修正ピリオドグラムの場合には

$$\beta = \begin{cases} \sum_{j=0}^{n-1} w_j^2 & (\text{窓関数によるパワー補正式を用いる場合}) \\ n & (\text{それ以外}) \end{cases}$$

とする. なお, ピリオドグラム  $p_k$  は両側パワー・スペクトルの半周期分 (周期  $n$ ) に相当し, 残りは  $p_{-k} = p_k$  の関係より得られる. また, 対応する系列の全パワーは,

$$\frac{\sum_{j=0}^{n-1} \{u_j\}^2}{n}$$

である.

##### (2) 使用法

倍精度サブルーチン:

CALL DFPS1D (N, R, LD, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFPS1D (N, R, LD, ISW, IWK, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{array} \right\}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	N	I	1	入 力	系列 $u_j$ の長さ $n$ (注意事項 (d) 参照)
2	R	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LD	入 力	系列 $u_j$ の値 (注意事項 (a) 参照)
				出 力	系列 $u_j$ のフーリエ・ピリオドグラム $p_k$ の値 (注意事項 (b), (c) 参照)
3	LD	I	1	入 力	配列 R の大きさ
4	ISW	I	1	入 力	処理スイッチ (注意事項 (e) 参照) ISW=0:生のフーリエ・ピリオドグラムを計算する ISW=±1:ユーザ定義窓関数を利用して計算する ISW=±2:Hanning 窓関数を利用して計算する ISW=±3:Bartlett 窓関数を利用して計算する ISW=±4:Welch 窓関数を利用して計算する ISW=±5:Parzen 窓関数を利用して計算する なお、窓関数によるパワー補正式を用いる場合は ISW > 0, それ以外の場合は ISW < 0 とする.
5	IWK	I	20	ワ ーク	作業領域
6	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	N+LD	ワ ーク	作業領域 ISW=±1 の場合には、ユーザ定義窓関数の値を入力する. (注意事項 (e) 参照)
7	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a)  $ISW \in \{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5\}$

(b)  $N > 1$

(c) N が奇数の時 :

$$LD \geq N + 1$$

N が偶数の時 :

$$LD \geq N + 2$$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3030	制限条件 (c) を満足しなかった.	
4000	ISW = 1 の時に、ユーザ定義窓関数が $w_j = 0$ ( $j = 0, \dots, n-1$ ) であった.	

## (6) 注意事項

- (a) 配列 R にはそれぞれ系列
- $u_j$
- の値を次のように格納する.

$$\begin{aligned} u_0 &\rightarrow R(1) \\ u_1 &\rightarrow R(2) \\ \dots &\dots \dots \\ u_{n-1} &\rightarrow R(N) \end{aligned}$$

なお, 配列 R の R (N+1) 以降の要素には値を入力する必要が無い.

- (b) フーリエ・ピリオドグラム
- $p_k$
- の値は配列 R に以下のように得られる.

$$\begin{aligned} p_0 &\rightarrow R(1) \\ p_1 &\rightarrow R(2) \\ \dots &\dots \dots \\ p_{\lfloor \frac{n+1}{2} \rfloor - 1} &\rightarrow R(\lfloor \frac{n+1}{2} \rfloor) \end{aligned}$$

なお,  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す.

- (c) 得られるフーリエ・ピリオドグラム  $p_k$  は両側パワー・スペクトル(負の周波数を考える場合に相当)の半周期に対応し, 対応する周波数  $\xi_k$  は  $\xi_k = \frac{k}{n\Delta x}$  ( $\Delta x$ : 標本化間隔) で与えられる. このとき  $-\xi_k$  に対応する成分は  $p_k$  となる. 片側スペクトルに対応するフーリエ・ピリオドグラム  $\hat{p}_k$  は  $\hat{p}_0 = p_0$ ;  $\hat{p}_k = 2p_k$  ( $k = 1, 2, \dots, m-1$ ) とすることによって得られる. ただし,  $n$  が偶数の場合には  $m = \frac{n}{2}$ ,  $\hat{p}_m = p_m$  とし,  $n$  が奇数の場合は  $m = \frac{n+1}{2}$  とする.
- (d) 系列  $u_j$  の長さ  $N$  の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば,  $N = 289 (= 17^2)$  とするよりも  $N = 300 (= 2^2 \times 3 \times 5^2)$  や  $320 (= 2^6 \times 5)$ ,  $384 (= 2^7 \times 3)$  などとした方が効率が良い. なお, データ数を大きくできない場合には, データの最後に 0 を必要なだけ補って  $N$  を調整して計算を行う.
- (e) 処理スイッチ ISW の値によって, 以下の様に打ち切り関数 (窓関数) を変更することができる.

$$w_j = \begin{cases} \begin{cases} \sin^2(\pi v_j) & \text{ISW} = \pm 2 \text{ (Hanning 窓)} \\ 1 - |2v_j - 1| & \text{ISW} = \pm 3 \text{ (Bartlett 窓)} \\ 1 - (2v_j - 1)^2 & \text{ISW} = \pm 4 \text{ (Welch 窓)} \end{cases} \\ \begin{cases} \begin{cases} 16v_j^3 & 0 \leq v_j < \frac{1}{4} \\ 1 - 6v_j(v_j - 1)^2 & \frac{1}{4} \leq v_j \leq \frac{1}{2} \\ 1 - 6v_j(v_{n-j+1} - 1)^2 & \frac{1}{2} \leq v_j \leq \frac{3}{4} \\ 16v_{n-j+1}^3 & \frac{3}{4} \leq v_j < 1 \end{cases} & \text{ISW} = \pm 5 \text{ (Parzen 窓)} \end{cases} \end{cases}$$

ただし,  $v_j = \frac{j}{n}$ . したがって, 上述のような窓関数を用いる場合には系列  $u_j$  の最初の要素  $u_0$  は修正ピリオドグラムの計算に影響しない. これを避けたい場合には, 実際に計算したい系列の長さよりも 1 大きい数を  $N$  に指定し,  $u_1$  以降に有効なデータを設定すれば良い. なお, 窓関数は  $|x| \leq 1$  でのみ非ゼロとなる時間 (または空間) 領域関数としてそれぞれ次の様に表される.

$$w(x) = \begin{cases} \frac{1 + \cos \pi x}{2} = \cos^2 \frac{\pi x}{2} & \text{Hanning 窓} \\ 1 - |x| & \text{Bartlett 窓} \\ 1 - x^2 & \text{Welch 窓} \\ \begin{cases} 1 - 6x^2 + 6|x|^3 & |x| \leq \frac{1}{2} \\ 2(1 - |x|)^3 & \frac{1}{2} \leq |x| \leq 1 \end{cases} & \text{Parzen 窓} \end{cases}$$

また, ユーザ独自の窓関数値  $w_j$  を利用したい場合には  $\text{ISW} = \pm 1$  として作業配列 WK に

$$\text{WK}(j+1) = w_j \quad (j = 0, \dots, n-1)$$

と設定してこのサブルーチンを呼び出す。

- (f) 生のピリオドグラムはその定義から自己相関関数の離散フーリエ変換近似とみなせる。有効データ数  $n$  の離散関数の自己相関関数の有効データ長は  $2n - 1$  であるので、一般の関数のパワー・スペクトルを生々のピリオドグラムで近似することは、1つの周期が以下のように与えられる方形打ち切り関数  $w(k)$  で関数を打ち切ったことに相当する。

$$w(k) = \begin{cases} 1 & k = 0, 1, \dots, n-1 \\ 0 & \text{それ以外} \end{cases}$$

方形関数のフーリエ変換は周波数を  $f$  とした場合、 $\frac{\sin f}{f}$  型の関数形をしており、中心周波数の周りに小さくないサイドローブを持っている。したがって、たとえば、周期関数を1周期の整数倍でない幅で単純に打ち切って標準化した場合、周波数領域では、生のピリオドグラムはパワー・スペクトルを求めたい周期関数のフーリエ変換と  $\frac{\sin f}{f}$  型関数との畳み込みとなるので、漏れ (leakage) と呼ばれる余分な周波数成分が発生する。このような漏れを抑止するためには、単純な打ち切りを行わずに Hanning 窓関数のような周波数領域でのサイドローブが小さい打ち切り関数を用いる。ただし、一般に漏れを抑圧すればする程離散フーリエ変換の結果は拡がりばやけたものとなる。したがって、パワー・スペクトルを推定する場合、目的に応じて、すなわち、スペクトル幅を問題としているの中心周波数を問題としているのか等に応じて、適切な打ち切り関数を選択する必要がある。

- (g) 離散フーリエ変換の分解度 (周波数領域での標本間隔)  $\frac{1}{nT}$  を上げるには標本数  $n$  を増やすか標本間隔  $T$  を増やせば良いが、標本間隔と分解度を一定に保った状態でパワー・スペクトルの推定値の精度を上げるために、標本数  $n$  の標本を  $m$  組とって  $m$  組それぞれについて修正ピリオドグラムを求めてその平均をとるという手法がよく取られる。この場合、系列から  $m$  組の標本データをオーバーラップして取るというような手法も提案されている。詳細は参考文献等を参照されたい。
- (h) パワー・スペクトルを求める場合、フーリエ変換の周波数推移に関する性質すなわち時間 (または空間) 領域で  $e^{2\pi\sqrt{-1}f_0t}$  を掛けることは周波数領域では周波数を  $f_0$  だけシフトすることに対応し、関数の形状は変わらないという性質を利用して、パワー・スペクトルの中心周波数をあらかじめシフトして計算することで計算に必要なデータ点数を削減するという手法も良く用いられる。なお、このような操作は変調 (modulation) として知られている。ただし、 $N$  が奇数ならば  $LD=N+1$ ,  $N$  が偶数ならば  $LD=N+2$  である。
- (i) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない。

## (7) 使用例

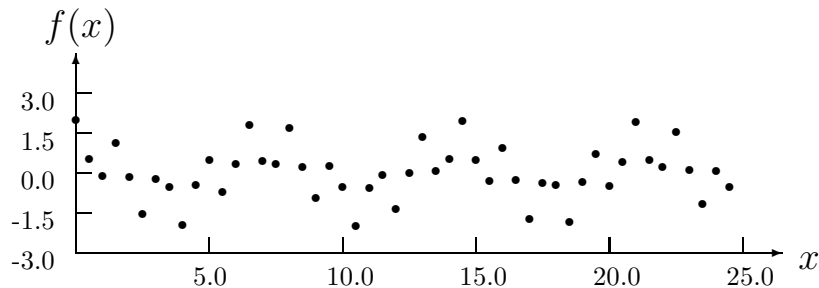
## (a) 問題

次式で定義される波形を標本化間隔  $\Delta x$  で離散化し、フーリエ・ピリオドグラムを計算し、パワー・スペクトルを推定する。

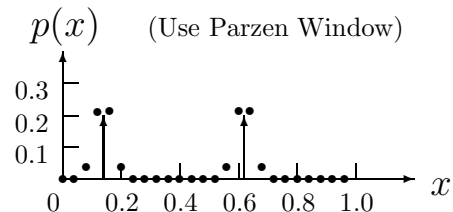
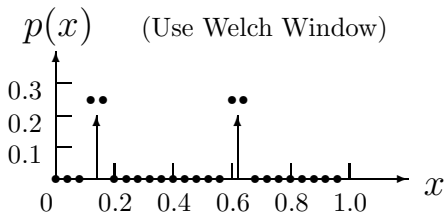
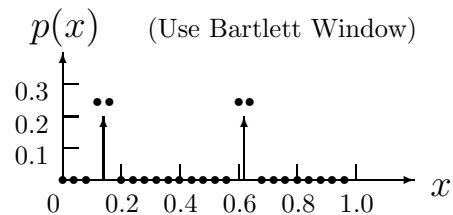
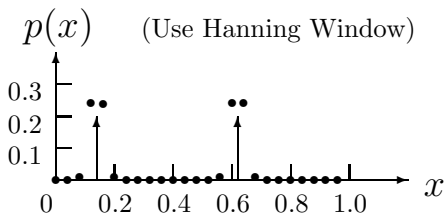
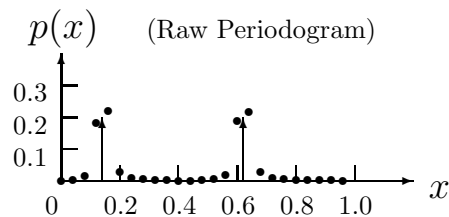
$$f(x) = \cos 2\pi f_1 x + \cos 2\pi f_2 x$$

## 備考

$f_1 = 0.62$ ,  $f_2 = 0.14$  とした場合,  $f(x)$  を区間  $[0, 25)$ ,  $\Delta x = 0.5$  で標本化すると, 以下のようなグラフとなる. 本来は, 標本化定理を参考に目的に応じてより細かく標本化すべきであるが, この程度の標本化でも窓関数の選択による違いについての傾向はわかる.



また, 対応するフーリエ・ピリオドグラムは以下の様なグラフとなる (上向き矢印は信号周波数). 信号周波数としてわざと打ち切りによる不連続が大きくなる周波数を用いているので生のフーリエ・ピリオドグラムでは漏れが大きくなっている.



(b) 入力データ

標本化データ

$$R(j) = f((j-1)\Delta x) \quad (j = 1, 2, \dots, N)$$

ただし,  $\Delta x = 0.5$

N, ISW

(c) 主プログラム

```

PROGRAM BFPS1D
! *** EXAMPLE OF DFPS1D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER N,LD,ISW,IERR,IWK(20)
INTEGER I,M,ND2,IS
INTEGER NO,ISWO
PARAMETER (NO=50)
PARAMETER (ISWO=4)
PARAMETER (LD=NO+2)
REAL(8) R(LD,-1:ISWO),WK(NO+LD)
REAL(8) P(-1:ISWO)
REAL(8) T,DT
REAL(8) PAI,F0,F1,F2
PARAMETER(PAI=3.141592D0)
!
N=NO
WRITE (6,1000) ISWO+1,N
DT=0.5D0
F0=1.0D0/(2.0*DT)
F1=0.62D0*F0
F2=0.14D0*F0
ND2=(N+1)/2
P(-1)=0.0D0
DO 100 I=1,N
    T=DBLE(I-1)*DT
    R(I,-1)=COS(2*PAI*F1*T)+COS(2*PAI*F2*T)
    P(-1)=P(-1)+R(I,-1)*R(I,-1)
100 CONTINUE
P(-1)=P(-1)/DBLE(N)
WRITE (6,1100) (I,DBLE(I-1)*DT,R(I,-1),I=1,N)
WRITE (6,1150) P(-1)
WRITE (6,1160) F1,F2
IS=0
DO 400 ISW=0,ISWO
    DO 200 I=1,N
        R(I,ISW)=R(I,-1)
200 CONTINUE
IF (ISW.NE.0) IS=ISW+1
CALL DFPS1D(N,R(1,ISW),LD,IS,IWK,WK,IERR)
!**** FOR ONE-SIDED POWER SPECTRAL DENSITIES
IF (MOD(N,2).EQ.0) THEN
    M=ND2-1
ELSE
    M=ND2
ENDIF
DO 300 I=2,M
    R(I,ISW)=2.0D0*R(I,ISW)
300 CONTINUE
P(ISW)=0.0D0
DO 500 I=1,ND2
    P(ISW)=P(ISW)+R(I,ISW)
500 CONTINUE
400 CONTINUE
WRITE (6,1200)
WRITE (6,1300) IERR
WRITE (6,1400)&
    (I,(I-1)/(DT*N),(R(I,ISW),ISW=0,ISWO),I=1,ND2)
WRITE (6,1500) (P(ISW),ISW=0,ISWO)
1000 FORMAT(' ',/,/,&
    ' *** DFPS1D ***',/,&
    2X,'** INPUT **',/,&
    6X,'ISW =0, 2 TO ',I3,/,&
    6X,'N =',I3)
1100 FORMAT(12X,' TIME SERIES DATA',/,&
    9X,'I',4X,' TIME ',4X,'R(I)',/,&
    50(8X,I2,2F9.4,/) )
1150 FORMAT(6X,' TIME DOMAIN POWER =',F10.4)
1160 FORMAT(6X,' SIGNAL FREQUENCY =',2F10.4)
1200 FORMAT(2X,'** OUTPUT **')
1300 FORMAT(6X,' IERR =',I5)
1400 FORMAT(10X,&
    '(MODIFIED) PERIODOGRAM/ONE-SIDED POWER SPECTRUM ESTIMATION',/,&
    6X,'I',3X,' FREQ.',/,&
    1X,' RAW ',1X,' HANNING ',&
    1X,' BARTLETT ',1X,' WELCH ',1X,' PARZEN ',/,&
    25(5X,I2,6F10.4,/) )
1500 FORMAT(17X,' FREQUENCY DOMAIN POWER',/,&
    20X,' RAW ',1X,' HANNING ',&
    1X,' BARTLETT ',1X,' WELCH ',1X,' PARZEN ',/,&
    (17X,5F10.4) )
END

```



## (d) 出力結果

```

*** DFPS1D ***
** INPUT **
ISW =0, 2 TO 5
N = 50
TIME SERIES DATA
I TIME R(I)
1 0.0000 2.0000
2 0.5000 0.5367
3 1.0000 -0.0915
4 1.5000 1.1535
5 2.0000 -0.1246
6 2.5000 -1.5388
7 3.0000 -0.2389
8 3.5000 -0.5163
9 4.0000 -1.9219
10 4.5000 -0.4359
11 5.0000 0.5000
12 5.5000 -0.7190
13 6.0000 0.3484
14 6.5000 1.8266
15 7.0000 0.4563
16 7.5000 0.3633
17 8.0000 1.6976
18 8.5000 0.2428
19 9.0000 -0.9391
20 9.5000 0.2888
21 10.0000 -0.5000
22 10.5000 -1.9803
23 11.0000 -0.5428
24 11.5000 -0.0860
25 12.0000 -1.3556
26 12.5000 -0.0000
27 13.0000 1.3556
28 13.5000 0.0860
29 14.0000 0.5428
30 14.5000 1.9803
31 15.0000 0.5000
32 15.5000 -0.2888
33 16.0000 0.9391
34 16.5000 -0.2428
35 17.0000 -1.6976
36 17.5000 -0.3633
37 18.0000 -0.4563
38 18.5000 -1.8266
39 19.0000 -0.3485
40 19.5000 0.7190
41 20.0000 -0.5000
42 20.5000 0.4358
43 21.0000 1.9219
44 21.5000 0.5163
45 22.0000 0.2389
46 22.5000 1.5388
47 23.0000 0.1246
48 23.5000 -1.1535
49 24.0000 0.0915
50 24.5000 -0.5367

TIME DOMAIN POWER = 1.0000
SIGNAL FREQUENCY = 0.6200 0.1400
** OUTPUT **
IERR = 0
(MODIFIED) PERIODOGRAM/ONE-SIDED POWER SPECTRUM ESTIMATION
I FREQ. RAW HANNING BARTLETT WELCH PARZEN
1 0.0000 0.0016 0.0000 0.0000 0.0000 0.0000
2 0.0400 0.0051 0.0001 0.0002 0.0000 0.0006
3 0.0800 0.0166 0.0094 0.0026 0.0003 0.0369
4 0.1200 0.1841 0.2408 0.2437 0.2494 0.2116
5 0.1600 0.2211 0.2398 0.2446 0.2498 0.2121
6 0.2000 0.0286 0.0096 0.0029 0.0003 0.0373
7 0.2400 0.0117 0.0002 0.0004 0.0000 0.0006
8 0.2800 0.0068 0.0000 0.0001 0.0000 0.0000
9 0.3200 0.0047 0.0000 0.0000 0.0000 0.0000
10 0.3600 0.0036 0.0000 0.0000 0.0000 0.0000
11 0.4000 0.0032 0.0000 0.0000 0.0000 0.0000
12 0.4400 0.0033 0.0000 0.0001 0.0000 0.0000
13 0.4800 0.0042 0.0000 0.0001 0.0000 0.0000
14 0.5200 0.0072 0.0002 0.0004 0.0000 0.0006
15 0.5600 0.0197 0.0096 0.0031 0.0003 0.0373
16 0.6000 0.1906 0.2403 0.2463 0.2496 0.2121
17 0.6400 0.2177 0.2401 0.2462 0.2497 0.2121
18 0.6800 0.0285 0.0096 0.0030 0.0003 0.0373
19 0.7200 0.0122 0.0002 0.0004 0.0000 0.0006
20 0.7600 0.0075 0.0000 0.0001 0.0000 0.0000
21 0.8000 0.0054 0.0000 0.0000 0.0000 0.0000
22 0.8400 0.0044 0.0000 0.0000 0.0000 0.0000
23 0.8800 0.0038 0.0000 0.0000 0.0000 0.0000
24 0.9200 0.0034 0.0000 0.0000 0.0000 0.0000
25 0.9600 0.0016 0.0000 0.0000 0.0000 0.0000

FREQUENCY DOMAIN POWER
RAW HANNING BARTLETT WELCH PARZEN
0.9968 1.0000 0.9943 0.9999 0.9991

```

## 2.16.2 DFPS2D, RFPS2D 2次元フーリエ・ピリオドグラム

### (1) 機能

系列  $u_{j_x, j_y}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ) の (修正) フーリエ・ピリオドグラムを求める。フーリエ・ピリオドグラム  $p_{k_x, k_y}$  は次式で定義される。

$$p_{k_x, k_y} = \frac{\left| \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} w_{j_x}^{(x)} w_{j_y}^{(y)} u_{j_x, j_y} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y}\right)} \right|^2}{n_x n_y \beta} \quad (k_x = 0, 1, \dots, \lfloor \frac{n_x}{2} \rfloor; k_y = 0, 1, \dots, n_y - 1)$$

ただし,  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。  $w_{j_x}^{(x)}$ ,  $w_{j_y}^{(y)}$  は打ち切り関数 (窓関数) であり, 生のフーリエ・ピリオドグラムの場合には,  $w_{j_x}^{(x)} = w_{j_y}^{(y)} = 1$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ),  $\beta = n_x n_y$  とし, 修正ピリオドグラムの場合には

$$\beta = \begin{cases} \left( \sum_{j_x=0}^{n_x-1} (w_{j_x}^{(x)})^2 \right) \left( \sum_{j_y=0}^{n_y-1} (w_{j_y}^{(y)})^2 \right) & \text{(窓関数によるパワー補正式を用いる場合)} \\ n_x n_y & \text{(それ以外)} \end{cases}$$

とする。なお, ピリオドグラム  $p_{k_x, k_y}$  は  $k_x$  についての半周期分 (周期  $(n_x, n_y)$ ) に相当し, 残りの半周期分は以下の関係から得られる。

$$\begin{aligned} p_{n_x - k_x, n_y - k_y} &= p_{k_x, k_y} \\ p_{n_x - k_x, k_y} &= p_{k_x, n_y - k_y} \end{aligned}$$

また, 対応する系列の全パワーは,

$$\frac{\sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} \{u_{j_x, j_y}\}^2}{n_x n_y}$$

である。

### (2) 使用法

倍精度サブルーチン:

CALL DFPS2D (NX, NY, R, LX, LY, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFPS2D (NX, NY, R, LX, LY, ISW, IWK, WK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{array} \right\}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	系列 $u_{j_x, j_y}$ の $j_x$ 方向の長さ $n_x$ (注意事項 (d) 参照)
2	NY	I	1	入 力	系列 $u_{j_x, j_y}$ の $j_y$ 方向の長さ $n_y$ (注意事項 (d) 参照)
3	R	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	LX, LY	入 力	系列 $u_{j_x, j_y}$ の値 (注意事項 (a) 参照)
				出 力	系列 $u_{j_x, j_y}$ のフーリエ・ピリオドグラム $p_{k_x, k_y}$ の値 (注意事項 (b), (c) 参照)
4	LX	I	1	入 力	配列 R の整合寸法
5	LY	I	1	入 力	配列 R の第 2 寸法
6	ISW	I	1	入 力	処理スイッチ (注意事項 (e) 参照) ISW=0:生のフーリエ・ピリオドグラムを計算する ISW=±1:ユーザ定義窓関数を利用して計算する ISW=±2:Hanning 窓関数を利用して計算する ISW=±3:Bartlett 窓関数を利用して計算する ISW=±4:Welch 窓関数を利用して計算する ISW=±5:Parzen 窓関数を利用して計算する なお, 窓関数によるパワー補正式を用いる場合は ISW > 0, それ以外の場合は ISW < 0 とする.
7	IWK	I	40	ワーク	作業領域
8	WK	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	内容参照	ワーク	作業領域 ISW=±1 の場合には, ユーザ定義窓関数の値を入力する (注意事項 (e) 参照). 大きさ: $NX + 2 \times NY + LX \times LY$
9	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5\}$

(b)  $NX > 1$

$NY > 1$

(c) NX が奇数の時:

$LX \geq NX + 1$

$LY \geq NY$

NX が偶数の時:

$LX \geq NX + 2$

$LY \geq NY$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3030	制限条件 (c) を満足しなかった.	
4000	ISW = 1 の場合にユーザ定義窓関数が $w_{j_x}^{(x)} = 0 \quad (j_x = 0, \dots, n_x - 1)$ であつた.	
4010	ISW = 1 の場合にユーザ定義窓関数が $w_{j_y}^{(y)} = 0 \quad (j_y = 0, \dots, n_y - 1)$ であつた.	

(6) 注意事項

- (a) 配列 R の各要素と系列  $u_{j_x, j_y}$  の値は以下の様に対応する.

$$u_{j_x, j_y} \leftrightarrow R(j_x + 1, j_y + 1)$$

ただし,  $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$  であり, それ以外の要素には値を入力する必要が無い. なお, 主記憶のバンク競合を避けるために配列 R の整合寸法について LX/2, LY が奇数になるように設定するのが望ましい. 通常, たとえば NX が (4 の倍数)+2 のときは LX=NX+4 とする.

- (b) フーリエ・ピリオドグラム  $p_{k_x, k_y}$  の値は配列 R の各要素と以下の様に対応する.

$$p(k_x, k_y) \leftrightarrow R(k_x + 1, k_y + 1) \quad (k_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor; k_y = 0, \dots, n_y - 1)$$

なお,  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す.

- (c) 得られるフーリエ・ピリオドグラム  $p_{k_x, k_y}$  ( $k_x = 0, 1, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $k_y = 0, 1, \dots, n_y - 1$ ) に対応する周波数  $(\xi_{k_x}, \eta_{k_y})$  は

$$\xi_{k_x} = \frac{k_x}{n_x \Delta} \quad (k_x = 0, 1, \dots, \lfloor \frac{n_x}{2} \rfloor)$$

$$\eta_{k_y} = \begin{cases} \frac{k_y}{n_y \Delta} & (k_y = 0, 1, \dots, \lfloor \frac{n_y}{2} \rfloor) \\ \frac{k_y - n_y}{n_y \Delta} & (k_y = \lfloor \frac{n_y}{2} \rfloor + 1, \dots, n_y - 1) \end{cases}$$

( $\Delta$ : 標本化間隔) で与えられる.

- (d) 系列  $u_{j_x, j_y}$  の長さ NX, NY の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える. たとえば, NX = 289(=17<sup>2</sup>) とするよりも NX = 300(=2<sup>2</sup> × 3 × 5<sup>2</sup>) や 320(=2<sup>6</sup> × 5), 384(=2<sup>7</sup> × 3) などとした方が効率が良い. なお, データ数を大きくできない場合には, データの最後に 0 を必要なだけ補って NX を調整して計算を行う.

- (e) 処理スイッチ ISW の値によって, 以下の様に打ち切り関数 (窓関数) を変更することができる.

$$w_j = \begin{cases} \sin^2(\pi v_j) & \text{ISW} = \pm 2 \text{ (Hanning 窓)} \\ 1 - |2v_j - 1| & \text{ISW} = \pm 3 \text{ (Bartlett 窓)} \\ 1 - (2v_j - 1)^2 & \text{ISW} = \pm 4 \text{ (Welch 窓)} \\ \left\{ \begin{array}{ll} 16v_j^3 & 0 \leq v_j < \frac{1}{4} \\ 1 - 6v_j(v_j - 1)^2 & \frac{1}{4} \leq v_j \leq \frac{1}{2} \\ 1 - 6v_j(v_{n-j+1} - 1)^2 & \frac{1}{2} \leq v_j \leq \frac{3}{4} \\ 16v_{n-j+1}^3 & \frac{3}{4} \leq v_j < 1 \end{array} \right\} & \text{ISW} = \pm 5 \text{ (Parzen 窓)} \end{cases}$$

ただし,  $v_j = \frac{j}{n}$  で,  $w_{j_x}^{(x)}$  については  $j = j_x, n = n_x$  とし,  $w_{j_y}^{(y)}$  については  $j = j_y, n = n_y$  とする. したがって, 上述のような窓関数を用いる場合には系列  $u_{j_x, j_y}$  の要素  $u_{0, j_y}, u_{j_x, 0}$  は修正ピリオドグラムの計算に影響しない. これを避けたい場合には, 実際に計算したい系列の長さよりも 1 大きい数を  $NX, NY$  に指定し,  $j_x, j_y$  について 1 以降の対応する要素に有効なデータを設定すれば良い. なお, 窓関数は  $|x| \leq 1$  のみ非ゼロとなる時間 (または空間) 領域関数としてそれぞれ次の様に表される.

$$w(x) = \begin{cases} \frac{1 + \cos \pi x}{2} = \cos^2 \frac{\pi x}{2} & \text{Hanning 窓} \\ 1 - |x| & \text{Bartlett 窓} \\ 1 - x^2 & \text{Welch 窓} \\ \left\{ \begin{array}{ll} 1 - 6x^2 + 6|x|^3 & |x| \leq \frac{1}{2} \\ 2(1 - |x|)^3 & \frac{1}{2} \leq |x| \leq 1 \end{array} \right\} & \text{Parzen 窓} \end{cases}$$

また, ユーザ独自の窓関数値  $w_{j_x}^{(x)}, w_{j_y}^{(y)}$  を利用したい場合には  $ISW = \pm 1$  として作業配列  $WK$  に

$$WK(j_x + 1) = w_{j_x}^{(x)} \quad (j_x = 0, \dots, n_x - 1), WK(n_x + j_y + 1) = w_{j_y}^{(y)} \quad (j_y = 0, \dots, n_y - 1)$$

と設定してこのサブルーチンを呼び出す.

- (f) 生のピリオドグラムはその定義から自己相関関数の離散フーリエ変換近似とみなせる. 有効データ数  $n$  の離散関数の自己相関関数の有効データ長は  $2n - 1$  であるので, 一般の関数のパワー・スペクトルを生みのピリオドグラムで近似することは, 1 つの周期が以下のように与えられる方形打ち切り関数  $w(k)$  で関数を打ち切ったことに相当する.

$$w(k) = \begin{cases} 1 & k = 0, 1, \dots, n - 1 \\ 0 & \text{それ以外} \end{cases}$$

方形関数のフーリエ変換は周波数を  $f$  とした場合,  $\frac{\sin f}{f}$  型の関数形をしており, 中心周波数の周りに小さくないサイドローブを持っている. したがって, たとえば, 周期関数を 1 周期の整数倍でない幅で単純に打ち切って標準化した場合, 周波数領域では, 生のピリオドグラムはパワー・スペクトルを求めたい周期関数のフーリエ変換と  $\frac{\sin f}{f}$  型関数との畳み込みとなるので, 漏れ (leakage) と呼ばれる余分な周波数成分が発生する. このような漏れを抑止するためには, 単純な打ち切りを行わずに Hanning 窓関数のような周波数領域でのサイドローブが小さい打ち切り関数を用いる. ただし, 一般に漏れを抑圧すればする程離散フーリエ変換の結果は拡がりばやけたものとなる. したがって, パワー・スペクトルを推定する場合, 目的に応じて, すなわち, スペクトル幅を問題としているの中心周波数を問題としているのか等に応じて, 適切な打ち切り関数を選択する必要がある.

- (g) 離散フーリエ変換の分解度 (周波数領域での標本間隔)  $\frac{1}{nT}$  を上げるには標本数  $n$  を増やすか標本間隔  $T$  を増やせば良いが, 標本間隔と分解度を一定に保った状態でパワー・スペクトルの推定値の精度を上げるために, 標本数  $n$  の標本を  $m$  組とって  $m$  組それぞれについて修正ピリオドグラムを求めてその平均をとるという手法がよく取られる. この場合, 系列から  $m$  組の標本データをオーバーラップして取るというような手法も提案されている. 詳細は参考文献等を参照されたい.
- (h) パワー・スペクトルを求める場合, フーリエ変換の周波数推移に関する性質すなわち時間 (または空間) 領域で  $e^{2\pi\sqrt{-1}f_0 t}$  を掛けることは周波数領域では周波数を  $f_0$  だけシフトすることに対応し, 関数の形状は変わらないという性質を利用して, パワー・スペクトルの中心周波数をあらかじめシフトして計算することで計算に必要なデータ点数を削減するという手法も良く用いられる. なお, このような操作は変調 (modulation) として知られている. ただし,  $LX = NX + 1$  ( $NX$  が奇数のとき) または  $LX = NX + 2$  ( $NX$  が偶数のとき)  $LY = NY$  である.
- (i) この機能は逐次版および OpenMP 不使用の MPI 版ライブラリにおいてスレッドセーフではない.

(7) 使用例

(a) 問題

次式で定義される波形を標本化間隔  $\Delta$  で離散化し、フーリエ・ピリオドグラムを計算し、パワー・スペクトルを推定する。

$$f(x, y) = \cos 2\pi f_1 x + \cos 2\pi f_2 y$$

(b) 入力データ

標本化データ

$$R(j_x + 1, j_y + 1) = f(j_x \Delta, j_y \Delta) \quad (j_x = 0, 1, \dots, NX - 1; j_y = 0, 1, \dots, NY - 1)$$

ただし,  $\Delta = 0.5$

NX, NY, ISW

(c) 主プログラム

```

PROGRAM BFPS2D
! *** EXAMPLE OF DFPS2D ***
IMPLICIT REAL(8) (A-H,O-Z)
INTEGER NX,NY,LX,LY,ISW,IERR,IWK(40)
INTEGER I,J,M,ND2,IS
INTEGER NO,ISWO
PARAMETER (NO=8)
PARAMETER (ISWO=4)
PARAMETER (LX=NO+2,LY=NO)
REAL(8) R(LX,LY,-1:ISWO),WK(3*NO+LX*LY)
REAL(8) P(-1:ISWO)
REAL(8) TX,TY,DT,DFX,DFY
REAL(8) PAI,F0,F1,F2
PARAMETER(PAI=3.141592D0)
!
NX=NO
NY=NO
WRITE (6,1000) ISWO+1,NX,NY
DT=0.5D0
F0=1.0D0/(2.0*DT)
F1=0.62D0*F0
F2=0.14D0*F0
ND2=(NX+1)/2
DFX=1.0D0/(DT*NX)
DFY=1.0D0/(DT*NY)
P(-1)=0.0D0
DO 100 J=1,NY
  TY=DBLE(J-1)*DT
DO 101 I=1,NX
  TX=DBLE(I-1)*DT
  R(I,J,-1)=COS(2*PAI*F1*TX)+COS(2*PAI*F2*TY)
  P(-1)=P(-1)+R(I,J,-1)*R(I,J,-1)
101 CONTINUE
100 CONTINUE
P(-1)=P(-1)/(DBLE(NX)*DBLE(NY))
WRITE (6,1100) (I,(R(I,J,-1),J=1,NY),I=1,NX)
WRITE (6,1150) P(-1)
WRITE (6,1160) F1,F2
IS=0
DO 400 ISW=0,ISWO
DO 200 J=1,NY
DO 201 I=1,NX
  R(I,J,ISW)=R(I,J,-1)
201 CONTINUE
200 CONTINUE
IF (ISW.NE.0) IS=ISW+1
CALL DFPS2D(NX,NY,R(1,1,ISW),LX,LY,IS,IWK,WK,IERR)
P(ISW)=0.0D0
IF (MOD(NX,2).EQ.0) THEN
  M=ND2-1
  DO 500 J=1,NY
  DO 501 I=2,M
    P(ISW)=P(ISW)+2.0D0*R(I,J,ISW)
501 CONTINUE
500 CONTINUE
  DO 300 J=1,NY
    P(ISW)=P(ISW)+R(1,J,ISW)+R(ND2,J,ISW)
300 CONTINUE
ELSE
  M=ND2
  DO 600 J=1,NY
  DO 601 I=2,M
    P(ISW)=P(ISW)+2.0D0*R(I,J,ISW)
601 CONTINUE
600 CONTINUE
  DO 700 J=1,NY
    P(ISW)=P(ISW)+R(1,J,ISW)
700 CONTINUE
ENDIF
400 CONTINUE
WRITE (6,1200)
WRITE (6,1300) IERR
!

```

```

ISW=0
WRITE (6,1410) 'RAW',P(ISW),&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,ISW),J=1,(NY+1)/2),I=1,ND2)
!
ISW=1
WRITE (6,1410) 'HANNING',P(ISW),&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,ISW),J=1,(NY+1)/2),I=1,ND2)
!
ISW=2
WRITE (6,1410) 'BARTLETT',P(ISW),&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,ISW),J=1,(NY+1)/2),I=1,ND2)
!
ISW=3
WRITE (6,1410) 'WELCH',P(ISW),&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,ISW),J=1,(NY+1)/2),I=1,ND2)
!
ISW=4
WRITE (6,1410) 'PARZEN',P(ISW),&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,ISW),J=1,(NY+1)/2),I=1,ND2)
!
1000 FORMAT(' ',/,/,&
  ' *** DFPS2D ***',/,&
  2X,'** INPUT **',/,&
  6X,'ISW =0, 2 TO ',I3,/,&
  6X,'NX = ',I3,/,&
  6X,'NY = ',I3)
1100 FORMAT(12X,'DATA R(I,J)',/,&
  4X,'I/J      1      2      3      4',&
  '          5      6      7      8',/,&
  4X,'-----',/,&
  '-----',/,&
  8(2X,I3,8F9.4,/) )
1150 FORMAT(6X,'TIME DOMAIN POWER =',F10.4)
1160 FORMAT(6X,'SIGNAL FREQUENCY =( ',F10.4,', ',F10.4,', )')
1200 FORMAT(2X,'** OUTPUT **')
1300 FORMAT(6X,'IERR =',I5)
1410 FORMAT(6X,'(MODIFIED) PERIODOGRAM( ',A,', )',/,&
  3X,'FREQUENCY DOMAIN POWER=',F8.4,/,&
  2X,'X/Y-FRQ',8F8.2,/,&
  '-----',/,&
  '-----')
1420 FORMAT(8(2X,F7.2,8F8.4,/) )
END

```

## (d) 出力結果

```

*** DFPS2D ***
** INPUT **
ISW =0, 2 TO 5
NX = 8
NY = 8
DATA R(I,J)
I/J  1      2      3      4      5      6      7      8
-----
1  2.0000  1.9048  1.6374  1.2487  0.8126  0.4122  0.1237  0.0020
2  0.6319  0.5367  0.2693  -0.1194  -0.5555  -0.9559  -1.2444  -1.3662
3  0.2710  0.1759  -0.0915  -0.4803  -0.9164  -1.3168  -1.6053  -1.7270
4  1.9048  1.8097  1.5423  1.1535  0.7174  0.3170  0.0285  -0.0932
5  1.0628  0.9676  0.7002  0.3115  -0.1246  -0.5250  -0.8135  -0.9352
6  0.0489  -0.0462  -0.3136  -0.7024  -1.1384  -1.5388  -1.8274  -1.9491
7  1.6374  1.5422  1.2748  0.8861  0.4500  0.0496  -0.2389  -0.3606
8  1.4818  1.3866  1.1192  0.7304  0.2944  -0.1060  -0.3946  -0.5163

TIME DOMAIN POWER = 1.0626
SIGNAL FREQUENCY =( 0.6200, 0.1400)
** OUTPUT **
IERR = 0
(MODIFIED) PERIODOGRAM(RAW)
FREQUENCY DOMAIN POWER= 0.9717
X/Y-FRQ  -1.00  -0.75  -0.50  -0.25  0.00  0.25  0.50  0.75
-----
0.00  0.0158  0.0188  0.0350  0.2150  0.0218  0.2150  0.0350  0.0188
0.25  0.0000  0.0000  0.0000  0.0000  0.0239  0.0000  0.0000  0.0000
0.50  0.0000  0.0000  0.0000  0.0000  0.1352  0.0000  0.0000  0.0000
0.75  0.0000  0.0000  0.0000  0.0000  0.0781  0.0000  0.0000  0.0000

(MODIFIED) PERIODOGRAM(HANNING)
FREQUENCY DOMAIN POWER= 0.5980
X/Y-FRQ  -1.00  -0.75  -0.50  -0.25  0.00  0.25  0.50  0.75
-----

```

0.00	0.0000	0.0001	0.0054	0.0632	0.0105	0.0632	0.0054	0.0001
0.25	0.0000	0.0000	0.0013	0.0095	0.0056	0.0236	0.0013	0.0000
0.50	0.0000	0.0000	0.0000	0.0204	0.0814	0.0204	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0205	0.0820	0.0205	0.0000	0.0000

(MODIFIED) PERIODOGRAM(BARTLETT)  
FREQUENCY DOMAIN POWER= 0.5835  
X/Y-FRQ -1.00 -0.75 -0.50 -0.25 0.00 0.25 0.50 0.75

---

0.00	0.0000	0.0000	0.0009	0.0820	0.0109	0.0820	0.0009	0.0000
0.25	0.0000	0.0000	0.0002	0.0122	0.0025	0.0178	0.0002	0.0000
0.50	0.0000	0.0005	0.0000	0.0156	0.0855	0.0156	0.0000	0.0005
0.75	0.0000	0.0004	0.0000	0.0095	0.0762	0.0191	0.0000	0.0004

(MODIFIED) PERIODOGRAM(WELCH)  
FREQUENCY DOMAIN POWER= 0.7072  
X/Y-FRQ -1.00 -0.75 -0.50 -0.25 0.00 0.25 0.50 0.75

---

0.00	0.0000	0.0000	0.0001	0.1263	0.0124	0.1263	0.0001	0.0000
0.25	0.0000	0.0000	0.0000	0.0140	0.0014	0.0127	0.0000	0.0000
0.50	0.0002	0.0003	0.0010	0.0195	0.1065	0.0054	0.0009	0.0003
0.75	0.0002	0.0003	0.0008	0.0064	0.0941	0.0142	0.0009	0.0003

(MODIFIED) PERIODOGRAM(PARZEN)  
FREQUENCY DOMAIN POWER= 0.4909  
X/Y-FRQ -1.00 -0.75 -0.50 -0.25 0.00 0.25 0.50 0.75

---

0.00	0.0000	0.0002	0.0093	0.0253	0.0070	0.0253	0.0093	0.0002
0.25	0.0000	0.0001	0.0022	0.0022	0.0127	0.0279	0.0064	0.0001
0.50	0.0000	0.0000	0.0006	0.0171	0.0558	0.0323	0.0030	0.0000
0.75	0.0000	0.0000	0.0013	0.0206	0.0485	0.0214	0.0014	0.0000



## 2.16.3 DFPS3D, RFPS3D

## 3次元フーリエ・ピリオドグラム

## (1) 機能

系列  $u_{j_x, j_y, j_z}$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ ) の (修正) フーリエ・ピリオドグラムを求める。フーリエ・ピリオドグラム  $p_{k_x, k_y, k_z}$  は次式で定義される。

$$p_{k_x, k_y, k_z} = \frac{\left| \sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} \sum_{j_z=0}^{n_z-1} w_{j_x}^{(x)} w_{j_y}^{(y)} w_{j_z}^{(z)} u_{j_x, j_y, j_z} e^{-2\pi\sqrt{-1}\left(\frac{j_x k_x}{n_x} + \frac{j_y k_y}{n_y} + \frac{j_z k_z}{n_z}\right)} \right|^2}{n_x n_y n_z \beta}$$

$$(k_x = 0, 1, \dots, \lfloor \frac{n_x}{2} \rfloor; k_y = 0, 1, \dots, n_y - 1; k_z = 0, 1, \dots, n_z - 1)$$

ただし、 $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。 $w_{j_x}^{(x)}$ ,  $w_{j_y}^{(y)}$ ,  $w_{j_z}^{(z)}$  は打ち切り関数 (窓関数) であり、生のフーリエ・ピリオドグラムの場合には、 $w_{j_x}^{(x)} = w_{j_y}^{(y)} = w_{j_z}^{(z)} = 1$  ( $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$ )、 $\beta = n_x n_y n_z$  とし、修正ピリオドグラムの場合には

$$\beta = \begin{cases} \left( \sum_{j_x=0}^{n_x-1} (w_{j_x}^{(x)})^2 \right) \left( \sum_{j_y=0}^{n_y-1} (w_{j_y}^{(y)})^2 \right) \left( \sum_{j_z=0}^{n_z-1} (w_{j_z}^{(z)})^2 \right) & \text{(窓関数によるパワー補正式を用いる場合)} \\ n_x n_y n_z & \text{(それ以外)} \end{cases}$$

とする。なお、ピリオドグラム  $p_{k_x, k_y, k_z}$  は  $k_x$  についての半周期分 (周期  $(n_x, n_y, n_z)$ ) に相当し、残りの半周期分は以下の関係から得られる。

$$\begin{aligned} p_{n_x - k_x, n_y - k_y, n_z - k_z} &= p_{k_x, k_y, k_z} \\ p_{n_x - k_x, k_y, k_z} &= p_{k_x, n_y - k_y, n_z - k_z} \\ p_{n_x - k_x, n_y - k_y, k_z} &= p_{k_x, k_y, n_z - k_z} \end{aligned}$$

また、対応する系列の全パワーは、

$$\frac{\sum_{j_x=0}^{n_x-1} \sum_{j_y=0}^{n_y-1} \sum_{j_z=0}^{n_z-1} \{u_{j_x, j_y, j_z}\}^2}{n_x n_y n_z}$$

である。

## (2) 使用法

倍精度サブルーチン:

CALL DFPS3D (NX, NY, NZ, R, LX, LY, LZ, ISW, IWK, WK, IERR)

単精度サブルーチン:

CALL RFPS3D (NX, NY, NZ, R, LX, LY, LZ, ISW, IWK, WK, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NX	I	1	入 力	系列 $u_{j_x, j_y, j_z}$ の $j_x$ 方向の長さ $n_x$ (注意事項 (d) 参照)
2	NY	I	1	入 力	系列 $u_{j_x, j_y, j_z}$ の $j_y$ 方向の長さ $n_y$ (注意事項 (d) 参照)
3	NZ	I	1	入 力	系列 $u_{j_x, j_y, j_z}$ の $j_z$ 方向の長さ $n_z$ (注意事項 (d) 参照)
4	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	LX, LY, LZ	入 力	系列 $u_{j_x, j_y, j_z}$ の値 (注意事項 (a) 参照)
				出 力	系列 $u_{j_x, j_y, j_z}$ のフーリエ・ピリオドグラム $p_{k_x, k_y, k_z}$ の値 (注意事項 (b), (c) 参照)
5	LX	I	1	入 力	配列 R の整合寸法
6	LY	I	1	入 力	配列 R の第 2 寸法
7	LZ	I	1	入 力	配列 R の第 3 寸法
8	ISW	I	1	入 力	処理スイッチ (注意事項 (e) 参照) ISW=0:生のフーリエ・ピリオドグラムを計算する ISW=±1:ユーザ定義窓関数を利用して計算する ISW=±2:Hanning 窓関数を利用して計算する ISW=±3:Bartlett 窓関数を利用して計算する ISW=±4:Welch 窓関数を利用して計算する ISW=±5:Parzen 窓関数を利用して計算する なお、窓関数によるパワー補正式を用いる場合は ISW > 0, それ以外の場合は ISW < 0 とする.
9	IWK	I	60	ワーク	作業領域
10	WK	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	内容参照	ワーク	作業領域 ISW=±1 の場合には、ユーザ定義窓関数の値を入力する (注意事項 (e) 参照). 大きさ: $NX + 2 \times (NY + NZ) + LX \times LY \times LZ$
11	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $ISW \in \{0, \pm 1, \pm 2, \pm 3, \pm 4, \pm 5\}$

(b)  $NX > 1$

$NY > 1$

$NZ > 1$

(c) NX が奇数の時 :

$LX \geq NX + 1$ , LX は偶数

$LY \geq NY$

$LZ \geq NZ$

NX が偶数の時 :

$LX \geq NX + 2$ , LX は偶数

$LY \geq NY$

$LZ \geq NZ$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3030	制限条件 (c) を満足しなかった.	
4000	ISW = 1 の場合にユーザ定義窓関数が $w_{j_x}^{(x)} = 0$ ( $j_x = 0, \dots, n_x - 1$ ) であった.	
4010	ISW=1 の場合にユーザ定義窓関数が $w_{j_y}^{(y)} = 0$ ( $j_y = 0, \dots, n_y - 1$ ) であった.	
4020	ISW=1 の場合にユーザ定義窓関数が $w_{j_z}^{(z)} = 0$ ( $j_z = 0, \dots, n_z - 1$ ) であった.	

(6) 注意事項

- (a) 配列 R の各要素と系列  $u_{j_x, j_y, j_z}$  の値は以下の様に対応する。

$$u_{j_x, j_y, j_z} \leftrightarrow R(j_x + 1, j_y + 1, j_z + 1)$$

ただし,  $j_x = 0, \dots, n_x - 1$ ;  $j_y = 0, \dots, n_y - 1$ ;  $j_z = 0, \dots, n_z - 1$  であり, それ以外の要素には値を入力する必要が無い。なお, 主記憶のバンク競合を避けるために配列 R の整合寸法について LX/2, LY, LZ が奇数になるように設定するのが望ましい。また, 高速化のために配列 R 内のデータ設定領域以外の要素に対しても演算を実行する。通常, たとえば NX が (4 の倍数)+2 のときは LX=NX+4 とする。

- (b) フーリエ・ピリオドグラム  $p_{k_x, k_y, k_z}$  の値は配列 R の各要素と以下の様に対応する。

$$p(k_x, k_y, k_z) \leftrightarrow R(k_x + 1, k_y + 1, k_z + 1) \\ (k_x = 0, \dots, \lfloor \frac{n_x}{2} \rfloor; k_y = 0, \dots, n_y - 1; k_z = 0, \dots, n_z - 1)$$

なお,  $\lfloor x \rfloor$  は  $x$  を超えない最大の整数を表す。

- (c) 得られるフーリエ・ピリオドグラム  $p_{k_x, k_y, k_z}$  ( $k_x = 0, 1, \dots, \lfloor \frac{n_x}{2} \rfloor$ ;  $k_y = 0, 1, \dots, n_y - 1$ ;  $k_z = 0, \dots, n_z - 1$ ) に対応する周波数 ( $\xi_{k_x}, \eta_{k_y}, \zeta_{k_z}$ ) は

$$\xi_{k_x} = \frac{k_x}{n_x \Delta} \quad (k_x = 0, 1, \dots, \lfloor \frac{n_x}{2} \rfloor) \\ \eta_{k_y} = \begin{cases} \frac{k_y}{n_y \Delta} & (k_y = 0, 1, \dots, \lfloor \frac{n_y}{2} \rfloor) \\ \frac{k_y - n_y}{n_y \Delta} & (k_y = \lfloor \frac{n_y}{2} \rfloor + 1, \dots, n_y - 1) \end{cases} \\ \zeta_{k_z} = \begin{cases} \frac{k_z}{n_z \Delta} & (k_z = 0, 1, \dots, \lfloor \frac{n_z}{2} \rfloor) \\ \frac{k_z - n_z}{n_z \Delta} & (k_z = \lfloor \frac{n_z}{2} \rfloor + 1, \dots, n_z - 1) \end{cases}$$

( $\Delta$ : 標本化間隔) で与えられる。

- (d) 系列  $u_{j_x, j_y, j_z}$  の長さ NX, NY, NZ の値は混合基数 FFT アルゴリズムが有効に働く数 (FFT の混合基数である 2, 3, 5 等の倍数) となるように設定した方が効率良い計算を行える。たとえば, NX = 289 (=17<sup>2</sup>) とするよりも NX = 300 (=2<sup>2</sup> × 3 × 5<sup>2</sup>) や 320 (=2<sup>6</sup> × 5), 384 (=2<sup>7</sup> × 3) などとした方が効率が良い。なお, データ数を大きくできない場合には, データの最後に 0 を必要なだけ補って NX を調整して計算を行う。

- (e) 処理スイッチ ISW の値によって, 以下の様に打ち切り関数 (窓関数) を変更することができる。

$$w_j = \begin{cases} \sin^2(\pi v_j) & \text{ISW} = \pm 2 \text{ (Hanning 窓)} \\ 1 - |2v_j - 1| & \text{ISW} = \pm 3 \text{ (Bartlett 窓)} \\ 1 - (2v_j - 1)^2 & \text{ISW} = \pm 4 \text{ (Welch 窓)} \\ \left. \begin{cases} 16v_j^3 & 0 \leq v_j < \frac{1}{4} \\ 1 - 6v_j(v_j - 1)^2 & \frac{1}{4} \leq v_j < \frac{1}{2} \\ 1 - 6v_j(v_{n-j+1} - 1)^2 & \frac{1}{2} \leq v_j < \frac{3}{4} \\ 16v_{n-j+1}^3 & \frac{3}{4} \leq v_j < 1 \end{cases} \right\} \text{ISW} = \pm 5 \text{ (Parzen 窓)} \end{cases}$$

ただし,  $v_j = \frac{j}{n}$  で,  $w_{j_x}^{(x)}$  については  $j = j_x, n = n_x$  とし,  $w_{j_y}^{(y)}$  については  $j = j_y, n = n_y$ ,  $w_{j_z}^{(z)}$  については  $j = j_z, n = n_z$  とする。したがって, 上述のような窓関数を用いる場合には系列  $u_{j_x, j_y, j_z}$  の要素  $u_{0, j_y, j_z}$ ,  $u_{j_x, 0, j_z}$ ,  $u_{j_x, j_y, 0}$  は修正ピリオドグラムの計算に影響しない。これを避けたい場合には, 実際に計算したい系列の長さよりも 1 大きい数を NX, NY, NZ に指定し,  $j_x, j_y, j_z$  について 1 以降の対応する要素に有効なデータを設定すれば良い。なお, 窓関数は  $|x| \leq 1$  でのみ非ゼロとなる時間 (または空間) 領域関数とし

それぞれ次の様に表される.

$$w(x) = \begin{cases} \frac{1 + \cos \pi x}{2} = \cos^2 \frac{\pi x}{2} & \text{Hanning 窓} \\ 1 - |x| & \text{Bartlett 窓} \\ 1 - x^2 & \text{Welch 窓} \\ \left\{ \begin{array}{ll} 1 - 6x^2 + 6|x|^3 & |x| \leq \frac{1}{2} \\ 2(1 - |x|)^3 & \frac{1}{2} \leq |x| \leq 1 \end{array} \right\} & \text{Parzen 窓} \end{cases}$$

また, ユーザ独自の窓関数値  $w_{j_x}^{(x)}, w_{j_y}^{(y)}, w_{j_z}^{(z)}$  を利用したい場合には  $ISW = \pm 1$  として作業配列 WK に

$$\begin{aligned} WK(j_x + 1) &= w_{j_x}^{(x)} \quad (j_x = 0, \dots, n_x - 1), \\ WK(n_x + j_y + 1) &= w_{j_y}^{(y)} \quad (j_y = 0, \dots, n_y - 1), \\ WK(n_x + n_y + j_z + 1) &= w_{j_z}^{(z)} \quad (j_z = 0, \dots, n_z - 1) \end{aligned}$$

と設定してこのサブルーチンを呼び出す.

- (f) 生のピリオドグラムはその定義から自己相関関数の離散フーリエ変換近似とみなせる. 有効データ数  $n$  の離散関数の自己相関関数の有効データ長は  $2n - 1$  であるので, 一般の関数のパワー・スペクトルを生ピリオドグラムで近似することは, 1つの周期が以下のように与えられる方形打ち切り関数  $w(k)$  で関数を打ち切ったことに相当する.

$$w(k) = \begin{cases} 1 & k = 0, 1, \dots, n - 1 \\ 0 & \text{それ以外} \end{cases}$$

方形関数のフーリエ変換は周波数を  $f$  とした場合,  $\frac{\sin f}{f}$  型の関数形をしており, 中心周波数の周りに小さくないサイドローブを持っている. したがって, たとえば, 周期関数を 1 周期の整数倍でない幅で単純に打ち切って標本化した場合, 周波数領域では, 生のピリオドグラムはパワー・スペクトルを求めたい周期関数のフーリエ変換と  $\frac{\sin f}{f}$  型関数との畳み込みとなるので, 漏れ (leakage) と呼ばれる余分な周波数成分が発生する. このような漏れを抑止するためには, 単純な打ち切りを行わずに Hanning 窓関数のような周波数領域でのサイドローブが小さい打ち切り関数を用いる. ただし, 一般に漏れを抑圧すればする程離散フーリエ変換の結果は拡がりぼやけたものとなる. したがって, パワー・スペクトルを推定する場合, 目的に応じて, すなわち, スペクトル幅を問題としているの中心周波数を問題としているのか等に応じて, 適切な打ち切り関数を選択する必要がある.

- (g) 離散フーリエ変換の分解度 (周波数領域での標本間隔)  $\frac{1}{nT}$  を上げるには標本数  $n$  を増やすか標本間隔  $T$  を増やせば良いが, 標本間隔と分解度を一定に保った状態でパワー・スペクトルの推定値の精度を上げるために, 標本数  $n$  の標本を  $m$  組とって  $m$  組それぞれについて修正ピリオドグラムを求めてその平均をとるという手法がよく取られる. この場合, 系列から  $m$  組の標本データをオーバーラップして取るというような手法も提案されている. 詳細は参考文献等を参照されたい.
- (h) パワー・スペクトルを求める場合, フーリエ変換の周波数推移に関する性質すなわち時間 (または空間) 領域で  $e^{2\pi\sqrt{-1}f_0 t}$  を掛けることは周波数領域では周波数を  $f_0$  だけシフトすることに対応し, 関数の形状は変わらないという性質を利用して, パワー・スペクトルの中心周波数をあらかじめシフトして計算することで計算に必要なデータ点数を削減するという手法も良く用いられる. なお, このような操作は変調 (modulation) として知られている. ただし,

$$LX = NX + 1 \quad (NX \text{ が奇数のとき}) \text{ または}$$

$$LX = NX + 2 \quad (NX \text{ が偶数のとき})$$

$$LY = NY, LZ = NZ$$

である.

- (i) この機能は逐次版および OpenMP 不利用の MPI 版ライブラリにおいてスレッドセーフではない.

(7) 使用例

(a) 問題

次式で定義される波形を標本化間隔  $\Delta$  で離散化し、フーリエ・ピリオドグラムを計算し、パワー・スペクトルを推定する。

$$f(x, y, z) = \cos 2\pi f_1 x + \cos 2\pi f_2 y + \cos 2\pi f_3 z$$

(b) 入力データ

標本化データ

$R(j_x+1, j_y+1, j_z+1) = f(j_x\Delta, j_y\Delta, j_z\Delta)$  ( $j_x = 0, 1, \dots, NX-1$ ;  $j_y = 0, 1, \dots, NY-1$ ;  $j_z = 0, 1, \dots, NZ-1$ )  
ただし,  $\Delta = 0.5$

$NX, NY, NZ, ISW$

(c) 主プログラム

```

PROGRAM BFPS3D
! *** EXAMPLE OF DFPS3D ***
! IMPLICIT REAL(8) (A-H,O-Z)
! IMPLICIT NONE
INTEGER NX, NY, NZ, LX, LY, LZ, ISW, IERR, IWK(60)
INTEGER I, J, K, M, ND2, IS
INTEGER NO, ISWO
PARAMETER (NO =8)
PARAMETER (ISWO=4)
PARAMETER (LX = (NO+2)/2*2, LY=NO, LZ=NO)
REAL(8) R(LX, LY, LZ, -1: ISWO), WK(5*NO+LX*LY*LZ)
REAL(8) P(-1: ISWO)
REAL(8) TX, TY, TZ, DT, DFX, DFY, DFZ
REAL(8) PAI, F0, F1, F2, F3
PARAMETER(PAI=3.141592D0)
!
NX=NO
NY=NO
NZ=NO
WRITE (6, 1000) ISWO+1, NX, NY, NZ
DT=0.5D0
F0=1.0D0/(2.0*DT)
F1=0.62D0*F0
F2=0.14D0*F0
F3=0.55D0*F0
ND2=(NX+1)/2
DFX=1.0D0/(DT*NX)
DFY=1.0D0/(DT*NY)
DFZ=1.0D0/(DT*NZ)
P(-1)=0.0D0
DO 100 K=1, NZ
  TZ=DBLE(K-1)*DT
DO 101 J=1, NY
  TY=DBLE(J-1)*DT
DO 102 I=1, NX
  TX=DBLE(I-1)*DT
  R(I, J, K, -1)=COS(2*PAI*F1*TX)+COS(2*PAI*F2*TY)+COS(2*PAI*F3*TZ)
  P(-1)=P(-1)+R(I, J, K, -1)*R(I, J, K, -1)
102 CONTINUE
101 CONTINUE
100 CONTINUE
P(-1)=P(-1)/(DBLE(NX)*DBLE(NY)*DBLE(NZ))
DO 110 K=1, NZ
  WRITE (6, 1100) K, (I, (R(I, J, K, -1), J=1, NY), I=1, NX)
110 CONTINUE
WRITE (6, 1150) P(-1)
WRITE (6, 1160) F1, F2, F3
IS=0
DO 400 ISW=0, ISWO
DO 200 K=1, NZ
DO 201 J=1, NY
DO 202 I=1, NX
  R(I, J, K, ISW)=R(I, J, K, -1)
202 CONTINUE
201 CONTINUE
200 CONTINUE
IF (ISW.NE.0) IS=ISW+1
CALL DFPS3D(NX, NY, NZ, R(1, 1, 1, ISW), LX, LY, LZ, IS, IWK, WK, IERR)
P(ISW)=0.0D0
IF (MOD(NX, 2).EQ.0) THEN
  M=ND2-1
  DO 500 K=1, NZ
  DO 501 J=1, NY
  DO 502 I=2, M
    P(ISW)=P(ISW)+2.0D0*R(I, J, K, ISW)
502 CONTINUE
501 CONTINUE
500 CONTINUE
DO 300 K=1, NZ
DO 301 J=1, NY
  P(ISW)=P(ISW)+R(1, J, K, ISW)+R(ND2, J, K, ISW)
301 CONTINUE
300 CONTINUE

```

```

ELSE
  M=ND2
  DO 600 K=1,NZ
  DO 601 J=1,NY
  DO 602 I=2,M
    P(ISW)=P(ISW)+2.0DO*R(I,J,K,ISW)
602 CONTINUE
601 CONTINUE
600 CONTINUE
  DO 700 K=1,NZ
  DO 701 J=1,NY
    P(ISW)=P(ISW)+R(1,J,K,ISW)
701 CONTINUE
700 CONTINUE
ENDIF
400 CONTINUE
WRITE (6,1200)
WRITE (6,1300) IERR
!
ISW=0
WRITE (6,1410) 'RAW',P(ISW)
DO 800 K=(NZ+1)/2+1,NZ
WRITE (6,1430) (K-(NZ+1))*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
800 CONTINUE
DO 805 K=1,(NZ+1)/2
WRITE (6,1430) (K-1)*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
805 CONTINUE
!
ISW=1
WRITE (6,1410) 'HANNING',P(ISW)
DO 810 K=(NZ+1)/2+1,NZ
WRITE (6,1430) (K-(NZ+1))*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
810 CONTINUE
DO 815 K=1,(NZ+1)/2
WRITE (6,1430) (K-1)*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
815 CONTINUE
!
ISW=2
WRITE (6,1410) 'BARTLETT',P(ISW)
DO 820 K=(NZ+1)/2+1,NZ
WRITE (6,1430) (K-(NZ+1))*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
820 CONTINUE
DO 825 K=1,(NZ+1)/2
WRITE (6,1430) (K-1)*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
825 CONTINUE
!
ISW=3
WRITE (6,1410) 'WELCH',P(ISW)
DO 830 K=(NZ+1)/2+1,NZ
WRITE (6,1430) (K-(NZ+1))*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
830 CONTINUE
DO 835 K=1,(NZ+1)/2
WRITE (6,1430) (K-1)*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
835 CONTINUE
!
ISW=4
WRITE (6,1410) 'PARZEN',P(ISW)
DO 840 K=(NZ+1)/2+1,NZ
WRITE (6,1430) (K-(NZ+1))*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)

```

```

WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
840 CONTINUE
DO 845 K=1,(NZ+1)/2
WRITE (6,1430) (K-1)*DFZ,&
  ((J-(NY+1))*DFY,J=(NY+1)/2+1,NY),&
  ((J-1)*DFY,J=1,(NY+1)/2)
WRITE (6,1420)&
  ((I-1)*DFX,(R(I,J,K,ISW),J=(NY+1)/2+1,NY),&
  (R(I,J,K,ISW),J=1,(NY+1)/2),I=1,ND2)
845 CONTINUE
!
1000 FORMAT(' ',/,/,&
  ' *** DFPS3D ***',/,&
  2X,'** INPUT **',/,&
  6X,'ISW =0, 2 TO ',I3,/,&
  6X,'NX =',I3,/,&
  6X,'NY =',I3,/,&
  6X,'NZ =',I3)
1100 FORMAT(12X,'DATA R(I,J, ',I3,')',/,&
  4X,'I/J 1 2 3 4',&
  4X,' 5 6 7 8',/,&
  '-----',/,&
  8(2X,I3,8F9.4,/) )
1150 FORMAT(6X,'TIME DOMAIN POWER =',F10.4)
1160 FORMAT(6X,'SIGNAL FREQUENCY =( ',F10.4,', ',F10.4,', ',F10.4,')')
1200 FORMAT(2X,'** OUTPUT **')
1300 FORMAT(6X,'IERR =',I5)
1410 FORMAT(6X,'(MODIFIED) PERIODOGRAM(' ,A,')',/,&
  3X,'FREQUENCY DOMAIN POWER=',F8.4)
1430 FORMAT(17X,' Z-FRQ=',F8.2,/,&
  2X,'X/Y-FRQ',8F8.2,/,&
  2X,'-----',/,&
  '-----')
1420 FORMAT(8(2X,F7.2,8F8.4,/) )
END

```

## (d) 出力結果

```

*** DFPS3D ***
** INPUT **
ISW =0, 2 TO 5
NX = 8
NY = 8
NZ = 8

```

DATA R(I,J, 1)								
I/J	1	2	3	4	5	6	7	8
1	3.0000	2.9048	2.6374	2.2487	1.8126	1.4122	1.1237	1.0020
2	1.6319	1.5367	1.2693	0.8806	0.4445	0.0441	-0.2444	-0.3662
3	1.2710	1.1759	0.9085	0.5197	0.0836	-0.3168	-0.6053	-0.7270
4	2.9048	2.8097	2.5423	2.1535	1.7174	1.3170	1.0285	0.9068
5	2.0628	1.9676	1.7002	1.3115	0.8754	0.4750	0.1865	0.0648
6	1.0489	0.9538	0.6864	0.2976	-0.1384	-0.5388	-0.8274	-0.9491
7	2.6374	2.5422	2.2748	1.8861	1.4500	1.0496	0.7611	0.6394
8	2.4818	2.3866	2.1192	1.7304	1.2944	0.8940	0.6054	0.4837

DATA R(I,J, 2)								
I/J	1	2	3	4	5	6	7	8
1	1.8436	1.7484	1.4810	1.0923	0.6562	0.2558	-0.0327	-0.1545
2	0.4754	0.3803	0.1129	-0.2759	-0.7119	-1.1123	-1.4009	-1.5226
3	0.1146	0.0194	-0.2480	-0.6367	-1.0728	-1.4732	-1.7617	-1.8834
4	1.7484	1.6532	1.3858	0.9971	0.5610	0.1606	-0.1279	-0.2496
5	0.9064	0.8112	0.5438	0.1550	-0.2810	-0.6814	-0.9699	-1.0917
6	-0.1075	-0.2027	-0.4701	-0.8588	-1.2949	-1.6953	-1.9838	-2.1055
7	1.4810	1.3858	1.1184	0.7297	0.2936	-0.1068	-0.3953	-0.5170
8	1.3253	1.2301	0.9627	0.5740	0.1379	-0.2625	-0.5510	-0.6727

DATA R(I,J, 3)								
I/J	1	2	3	4	5	6	7	8
1	1.0489	0.9538	0.6864	0.2976	-0.1384	-0.5388	-0.8274	-0.9491
2	-0.3192	-0.4144	-0.6818	-1.0705	-1.5066	-1.9070	-2.1955	-2.3172
3	-0.6800	-0.7752	-1.0426	-1.4313	-1.8674	-2.2678	-2.5563	-2.6781
4	0.9538	0.8586	0.5912	0.2025	-0.2336	-0.6340	-0.9225	-1.0443
5	0.1117	0.0166	-0.2508	-0.6396	-1.0756	-1.4760	-1.7646	-1.8863
6	-0.9021	-0.9973	-1.2647	-1.6534	-2.0895	-2.4899	-2.7784	-2.9001
7	0.6864	0.5912	0.3238	-0.0649	-0.5010	-0.9014	-1.1899	-1.3117
8	0.5307	0.4355	0.1681	-0.2206	-0.6567	-1.0571	-1.3456	-1.4673

DATA R(I,J, 4)								
I/J	1	2	3	4	5	6	7	8
1	2.4540	2.3588	2.0914	1.7027	1.2666	0.8662	0.5777	0.4560
2	1.0859	0.9907	0.7233	0.3346	-0.1015	-0.5019	-0.7904	-0.9122
3	0.7250	0.6298	0.3624	-0.0263	-0.4624	-0.8628	-1.1513	-1.2730
4	2.3588	2.2636	1.9962	1.6075	1.1714	0.7710	0.4825	0.3608
5	1.5168	1.4216	1.1542	0.7655	0.3294	-0.0710	-0.3595	-0.4812
6	0.5029	0.4078	0.1404	-0.2484	-0.6844	-1.0849	-1.3734	-1.4951
7	2.0914	1.9962	1.7288	1.3401	0.9040	0.5036	0.2151	0.0934
8	1.9357	1.8406	1.5732	1.1844	0.7484	0.3480	0.0594	-0.0623

DATA R(I,J, 5)								
I/J	1	2	3	4	5	6	7	8
1	2.8090	2.7138	2.4464	2.0577	1.6216	1.2212	0.9327	0.8110
2	1.4409	1.3457	1.0783	0.6896	0.2535	-0.1469	-0.4354	-0.5571



## 3次元フーリエ・ピリオドグラム

3	1.0800	0.9849	0.7175	0.3287	-0.1073	-0.5077	-0.7963	-0.9180
4	2.7138	2.6187	2.3513	1.9625	1.5265	1.1261	0.8375	0.7158
5	1.8718	1.7766	1.5092	1.1205	0.6844	0.2840	-0.0045	-0.1262
6	0.8580	0.7628	0.4954	0.1067	-0.3294	-0.7298	-1.0183	-1.1401
7	2.4464	2.3513	2.0839	1.6951	1.2591	0.8587	0.5701	0.4484
8	2.2908	2.1956	1.9282	1.5395	1.1034	0.7030	0.4145	0.2927

DATA R(I,J, 6)								
I/J	1	2	3	4	5	6	7	8
1	1.2929	1.1977	0.9303	0.5416	0.1055	-0.2949	-0.5834	-0.7051
2	-0.0752	-0.1704	-0.4378	-0.8265	-1.2626	-1.6630	-1.9515	-2.0733
3	-0.4361	-0.5312	-0.7987	-1.1874	-1.6235	-2.0239	-2.3124	-2.4341
4	1.1977	1.1025	0.8351	0.4464	0.0103	-0.3901	-0.6786	-0.8003
5	0.3557	0.2605	-0.0069	-0.3956	-0.8317	-1.2321	-1.5206	-1.6423
6	-0.6582	-0.7533	-1.0207	-1.4095	-1.8455	-2.2459	-2.5345	-2.6562
7	0.9303	0.8351	0.5677	0.1790	-0.2571	-0.6575	-0.9460	-1.0677
8	0.7747	0.6795	0.4121	0.0233	-0.4127	-0.8131	-1.1017	-1.2234

DATA R(I,J, 7)								
I/J	1	2	3	4	5	6	7	8
1	1.4122	1.3170	1.0496	0.6609	0.2248	-0.1756	-0.4641	-0.5858
2	0.0441	-0.0511	-0.3185	-0.7072	-1.1433	-1.5437	-1.8322	-1.9539
3	-0.3168	-0.4119	-0.6793	-1.0681	-1.5041	-1.9045	-2.1931	-2.3148
4	1.3170	1.2219	0.9545	0.5657	0.1297	-0.2707	-0.5593	-0.6810
5	0.4750	0.3798	0.1124	-0.2763	-0.7124	-1.1128	-1.4013	-1.5230
6	-0.5388	-0.6340	-0.9014	-1.2902	-1.7262	-2.1266	-2.4152	-2.5369
7	1.0496	0.9545	0.6871	0.2983	-0.1377	-0.5381	-0.8267	-0.9484
8	0.8940	0.7988	0.5314	0.1427	-0.2934	-0.6938	-0.9823	-1.1041

DATA R(I,J, 8)								
I/J	1	2	3	4	5	6	7	8
1	2.8910	2.7958	2.5284	2.1397	1.7036	1.3032	1.0147	0.8930
2	1.5229	1.4277	1.1603	0.7716	0.3355	-0.0649	-0.3534	-0.4751
3	1.1620	1.0669	0.7995	0.4107	-0.0253	-0.4257	-0.7143	-0.8360
4	2.7958	2.7007	2.4333	2.0445	1.6085	1.2080	0.9195	0.7978
5	1.9538	1.8586	1.5912	1.2025	0.7664	0.3660	0.0775	-0.0442
6	0.9399	0.8448	0.5774	0.1886	-0.2474	-0.6478	-0.9364	-1.0581
7	2.5284	2.4333	2.1659	1.7771	1.3410	0.9406	0.6521	0.5304
8	2.3728	2.2776	2.0102	1.6215	1.1854	0.7850	0.4965	0.3747

TIME DOMAIN POWER = 1.6439  
SIGNAL FREQUENCY =( 0.6200, 0.1400, 0.5500)  
\*\* OUTPUT \*\*  
IERR = 0  
(MODIFIED) PERIODOGRAM(RAW)  
FREQUENCY DOMAIN POWER= 1.5531  
Z-FRQ= -1.00

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0000	0.0000	0.0000	0.0000	0.0007	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Z-FRQ= -0.75

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0000	0.0000	0.0000	0.0000	0.0071	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Z-FRQ= -0.50

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0000	0.0000	0.0000	0.0000	0.2513	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Z-FRQ= -0.25

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0000	0.0000	0.0000	0.0000	0.0136	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Z-FRQ= 0.00

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0158	0.0188	0.0350	0.2150	0.0583	0.2150	0.0350	0.0188
0.25	0.0000	0.0000	0.0000	0.0000	0.0239	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.1352	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0781	0.0000	0.0000	0.0000

Z-FRQ= 0.25

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0000	0.0000	0.0000	0.0000	0.0136	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Z-FRQ= 0.50

X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
0.00	0.0000	0.0000	0.0000	0.0000	0.2513	0.0000	0.0000	0.0000

## 3次元フーリエ・ピリオドグラム

0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
				Z-FRQ=	0.75			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0000	0.0071	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
(MODIFIED) PERIODOGRAM(HANNING)								
FREQUENCY DOMAIN POWER= 1.0699								
				Z-FRQ=	-1.00			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0001	0.0004	0.0001	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
				Z-FRQ=	-0.75			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0078	0.0310	0.0078	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0019	0.0078	0.0019	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
				Z-FRQ=	-0.50			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0176	0.0704	0.0176	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0044	0.0176	0.0044	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
				Z-FRQ=	-0.25			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0009	0.0164	0.0045	0.0053	0.0009	0.0000
0.25	0.0000	0.0000	0.0002	0.0027	0.0004	0.0023	0.0002	0.0000
0.50	0.0000	0.0000	0.0000	0.0034	0.0136	0.0034	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0034	0.0137	0.0034	0.0000	0.0000
				Z-FRQ=	0.00			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0001	0.0036	0.0427	0.0087	0.0427	0.0036	0.0001
0.25	0.0000	0.0000	0.0009	0.0064	0.0041	0.0158	0.0009	0.0000
0.50	0.0000	0.0000	0.0000	0.0136	0.0543	0.0136	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0137	0.0547	0.0137	0.0000	0.0000
				Z-FRQ=	0.25			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0009	0.0053	0.0045	0.0164	0.0009	0.0000
0.25	0.0000	0.0000	0.0002	0.0006	0.0031	0.0058	0.0002	0.0000
0.50	0.0000	0.0000	0.0000	0.0034	0.0136	0.0034	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0034	0.0137	0.0034	0.0000	0.0000
				Z-FRQ=	0.50			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0176	0.0704	0.0176	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0044	0.0176	0.0044	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
				Z-FRQ=	0.75			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0078	0.0310	0.0078	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0019	0.0078	0.0019	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
(MODIFIED) PERIODOGRAM(BARTLETT)								
FREQUENCY DOMAIN POWER= 1.0593								
				Z-FRQ=	-1.00			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
0.25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
				Z-FRQ=	-0.75			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0002	0.0000	0.0079	0.0346	0.0050	0.0000	0.0002
0.25	0.0000	0.0000	0.0000	0.0014	0.0062	0.0009	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0001	0.0003	0.0001	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0003	0.0001	0.0000	0.0000
				Z-FRQ=	-0.50			
X/Y-FRQ	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75
-----								
0.00	0.0000	0.0005	0.0000	0.0165	0.0907	0.0165	0.0000	0.0005
0.25	0.0000	0.0001	0.0000	0.0030	0.0165	0.0030	0.0000	0.0001

## 3次元フーリエ・ピリオドグラム

0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0001	0.0005	0.0001	0.0000	0.0000	0.0000
X/Y-FRQ	-1.00	Z-FRQ= -0.75	-0.25 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0001	0.0141	0.0037	0.0074	0.0001	0.0000	
0.25	0.0000	0.0000	0.0000	0.0022	0.0005	0.0017	0.0000	0.0000	
0.50	0.0000	0.0001	0.0000	0.0021	0.0113	0.0021	0.0000	0.0001	
0.75	0.0000	0.0001	0.0000	0.0012	0.0096	0.0024	0.0000	0.0001	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.00 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0007	0.0594	0.0070	0.0594	0.0007	0.0000	
0.25	0.0000	0.0000	0.0001	0.0089	0.0017	0.0129	0.0001	0.0000	
0.50	0.0000	0.0003	0.0000	0.0113	0.0622	0.0113	0.0000	0.0003	
0.75	0.0000	0.0003	0.0000	0.0069	0.0554	0.0139	0.0000	0.0003	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.25 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0001	0.0074	0.0037	0.0141	0.0001	0.0000	
0.25	0.0000	0.0000	0.0000	0.0011	0.0011	0.0030	0.0000	0.0000	
0.50	0.0000	0.0001	0.0000	0.0021	0.0113	0.0021	0.0000	0.0001	
0.75	0.0000	0.0001	0.0000	0.0014	0.0108	0.0027	0.0000	0.0001	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.50 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0005	0.0000	0.0165	0.0907	0.0165	0.0000	0.0005	
0.25	0.0000	0.0001	0.0000	0.0030	0.0165	0.0030	0.0000	0.0001	
0.50	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
0.75	0.0000	0.0000	0.0000	0.0001	0.0005	0.0001	0.0000	0.0000	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.75 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0002	0.0000	0.0050	0.0346	0.0079	0.0000	0.0002	
0.25	0.0000	0.0000	0.0000	0.0009	0.0065	0.0015	0.0000	0.0000	
0.50	0.0000	0.0000	0.0000	0.0001	0.0003	0.0001	0.0000	0.0000	
0.75	0.0000	0.0000	0.0000	0.0001	0.0008	0.0002	0.0000	0.0000	
(MODIFIED) PERIODOGRAM(WELCH)									
FREQUENCY DOMAIN POWER= 1.2154									
X/Y-FRQ	-1.00	Z-FRQ= -0.75	-1.00 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0000	0.0005	0.0030	0.0005	0.0000	0.0000	
0.25	0.0000	0.0000	0.0000	0.0001	0.0003	0.0001	0.0000	0.0000	
0.50	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000	
0.75	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	-0.75 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0001	0.0001	0.0004	0.0051	0.0357	0.0028	0.0003	0.0001	
0.25	0.0000	0.0000	0.0000	0.0005	0.0038	0.0003	0.0000	0.0000	
0.50	0.0000	0.0000	0.0000	0.0001	0.0009	0.0001	0.0000	0.0000	
0.75	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	-0.50 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0003	0.0004	0.0011	0.0103	0.1247	0.0186	0.0011	0.0004	
0.25	0.0000	0.0000	0.0001	0.0011	0.0131	0.0020	0.0001	0.0000	
0.50	0.0000	0.0000	0.0000	0.0001	0.0009	0.0001	0.0000	0.0000	
0.75	0.0000	0.0000	0.0000	0.0001	0.0017	0.0002	0.0000	0.0000	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	-0.25 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0001	0.0122	0.0012	0.0090	0.0000	0.0000	
0.25	0.0000	0.0000	0.0000	0.0013	0.0002	0.0009	0.0000	0.0000	
0.50	0.0000	0.0000	0.0001	0.0017	0.0095	0.0005	0.0001	0.0000	
0.75	0.0000	0.0000	0.0001	0.0005	0.0078	0.0012	0.0001	0.0000	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.00 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0000	0.1034	0.0186	0.1034	0.0000	0.0000	
0.25	0.0000	0.0000	0.0000	0.0115	0.0021	0.0104	0.0000	0.0000	
0.50	0.0002	0.0003	0.0008	0.0158	0.0862	0.0044	0.0007	0.0003	
0.75	0.0002	0.0002	0.0007	0.0052	0.0760	0.0115	0.0007	0.0002	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.25 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0000	0.0090	0.0012	0.0122	0.0001	0.0000	
0.25	0.0000	0.0000	0.0000	0.0010	0.0001	0.0012	0.0000	0.0000	
0.50	0.0000	0.0000	0.0001	0.0016	0.0086	0.0004	0.0001	0.0000	
0.75	0.0000	0.0000	0.0001	0.0006	0.0083	0.0012	0.0001	0.0000	
X/Y-FRQ	-1.00	Z-FRQ= -0.75	0.50 -0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0003	0.0004	0.0011	0.0186	0.1247	0.0103	0.0011	0.0004	
0.25	0.0000	0.0000	0.0001	0.0020	0.0133	0.0011	0.0001	0.0000	
0.50	0.0000	0.0000	0.0000	0.0004	0.0031	0.0003	0.0000	0.0000	
0.75	0.0000	0.0000	0.0000	0.0000	0.0004	0.0001	0.0000	0.0000	

X/Y-FRQ	Z-FRQ= 0.75		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0001	0.0001	0.0003	0.0028	0.0357	0.0051	0.0004	0.0001
0.25	0.0000	0.0000	0.0000	0.0003	0.0037	0.0005	0.0000	0.0000
0.50	0.0000	0.0000	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0005	0.0001	0.0000	0.0000

(MODIFIED) PERIODOGRAM(PARZEN)  
FREQUENCY DOMAIN POWER= 0.9132

X/Y-FRQ	Z-FRQ= -1.00		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0001	0.0023	0.0053	0.0023	0.0001	0.0000
0.25	0.0000	0.0000	0.0001	0.0010	0.0023	0.0010	0.0001	0.0000
0.50	0.0000	0.0000	0.0000	0.0001	0.0001	0.0001	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

X/Y-FRQ	Z-FRQ= -0.75		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0006	0.0092	0.0209	0.0089	0.0006	0.0000
0.25	0.0000	0.0000	0.0003	0.0039	0.0090	0.0038	0.0002	0.0000
0.50	0.0000	0.0000	0.0000	0.0002	0.0005	0.0002	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

X/Y-FRQ	Z-FRQ= -0.50		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0014	0.0162	0.0313	0.0112	0.0005	0.0000
0.25	0.0000	0.0000	0.0005	0.0062	0.0120	0.0044	0.0002	0.0000
0.50	0.0000	0.0000	0.0000	0.0003	0.0007	0.0004	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0003	0.0007	0.0003	0.0000	0.0000

X/Y-FRQ	Z-FRQ= -0.25		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0001	0.0030	0.0125	0.0054	0.0016	0.0012	0.0000
0.25	0.0000	0.0000	0.0008	0.0019	0.0010	0.0030	0.0010	0.0000
0.50	0.0000	0.0000	0.0001	0.0032	0.0106	0.0063	0.0006	0.0000
0.75	0.0000	0.0000	0.0003	0.0046	0.0108	0.0048	0.0003	0.0000

X/Y-FRQ	Z-FRQ= 0.00		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0001	0.0047	0.0117	0.0007	0.0117	0.0047	0.0001
0.25	0.0000	0.0000	0.0011	0.0006	0.0055	0.0140	0.0033	0.0001
0.50	0.0000	0.0000	0.0003	0.0089	0.0291	0.0168	0.0016	0.0000
0.75	0.0000	0.0000	0.0007	0.0107	0.0253	0.0111	0.0007	0.0000

X/Y-FRQ	Z-FRQ= 0.25		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0012	0.0016	0.0054	0.0125	0.0030	0.0001
0.25	0.0000	0.0000	0.0002	0.0005	0.0086	0.0110	0.0019	0.0000
0.50	0.0000	0.0000	0.0002	0.0048	0.0151	0.0085	0.0008	0.0000
0.75	0.0000	0.0000	0.0003	0.0047	0.0110	0.0049	0.0003	0.0000

X/Y-FRQ	Z-FRQ= 0.50		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0005	0.0112	0.0313	0.0162	0.0014	0.0000
0.25	0.0000	0.0000	0.0003	0.0055	0.0155	0.0081	0.0007	0.0000
0.50	0.0000	0.0000	0.0001	0.0010	0.0028	0.0014	0.0001	0.0000
0.75	0.0000	0.0000	0.0000	0.0003	0.0008	0.0003	0.0000	0.0000

X/Y-FRQ	Z-FRQ= 0.75		-0.25	0.00	0.25	0.50	0.75	
-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	
0.00	0.0000	0.0000	0.0006	0.0089	0.0209	0.0092	0.0006	0.0000
0.25	0.0000	0.0000	0.0002	0.0039	0.0091	0.0040	0.0003	0.0000
0.50	0.0000	0.0000	0.0000	0.0003	0.0006	0.0003	0.0000	0.0000
0.75	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

---

## 2.17 ラプラス変換

### 2.17.1 DFLARA, RFLARA

#### ラプラス逆変換 (有理関数)

(1) 機能

有理関数  $F(s) = \frac{Q(s)}{P(s)} = \frac{q_1 s^{nq} + q_2 s^{nq-1} + \cdots + q_{nq} s + q_{nq+1}}{p_1 s^{np} + p_2 s^{np-1} + \cdots + p_{np} s + p_{np+1}}$  ( $np \leq nq; p_1, \dots, p_{np+q}, q_1, \dots, q_{nq+1}$ :実数)  
のラプラス逆変換

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} F(s)e^{st} ds \quad (0 < t < \infty)$$

を求める.

(2) 使用法

倍精度サブルーチン:

CALL DFLARA (P, NP, Q, NQ, T, N, A, IP, K1, K2, R, F, ER, ISW, W1, IERR)

単精度サブルーチン:

CALL RFLARA (P, NP, Q, NQ, T, N, A, IP, K1, K2, R, F, ER, ISW, W1, IERR)

(3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	P	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	NP+1	入 力	分母多項式 $P(s)$ の係数
2	NP	I	1	入 力	分母多項式 $P(s)$ の次数
3	Q	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	NQ+1	入 力	分子多項式 $Q(s)$ の係数
4	NQ	I	1	入 力	分母多項式 $Q(s)$ の次数
5	T	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	入 力	原関数 $f(t)$ の計算点
6	N	I	1	入 力	配列 T の寸法
7	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	近似誤差をきめる $a$ の値 (2.1.2(1)(d) 参照)
8	IP	I	1	入 力	オイラー変換の次数 $p$ (2.1.2(1)(d) 参照)
9	K1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	打ち切り項数決定のためのパラメータ $k_1$ (2.1.2 (1)(d) 参照)
10	K2	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	打ち切り項数決定のためのパラメータ $k_2$ (2.1.2 (1)(d) 参照)
11	R	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力 出 力	ISW=1:R=0.0 を出力する ISW=2:収束座標を入力する ISW=3:収束座標の計算結果を出力する (注意事項 (b) 参照)
12	F	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	出 力	各 T (I) における原関数 $f(t)$ の値
13	ER	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	N	出 力	各 F (I) (I=1, ..., N) 計算時の打ち切り誤差
14	ISW	I	1	入 力	ISW=1: $\Re(s) > 0$ で正則な場合 ISW=2: R に収束座標の値を入力する ( $\Re(s) > 0$ で非正則で収束座標の値が既知の場合) ISW=3: R に収束座標の値を出力する (収束座標の値が未知の場合)
15	W1	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2 \times (IP + NP + 1)$	ワー ーク	作業領域
16	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $N > 0$
- (b)  $T(i) > 0.0 \quad (i = 1, \dots, N)$
- (c)  $P(1) > 0.0$
- (d)  $0 < NQ \leq NP$
- (e)  $A > 0.0$
- (f)  $IP > 0$
- (g)  $K1 > 0.0$   
 $K2 \geq 0.0$
- (h)  $R \geq 0.0$
- (i)  $ISW \in \{1, 2, 3\}$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$T(i) = 0.0$ だった. ( $i = 1, \dots, N$ )	注意事項 (a), (b) 参照.
1100	$T(i) < 0.0$ だった. ( $i = 1, \dots, N$ )	$T(i)$ の処理を打ち切り, $T(i+1)$ 以降の処理を行う.
2000	$R < 0.0$ だった ( $ISW = 2$ の時).	$R = 0.0$ として処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (c) を満足しなかった.	
3020	制限条件 (d) を満足しなかった.	
3030	制限条件 (e) を満足しなかった.	
3040	制限条件 (f) を満足しなかった.	
3050	制限条件 (g) を満足しなかった.	
3060	制限条件 (i) を満足しなかった.	

(6) 注意事項

- (a) 利用者は,  $A, IP, K1, K2$  の値を変えることにより誤差の制御が簡単にできる.
- (b)  $ISW=1$  の場合,  $R=0.0$  として処理を行う.  
 $ISW=2$  の場合,  $F(s)$  が  $\Re(s) > \alpha$  で正則であるとき,  $\gamma \geq \alpha$  となる  $\gamma$  を  $R$  に入力する.  $R \leq 0.0$  の時は  $R=0.0$  として処理を行う.  
 $ISW=3$  の場合, 収束座標  $\gamma_0$  を求め,  $R=\gamma_0$  として処理を行う.
- (c)  $NP=NQ$  のとき

$$F(s) = \frac{Q(s)}{P(s)} = \frac{q_1}{p_1} + G(s)$$

$$\text{ただし, } G(s) = \frac{o_2 s^{np-1} + o_3 s^{np-2} + \dots + o_{np-1}}{p_1 s^{np} + p_2 s^{np-1} + \dots + p_{np+1}}$$

と分解できる.  $G(s)$  の逆変換を  $g(t)$  とすると

$$f(t) = \frac{q_1}{p_1} \delta(t) + g(t)$$

となる. ここで  $\delta(t)$  は, Dirac の  $\delta$ -関数である.

よって

$$f(t) = \begin{cases} \text{正の最大値} & (t = 0) \\ g(t) & (t > 0) \end{cases}$$

(d)  $T(I) = 0$  を与えたとき,  $f(0)$  の値は  $f(0) = [sF(s)]_{s=\infty}$  という公式から計算する.

$$f(0) = \begin{cases} \text{正の最大値} & (np = nq) \\ \frac{q_1}{p_1} & (np = nq + 1) \\ 0 & (np > nq + 1) \end{cases}$$

## (7) 使用例

### (a) 問題

$\Re(s) > 0$  で正則な有理関数

$$F(s) = \frac{1}{s+1}$$

のラプラス逆変換  $f(t)$  を  $t = 1.0, 2.0, 3.0, 4.0, 5.0$  について求める.

### (b) 入力データ

$P = \{1, 1\}$ ,  $NP = 1$ ,  $Q = 1$ ,  $NQ = 0$ ,  $N = 5$ ,  $T = \{1.0, 2.0, 3.0, 4.0, 5.0\}$ ,  $A = 10$ ,  $IP = 10$ ,  $K1 = 10.0$ ,  $K2 = 0.0$ ,  $ISW = 1$

### (c) 主プログラム

```

PROGRAM BFLARA
INTEGER NP,NQ,N,IP,ISW,IERR
REAL(8) P(20),Q(20),T(20),A,R,F(20),E(20),K1,K2&
        ,W1(2,21)
READ(*,*)NP
READ(*,*)P(I),I=1,NP+1)
READ(*,*)NQ
READ(*,*)Q(I),I=1,NQ+1)
READ(*,*)N
READ(*,*)T(I),I=1,N)
READ(*,*)IP
READ(*,*)A
READ(*,*)K1,K2
READ(*,*)ISW
WRITE(*,1000)
WRITE(*,1100)NP
DO 100 I=1,NP
  WRITE(*,1110)I,P(I)
100 CONTINUE
WRITE(*,1200)NQ
DO 110 I=1,NP
  WRITE(*,1210)I,Q(I)
110 CONTINUE
WRITE(*,1300)N
DO 120 I=1,N
  WRITE(*,1310)I,T(I)
120 CONTINUE
WRITE(*,1400)IP
WRITE(*,1500)A
WRITE(*,1600)K1,K2
CALL DFLARA(P,NP,Q,NQ,T,N,A,IP,K1,K2,R,F,E,ISW,W1,IERR)
WRITE(*,2000) IERR
DO I=1,N
  WRITE(*,2100)I,T(I),F(I),E(I)
ENDDO
1000 FORMAT(4X,' *** DFLARA *** ',/,/,4X,' ** INPUT ** ')
1100 FORMAT(6X,'NP      =',I3)
1110 FORMAT(6X,'P(',I3,')=',F8.3)
1200 FORMAT(' ',/,6X,'NQ      =',I3)
1210 FORMAT(6X,'Q(',I3,')=',F8.3)
1300 FORMAT(' ',/,6X,'N      =',I3)
1310 FORMAT(6X,'T(',I3,')=',F8.3)
1400 FORMAT(' ',/,6X,'IP     =',I3)
1500 FORMAT(6X,'A      =',F8.3)
1600 FORMAT(6X,'K1     =',F8.3,5X,'K2     =',F8.3)
2000 FORMAT(4X,' ** OUTPUT ** ',/,/,6X,' IERR =',I4,/,/,&
        6X,' I ',10X,'T(I)',7X,'F(T(I))',2X,'TRUNCATION ERROR')
2100 FORMAT(6X,I2,F14.6,F14.6,D18.8)
END

```



## (d) 出力結果

```

*** DFLARA ***
** INPUT **
NP   = 1
P( 1)= 1.000

NQ   = 0
Q( 1)= 1.000

N    = 5
T( 1)= 1.000
T( 2)= 2.000
T( 3)= 3.000
T( 4)= 4.000
T( 5)= 5.000

IP   = 10
A    = 10.000
K1   = 10.000      K2   = 0.000
** OUTPUT **

IERR = 0

I      T(I)      F(T(I))  TRUNCATION ERROR
1      1.000000   0.367881  -0.15402395D-05
2      2.000000   0.135336  -0.16370386D-05
3      3.000000   0.049788  -0.16358601D-05
4      4.000000   0.018317  -0.15500555D-05
5      5.000000   0.006739  -0.13969849D-05

```

## 2.17.2 DFLAGE, RFLAGE ラプラス逆変換 (一般関数)

(1) 機能

一般関数  $F(s)$  のラプラス逆変換

$$f(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} F(s)e^{st} ds \quad (0 < t < \infty)$$

を求める.

(2) 使用法

倍精度サブルーチン:

CALL DFLAGE (FI, T, N, A, IP, K1, K2, R, F, ER, W1, IERR)

単精度サブルーチン:

CALL RFLAGE (FI, T, N, A, IP, K1, K2, R, F, ER, W1, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\left\{ \begin{array}{l} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{array} \right\}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	FI	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	—	入 力	像関数 $F(s)$ の虚数部を求める関数副プログラム名
2	T	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	N	入 力	原関数 $f(t)$ の計算点
3	N	I	1	入 力	配列 T の寸法
4	A	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	1	入 力	近似誤差をきめる $a$ の値 (2.1.2(1) (d) 参照)
5	IP	I	1	入 力	オイラー変換の次数 $p$ (2.1.2(1) (d) 参照)
6	K1	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	1	入 力	打ち切り項数決定のためのパラメータ $k_1$ (2.1.2 (1) (d) 参照)
7	K2	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	1	入 力	打ち切り項数決定のためのパラメータ $k_2$ (2.1.2 (1) (d) 参照)
8	R	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	1	入 力	収束座標
9	F	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	N	出 力	各 T (I) における原関数 $f(t)$ の値
10	ER	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	N	出 力	各 F (I) (I=1, ..., N) 計算時の打ち切り誤差
11	W1	$\left\{ \begin{array}{l} D \\ R \end{array} \right\}$	$2 \times (IP+2)$	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $N > 0$
- (b)  $T(i) > 0.0 \quad (i = 1, \dots, N)$
- (c)  $A > 0.0$
- (d)  $IP > 0$
- (e)  $K1 > 0.0$   
 $K2 \geq 0.0$
- (f)  $R \geq 0.0$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
1000	$T(i) \leq 0.0$ だった. ( $i = 1, \dots, N$ )	$T(i)$ の処理を打ち切り, $T(i+1)$ 以降の処理を行う.
2000	$R < 0.0$ だった.	$R = 0.0$ として処理を続ける.
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (c) を満足しなかった.	
3020	制限条件 (d) を満足しなかった.	
3030	制限条件 (e) を満足しなかった.	

## (6) 注意事項

- (a) 引数第 1 項 FI の実際の名前は, 使用者側のプログラムで EXTERNAL 宣言し, 実際の名前の関数副プログラムを作っておかなければならない.

関数副プログラム (倍精度) の作り方は次のようになる.

```

REAL(8) FUNCTION FI(S)
COMPLEX(8) S
FI=~
RETURN
END

```

なお, 本サブルーチンは, 関数  $F(s)$  の虚数部の値を  $\Re(s) > 0.0$ ,  $\Im(s) > 0.0$  の点で評価する.  $F(s)$  が多価関数である場合には, 関数副プログラムの中で正しい枝を計算するように注意が必要である.

- (b) 利用者は, A, IP, K1, K2 の値を変えることにより誤差の制御が簡単にできる.
- (c)  $F(s)$  が  $\Re(s) > \alpha$  で正則であるとき,  $\gamma \geq \alpha$  となる  $\gamma$  を R に入力する.  $R \leq 0.0$  の時は  $R=0.0$  として処理を行う.

(7) 使用例

(a) 問題

$\Re(s) > 0$  で正則な関数

$$F(s) = e^{-\sqrt{s}}$$

のラプラス逆変換  $f(t)$  を  $t=1.0, 2.0, 3.0, 4.0, 5.0$  について求める.

(b) 入力データ

$T=\{1.0, 2.0, 3.0, 4.0, 5.0\}$ ,  $N=5$ ,  $A=10$ ,  $IP=10$ ,  $K1=10.0$ ,  $K2=0.0$ ,  $R=0.0$

(c) 主プログラム

```

PROGRAM BFLAGE
! *** EXAMPLE OF DFLAGE ***
INTEGER N,IP,IERR
REAL(8) T(20),A,R,F(20),E(20),K1,K2&
           ,W1(44),DWCNST
REAL(8) FI
EXTERNAL FI,DWCNST
!
PI=DWCNST(1)
READ(*,*)N
READ(*,*)(T(I),I=1,N)
READ(*,*)IP
READ(*,*)A
READ(*,*)K1,K2
READ(*,*)R
WRITE(*,1000)
WRITE(*,1300)N
DO 120 I=1,N
  WRITE(*,1310)I,T(I)
120 CONTINUE
WRITE(*,1400)IP
WRITE(*,1500)A
WRITE(*,1600)K1,K2
WRITE(*,1700)R
CALL DFLAGE(FI,T,N,A,IP,K1,K2,R,F,E,W1,IERR)
WRITE(*,2000)IERR
DO I=1,N
  WRITE(*,2100)I,T(I),F(I),E(I)
ENDDO
1000 FORMAT(4X,' *** DFLARA *** ',/,/,4X,' ** INPUT ** ')
1300 FORMAT(' ',/,6X,'N      =',I3)
1310 FORMAT(6X,'T(',I3,')=',F8.3)
1400 FORMAT(' ',/,6X,'IP     =',I3)
1500 FORMAT(6X,'A      =',F8.3)
1600 FORMAT(6X,'K1     =',F8.3,5X,'K2    =',F8.3)
1700 FORMAT(6X,'R      =',F8.3,/)
2000 FORMAT(4X,' ** OUTPUT ** ',/,/,6X,' IERR = ',I4,/,/,6X,&
           ' I ',10X,'T(I)',7X,'F(T(I))',2X,'TRUNCATION ERROR')
2100 FORMAT(6X,I2,F14.6,F14.6,D18.8)
END
!
REAL(8) FUNCTION FI(S)
COMPLEX(8) S,SQ
SQ=SQRT(S)
IF(DBLE(SQ).LT.0.0D0) THEN
  SQ=-SQ
ENDIF
FI=AIMAG(EXP(-SQ))
RETURN
END

```

(d) 出力結果

```

*** DFLARA ***
** INPUT **
N      = 5
T( 1)= 1.000
T( 2)= 2.000
T( 3)= 3.000
T( 4)= 4.000
T( 5)= 5.000
IP     = 10
A      = 10.000
K1     = 10.000    K2    = 0.000
R      = 0.000
** OUTPUT **
IERR = 0
I      T(I)      F(T(I))  TRUNCATION ERROR
1      1.000000  0.219698  -0.40345965D-05
2      2.000000  0.088017  -0.15425310D-05
3      3.000000  0.049949  -0.75899467D-06
4      4.000000  0.033126  -0.45183668D-06
5      5.000000  0.024001  -0.30215943D-06

```

## 2.18 ウェーブレット変換

### 2.18.1 DFWTH1, RFWTH1

#### Haar 関数の生成

(1) 機能

1 次元ウェーブレット変換に必要な Haar 関数

$$H_{mn}(x) = \begin{cases} \sqrt{\frac{2^m}{a}} & \frac{a}{2^m}(n-1) \leq x \leq \frac{a}{2^m}(n-1/2) \\ -\sqrt{\frac{2^m}{a}} & \frac{a}{2^m}(n-1/2) < x \leq \frac{a}{2^m}n \\ 0 & \text{その他} \end{cases}$$

を区間  $[0, a](a \geq 0.0)$  で生成する.

(2) 使用法

倍精度サブルーチン:

CALL DFWTH1 (A, M, N, C, BL, BM, BR, IERR)

単精度サブルーチン:

CALL RFWTH1 (A, M, N, C, BL, BM, BR, IERR)

(3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	Haar 関数 $H_{mn}(x)$ を生成する区間 $[0, a]$ の上限 $a$ .
2	M	I	1	入 力	Haar 関数 $H_{mn}(x)$ の世代 $m$ .
3	N	I	1	入 力	Haar 関数 $H_{mn}(x)$ のインデックス $n$ . (注意事項 (a) 参照)
4	C	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	Haar 関数 $H_{mn}(x)$ の区間 $[\frac{a}{2^m}(n-1), \frac{a}{2^m}n]$ での絶対値 $\sqrt{\frac{2^m}{a}}$ .
5	BL	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	Haar 関数 $H_{mn}(x)$ の小さい方の 立ち上がり位置 $\frac{a}{2^m}(n-1)$ .
6	BM	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	Haar 関数 $H_{mn}(x)$ の値が正から負に変化する 位置 $\frac{a}{2^m}(n-1/2)$
7	BR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	Haar 関数 $H_{mn}(x)$ の大きい方の 立ち上がり位置 $\frac{a}{2^m}n$ .
8	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $A > 0$
- (b)  $M \geq 0$
- (c)  $N \leq 2^M$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) このサブルーチンでは定数項を与えない. 基底としての Haar 関数には, このサブルーチンで与えられるものの他に, 区間  $[0, a]$  で一定値  $1/\sqrt{a}$  が必要である. 入力データが連続, あるいはサンプリングされた間隔が不規則な場合, ウェーブレット変換において本サブルーチンを用いる. 変換されたデータのサンプリングされた間隔が一定で, かつ  $2^k$  個 ( $k$  は自然数) である場合, より簡単な 2.18.4  $\left\{ \begin{array}{l} \text{DFWTH2} \\ \text{RFWTH2} \end{array} \right\}$  を用いてウェーブレット変換のための Haar 関数を生成することができる.

## (7) 使用例

## (a) 問題

$a = 2$  の場合に Haar 関数  $H_{34}(x)$  の値を計算する.

## (b) 入力データ

$A=2, M=3, N=4$

## (c) 主プログラム

```

PROGRAM BFWTH1
! *** EXAMPLE OF DFWTH1 ***
REAL(8) A,C,BL,BM,BR
INTEGER M,N,IERR
!
  READ(5,*) A
  READ(5,*) M
  READ(5,*) N
  WRITE(6,1000) ' *** DFWTH1 **** '
  WRITE(6,1000) ' ** INPUT ** '
  WRITE(6,1100) 'A=',A,' M=',M,' N=',N
!
  CALL DFWTH1(A,M,N,C,BL,BM,BR,IERR)
!
  WRITE(6,1000) ' ** OUTPUT ** '
  WRITE(6,1200) ' IERR=',IERR
  WRITE(6,1300) ' C=',C
  WRITE(6,1400) ' BL=',BL,' BM=',BM,' BR=',BR
1000 FORMAT(2X,A14)
1100 FORMAT(2X,A4,D15.8,A4,I5,A4,I5)
1200 FORMAT(2X,A5,I5)
1300 FORMAT(2X,A3,D15.8)
1400 FORMAT(2X,A5,D15.8,A5,D15.8,A5,D15.8)
STOP
END

```

## (d) 出力結果

```

*** DFWTH1 **
** INPUT **
A= 0.10000000D+01 M= 2 N= 2
** OUTPUT **
IERR= 0
C= 0.20000000D+01
BL= 0.25000000D+00 BM= 0.37500000D+00 BR= 0.50000000D+00

```

## 2.18.2 DFWTHR, RFWTHR

## Haar 関数によるウェーブレット変換

## (1) 機能

入力データ  $\{(x_i, f(x_i))\}$  のサンプリングされた間隔が等間隔でない場合や、サンプリングされたデータ数が  $2^k$  ( $k$  は自然数) 個でない場合に Haar 関数によるウェーブレット変換

$$C_{mn} = \int_0^a f(x)H_{mn}(x)dx$$

を計算する.

## (2) 使用法

倍精度サブルーチン:

CALL DFWTHR (XD, YD, ND, MR, NR, DR, IMR, INR, IERR)

単精度サブルーチン:

CALL RFWTHR (XD, YD, ND, MR, NR, DR, IMR, INR, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	XD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの $x$ 座標の組 $\{x_i\}$ .
2	YD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの関数値の組 $\{f(x_i)\}$ .
3	ND	I	1	入 力	入力データの個数 $N$ .
4	MR	I	1	入 力	ウェーブレット変換に用いる Haar 関数 $H_{mn}(x)$ のインデックス $m$ の最大値.
5	NR	I	1	入 力	ウェーブレット変換に用いる Haar 関数 $H_{mn}(x)$ のインデックス $n$ の最大値. $m$ 世代でインデックス $N$ までの Haar 関数をウェーブレット変換に用いる.
6	DR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$0 : 2^{\text{MR}} + \text{NR} - 1$	出 力	ウェーブレット変換 $C_{mn}$ . (注意事項 (a) および (b) 参照)
7	IMR	I	$2^{\text{MR}} + \text{NR} - 1$	出 力	DR に格納されているウェーブレット変換 $C_{mn}$ のインデックス $m$ の情報 (注意事項 (b) 参照)
8	INR	I	$2^{\text{MR}} + \text{NR} - 1$	出 力	DR に格納されているウェーブレット変換 $C_{mn}$ のインデックス $n$ の情報 (注意事項 (b) 参照)
9	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

- (a)  $MR \geq 0$
- (b)  $NR \leq 2^{MR}$
- (c)  $ND \leq 100000$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) DR (0) には入力データ  $\{f(x_i)\}$  の平均値が格納される.
- (b) DR ( $i$ ) ( $i = 1, 2, \dots, 2^{MR} + NR - 1$ ) には, ウェーブレット変換  $C_{mn}$  の値が格納されている. 対応するインデックス  $m$  は IMR ( $i$ ) に, インデックス  $n$  は INR ( $i$ ) に, それぞれ格納されている.

(7) 使用例

- (a) 問題  
関数

$$f(x) = \sin(2\pi x) + \frac{1}{5} \sin(6\pi x) + \frac{1}{2}$$

を区間  $[0, 1]$  で等間隔にサンプリングして得たデータを用いて,  $m = 3, n = 8$  までのウェーブレット変換を行う.

- (b) 入力データ

$\{(x_i, f(x_i))\}$ ,  $ND=10$ ,  $MR=4$ ,  $NR=2$

- (c) 主プログラム

```

PROGRAM BFWTHR
! *** EXAMPLE OF DFWTHR ***
INTEGER ND
PARAMETER( ND = 10 )
REAL(8) XD(1:ND), YD(1:ND)
INTEGER MR, NR
PARAMETER( MR = 3, NR = 8 )
REAL(8) DR(0:2**MR-1+NR)
INTEGER IMR(2**MR-1+NR), INR(2**MR-1+NR)
INTEGER IERR
INTEGER NUMRESULT
!
NUMRESULT = 2**MR-1+NR
WRITE(6,1000) '*** DFWTHR ***'
WRITE(6,1000) ' ** INPUT ** '
WRITE(6,1500) 'ND=', ND, ' MR=', MR, ' NR=', NR
WRITE(6,1100) ' I   XD          YD'
DO 100 I=1, ND
  READ(5,*) XD(I), YD(I)
  WRITE(6,1200) I, XD(I), YD(I)
100 CONTINUE
!
CALL DFWTHR(XD, YD, ND, MR, NR, DR, IMR, INR, IERR)
!
WRITE(6,1000) ' ** OUTPUT ** '
WRITE(6,1300) 'IERR=', IERR
WRITE(6,1400) ' DR      IMR  INR'
DO 110 I=1, NUMRESULT
  WRITE(6,1600) DR(I), IMR(I), INR(I)
110 CONTINUE
!
1000 FORMAT(2X, A14)
1100 FORMAT(2X, A25)
1200 FORMAT(2X, I5, D15.8, D15.8)

```



```

1300 FORMAT(2X,A5,I5)
1400 FORMAT(2X,A25)
1500 FORMAT(2X,A5,I5,A5,I5,A5,I5)
1600 FORMAT(2X,D15.8,I5,I5)
!
      STOP
      END

```

## (d) 出力結果

```

*** DFWTHR ***
** INPUT **
ND= 10 MR= 3 NR= 8
I   XD           YD
1  0.10000000D+00 0.12779900D+01
2  0.20000000D+00 0.13335000D+01
3  0.30000000D+00 0.13335000D+01
4  0.40000000D+00 0.12779966D+01
5  0.50000000D+00 0.50000000D+00
6  0.60000000D+00-0.27799660D+00
7  0.70000000D+00-0.33350000D+00
8  0.80000000D+00-0.33350000D+00
9  0.90000000D+00-0.27799660D+00
10 0.10000000D+01 0.50000000D+00
** OUTPUT **
IERR= 0
      DR           IMR  INR
0.63846226D+00    0    1
0.12838733D+00    1    1
-0.62125422D-01   1    2
-0.58512678D-02   2    1
0.16694137D+00    2    2
0.11701144D-01    2    3
-0.90783900D-01   2    4
-0.82749422D-02   3    1
0.41375569D-16    3    2
0.41369792D-02    3    3
0.28994221D-01    3    4
0.20684896D-02    3    5
0.51719462D-17    3    6
-0.62054688D-02   3    7
-0.11597689D+00   3    8

```

### 2.18.3 DFWTHS, RFWTHS

#### Haar 関数による逆ウェーブレット変換

(1) 機能

入力データ  $\{(x_i, f(x_i))\}$  のサンプリングされた間隔が等間隔でない場合や、サンプリングされたデータ数が  $2^k$  ( $k$  は自然数) 個でない場合の Haar 関数によるウェーブレット変換  $C_{mn}$  に対して逆ウェーブレット変換

$$\sum_{m,n} C_{mn} H_{mn}(x)$$

により、 $f(x)$  の近似値を計算する

(2) 使用法

倍精度サブルーチン:

CALL DFWTHS (A, DR, MR, NR, MR2, NR2, IMR, INR, FR, XR, IERR)

単精度サブルーチン:

CALL RFWTHS (A, DR, MR, NR, MR2, NR2, IMR, INR, FR, XR, IERR)

(3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	入力データの $x$ 座標の最大値と最小値の差。データは区間 $[0, a]$ で再生される。
2	DR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$0 : 2^{\text{MR}} + \text{NR} - 1$	入 力	ウェーブレット変換 $C_{mn}$ 。(注意事項 (a) 参照)
3	MR	I	1	入 力	ウェーブレット変換時に使った Haar 関数 $H_{mn}(x)$ の $m$ の最大値。
4	NR	I	1	入 力	ウェーブレット変換時に使った Haar 関数 $H_{mn}(x)$ の $n$ の最大値。
5	MR2	I	1	入 力	逆ウェーブレット変換に用いる Haar 関数 $H_{mn}(x)$ のインデックス $m$ の最大値。
6	NR2	I	1	入 力	逆ウェーブレット変換に用いる Haar 関数 $H_{mn}(x)$ のインデックス $n$ の最大値。
7	IMR	I	$2^{\text{MR}} + \text{NR} - 1$	入 力	ウェーブレット変換 $C_{mn}$ のインデックス $m$ の情報。(注意事項 (a) 参照)
8	INR	I	$2^{\text{MR}} + \text{NR} - 1$	入 力	ウェーブレット変換 $C_{mn}$ のインデックス $n$ の情報。(注意事項 (a) 参照)
9	FR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2^{\text{MR}2+1}$	出 力	ウェーブレット変換 $C_{mn}$ から再構成された $f(x)$ の値。(注意事項 (b) 参照)
10	XR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2^{\text{MR}2+1}$	出 力	ウェーブレット変換 $C_{mn}$ から再構成した $f(x)$ の $x$ 座標の値。(注意事項 (b) 参照)
11	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $MR \geq 0$
- (b)  $NR \leq 2^{MR}$
- (c)  $MR2 \leq MR$
- (d)  $NR2 \leq 2^{MR2}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

## (6) 注意事項

- (a) DR ( $i$ ) ( $i = 1, 2, \dots, 2^{MR} + NR - 1$ ) には, ウェーブレット変換  $C_{mn}$  の値が格納されている. 対応するインデックス  $m$  は IMR ( $i$ ) に, インデックス  $n$  は INR ( $i$ ) に, それぞれ格納されている.
- (b) FR に格納されるのは元のデータの近似値であり, XR に格納されているその  $x$  座標値は等間隔である.

## (7) 使用例

## (a) 問題

2.18.2  $\left\{ \begin{array}{l} \text{DFWTHR} \\ \text{RFWTHR} \end{array} \right\}$  の例題を実行して得られたウェーブレット変換を入力データとして  $m = 3, n = 8$  までの逆ウェーブレット変換を行う.

## (b) 入力データ

ウェーブレット変換  $\{C_{mn}\}$ ,  $A=1$ ,  $MR=3$ ,  $NR=8$ ,  $MR2=3$ ,  $NR2=8$

## (c) 主プログラム

```

PROGRAM BFWTHS
! *** EXAMPLE OF DFWTHS ***
INTEGER MR, NR
PARAMETER( MR = 3, NR = 8 )
INTEGER MR2, NR2
PARAMETER( MR2 = 3, NR2 = 8 )
INTEGER IMR(2**MR-1+NR), INR(2**MR-1+NR)
REAL(8) A
REAL(8) DR(0 : 2**MR-1+NR)
REAL(8) FR(2**(MR2+1)), XR(2**(MR2+1))
INTEGER IERR

!
! INTEGER NUMDATA, NUMRESULT
!
! NUMDATA = 2**MR-1+NR
! NUMRESULT = 2**(MR2+1)
!
WRITE(6,1000) '*** DFWTHS ***'
WRITE(6,1000) ' ** INPUT ** '
READ(5,*) A
WRITE(6,1100) 'A=', A
WRITE(6,1200) 'MR=', MR, ' NR=', NR, ' MR2=', MR2, ' NR2=', NR2
DR(0)=0.0D0
WRITE(6,1300) 'DR(0)=', DR(0)
WRITE(6,1400) ' DR IMR INR'
DO 100 I=1, NUMDATA
READ(5,*) DR(I), IMR(I), INR(I)
WRITE(6,1500) DR(I), IMR(I), INR(I)
100 CONTINUE
!
! CALL DFWTHS(A, DR, MR, NR, MR2, NR2, IMR, INR, FR, XR, IERR)
!
!
WRITE(6,1000) ' ** OUTPUT ** '

```

```

      WRITE(6,1600) 'IERR = ',IERR
      WRITE(6,1700) '  XR          FR'
      DO 200 I=1,NUMRESULT
        WRITE(6,1800) XR(I),FR(I)
      200 CONTINUE
!
1000 FORMAT(2X,A14)
1100 FORMAT(2X,A3,D15.8)
1200 FORMAT(2X,A3,I5,A4,I5,A5,I5,A5,I5)
1300 FORMAT(2X,A6,D15.8)
1400 FORMAT(2X,A25)
1500 FORMAT(2X,D15.8,I5,I5)
1600 FORMAT(2X,A7,I5)
1700 FORMAT(2X,A20)
1800 FORMAT(2X,D15.8,D15.8)
!
      STOP
      END

```

## (d) 出力結果

```

*** DFWTHS ***
** INPUT **
A= 0.10000000D+01
MR= 3 NR= 8 MR2= 3 NR2= 8
DR(0)= 0.00000000D+00
      DR      IMR  INR
0.63846226D+00  0  1
0.12838733D+00  1  1
-0.62125422D-01  1  2
-0.58512678D-02  2  1
0.16694137D+00  2  2
0.11701144D-01  2  3
-0.90783900D-01  2  4
-0.82749422D-02  3  1
0.41375569D-16  3  2
0.41369792D-02  3  3
0.28994221D-01  3  4
0.20684896D-02  3  5
0.00000000D+00  3  6
-0.62054688D-02  3  7
-0.11597689D+00  3  8
** OUTPUT **
IERR = 0
      XR      FR
0.31250000D-01  0.78492176D+00
0.93750000D-01  0.83173190D+00
0.15625000D+00  0.83173190D+00
0.21875000D+00  0.83173190D+00
0.28125000D+00  0.80247904D+00
0.34375000D+00  0.77907675D+00
0.40625000D+00  0.20502046D+00
0.46875000D+00  0.41004376D-01
0.53125000D+00-0.69706801D+00
0.59375000D+00-0.70876916D+00
0.65625000D+00-0.74972316D+00
0.71875000D+00-0.74972316D+00
0.78125000D+00-0.74972316D+00
0.84375000D+00-0.71461973D+00
0.90625000D+00-0.69706803D+00
0.96875000D+00-0.41003664D-01

```

## 2.18.4 DFWTH2, RFWTH2

## Haar 関数の生成 (等間隔サンプリングデータ)

## (1) 機能

ウェーブレット変換する対象のデータのサンプリングされた間隔が一定で、かつ個数が  $N = 2^k$  個 ( $k$  は自然数) である場合、1次元ウェーブレット変換に必要な Haar 関数

$$H_{mn}(x) = \begin{cases} \sqrt{\frac{2^m}{a}} & \frac{a}{2^m}(n-1) \leq x \leq \frac{a}{2^m}(n-1/2) \\ -\sqrt{\frac{2^m}{a}} & \frac{a}{2^m}(n-1/2) < x \leq \frac{a}{2^m}n \\ 0 & \text{その他} \end{cases}$$

を区間  $[0, 1]$  で生成する.

## (2) 使用法

倍精度サブルーチン:

CALL DFWTH2 (NA, M, N, C, LR, IERR)

単精度サブルーチン:

CALL RFWTH2 (NA, M, N, C, LR, IERR)

## (3) 引数

D:倍精度実数型    Z:倍精度複素数型    I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型    C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	NA	I	1	入 力	入力データの個数 $N$ .
2	M	I	1	入 力	Haar 関数 $H_{mn}(x)$ の世代 $m$ .
3	N	I	1	入 力	Haar 関数 $H_{mn}(x)$ のインデックス $n$ . (注意事項 (a) 参照)
4	C	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	Haar 関数 $H_{mn}(x)$ の値.
5	LR	I	NA	出 力	Haar 関数 $H_{mn}(x)$ の値の符号. Haar 関数が正のときには 1 が, 負のときには -1 が, 値として格納される.
6	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $NA = 2^k$  ( $k$  は 20 以下の自然数.)

(b)  $0 \leq M \leq k$

(c)  $N \leq 2^M$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

(6) 注意事項

- (a) このサブルーチンでは定数項を与えない. 基底としての Haar 関数には, このサブルーチンで与えられるものの他に,  $[0, 1]$  区間で一定値 1 が必要である.
- (b) 変換されるべきデータが連続, あるいはサンプリングされた間隔が不規則な場合, Haar 関数の生成には 2.18.1  $\left\{ \begin{matrix} \text{DFWTH1} \\ \text{RFWTH1} \end{matrix} \right\}$  を用いなければならない.

(7) 使用例

(a) 問題

区間  $[0, 1]$  での等間隔にならんだ 16 個の  $x$  座標について Haar 関数  $H_{34}(x)$  を計算する.

(b) 入力データ

NA=16, M=3, N=4

(c) 主プログラム

```

PROGRAM BFWTH2
! *** EXAMPLE OF DFWTH2 ***
INTEGER NA, M,N
PARAMETER( NA = 16)
!
REAL(8) C
INTEGER LR(NA)
INTEGER IERR
READ(5,*) M
READ(5,*) N
WRITE(6,1000) ' *** DFWTH2 **** '
WRITE(6,1000) ' ** INPUT ** '
WRITE(6,1100) 'NA=',NA,' M=',M,' N=',N
!
CALL DFWTH2(NA,M,N,C,LR,IERR)
!
WRITE(6,1000) ' ** OUTPUT ** '
WRITE(6,1200) 'IERR=',IERR
WRITE(6,1300) 'C=',C
WRITE(6,1400) 'I LR '
DO 10 I=1,NA
WRITE(6,1500) I, LR(I)
10 CONTINUE
1000 FORMAT(2X,A14)
1100 FORMAT(2X,A4,I5,A4,I5,A4,I5)
1200 FORMAT(2X,A6,I5)
1300 FORMAT(2X,A4,E15.8)
1400 FORMAT(2X,A15)
1500 FORMAT(2X,I5,I5)
!
STOP
END

```

(d) 出力結果

```
*** DFWTH2 **
** INPUT **
NA= 16 M= 3 N= 8
** OUTPUT **
IERR= 0
C= 0.28284271E+01
I LR
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0
12 0
13 0
14 0
15 1
16 -1
```

## 2.18.5 DFWTHT, RFWTHT

## Haar 関数によるウェーブレット変換 (等間隔サンプリングデータ)

## (1) 機能

入力データ  $\{(x_i, f(x_i))\}$  のサンプリングされた間隔が等間隔でかつその回数が  $2^k$  ( $k$  は自然数) 個である場合に Haar 関数によるウェーブレット変換

$$C_{mn} = \int_0^a f(x)H_{mn}(x)dx$$

を計算する.

## (2) 使用法

倍精度サブルーチン:

CALL DFWTHT (XD, YD, ND, MR, NR, A, DR, IMR, INR, IWK, IERR)

単精度サブルーチン:

CALL RFWTHT (XD, YD, ND, MR, NR, A, DR, IMR, INR, IWK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	XD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの $x$ 座標の組 $\{x_i\}$ .
2	YD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの関数値の組 $\{f(x_i)\}$ .
3	ND	I	1	入 力	入力データの個数 $N$ .
4	MR	I	1	入 力	ウェーブレット変換に用いる Haar 関数 $H_{mn}$ のインデックス $m$ の最大値.
5	NR	I	1	入 力	ウェーブレット変換に用いる Haar 関数 $H_{mn}$ のインデックス $n$ の最大値.
6	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	ウェーブレット変換に際しての積分の範囲 $[0, a]$ の上限 $a$ .
7	DR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$0 : 2^{MR} + NR - 1$	出 力	ウェーブレット変換 $C_{mn}$ . (注意事項 (a) および (b) 参照)
8	IMR	I	$2^{MR} + NR - 1$	出 力	DR に格納されているウェーブレット変換 $C_{mn}$ のインデックス $m$ の情報 (注意事項 (b) 参照)
9	INR	I	$2^{MR} + NR - 1$	出 力	DR に格納されているウェーブレット変換 $C_{mn}$ のインデックス $n$ の情報 (注意事項 (b) 参照)
10	IWK	I	ND	ワーク	作業領域
11	IERR	I	1	出 力	エラーインディケータ



## (4) 制限条件

- (a)  $ND = 2^k$  ( $k$  は 20 以下の自然数.)  
 (b)  $0 \leq MR \leq k - 1$   
 (c)  $NR \leq 2^{MR}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	

## (6) 注意事項

- (a) DR (0) には入力データ  $\{f(x_i)\}$  の平均値が格納される.  
 (b) DR ( $i$ ) ( $i = 1, 2, \dots, 2^{MR} + NR - 1$ ) には, ウェーブレット変換  $C_{mn}$  の値が格納されている. 対応するインデックス  $m$  は IMR ( $i$ ) に, インデックス  $n$  は INR ( $i$ ) に, それぞれ格納されている.

## (7) 使用例

- (a) 問題  
関数

$$f(x) = \sin(2\pi x) + \frac{1}{5} \sin(6\pi x) + \frac{1}{2}$$

を区間  $[0, 1]$  で等間隔にサンプリングして得たデータを用いて,  $m = 3, n = 8$  までのウェーブレット変換を行う.

- (b) 入力データ

$\{(x_i, f(x_i))\}$ ,  $ND=16$ ,  $MR=3$ ,  $NR=8$

- (c) 主プログラム

```

PROGRAM BFWHT
! *** EXAMPLE OF DFWHT ***
!
  INTEGER ND
  PARAMETER( ND = 16 )
  REAL(8) XD(1:ND), YD(1:ND)
  REAL(8) A
  INTEGER MR, NR
  PARAMETER( MR = 3, NR = 8 )
  REAL(8) DR(0:2**MR-1+NR)
  INTEGER IMR(2**MR-1+NR), INR(2**MR-1+NR)
  INTEGER IWK(ND)
  INTEGER IERR
!
  NUMRESULT = 2**MR-1+NR
  WRITE(6,1000) '*** DFWHT ***'
  WRITE(6,1000) ' ** INPUT ** '
  WRITE(6,1100) 'ND=', ND, ' MR=', MR, ' NR=', NR
  WRITE(6,1200) ' XD          YD'
  DO 100 I=1, ND
    READ(5,*) XD(I), YD(I)
    WRITE(6,1300) XD(I), YD(I)
  100 CONTINUE
!
  CALL DFWHT(XD, YD, ND, MR, NR, A, DR, IMR, INR, IWK, IERR)
!
  WRITE(6,1000) ' ** OUTPUT ** '
  WRITE(6,1400) ' IERR=', IERR
  WRITE(6,1500) ' A=', A
  WRITE(6,1600) ' DR          IMR  INR'
  DO 110 I=1, NUMRESULT
    WRITE(6,1700) DR(I), IMR(I), INR(I)
  110 CONTINUE
!
```

```

1000 FORMAT(2X,A14)
1100 FORMAT(2X,A4,I5,A4,I5,A4,I5)
1200 FORMAT(2X,A20)
1300 FORMAT(2X,D15.8,D15.8)
1400 FORMAT(2X,A6,I5)
1500 FORMAT(2X,A4,D15.8)
1600 FORMAT(2X,A25)
1700 FORMAT(2X,D15.8,I5,I5)
!
      STOP
      END

```

## (d) 出力結果

```

*** DFWTHT ***
** INPUT **
ND= 16 MR= 3 NR= 8
  XD      YD
0.62500000D-01 0.10674593D+01
0.12500000D+00 0.13485281D+01
0.18750000D+00 0.13473428D+01
0.25000000D+00 0.13000000D+01
0.31250000D+00 0.13473428D+01
0.37500000D+00 0.13485281D+01
0.43750000D+00 0.10674593D+01
0.50000000D+00 0.50000000D+00
0.56250000D+00-0.67459339D-01
0.62500000D+00-0.34852814D+00
0.68750000D+00-0.34734285D+00
0.75000000D+00-0.30000000D+00
0.81250000D+00 0.00000000D+00
0.87500000D+00 0.00000000D+00
0.93750000D+00 0.00000000D+00
0.10000000D+01 0.49994516D+00
** OUTPUT **
IERR= 0
A= 0.10000000D+01
  DR      IMR  INR
0.61812785D+00 0 1
0.70710678D-01 1 1
-0.13817534D+00 1 2
-0.28919425D-01 2 1
0.14105145D+00 2 2
0.28919421D-01 2 3
-0.62493145D-01 2 4
-0.49686414D-01 3 1
0.83691037D-02 3 2
-0.20953342D-03 3 3
0.10031358D+00 3 4
0.49686414D-01 3 5
-0.83691126D-02 3 6
0.00000000D+00 3 7
-0.88378653D-01 3 8

```

## 2.18.6 DFWTHI, RFWTHI

## Haar 関数による逆ウェーブレット変換 (等間隔サンプリングデータ)

## (1) 機能

入力データ  $\{(x_i, f(x_i))\}$  のサンプリングされた間隔が等間隔でかつその回数が  $2^k$  ( $k$  は自然数) 個である場合の Haar 関数によるウェーブレット変換  $C_{mn}$  に対して逆ウェーブレット変換

$$\sum_{m,n} C_{mn} H_{mn}(x)$$

により,  $f(x)$  を計算する

## (2) 使用法

倍精度サブルーチン:

CALL DFWTHI (A, DR, MR, NR, MR2, NR2, IMR, INR, FR, XR, IWK, IERR)

単精度サブルーチン:

CALL RFWTHI (A, DR, MR, NR, MR2, NR2, IMR, INR, FR, XR, IWK, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER(4)} \\ 64 \text{ ビット整数版では INTEGER(8)} \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	入力データの $x$ 座標の最大値と最小値の差にデータのサンプリング間隔を足した値. データは $[0, A]$ の間で再生される.
2	DR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$0 : 2^{\text{MR}} + \text{NR} - 1$	入 力	ウェーブレット変換 $C_{mn}$ . (注意事項 (a) 参照)
3	MR	I	1	入 力	ウェーブレット変換時に使った Haar 関数 $H_{mn}$ の $m$ の最大値.
4	NR	I	1	入 力	ウェーブレット変換時に使った Haar 関数 $H_{mn}$ の $n$ の最大値.
5	MR2	I	1	入 力	逆ウェーブレット変換に用いる Haar 関数 $H_{mn}$ のインデックス $n$ の最大値.
6	NR2	I	1	入 力	逆ウェーブレット変換に用いる Haar 関数 $H_{mn}$ のインデックス $n$ の最大値.
7	IMR	I	$2^{\text{MR}} + \text{NR} - 1$	入 力	ウェーブレット変換 $C_{mn}$ のインデックス $m$ の情報. (注意事項 (a) 参照)
8	INR	I	$2^{\text{MR}} + \text{NR} - 1$	入 力	ウェーブレット変換 $C_{mn}$ のインデックス $n$ の情報. (注意事項 (a) 参照)
9	FR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2^{\text{MR}2+1}$	出 力	ウェーブレット変換 $C_{mn}$ から再構成された $f(x)$ の値. (注意事項 (b) 参照)
10	XR	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	$2^{\text{MR}2+1}$	出 力	ウェーブレット変換 $C_{mn}$ から再構成した $f(x)$ の $x$ 座標の値. (注意事項 (b) 参照)
11	IWK	I	$2^{\text{MR}2+1}$	ワーク	作業領域
12	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

- (a)  $\text{MR} \geq 0$
- (b)  $\text{NR} \leq 2^{\text{MR}}$
- (c)  $\text{MR}2 \leq \text{MR}$
- (d)  $\text{NR}2 \leq 2^{\text{MR}2}$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.
3010	制限条件 (b) を満足しなかった.	
3020	制限条件 (c) を満足しなかった.	
3030	制限条件 (d) を満足しなかった.	

## (6) 注意事項

- (a) DR ( $i$ ) ( $i = 1, 2, \dots, 2^{\text{MR}} + \text{NR} - 1$ ) には, ウェーブレット変換  $C_{mn}$  の値が格納されている. 対応するインデックス  $m$  は IMR ( $i$ ) に, インデックス  $n$  は INR ( $i$ ) に, それぞれ格納されている.
- (b) FR に格納されるのは元のデータの近似値であり, XR に格納されているその  $x$  座標値は等間隔である.

## (7) 使用例

## (a) 問題

2.18.5  $\begin{cases} \text{DFWTHI} \\ \text{RFWTHI} \end{cases}$  の例題を実行して得られたウェーブレット変換を入力データとして  $m = 3, n = 8$  までの逆ウェーブレット変換を行う.

## (b) 入力データ

ウェーブレット変換  $\{C_{mn}\}$ ,  $A=1$ ,  $\text{MR}=3$ ,  $\text{NR}=8$ ,  $\text{MR2}=3$ ,  $\text{NR2}=8$

## (c) 主プログラム

```

PROGRAM BFWTHI
! *** EXAMPLE OF DFWTHI ***
INTEGER MR, NR, MR2, NR2
PARAMETER (MR=3, NR=8, MR2=3, NR2=8)
INTEGER IMR(2**MR-1+NR), INR(2**MR-1+NR)
REAL(8) A
REAL(8) DR(0 : 2**MR-1+NR)
REAL(8) FR(2**(MR2+1)), XR(2**(MR2+1))
INTEGER IWK(2**(MR2+1))
INTEGER I, IERR

!
INTEGER NUMDATA, NUMRESULT
!
NUMDATA = 2**MR-1+NR
NUMRESULT = 2**(MR2+1)
!
!
WRITE(6,1000) ' *** DFWTHI *** '
WRITE(6,1000) ' ** INPUT ** '
READ(5,*) A
WRITE(6,1100) ' A=', A
WRITE(6,1200) ' MR=', MR, ' NR=', NR, ' MR2=', MR2, ' NR2=', NR2
DR(0)=0.0D0
WRITE(6,1300) ' DR(0)=', DR(0)
WRITE(6,1400) ' DR          IMR  INR'
DO 100 I = 1, NUMDATA
  READ(5,*) DR(I), IMR(I), INR(I)
  WRITE(6,1500) DR(I), IMR(I), INR(I)
100 CONTINUE
!
CALL DFWTHI (A, DR, MR, NR, MR2, NR2, IMR, INR, FR, XR, IWK, IERR)
!
WRITE(6,1000) ' ** OUTPUT ** '
WRITE(6,1600) ' IERR = ', IERR
WRITE(6,1700) ' XR          FR'
DO 200 I=1, NUMRESULT
  WRITE(6,1800) XR(I), FR(I)
200 CONTINUE
!
1000 FORMAT(2X, A14)
1100 FORMAT(2X, A3, D15.8)
1200 FORMAT(2X, A3, I5, A4, I5, A5, I5, A5, I5)
1300 FORMAT(2X, A6, D15.8)
1400 FORMAT(2X, A25)
1500 FORMAT(2X, D15.8, I5, I5)
1600 FORMAT(2X, A7, I5)
1700 FORMAT(2X, A20)
1800 FORMAT(2X, D15.8, D15.8)
!

```

STOP  
END

## (d) 出力結果

```

*** DFWTHI **
** INPUT **
A= 0.10000000D+01
MR= 3 NR= 8 MR2= 3 NR2= 8
DR(0)= 0.0000000D+00
      DR      IMR  INR
0.58649342D+02  0    1
0.70710674D-01  1    1
0.81930354D+02  1    2
-0.28919414D-01  2    1
0.14105146D+00  2    2
0.28919422D-01  2    3
-0.38177975D+02  2    4
-0.49686414D-01  3    1
0.83691208D-02  3    2
-0.20953306D-03  3    3
0.10031359D+00  3    4
0.49686415D-01  3    5
-0.83691102D-02  3    6
-0.12247321D+01  3    7
-0.55205192D+02  3    8
** OUTPUT **
IERR = 0
      XR      FR
0.31250000D-01 0.58550969D+02
0.93750000D-01 0.58832038D+02
0.15625000D+00 0.58830852D+02
0.21875000D+00 0.58783509D+02
0.28125000D+00 0.58830852D+02
0.34375000D+00 0.58832038D+02
0.40625000D+00 0.58550969D+02
0.46875000D+00 0.57983509D+02
0.53125000D+00 0.57416049D+02
0.59375000D+00 0.57134980D+02
0.65625000D+00 0.57136166D+02
0.71875000D+00 0.57183508D+02
0.78125000D+00-0.25433638D+03
0.84375000D+00-0.24740824D+03
0.90625000D+00-0.25430427D+03
0.96875000D+00 0.57983453D+02

```

### 2.18.7 DFWTMF, RFWTMF メキシカンハット関数の計算

(1) 機能

メキシカンハット関数

$$\varphi_{MH}(x) = (1 - 2x^2)e^{-x^2}$$

によるウェーブレット変換の基底

$$\phi_{MH}(x; a, b) = \frac{1}{\sqrt{C}} \left( \frac{x - b}{a} \right)$$

を計算する. ここで

$$C = a \left( 1 - \frac{a^2}{2} + \frac{3a^4}{4} \right) \sqrt{\frac{\pi}{2}}$$

は規格化の定数である.

(2) 使用法

倍精度サブルーチン:

CALL DFWTMF (A, B, X, V, IERR)

単精度サブルーチン:

CALL RFWTMF (A, B, X, V, IERR)

(3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	メキシカンハット関数によるウェーブレット変換の基底 $\phi_{MH}(x; a, b)$ の周波数パラメータ $a$ .
2	B	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	メキシカンハット関数によるウェーブレット変換の基底 $\phi_{MH}(x; a, b)$ のシフトパラメータ $b$ .
3	X	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	変数値 $x$ .
4	V	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	メキシカンハット関数によるウェーブレット変換の基底 $\phi_{MH}(x; a, b)$ の値
5	IERR	I	1	出 力	エラーインディケータ

(4) 制限条件

(a)  $A > 0$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.

(6) 注意事項

なし.

(7) 使用例

(a) 問題

メキシカンハット関数によるウェーブレット変換の基底  $\phi_{MH}(4; 2, 3)$  を計算する.

(b) 入力データ

$A = 4.0, B = 2.0, X = 3.0$

(c) 主プログラム

```

PROGRAM BFWTMF
! *** EXAMPLE OF DFWTMF ***
REAL(8) A, B, X, V
INTEGER IERR
!
!   READ(5,*) A
!   READ(5,*) B
!   READ(5,*) X
!   WRITE(6,1000) '*** DFWTMF ***'
!   WRITE(6,1000) ' ** INPUT ** '
!   WRITE(6,1100) 'A=',A,' B=',B, ' X=',X
!
!   CALL DFWTMF(A,B,X,V,IERR)
!
!   WRITE(6,1000) ' ** OUTPUT ** '
!   WRITE(6,1200) 'IERR=',IERR
!   WRITE(6,1300) ' V = ',V
!
!1000 FORMAT(2X,A14)
!1100 FORMAT(2X,A5,D15.8,A4,D15.8,A4,D15.8)
!1200 FORMAT(2X,A5,I5)
!1300 FORMAT(2X,A5,D15.8)
!
!   STOP
!   END

```

(d) 出力結果

```

*** DFWTMF ***
** INPUT **
A= 0.40000000D+01 B= 0.20000000D+01 X= 0.30000000D+01
** OUTPUT **
IERR= 0
V = 0.28605127D+02

```



## 2.18.8 DFWTMT, RFWTMT

## メキシカンハット関数によるウェーブレット変換

## (1) 機能

入力データとして与えられた  $n$  個の  $x$  座標と関数値  $f(x)$  の組  $\{(x_i, f(x_i))\}$  ( $i = 1, 2, \dots, n$ ) について、メキシカンハット関数によるウェーブレット変換

$$(W_{\phi_{MH}}f)(b, a) = \int_{-\infty}^{\infty} \phi_{MH}(x; a, b)f(x)dx$$

を計算する。

## (2) 使用法

倍精度サブルーチン:

CALL DFWTMT (XD, YD, ND, A, B, C, IERR)

単精度サブルーチン:

CALL RFWTMT (XD, YD, ND, A, B, C, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	XD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの $x$ 座標の組 $\{x_i\}$ .
2	YD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの関数値の組 $\{f(x_i)\}$ .
3	ND	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	入力データの個数 $n$ .
4	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	ウェーブレット変換の周波数パラメータ $a$
5	B	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	ウェーブレット変換のシフトパラメータ $b$
6	C	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	ウェーブレット変換の値 $(W_{\phi_{MH}}f)(b, a)$ .
7	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $A > 0$

## (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.

## (6) 注意事項

なし

## (7) 使用例

## (a) 問題

関数

$$f(x) = \sin(2\pi x) + \frac{1}{5} \sin(6\pi x) + \frac{1}{2}$$

を区間  $[0, 1]$  で等間隔にサンプリングして得たデータを用いて、メキシカンハット関数によるウェーブレット変換  $(W_{\phi_{MH}} f)(2, 3)$  を計算する.

## (b) 入力データ

 $\{(x_i, f(x_i))\}, A=3, B=2$ 

## (c) 主プログラム

```

PROGRAM BFWTMT
! *** EXAMPLE OF DFWTMT ***
REAL(8) A, B, C
INTEGER ND
PARAMETER( ND = 10 )
REAL(8) XD(ND), YD(ND)
INTEGER IERR
!
WRITE(6,1000) '*** DFWTMT ***'
WRITE(6,1000) ' ** INPUT ** '
READ(5,*) A,B
WRITE(6,1100) 'A=',A, ' B=',B
WRITE(6,1200) ' XD'          YD'
DO 100 I=1,ND
  READ(5,*) XD(I),YD(I)
  WRITE(6,1300) XD(I),YD(I)
100 CONTINUE
!
CALL DFWTMT(XD,YD,ND,A,B,C,IERR)
!
WRITE(6,1000) ' ** OUTPUT ** '
WRITE(6,1400) 'IERR=',IERR
WRITE(6,1500) 'C=',C
!
1000 FORMAT(2X,A14)
1100 FORMAT(2X,A4,D15.8,A4,D15.8)
1200 FORMAT(2X,A20)
1300 FORMAT(2X,D15.8,D15.8)
1400 FORMAT(2X,A6,I5)
1500 FORMAT(2X,A4,D15.8)
!
STOP
END

```

## (d) 出力結果

```

*** DFWTMT ***
** INPUT **
A= 0.30000000D+01 B= 0.20000000D+01
XD          YD
0.10000000D+00 0.12779900D+01
0.20000000D+00 0.13335000D+01
0.30000000D+00 0.13335000D+01
0.40000000D+00 0.12779966D+01
0.50000000D+00 0.50000000D+00
0.60000000D+00-0.27799660D+00
0.70000000D+00-0.33350000D+00
0.80000000D+00-0.33350000D+00
0.90000000D+00-0.27799660D+00
0.10000000D+01 0.50000000D+00
** OUTPUT **
IERR= 0
C= 0.42057113D+01

```

## 2.18.9 DFWTFF, RFWTFF フレンチハット関数の計算

### (1) 機能

フレンチハット関数

$$\varphi_{FH}(x) = \begin{cases} 1 & -1 \leq x < 1 \\ -\frac{1}{2} - 3 \leq x < -1 \text{ または } 1 \leq x < 3 \\ 0 & \text{それ以外} \end{cases}$$

によるウェーブレット変換の基底

$$\phi_{FH}(x; a, b) = \frac{1}{\sqrt{3a}} \varphi\left(\frac{x-b}{a}\right)$$

を計算する.

### (2) 使用法

倍精度サブルーチン:

CALL DFWTFF (A, B, X, V, IERR)

単精度サブルーチン:

CALL RFWTFF (A, B, X, V, IERR)

### (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	A	$\begin{cases} D \\ R \end{cases}$	1	入 力	フレンチハット関数によるウェーブレット変換の基底 $\phi_{FH}(x; a, b)$ の周波数パラメータ $a$ .
2	B	$\begin{cases} D \\ R \end{cases}$	1	入 力	フレンチハット関数によるウェーブレット変換の基底 $\phi_{FH}(x; a, b)$ のシフトパラメータ $b$
3	X	$\begin{cases} D \\ R \end{cases}$	1	入 力	変数値 $x$
4	V	$\begin{cases} D \\ R \end{cases}$	1	出 力	フレンチハット関数によるウェーブレット変換の基底 $\phi_{FH}(x; a, b)$ の値.
5	IERR	I	1	出 力	エラーインディケータ

### (4) 制限条件

(a)  $A > 0$

### (5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.

(6) 注意事項  
なし

(7) 使用例

(a) 問題

フレンチハット関数によるウェーブレット変換の基底  $\phi_{FH}(1.5; 2, 1)$  を計算する.

(b) 入力データ

X=1.5, A=2, B=1

(c) 主プログラム

```
PROGRAM BFWTFF
! *** EXAMPLE OF DFWFF ***
REAL(8) A,B,X,V
INTEGER IERR
!
READ(5,*) A
READ(5,*) B
READ(5,*) X
WRITE(6,1000) '*** DFWTFF ***'
WRITE(6,1000) ' ** INPUT ** '
WRITE(6,1100) 'A=',A,' B=',B,' X=',X
!
CALL DFWTFF(A,B,X,V,IERR)
!
WRITE(6,1000) ' ** OUTPUT ** '
WRITE(6,1200) 'IERR=',IERR
WRITE(6,1300) ' V = ',V
!
1000 FORMAT(2X,A14)
1100 FORMAT(2X,A3,D15.8,A4,D15.8,A4,D15.8)
1200 FORMAT(2X,A5,I5)
1300 FORMAT(2X,A5,D15.8)
!
STOP
END
```

(d) 出力結果

```
*** DFWTFF ***
** INPUT **
A= 0.20000000D+01 B= 0.10000000D+01 X= 0.15000000D+01
** OUTPUT **
IERR= 0
V = 0.00000000D+00
```

## 2.18.10 DFWTFT, RFWTFT

## フレンチハット関数によるウェーブレット変換

## (1) 機能

入力データとして与えられた  $n$  個の  $x$  座標と関数値  $f(x)$  の組  $\{(x_i, f(x_i))\}$  ( $i = 1, 2, \dots, n$ ) について、フレンチハット関数によるウェーブレット変換

$$(W_{\phi_{FH}} f)(b, a) = \int_{-\infty}^{\infty} \phi_{FH}(x; a, b) f(x) dx$$

を計算する.

## (2) 使用法

倍精度サブルーチン:

CALL DFWTFT (XD, YD, ND, A, B, C, IERR)

単精度サブルーチン:

CALL RFWTFT (XD, YD, ND, A, B, C, IERR)

## (3) 引数

D:倍精度実数型 Z:倍精度複素数型 I:  $\begin{cases} 32 \text{ ビット整数版では INTEGER}(4) \\ 64 \text{ ビット整数版では INTEGER}(8) \end{cases}$   
 R:単精度実数型 C:単精度複素数型

項番	引数名	型	大きさ	入出力	内 容
1	XD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの $x$ 座標の組 $\{x_i\}$ .
2	YD	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	ND	入 力	入力データの関数値の組 $\{f(x_i)\}$ .
3	ND	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	入力データの個数 $n$
4	A	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	ウェーブレット変換の周波数パラメータ $a$
5	B	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	入 力	ウェーブレット変換のシフトパラメータ $b$
6	C	$\begin{Bmatrix} D \\ R \end{Bmatrix}$	1	出 力	ウェーブレット変換の値 $(W_{\phi_{FH}} f)(b, a)$ .
7	IERR	I	1	出 力	エラーインディケータ

## (4) 制限条件

(a)  $A > 0$

(5) エラーインディケータ

IERR の値	意 味	処 理 内 容
0	正常終了.	
3000	制限条件 (a) を満足しなかった.	処理を打ち切る.

(6) 注意事項  
なし

(7) 使用例

(a) 問題  
関数

$$f(x) = \sin(2\pi x) + \frac{1}{5} \sin(6\pi x) + \frac{1}{2}$$

を区間  $[0, 1]$  で等間隔にサンプリングして得たデータを用いて、フレンチハット関数によるウェーブレット変換  $(W_{\phi_{FH}} f)(1, 2)$  を計算する.

(b) 入力データ

$\{(x_i, f(x_i))\}$ ,  $A=2$ ,  $B=1$

(c) 主プログラム

```

PROGRAM BFWTFT
! *** EXAMPLE OF DFWTFT ***
INTEGER NN,ND
PARAMETER(NN=10)
REAL(8) XD(NN),YD(NN)
REAL(8) A,B,C
INTEGER IERR
!
!
INTEGER I
!
READ(5,*) ND
READ(5,*) A
READ(5,*) B
!
WRITE(6,1000) '*** DFWTFT ***'
WRITE(6,1000) ' ** INPUT ** '
WRITE(6,1100) 'A=',A,' B=',B,' ND=',ND
WRITE(6,1200) ' XD' YD'
DO 100 I=1,ND
READ(5,*) XD(I),YD(I)
WRITE(6,1300) XD(I),YD(I)
100 CONTINUE
!
CALL DFWTFT(XD,YD,ND,A,B,C,IERR)
!
WRITE(6,1000) ' ** OUTPUT ** '
WRITE(6,1400) 'IERR=',IERR
WRITE(6,1500) 'C=',C
!
1000 FORMAT(2X,A14)
1100 FORMAT(2X,A4,D15.8,A4,D15.8,A4,I5)
1200 FORMAT(2X,A22)
1300 FORMAT(2X,D15.8,2X,D15.8)
1400 FORMAT(2X,A6,I5)
1500 FORMAT(2X,A4,D15.8)
!
STOP
END

```

(d) 出力結果

```

*** DFWTFT ***
** INPUT **
A= 0.20000000D+01 B= 0.10000000D+01 ND= 10
XD YD
0.10000000D+00 0.12779900D+01
0.20000000D+00 0.13335000D+01
0.30000000D+00 0.13335000D+01
0.40000000D+00 0.12779966D+01
0.50000000D+00 0.50000000D+00
0.60000000D+00 -0.27799660D+00
0.70000000D+00 -0.33350000D+00
0.80000000D+00 -0.33350000D+00
0.90000000D+00 -0.27799660D+00
0.10000000D+01 0.50000000D+00
** OUTPUT **
IERR= 0
C=-0.12500000D-01

```



# 付録 A ASL で使用している計算機依存定数

## A.1 誤差判定のための単位

ASL では、浮動小数点演算における誤差判定のための単位として次の値を設定している。誤差判定のための単位は、浮動小数点データの内部表現によって決まる数値であり、ASL ではこの単位を収束判定、零判定などに用いることがある。

表 A-1 誤差判定のための単位

単精度演算	倍精度演算
$2^{-23} (\approx 1.19 \times 10^{-7})$	$2^{-52} (\approx 2.22 \times 10^{-16})$

備考 誤差判定の単位  $\epsilon$  はマシン  $\epsilon$  と呼ばれることもあり、通常、対応する浮動小数点形式で  $1 + \epsilon$  の計算結果が 1 と異なるような最小の正の定数として定義される。したがって、誤差判定の単位を見れば、その浮動小数点形式での (仮数部の) 演算の最大有効桁数がわかる。

## A.2 浮動小数点データの値の最大値・最小値

ASL の内部で定義している浮動小数点データの値の最大値、最小値を以下に示す。

なお、以下の最大値、最小値はハードウェアが実際に採用している浮動小数点形式のそれとは異なる場合があるので注意されたい。

表 A-2 浮動小数点データの値の最大値・最小値

	単精度演算	倍精度演算
最大値	$2^{127}(2 - 2^{-23}) (\approx 3.40 \times 10^{38})$	$2^{1023}(2 - 2^{-52}) (\approx 1.80 \times 10^{308})$
正の最小値	$2^{-126} (\approx 1.17 \times 10^{-38})$	$2^{-1022} (\approx 2.23 \times 10^{-308})$
負の最大値	$-2^{-126} (\approx -1.17 \times 10^{-38})$	$-2^{-1022} (\approx -2.23 \times 10^{-308})$
最小値	$-2^{127}(2 - 2^{-23}) (\approx -3.40 \times 10^{38})$	$-2^{1023}(2 - 2^{-52}) (\approx -1.80 \times 10^{308})$



# 索引

- CAM1HH : 第 1 分册, 83  
CAM1HM : 第 1 分册, 80  
CAM1MH : 第 1 分册, 77  
CAM1MM : 第 1 分册, 74  
CAN1HH : 第 1 分册, 95  
CAN1HM : 第 1 分册, 92  
CAN1MH : 第 1 分册, 89  
CAN1MM : 第 1 分册, 86  
CANVJ1 : 第 1 分册, 123  
CARGJM : 第 1 分册, 36  
CARSJD : 第 1 分册, 31  
CBGMDI : 第 2 分册, 71  
CBGMLC : 第 2 分册, 64  
CBGMLS : 第 2 分册, 66  
CBGMLU : 第 2 分册, 62  
CBGMLX : 第 2 分册, 73  
CBGMMS : 第 2 分册, 68  
CBGMSL : 第 2 分册, 58  
CBGMSM : 第 2 分册, 54  
CBGNDI : 第 2 分册, 90  
CBGNLC : 第 2 分册, 83  
CBGNLS : 第 2 分册, 85  
CBGNLU : 第 2 分册, 81  
CBGNLX : 第 2 分册, 92  
CBGNMS : 第 2 分册, 87  
CBGNSL : 第 2 分册, 78  
CBGNSM : 第 2 分册, 75  
CBHEDI : 第 2 分册, 208  
CBHELX : 第 2 分册, 203  
CBHELS : 第 2 分册, 203  
CBHELX : 第 2 分册, 210  
CBHEMS : 第 2 分册, 205  
CBHESL : 第 2 分册, 196  
CBHEUC : 第 2 分册, 201  
CBHEUD : 第 2 分册, 199  
CBHFDI : 第 2 分册, 192  
CBHFLLS : 第 2 分册, 187  
CBHFLX : 第 2 分册, 194  
CBHFMS : 第 2 分册, 189  
CBHFSL : 第 2 分册, 179  
CBHFUC : 第 2 分册, 185  
CBHFUD : 第 2 分册, 183  
CBHPDI : 第 2 分册, 158  
CBHPLS : 第 2 分册, 153  
CBHPLX : 第 2 分册, 160  
CBHPMS : 第 2 分册, 155  
CBHPSL : 第 2 分册, 145  
CBHPUC : 第 2 分册, 151  
CBHPUD : 第 2 分册, 149  
CBHRDI : 第 2 分册, 175  
CBHRLS : 第 2 分册, 170  
CBHRLX : 第 2 分册, 177  
CBHRMS : 第 2 分册, 172  
CBHRSL : 第 2 分册, 162  
CBHRUC : 第 2 分册, 168  
CBHRUD : 第 2 分册, 166  
CCGEAA : 第 1 分册, 155  
CCGEAN : 第 1 分册, 158  
CCGHAA : 第 1 分册, 306  
CCGHAN : 第 1 分册, 310  
CCGJAA : 第 1 分册, 312  
CCGJAN : 第 1 分册, 316  
CCGKAA : 第 1 分册, 318  
CCGKAN : 第 1 分册, 322  
CCGNAA : 第 1 分册, 160  
CCGNAN : 第 1 分册, 163  
CCGRAA : 第 1 分册, 300  
CCGRAN : 第 1 分册, 304  
CCHEAA : 第 1 分册, 197  
CCHEAN : 第 1 分册, 200  
CCHEEE : 第 1 分册, 208  
CCHEEN : 第 1 分册, 212  
CCHESN : 第 1 分册, 206  
CCHESS : 第 1 分册, 202  
CCHJSS : 第 1 分册, 258  
CCHRAA : 第 1 分册, 179  
CCHRAN : 第 1 分册, 182  
CCHREE : 第 1 分册, 190  
CCHREN : 第 1 分册, 195

- CCHRSN : 第 1 分册, 188  
CCHRSS : 第 1 分册, 184  
CFC1BF : 第 3 分册, 53  
CFC1FB : 第 3 分册, 50  
CFC2BF : 第 3 分册, 103  
CFC2FB : 第 3 分册, 100  
CFC3BF : 第 3 分册, 128  
CFC3FB : 第 3 分册, 125  
CFCMBF : 第 3 分册, 79  
CFCMFB : 第 3 分册, 76  
CIBH1N : 第 5 分册, 131  
CIBH2N : 第 5 分册, 133  
CIBINZ : 第 5 分册, 118  
CIBJNZ : 第 5 分册, 85  
CIBKNZ : 第 5 分册, 120  
CIBYNZ : 第 5 分册, 87  
CIGAMZ : 第 5 分册, 168  
CIGLGZ : 第 5 分册, 170  
CLACHA : 第 5 分册, 327  
CLNCIS : 第 5 分册, 342
- D1CDBN : 第 6 分册, 71  
D1CDBT : 第 6 分册, 111  
D1CDCC : 第 6 分册, 142  
D1CDCH : 第 6 分册, 75  
D1CDEX : 第 6 分册, 128  
D1CDFB : 第 6 分册, 99  
D1CDGM : 第 6 分册, 105  
D1CDGU : 第 6 分册, 131  
D1CDIB : 第 6 分册, 114  
D1CDIC : 第 6 分册, 78  
D1CDIF : 第 6 分册, 102  
D1CDIG : 第 6 分册, 108  
D1CDIN : 第 6 分册, 68  
D1CDIS : 第 6 分册, 96  
D1CDIT : 第 6 分册, 90  
D1CDIX : 第 6 分册, 84  
D1CDLD : 第 6 分册, 133  
D1CDLG : 第 6 分册, 139  
D1CDLN : 第 6 分册, 136  
D1CDNC : 第 6 分册, 81  
D1CDNO : 第 6 分册, 65  
D1CDNT : 第 6 分册, 93  
D1CDPA : 第 6 分册, 122  
D1CDTB : 第 6 分册, 87
- D1CDTR : 第 6 分册, 119  
D1CDUF : 第 6 分册, 117  
D1CDWE : 第 6 分册, 125  
D1DDBP : 第 6 分册, 145  
D1DDGO : 第 6 分册, 149  
D1DDHG : 第 6 分册, 153  
D1DDHN : 第 6 分册, 156  
D1DDPO : 第 6 分册, 151  
D2BA1T : 第 6 分册, 166  
D2BA2S : 第 6 分册, 171  
D2BAGM : 第 6 分册, 182  
D2BAHM : 第 6 分册, 190  
D2BAMO : 第 6 分册, 186  
D2BAMS : 第 6 分册, 178  
D2BASM : 第 6 分册, 193  
D2CCMA : 第 6 分册, 213  
D2CCMT : 第 6 分册, 208  
D2CCPR : 第 6 分册, 218  
D2VCGR : 第 6 分册, 201  
D2VCMT : 第 6 分册, 196  
D3IECD : 第 6 分册, 291  
D3IEME : 第 6 分册, 278  
D3IERA : 第 6 分册, 275  
D3IESR : 第 6 分册, 295  
D3IESU : 第 6 分册, 281  
D3IETC : 第 6 分册, 288  
D3IEVA : 第 6 分册, 285  
D3TSCD : 第 6 分册, 329  
D3TSME : 第 6 分册, 309  
D3TSRA : 第 6 分册, 300  
D3TSRD : 第 6 分册, 304  
D3TSSR : 第 6 分册, 332  
D3TSSU : 第 6 分册, 314  
D3TSTC : 第 6 分册, 324  
D3TSVA : 第 6 分册, 320  
D41WR1 : 第 6 分册, 345  
D42WR1 : 第 6 分册, 365  
D42WRM : 第 6 分册, 357  
D42WRN : 第 6 分册, 351  
D4BI01 : 第 6 分册, 420  
D4GL01 : 第 6 分册, 416  
D4MU01 : 第 6 分册, 398  
D4MWRF : 第 6 分册, 373  
D4MWRM : 第 6 分册, 385  
D4RBO1 : 第 6 分册, 412

- D5CHEF : 第 6 分册, 428  
D5CHMD : 第 6 分册, 437  
D5CHMN : 第 6 分册, 434  
D5CHTT : 第 6 分册, 431  
D5TEMH : 第 6 分册, 447  
D5TESG : 第 6 分册, 440  
D5TESP : 第 6 分册, 451  
D5TEWL : 第 6 分册, 443  
D6CLAN : 第 6 分册, 495  
D6CLDA : 第 6 分册, 499  
D6CLDS : 第 6 分册, 491  
D6CPCC : 第 6 分册, 463  
D6CPSC : 第 6 分册, 465  
D6CVAN : 第 6 分册, 475  
D6CVSC : 第 6 分册, 478  
D6DAFN : 第 6 分册, 482  
D6DASC : 第 6 分册, 485  
D6FALD : 第 6 分册, 469  
D6FAVR : 第 6 分册, 471  
DABMCS : 第 1 分册, 13  
DABMEL : 第 1 分册, 15  
DAM1AD : 第 1 分册, 46  
DAM1MM : 第 1 分册, 62  
DAM1MS : 第 1 分册, 55  
DAM1MT : 第 1 分册, 65  
DAM1MU : 第 1 分册, 52  
DAM1SB : 第 1 分册, 49  
DAM1TM : 第 1 分册, 68  
DAM1TP : 第 1 分册, 107  
DAM1TT : 第 1 分册, 71  
DAM1VM : 第 1 分册, 98  
DAM3TP : 第 1 分册, 109  
DAM3VM : 第 1 分册, 101  
DAM4VM : 第 1 分册, 104  
DAMT1M : 第 1 分册, 58  
DAMVJ1 : 第 1 分册, 112  
DAMVJ3 : 第 1 分册, 115  
DAMVJ4 : 第 1 分册, 119  
DARGJM : 第 1 分册, 26  
DARSJD : 第 1 分册, 21  
DASBCS : 第 1 分册, 17  
DASBEL : 第 1 分册, 19  
DATM1M : 第 1 分册, 60  
DBBDDI : 第 2 分册, 221  
DBBDLC : 第 2 分册, 217  
DBBDLS : 第 2 分册, 219  
DBBDLU : 第 2 分册, 215  
DBBDLX : 第 2 分册, 223  
DBBDSL : 第 2 分册, 212  
DBBPDI : 第 2 分册, 234  
DBBPLS : 第 2 分册, 232  
DBBPLX : 第 2 分册, 236  
DBBPSL : 第 2 分册, 226  
DBBPUC : 第 2 分册, 230  
DBBPUU : 第 2 分册, 229  
DBGMDI : 第 2 分册, 49  
DBGMLC : 第 2 分册, 42  
DBGMLS : 第 2 分册, 44  
DBGMLU : 第 2 分册, 40  
DBGMLX : 第 2 分册, 51  
DBGMMS : 第 2 分册, 46  
DBGMSL : 第 2 分册, 36  
DBGMSM : 第 2 分册, 32  
DBPDDI : 第 2 分册, 102  
DBPDLS : 第 2 分册, 100  
DBPDLX : 第 2 分册, 104  
DBPDSL : 第 2 分册, 94  
DBPDUC : 第 2 分册, 98  
DBPDUU : 第 2 分册, 97  
DBSMDI : 第 2 分册, 134  
DBSMLS : 第 2 分册, 129  
DBSMLX : 第 2 分册, 136  
DBSMMS : 第 2 分册, 131  
DBSMSL : 第 2 分册, 122  
DBSMUC : 第 2 分册, 127  
DBSMUD : 第 2 分册, 125  
DBSNLS : 第 2 分册, 143  
DBSNSL : 第 2 分册, 138  
DBSNUD : 第 2 分册, 141  
DBSPDI : 第 2 分册, 118  
DBSPLS : 第 2 分册, 113  
DBSPLX : 第 2 分册, 120  
DBSPMS : 第 2 分册, 115  
DBSPSL : 第 2 分册, 106  
DBSPUC : 第 2 分册, 111  
DBSPUD : 第 2 分册, 109  
DBTDSL : 第 2 分册, 238  
DBTLCO : 第 2 分册, 275  
DBTLDI : 第 2 分册, 277  
DBTLSL : 第 2 分册, 273

- DBTOSL : 第 2 分册, 256  
DBTPSL : 第 2 分册, 240  
DBTSSL : 第 2 分册, 260  
DBTUCO : 第 2 分册, 269  
DBTUDI : 第 2 分册, 271  
DBTUSL : 第 2 分册, 267  
DBVMSL : 第 2 分册, 263  
DCGBFF : 第 1 分册, 324  
DCGEAA : 第 1 分册, 144  
DCGEAN : 第 1 分册, 148  
DCGGAA : 第 1 分册, 264  
DCGGAN : 第 1 分册, 269  
DCGJAA : 第 1 分册, 288  
DCGJAN : 第 1 分册, 292  
DCGKAA : 第 1 分册, 294  
DCGKAN : 第 1 分册, 298  
DCGNAA : 第 1 分册, 150  
DCGNAN : 第 1 分册, 153  
DCGSAA : 第 1 分册, 271  
DCGSAN : 第 1 分册, 274  
DCGSEE : 第 1 分册, 282  
DCGSEN : 第 1 分册, 286  
DCGSSN : 第 1 分册, 280  
DCGSSS : 第 1 分册, 276  
DCSBAA : 第 1 分册, 214  
DCSBAN : 第 1 分册, 217  
DCSBFF : 第 1 分册, 225  
DCSBSN : 第 1 分册, 223  
DCSBSS : 第 1 分册, 219  
DCSJSS : 第 1 分册, 251  
DCSMAA : 第 1 分册, 164  
DCSMAN : 第 1 分册, 167  
DCSMEE : 第 1 分册, 173  
DCSMEN : 第 1 分册, 177  
DCSMSN : 第 1 分册, 171  
DCSMSS : 第 1 分册, 168  
DCSRSS : 第 1 分册, 245  
DCSTAA : 第 1 分册, 229  
DCSTAN : 第 1 分册, 232  
DCSTEE : 第 1 分册, 239  
DCSTEN : 第 1 分册, 243  
DCSTSN : 第 1 分册, 237  
DCSTSS : 第 1 分册, 233  
DFASMA : 第 6 分册, 242  
DFC1BF : 第 3 分册, 46  
DFC1FB : 第 3 分册, 43  
DFC2BF : 第 3 分册, 96  
DFC2FB : 第 3 分册, 93  
DFC3BF : 第 3 分册, 120  
DFC3FB : 第 3 分册, 116  
DFCMBF : 第 3 分册, 70  
DFCMFB : 第 3 分册, 66  
DFCN1D : 第 3 分册, 143  
DFCN2D : 第 3 分册, 152  
DFCN3D : 第 3 分册, 159  
DFCR1D : 第 3 分册, 169  
DFCR2D : 第 3 分册, 177  
DFCR3D : 第 3 分册, 184  
DFCRCS : 第 6 分册, 240  
DFCRCZ : 第 6 分册, 238  
DFCRSC : 第 6 分册, 236  
DFCVCS : 第 6 分册, 232  
DFCVSC : 第 6 分册, 229  
DFDPED : 第 6 分册, 248  
DFDPES : 第 6 分册, 246  
DFDPET : 第 6 分册, 251  
DFLAGE : 第 3 分册, 225  
DFLARA : 第 3 分册, 220  
DFPS1D : 第 3 分册, 194  
DFPS2D : 第 3 分册, 201  
DFPS3D : 第 3 分册, 208  
DFR1BF : 第 3 分册, 61  
DFR1FB : 第 3 分册, 57  
DFR2BF : 第 3 分册, 111  
DFR2FB : 第 3 分册, 107  
DFR3BF : 第 3 分册, 137  
DFR3FB : 第 3 分册, 133  
DFRMBF : 第 3 分册, 88  
DFRMFB : 第 3 分册, 84  
DFWTFF : 第 3 分册, 250  
DFWTFT : 第 3 分册, 252  
DFWTH1 : 第 3 分册, 228  
DFWTH2 : 第 3 分册, 236  
DFWTHI : 第 3 分册, 242  
DFWTHR : 第 3 分册, 230  
DFWTHS : 第 3 分册, 233  
DFWTHT : 第 3 分册, 239  
DFWTMF : 第 3 分册, 246  
DFWTMT : 第 3 分册, 248  
DGICBP : 第 4 分册, 410

- DGICBS : 第 4 分册, 430  
DGICCM : 第 4 分册, 388  
DGICCN : 第 4 分册, 391  
DGICCO : 第 4 分册, 384  
DGICCP : 第 4 分册, 377  
DGICCCQ : 第 4 分册, 378  
DGICCR : 第 4 分册, 380  
DGICCS : 第 4 分册, 382  
DGICCT : 第 4 分册, 386  
DGIDBY : 第 4 分册, 414  
DGIDCY : 第 4 分册, 396  
DGIDMC : 第 4 分册, 360  
DGIDPC : 第 4 分册, 352  
DGIDSC : 第 4 分册, 355  
DGIDYB : 第 4 分册, 403  
DGIIBZ : 第 4 分册, 416  
DGIICZ : 第 4 分册, 398  
DGIIMC : 第 4 分册, 372  
DGIIPC : 第 4 分册, 365  
DGIISC : 第 4 分册, 368  
DGIIZB : 第 4 分册, 407  
DGISBX : 第 4 分册, 412  
DGISCX : 第 4 分册, 394  
DGISI1 : 第 4 分册, 433  
DGISI2 : 第 4 分册, 437  
DGISI3 : 第 4 分册, 444  
DGISMC : 第 4 分册, 347  
DGISPC : 第 4 分册, 339  
DGISPO : 第 4 分册, 418  
DGISPR : 第 4 分册, 421  
DGISS1 : 第 4 分册, 450  
DGISS2 : 第 4 分册, 454  
DGISS3 : 第 4 分册, 462  
DGISSC : 第 4 分册, 342  
DGISSO : 第 4 分册, 424  
DGISSR : 第 4 分册, 427  
DGISXB : 第 4 分册, 400  
DH2INT : 第 4 分册, 245  
DHBDFS : 第 4 分册, 217  
DHBSFC : 第 4 分册, 220  
DHEMNH : 第 4 分册, 223  
DHEMNI : 第 4 分册, 236  
DHEMNL : 第 4 分册, 187  
DHNANL : 第 4 分册, 214  
DHNEFL : 第 4 分册, 196  
DHNENH : 第 4 分册, 229  
DHNENL : 第 4 分册, 206  
DHNFML : 第 4 分册, 257  
DHNFMN : 第 4 分册, 251  
DHNIFL : 第 4 分册, 200  
DHNINH : 第 4 分册, 232  
DHNINI : 第 4 分册, 242  
DHNINL : 第 4 分册, 210  
DHNOFH : 第 4 分册, 226  
DHNOFI : 第 4 分册, 239  
DHN OFL : 第 4 分册, 193  
DHNPNL : 第 4 分册, 203  
DHN RML : 第 4 分册, 254  
DHN RNM : 第 4 分册, 248  
DHNSNL : 第 4 分册, 190  
DIB AID : 第 5 分册, 155  
DIB AIX : 第 5 分册, 151  
DIB BEI : 第 5 分册, 137  
DIB BER : 第 5 分册, 135  
DIB BID : 第 5 分册, 157  
DIB BIX : 第 5 分册, 153  
DIB IMX : 第 5 分册, 112  
DIB INX : 第 5 分册, 108  
DIB JMX : 第 5 分册, 79  
DIB JNX : 第 5 分册, 75  
DIB KEI : 第 5 分册, 141  
DIB KER : 第 5 分册, 139  
DIB KMX : 第 5 分册, 115  
DIB KNX : 第 5 分册, 110  
DIB SIN : 第 5 分册, 127  
DIB SJN : 第 5 分册, 123  
DIB SKN : 第 5 分册, 129  
DIB SYN : 第 5 分册, 125  
DIB YMX : 第 5 分册, 82  
DIB YNX : 第 5 分册, 77  
DIE II1 : 第 5 分册, 180  
DIE II2 : 第 5 分册, 182  
DIE II3 : 第 5 分册, 184  
DIE II4 : 第 5 分册, 186  
DIGIG1 : 第 5 分册, 164  
DIGIG2 : 第 5 分册, 166  
DIICOS : 第 5 分册, 212  
DI IERF : 第 5 分册, 228  
DI ISIN : 第 5 分册, 210  
DILEG1 : 第 5 分册, 232

- DILEG2 : 第 5 分册, 235  
DIMTCE : 第 5 分册, 252  
DIMTSE : 第 5 分册, 255  
DIOPC2 : 第 5 分册, 248  
DIOPTH : 第 5 分册, 246  
DIOPLG : 第 5 分册, 250  
DIOPLH : 第 5 分册, 244  
DIOPLA : 第 5 分册, 242  
DIOPLB : 第 5 分册, 237  
DIXEPS : 第 5 分册, 270  
DIZBS0 : 第 5 分册, 90  
DIZBS1 : 第 5 分册, 92  
DIZBSL : 第 5 分册, 98  
DIZBSN : 第 5 分册, 94  
DIZBYN : 第 5 分册, 96  
DIZGLW : 第 5 分册, 239  
DJTECC : 第 6 分册, 32  
DJTEEX : 第 6 分册, 29  
DJTEGM : 第 6 分册, 41  
DJTEGU : 第 6 分册, 35  
DJTELG : 第 6 分册, 44  
DJTENO : 第 6 分册, 26  
DJTEUN : 第 6 分册, 21  
DJTEWE : 第 6 分册, 38  
DKFNCS : 第 4 分册, 66  
DKHNCS : 第 4 分册, 70  
DKINCT : 第 4 分册, 51  
DKMNCN : 第 4 分册, 74  
DKSNCA : 第 4 分册, 45  
DKSNCS : 第 4 分册, 39  
DKSSCA : 第 4 分册, 60  
DLARHA : 第 5 分册, 324  
DLNRDS : 第 5 分册, 330  
DLNRIS : 第 5 分册, 333  
DLNRSA : 第 5 分册, 339  
DLNRSS : 第 5 分册, 336  
DLSRDS : 第 5 分册, 345  
DLSRIS : 第 5 分册, 350  
DMCLAF : 第 5 分册, 407  
DMCLCP : 第 5 分册, 427  
DMCLMC : 第 5 分册, 422  
DMCLMZ : 第 5 分册, 416  
DMCLSN : 第 5 分册, 402  
DMCLTP : 第 5 分册, 433  
DMCQAZ : 第 5 分册, 449  
DMCQLM : 第 5 分册, 444  
DMCQSN : 第 5 分册, 439  
DMCUSN : 第 5 分册, 399  
DMSP11 : 第 5 分册, 467  
DMSP1M : 第 5 分册, 460  
DMSPMM : 第 5 分册, 464  
DMSQPM : 第 5 分册, 455  
DMUMQG : 第 5 分册, 392  
DMUMQN : 第 5 分册, 389  
DMUSSN : 第 5 分册, 396  
DMUUSN : 第 5 分册, 386  
DNCBPO : 第 4 分册, 316  
DNDAAO : 第 4 分册, 296  
DNDANL : 第 4 分册, 302  
DNDAP0 : 第 4 分册, 299  
DNGAPL : 第 4 分册, 312  
DNLNMA : 第 6 分册, 525  
DNLNRG : 第 6 分册, 513  
DNLNRR : 第 6 分册, 518  
DNNLGF : 第 6 分册, 535  
DNNLPO : 第 6 分册, 530  
DNRAPL : 第 4 分册, 307  
DOFNNF : 第 4 分册, 98  
DOFNNV : 第 4 分册, 92  
DOHNLV : 第 4 分册, 117  
DOHNNF : 第 4 分册, 111  
DOHNNV : 第 4 分册, 105  
DOIEF2 : 第 4 分册, 127  
DOIEV1 : 第 4 分册, 130  
DOLNLV : 第 4 分册, 123  
DOPDH2 : 第 4 分册, 133  
DOPDH3 : 第 4 分册, 139  
DOSNNF : 第 4 分册, 85  
DOSNNV : 第 4 分册, 79  
DPDAPN : 第 4 分册, 284  
DPDOPL : 第 4 分册, 281  
DPGOPL : 第 4 分册, 293  
DPLOPL : 第 4 分册, 288  
DQFODX : 第 4 分册, 154  
DQMOGX : 第 4 分册, 157  
DQMOHX : 第 4 分册, 160  
DQMOJX : 第 4 分册, 163  
DSMGON : 第 5 分册, 290  
DSMGPA : 第 5 分册, 294  
DSSTA1 : 第 5 分册, 277

- DSSTA2 : 第 5 分冊, 280  
DSSTPT : 第 5 分冊, 287  
DSSTRA : 第 5 分冊, 284  
DXA005 : 第 1 分冊, 39
- GAM1HH : 共有メモリ並列機能編, 41  
GAM1HM : 共有メモリ並列機能編, 37  
GAM1MH : 共有メモリ並列機能編, 33  
GAM1MM : 共有メモリ並列機能編, 29  
GAN1HH : 共有メモリ並列機能編, 54  
GAN1HM : 共有メモリ並列機能編, 51  
GAN1MH : 共有メモリ並列機能編, 48  
GAN1MM : 共有メモリ並列機能編, 45  
GBHESL : 共有メモリ並列機能編, 126  
GBHEUD : 共有メモリ並列機能編, 130  
GBHFSL : 共有メモリ並列機能編, 120  
GBHFUD : 共有メモリ並列機能編, 124  
GBHPSL : 共有メモリ並列機能編, 108  
GBHPUD : 共有メモリ並列機能編, 112  
GBHRSL : 共有メモリ並列機能編, 114  
GBHRUD : 共有メモリ並列機能編, 118  
GCGJAA : 共有メモリ並列機能編, 244  
GCGJAN : 共有メモリ並列機能編, 248  
GCGKAA : 共有メモリ並列機能編, 250  
GCGKAN : 共有メモリ並列機能編, 254  
GCGRAA : 共有メモリ並列機能編, 238  
GCGRAN : 共有メモリ並列機能編, 242  
GCHEAA : 共有メモリ並列機能編, 202  
GCHEAN : 共有メモリ並列機能編, 206  
GCHESN : 共有メモリ並列機能編, 212  
GCHESS : 共有メモリ並列機能編, 208  
GCHRAA : 共有メモリ並列機能編, 189  
GCHRAN : 共有メモリ並列機能編, 193  
GCHRSN : 共有メモリ並列機能編, 200  
GCHRSS : 共有メモリ並列機能編, 195  
GFC2BF : 共有メモリ並列機能編, 301  
GFC2FB : 共有メモリ並列機能編, 298  
GFC3BF : 共有メモリ並列機能編, 325  
GFC3FB : 共有メモリ並列機能編, 322  
GFCMBF : 共有メモリ並列機能編, 276  
GFCMFB : 共有メモリ並列機能編, 272
- HAM1HH : 共有メモリ並列機能編, 41  
HAM1HM : 共有メモリ並列機能編, 37  
HAM1MH : 共有メモリ並列機能編, 33  
HAM1MM : 共有メモリ並列機能編, 29
- HAN1HH : 共有メモリ並列機能編, 54  
HAN1HM : 共有メモリ並列機能編, 51  
HAN1MH : 共有メモリ並列機能編, 48  
HAN1MM : 共有メモリ並列機能編, 45  
HBGMLC : 共有メモリ並列機能編, 86  
HBGMLU : 共有メモリ並列機能編, 84  
HBGMSL : 共有メモリ並列機能編, 80  
HBGMSM : 共有メモリ並列機能編, 76  
HBGNLC : 共有メモリ並列機能編, 96  
HBGNLU : 共有メモリ並列機能編, 94  
HBGNSL : 共有メモリ並列機能編, 91  
HBGNSM : 共有メモリ並列機能編, 88  
HBHESL : 共有メモリ並列機能編, 126  
HBHEUD : 共有メモリ並列機能編, 130  
HBHFSL : 共有メモリ並列機能編, 120  
HBHFUD : 共有メモリ並列機能編, 124  
HBHPSL : 共有メモリ並列機能編, 108  
HBHPUD : 共有メモリ並列機能編, 112  
HBHRSL : 共有メモリ並列機能編, 114  
HBHRUD : 共有メモリ並列機能編, 118  
HCGJAA : 共有メモリ並列機能編, 244  
HCGJAN : 共有メモリ並列機能編, 248  
HCGKAA : 共有メモリ並列機能編, 250  
HCGKAN : 共有メモリ並列機能編, 254  
HCGRAA : 共有メモリ並列機能編, 238  
HCGRAN : 共有メモリ並列機能編, 242  
HCHEAA : 共有メモリ並列機能編, 202  
HCHEAN : 共有メモリ並列機能編, 206  
HCHESN : 共有メモリ並列機能編, 212  
HCHESS : 共有メモリ並列機能編, 208  
HCHRAA : 共有メモリ並列機能編, 189  
HCHRAN : 共有メモリ並列機能編, 193  
HCHRSN : 共有メモリ並列機能編, 200  
HCHRSS : 共有メモリ並列機能編, 195  
HFC2BF : 共有メモリ並列機能編, 301  
HFC2FB : 共有メモリ並列機能編, 298  
HFC3BF : 共有メモリ並列機能編, 325  
HFC3FB : 共有メモリ並列機能編, 322  
HFCMBF : 共有メモリ並列機能編, 276  
HFCMFB : 共有メモリ並列機能編, 272
- IIIERF : 第 5 分冊, 230  
JIIERF : 第 5 分冊, 230  
PAM1MM : 共有メモリ並列機能編, 17

- PAM1MT : 共有メモリ並列機能編, 20  
 PAM1MU : 共有メモリ並列機能編, 14  
 PAM1TM : 共有メモリ並列機能編, 23  
 PAM1TT : 共有メモリ並列機能編, 26  
 PBSNSL : 共有メモリ並列機能編, 103  
 PBSNUD : 共有メモリ並列機能編, 106  
 PBSPSL : 共有メモリ並列機能編, 98  
 PBSPUD : 共有メモリ並列機能編, 101  
 PCGJAA : 共有メモリ並列機能編, 226  
 PCGJAN : 共有メモリ並列機能編, 230  
 PCGKAA : 共有メモリ並列機能編, 232  
 PCGKAN : 共有メモリ並列機能編, 236  
 PCGSAA : 共有メモリ並列機能編, 214  
 PCGSAN : 共有メモリ並列機能編, 217  
 PCGSSN : 共有メモリ並列機能編, 224  
 PCGSSS : 共有メモリ並列機能編, 219  
 PCSMAA : 共有メモリ並列機能編, 179  
 PCSMAN : 共有メモリ並列機能編, 182  
 PCSMSN : 共有メモリ並列機能編, 187  
 PCSMSS : 共有メモリ並列機能編, 184  
 PFC2BF : 共有メモリ並列機能編, 294  
 PFC2FB : 共有メモリ並列機能編, 291  
 PFC3BF : 共有メモリ並列機能編, 317  
 PFC3FB : 共有メモリ並列機能編, 314  
 PFCMBF : 共有メモリ並列機能編, 266  
 PFCMFB : 共有メモリ並列機能編, 262  
 PFCN2D : 共有メモリ並列機能編, 339  
 PFCN3D : 共有メモリ並列機能編, 346  
 PFCR2D : 共有メモリ並列機能編, 354  
 PFCR3D : 共有メモリ並列機能編, 361  
 PFPS2D : 共有メモリ並列機能編, 370  
 PFPS3D : 共有メモリ並列機能編, 377  
 PFR2BF : 共有メモリ並列機能編, 309  
 PFR2FB : 共有メモリ並列機能編, 305  
 PFR3BF : 共有メモリ並列機能編, 334  
 PFR3FB : 共有メモリ並列機能編, 330  
 PFRMBF : 共有メモリ並列機能編, 285  
 PFRMFB : 共有メモリ並列機能編, 281  
 PSSTA1 : 共有メモリ並列機能編, 393  
 PSSTA2 : 共有メモリ並列機能編, 396  
 PXE010 : 共有メモリ並列機能編, 143  
 PXE020 : 共有メモリ並列機能編, 150  
 PXE030 : 共有メモリ並列機能編, 157  
 PXE040 : 共有メモリ並列機能編, 164  
 QAM1MM : 共有メモリ並列機能編, 17  
 QAM1MT : 共有メモリ並列機能編, 20  
 QAM1MU : 共有メモリ並列機能編, 14  
 QAM1TM : 共有メモリ並列機能編, 23  
 QAM1TT : 共有メモリ並列機能編, 26  
 QBGMLC : 共有メモリ並列機能編, 74  
 QBGMLU : 共有メモリ並列機能編, 72  
 QBGMSL : 共有メモリ並列機能編, 68  
 QBGMSM : 共有メモリ並列機能編, 65  
 QBSNSL : 共有メモリ並列機能編, 103  
 QBSNUD : 共有メモリ並列機能編, 106  
 QBSPSL : 共有メモリ並列機能編, 98  
 QBSPUD : 共有メモリ並列機能編, 101  
 QCGJAA : 共有メモリ並列機能編, 226  
 QCGJAN : 共有メモリ並列機能編, 230  
 QCGKAA : 共有メモリ並列機能編, 232  
 QCGKAN : 共有メモリ並列機能編, 236  
 QCGSAA : 共有メモリ並列機能編, 214  
 QCGSAN : 共有メモリ並列機能編, 217  
 QCGSSN : 共有メモリ並列機能編, 224  
 QCGSSS : 共有メモリ並列機能編, 219  
 QCSMAA : 共有メモリ並列機能編, 179  
 QCSMAN : 共有メモリ並列機能編, 182  
 QCSMSN : 共有メモリ並列機能編, 187  
 QCSMSS : 共有メモリ並列機能編, 184  
 QFC2BF : 共有メモリ並列機能編, 294  
 QFC2FB : 共有メモリ並列機能編, 291  
 QFC3BF : 共有メモリ並列機能編, 317  
 QFC3FB : 共有メモリ並列機能編, 314  
 QFCMBF : 共有メモリ並列機能編, 266  
 QFCMFB : 共有メモリ並列機能編, 262  
 QFCN2D : 共有メモリ並列機能編, 339  
 QFCN3D : 共有メモリ並列機能編, 346  
 QFCR2D : 共有メモリ並列機能編, 354  
 QFCR3D : 共有メモリ並列機能編, 361  
 QFPS2D : 共有メモリ並列機能編, 370  
 QFPS3D : 共有メモリ並列機能編, 377  
 QFR2BF : 共有メモリ並列機能編, 309  
 QFR2FB : 共有メモリ並列機能編, 305  
 QFR3BF : 共有メモリ並列機能編, 334  
 QFR3FB : 共有メモリ並列機能編, 330  
 QFRMBF : 共有メモリ並列機能編, 285  
 QFRMFB : 共有メモリ並列機能編, 281  
 QSSTA1 : 共有メモリ並列機能編, 393  
 QSSTA2 : 共有メモリ並列機能編, 396  
 QXE010 : 共有メモリ並列機能編, 143



- QXE020 : 共有メモリ並列機能編, 150  
QXE030 : 共有メモリ並列機能編, 157  
QXE040 : 共有メモリ並列機能編, 164
- R1CDBN : 第 6 分冊, 71  
R1CDBT : 第 6 分冊, 111  
R1CDCC : 第 6 分冊, 142  
R1CDCH : 第 6 分冊, 75  
R1CDEX : 第 6 分冊, 128  
R1CDFB : 第 6 分冊, 99  
R1CDGM : 第 6 分冊, 105  
R1CDGU : 第 6 分冊, 131  
R1CDIB : 第 6 分冊, 114  
R1CDIC : 第 6 分冊, 78  
R1CDIF : 第 6 分冊, 102  
R1CDIG : 第 6 分冊, 108  
R1CDIN : 第 6 分冊, 68  
R1CDIS : 第 6 分冊, 96  
R1CDIT : 第 6 分冊, 90  
R1CDIX : 第 6 分冊, 84  
R1CDLD : 第 6 分冊, 133  
R1CDLG : 第 6 分冊, 139  
R1CDLN : 第 6 分冊, 136  
R1CDNC : 第 6 分冊, 81  
R1CDNO : 第 6 分冊, 65  
R1CDNT : 第 6 分冊, 93  
R1CDPA : 第 6 分冊, 122  
R1CDTB : 第 6 分冊, 87  
R1CDTR : 第 6 分冊, 119  
R1CDUF : 第 6 分冊, 117  
R1CDWE : 第 6 分冊, 125  
R1DDBP : 第 6 分冊, 145  
R1DDGO : 第 6 分冊, 149  
R1DDHG : 第 6 分冊, 153  
R1DDHN : 第 6 分冊, 156  
R1DDPO : 第 6 分冊, 151  
R2BA1T : 第 6 分冊, 166  
R2BA2S : 第 6 分冊, 171  
R2BAGM : 第 6 分冊, 182  
R2BAHM : 第 6 分冊, 190  
R2BAMO : 第 6 分冊, 186  
R2BAMS : 第 6 分冊, 178  
R2BASM : 第 6 分冊, 193  
R2CCMA : 第 6 分冊, 213  
R2CCMT : 第 6 分冊, 208
- R2CCPR : 第 6 分冊, 218  
R2VCGR : 第 6 分冊, 201  
R2VCMT : 第 6 分冊, 196  
R3IECD : 第 6 分冊, 291  
R3IEME : 第 6 分冊, 278  
R3IERA : 第 6 分冊, 275  
R3IESR : 第 6 分冊, 295  
R3IESU : 第 6 分冊, 281  
R3IETC : 第 6 分冊, 288  
R3IEVA : 第 6 分冊, 285  
R3TSCD : 第 6 分冊, 329  
R3TSME : 第 6 分冊, 309  
R3TSRA : 第 6 分冊, 300  
R3TSRD : 第 6 分冊, 304  
R3TSSR : 第 6 分冊, 332  
R3TSSU : 第 6 分冊, 314  
R3TSTC : 第 6 分冊, 324  
R3TSVA : 第 6 分冊, 320  
R41WR1 : 第 6 分冊, 345  
R42WR1 : 第 6 分冊, 365  
R42WRM : 第 6 分冊, 357  
R42WRN : 第 6 分冊, 351  
R4BI01 : 第 6 分冊, 420  
R4GL01 : 第 6 分冊, 416  
R4MU01 : 第 6 分冊, 398  
R4MWRF : 第 6 分冊, 373  
R4MWRM : 第 6 分冊, 385  
R4RB01 : 第 6 分冊, 412  
R5CHEF : 第 6 分冊, 428  
R5CHMD : 第 6 分冊, 437  
R5CHMN : 第 6 分冊, 434  
R5CHTT : 第 6 分冊, 431  
R5TEMH : 第 6 分冊, 447  
R5TESG : 第 6 分冊, 440  
R5TESP : 第 6 分冊, 451  
R5TEWL : 第 6 分冊, 443  
R6CLAN : 第 6 分冊, 495  
R6CLDA : 第 6 分冊, 499  
R6CLDS : 第 6 分冊, 491  
R6CPCC : 第 6 分冊, 463  
R6CPSC : 第 6 分冊, 465  
R6CVAN : 第 6 分冊, 475  
R6CVSC : 第 6 分冊, 478  
R6DAFN : 第 6 分冊, 482  
R6DASC : 第 6 分冊, 485

- R6FALD : 第 6 分册, 469  
R6FAVR : 第 6 分册, 471  
RABMCS : 第 1 分册, 13  
RABMEL : 第 1 分册, 15  
RAM1AD : 第 1 分册, 46  
RAM1MM : 第 1 分册, 62  
RAM1MS : 第 1 分册, 55  
RAM1MT : 第 1 分册, 65  
RAM1MU : 第 1 分册, 52  
RAM1SB : 第 1 分册, 49  
RAM1TM : 第 1 分册, 68  
RAM1TP : 第 1 分册, 107  
RAM1TT : 第 1 分册, 71  
RAM1VM : 第 1 分册, 98  
RAM3TP : 第 1 分册, 109  
RAM3VM : 第 1 分册, 101  
RAM4VM : 第 1 分册, 104  
RAMT1M : 第 1 分册, 58  
RAMVJ1 : 第 1 分册, 112  
RAMVJ3 : 第 1 分册, 115  
RAMVJ4 : 第 1 分册, 119  
RARGJM : 第 1 分册, 26  
RARSJD : 第 1 分册, 21  
RASBCS : 第 1 分册, 17  
RASBEL : 第 1 分册, 19  
RATM1M : 第 1 分册, 60  
RBBDDI : 第 2 分册, 221  
RBBDL C : 第 2 分册, 217  
RBBDL S : 第 2 分册, 219  
RBBDL U : 第 2 分册, 215  
RBBDL X : 第 2 分册, 223  
RBBDSL : 第 2 分册, 212  
RBBPDI : 第 2 分册, 234  
RBBPL S : 第 2 分册, 232  
RBBPL X : 第 2 分册, 236  
RBBPSL : 第 2 分册, 226  
RBBPUC : 第 2 分册, 230  
RBBPUU : 第 2 分册, 229  
RBGM DI : 第 2 分册, 49  
RBGM LC : 第 2 分册, 42  
RBGM LS : 第 2 分册, 44  
RBGM LU : 第 2 分册, 40  
RBGM LX : 第 2 分册, 51  
RBGM MS : 第 2 分册, 46  
RBGM SL : 第 2 分册, 36  
RBGM SM : 第 2 分册, 32  
RBPDDI : 第 2 分册, 102  
RBPDL S : 第 2 分册, 100  
RBPDL X : 第 2 分册, 104  
RBPDSL : 第 2 分册, 94  
RBPDUC : 第 2 分册, 98  
RBPDUU : 第 2 分册, 97  
RBSMDI : 第 2 分册, 134  
RBSMLS : 第 2 分册, 129  
RBSMLX : 第 2 分册, 136  
RBSMMS : 第 2 分册, 131  
RBSMSL : 第 2 分册, 122  
RBSMUC : 第 2 分册, 127  
RBSMUD : 第 2 分册, 125  
RBSNLS : 第 2 分册, 143  
RBSNSL : 第 2 分册, 138  
RBSNUD : 第 2 分册, 141  
RBSPDI : 第 2 分册, 118  
RBSPL S : 第 2 分册, 113  
RBSPL X : 第 2 分册, 120  
RBSPMS : 第 2 分册, 115  
RBSPSL : 第 2 分册, 106  
RBSPUC : 第 2 分册, 111  
RBSPUU : 第 2 分册, 109  
RBTDSL : 第 2 分册, 238  
RBTLCO : 第 2 分册, 275  
RBTLDI : 第 2 分册, 277  
RBTLSL : 第 2 分册, 273  
RBTOSL : 第 2 分册, 256  
RBTPSL : 第 2 分册, 240  
RBTSSL : 第 2 分册, 260  
RBTUCO : 第 2 分册, 269  
RBTUDI : 第 2 分册, 271  
RBTUSL : 第 2 分册, 267  
RBVMSL : 第 2 分册, 263  
RCGBFF : 第 1 分册, 324  
RCGEAA : 第 1 分册, 144  
RCGEAN : 第 1 分册, 148  
RCGGAA : 第 1 分册, 264  
RCGGAN : 第 1 分册, 269  
RCGJAA : 第 1 分册, 288  
RCGJAN : 第 1 分册, 292  
RCGKAA : 第 1 分册, 294  
RCGKAN : 第 1 分册, 298  
RCGNAA : 第 1 分册, 150

- RCGNAN : 第 1 分册, 153  
RCGSAA : 第 1 分册, 271  
RCGSAN : 第 1 分册, 274  
RCGSEE : 第 1 分册, 282  
RCGSEN : 第 1 分册, 286  
RCGSSN : 第 1 分册, 280  
RCGSSS : 第 1 分册, 276  
RCSBAA : 第 1 分册, 214  
RCSBAN : 第 1 分册, 217  
RCSBFF : 第 1 分册, 225  
RCSBSN : 第 1 分册, 223  
RCSBSS : 第 1 分册, 219  
RCSJSS : 第 1 分册, 251  
RCSMAA : 第 1 分册, 164  
RCSMAN : 第 1 分册, 167  
RCSMEE : 第 1 分册, 173  
RCSMEN : 第 1 分册, 177  
RCSMSN : 第 1 分册, 171  
RCSMSS : 第 1 分册, 168  
RCSRSS : 第 1 分册, 245  
RCSTAA : 第 1 分册, 229  
RCSTAN : 第 1 分册, 232  
RCSTEE : 第 1 分册, 239  
RCSTEN : 第 1 分册, 243  
RCSTSN : 第 1 分册, 237  
RCSTSS : 第 1 分册, 233  
RFASMA : 第 6 分册, 242  
RFC1BF : 第 3 分册, 46  
RFC1FB : 第 3 分册, 43  
RFC2BF : 第 3 分册, 96  
RFC2FB : 第 3 分册, 93  
RFC3BF : 第 3 分册, 120  
RFC3FB : 第 3 分册, 116  
RFCMBF : 第 3 分册, 70  
RFCMFB : 第 3 分册, 66  
RFCN1D : 第 3 分册, 143  
RFCN2D : 第 3 分册, 152  
RFCN3D : 第 3 分册, 159  
RFCR1D : 第 3 分册, 169  
RFCR2D : 第 3 分册, 177  
RFCR3D : 第 3 分册, 184  
RFCRCS : 第 6 分册, 240  
RFCRCZ : 第 6 分册, 238  
RFCRSC : 第 6 分册, 236  
RFCVCS : 第 6 分册, 232  
RFCVSC : 第 6 分册, 229  
RFDPED : 第 6 分册, 248  
RFDPEB : 第 6 分册, 246  
RFDPET : 第 6 分册, 251  
RFLAGE : 第 3 分册, 225  
RFLARA : 第 3 分册, 220  
RFPS1D : 第 3 分册, 194  
RFPS2D : 第 3 分册, 201  
RFPS3D : 第 3 分册, 208  
RFR1BF : 第 3 分册, 61  
RFR1FB : 第 3 分册, 57  
RFR2BF : 第 3 分册, 111  
RFR2FB : 第 3 分册, 107  
RFR3BF : 第 3 分册, 137  
RFR3FB : 第 3 分册, 133  
RFRMBF : 第 3 分册, 88  
RFRMFB : 第 3 分册, 84  
RFWTFF : 第 3 分册, 250  
RFWTFT : 第 3 分册, 252  
RFWTH1 : 第 3 分册, 228  
RFWTH2 : 第 3 分册, 236  
RFWTHI : 第 3 分册, 242  
RFWTHR : 第 3 分册, 230  
RFWTHS : 第 3 分册, 233  
RFWTHT : 第 3 分册, 239  
RFWTMF : 第 3 分册, 246  
RFWTMT : 第 3 分册, 248  
RGICBP : 第 4 分册, 410  
RGICBS : 第 4 分册, 430  
RGICCM : 第 4 分册, 388  
RGICCN : 第 4 分册, 391  
RGICCO : 第 4 分册, 384  
RGICCP : 第 4 分册, 377  
RGICCQ : 第 4 分册, 378  
RGICCR : 第 4 分册, 380  
RGICCS : 第 4 分册, 382  
RGICCT : 第 4 分册, 386  
RGIDBY : 第 4 分册, 414  
RGIDCY : 第 4 分册, 396  
RGIDMC : 第 4 分册, 360  
RGIDPC : 第 4 分册, 352  
RGIDSC : 第 4 分册, 355  
RGIDYB : 第 4 分册, 403  
RGIIBZ : 第 4 分册, 416  
RGIICZ : 第 4 分册, 398

- RGIIMC : 第 4 分册, 372  
RGIIPC : 第 4 分册, 365  
RGIISC : 第 4 分册, 368  
RGIIZB : 第 4 分册, 407  
RGISBX : 第 4 分册, 412  
RGISCX : 第 4 分册, 394  
RGISI1 : 第 4 分册, 433  
RGISI2 : 第 4 分册, 437  
RGISI3 : 第 4 分册, 444  
RGISMC : 第 4 分册, 347  
RGISPC : 第 4 分册, 339  
RGISPO : 第 4 分册, 418  
RGISPR : 第 4 分册, 421  
RGISS1 : 第 4 分册, 450  
RGISS2 : 第 4 分册, 454  
RGISS3 : 第 4 分册, 462  
RGISSC : 第 4 分册, 342  
RGISSO : 第 4 分册, 424  
RGISSR : 第 4 分册, 427  
RGISXB : 第 4 分册, 400  
RH2INT : 第 4 分册, 245  
RHBDFS : 第 4 分册, 217  
RHBSFC : 第 4 分册, 220  
RHEMNH : 第 4 分册, 223  
RHEMNI : 第 4 分册, 236  
RHEMNL : 第 4 分册, 187  
RHNANL : 第 4 分册, 214  
RHNEFL : 第 4 分册, 196  
RHNENH : 第 4 分册, 229  
RHNENL : 第 4 分册, 206  
RHNFML : 第 4 分册, 257  
RHNFMN : 第 4 分册, 251  
RHNIFL : 第 4 分册, 200  
RHNINH : 第 4 分册, 232  
RHNINI : 第 4 分册, 242  
RHNINL : 第 4 分册, 210  
RHNOFH : 第 4 分册, 226  
RHNOFI : 第 4 分册, 239  
RHNOFL : 第 4 分册, 193  
RHNPNL : 第 4 分册, 203  
RHNRMN : 第 4 分册, 254  
RHNRLM : 第 4 分册, 248  
RHNSNL : 第 4 分册, 190  
RIBAID : 第 5 分册, 155  
RIBAIX : 第 5 分册, 151  
RIBBEI : 第 5 分册, 137  
RIBBER : 第 5 分册, 135  
RIBBID : 第 5 分册, 157  
RIBBIX : 第 5 分册, 153  
RIBIMX : 第 5 分册, 112  
RIBINX : 第 5 分册, 108  
RIBJMX : 第 5 分册, 79  
RIBJNX : 第 5 分册, 75  
RIBKEI : 第 5 分册, 141  
RIBKER : 第 5 分册, 139  
RIBKMX : 第 5 分册, 115  
RIBKNX : 第 5 分册, 110  
RIBSIN : 第 5 分册, 127  
RIBSIN : 第 5 分册, 123  
RIBSKN : 第 5 分册, 129  
RIBSYN : 第 5 分册, 125  
RIBYMX : 第 5 分册, 82  
RIBYNX : 第 5 分册, 77  
RIEII1 : 第 5 分册, 180  
RIEII2 : 第 5 分册, 182  
RIEII3 : 第 5 分册, 184  
RIEII4 : 第 5 分册, 186  
RIGIG1 : 第 5 分册, 164  
RIGIG2 : 第 5 分册, 166  
RIICOS : 第 5 分册, 212  
RIIERF : 第 5 分册, 228  
RIISIN : 第 5 分册, 210  
RILEG1 : 第 5 分册, 232  
RILEG2 : 第 5 分册, 235  
RIMTCE : 第 5 分册, 252  
RIMTSE : 第 5 分册, 255  
RIOPC2 : 第 5 分册, 248  
RIOPCN : 第 5 分册, 246  
RIOPLA : 第 5 分册, 250  
RIOPLA : 第 5 分册, 244  
RIOPLA : 第 5 分册, 242  
RIOPLA : 第 5 分册, 237  
RIXEPS : 第 5 分册, 270  
RIZBS0 : 第 5 分册, 90  
RIZBS1 : 第 5 分册, 92  
RIZBSL : 第 5 分册, 98  
RIZBSN : 第 5 分册, 94  
RIZBYN : 第 5 分册, 96  
RIZGLW : 第 5 分册, 239  
RJTEBI : 第 6 分册, 47

- RJTECC : 第 6 分册, 32  
RJTEEX : 第 6 分册, 29  
RJTEGM : 第 6 分册, 41  
RJTEGU : 第 6 分册, 35  
RJTELG : 第 6 分册, 44  
RJTENG : 第 6 分册, 50  
RJTEN0 : 第 6 分册, 26  
RJTEPO : 第 6 分册, 53  
RJTEUN : 第 6 分册, 21  
RJTEWE : 第 6 分册, 38  
RKFNCS : 第 4 分册, 66  
RKHNCs : 第 4 分册, 70  
RKINCT : 第 4 分册, 51  
RKMNCN : 第 4 分册, 74  
RKSNCa : 第 4 分册, 45  
RKSNCs : 第 4 分册, 39  
RKSSCA : 第 4 分册, 60  
RLARHA : 第 5 分册, 324  
RLNRDS : 第 5 分册, 330  
RLNRIS : 第 5 分册, 333  
RLNRSA : 第 5 分册, 339  
RLNRSS : 第 5 分册, 336  
RLSRDS : 第 5 分册, 345  
RLSRIS : 第 5 分册, 350  
RMCLAF : 第 5 分册, 407  
RMCLCP : 第 5 分册, 427  
RMCLMC : 第 5 分册, 422  
RMCLMZ : 第 5 分册, 416  
RMCLSN : 第 5 分册, 402  
RMCLTP : 第 5 分册, 433  
RMCQAZ : 第 5 分册, 449  
RMCQLM : 第 5 分册, 444  
RMCQSN : 第 5 分册, 439  
RMCUSN : 第 5 分册, 399  
RMSP11 : 第 5 分册, 467  
RMSP1M : 第 5 分册, 460  
RMSPMM : 第 5 分册, 464  
RMSQPM : 第 5 分册, 455  
RMUMQG : 第 5 分册, 392  
RMUMQN : 第 5 分册, 389  
RMUSSN : 第 5 分册, 396  
RMUUSN : 第 5 分册, 386  
RNCBPO : 第 4 分册, 316  
RNDAAO : 第 4 分册, 296  
RNDANL : 第 4 分册, 302  
RNDAP0 : 第 4 分册, 299  
RNGAPL : 第 4 分册, 312  
RNLNMA : 第 6 分册, 525  
RNLNRG : 第 6 分册, 513  
RNLNRR : 第 6 分册, 518  
RNNLGF : 第 6 分册, 535  
RNRAPL : 第 4 分册, 307  
ROFNMF : 第 4 分册, 98  
ROFNNV : 第 4 分册, 92  
ROHNLV : 第 4 分册, 117  
ROHNNF : 第 4 分册, 111  
ROHNNV : 第 4 分册, 105  
ROIEF2 : 第 4 分册, 127  
ROIEV1 : 第 4 分册, 130  
ROLNLV : 第 4 分册, 123  
ROPDH2 : 第 4 分册, 133  
ROPDH3 : 第 4 分册, 139  
ROSNMF : 第 4 分册, 85  
ROSNNV : 第 4 分册, 79  
RPDAPN : 第 4 分册, 284  
RPDOPL : 第 4 分册, 281  
RPGOPL : 第 4 分册, 293  
RPLOPL : 第 4 分册, 288  
RQFODX : 第 4 分册, 154  
RQMOGX : 第 4 分册, 157  
RQMOHX : 第 4 分册, 160  
RQMOJX : 第 4 分册, 163  
RSMGON : 第 5 分册, 290  
RSMGPA : 第 5 分册, 294  
RSSTA1 : 第 5 分册, 277  
RSSTA2 : 第 5 分册, 280  
RSSTPT : 第 5 分册, 287  
RSSTRA : 第 5 分册, 284  
RXA005 : 第 1 分册, 39  
VIBHOX : 第 5 分册, 143  
VIBH1X : 第 5 分册, 145  
VIBHY0 : 第 5 分册, 147  
VIBHY1 : 第 5 分册, 149  
VIBIOX : 第 5 分册, 100  
VIBI1X : 第 5 分册, 104  
VIBJOX : 第 5 分册, 67  
VIBJ1X : 第 5 分册, 71  
VIBK0X : 第 5 分册, 102  
VIBK1X : 第 5 分册, 106

- VIBY0X : 第 5 分册, 69  
 VIBY1X : 第 5 分册, 73  
 VIDBEY : 第 5 分册, 261  
 VIECI1 : 第 5 分册, 176  
 VIECI2 : 第 5 分册, 178  
 VIEJAC : 第 5 分册, 188  
 VIEJEP : 第 5 分册, 198  
 VIEJTE : 第 5 分册, 200  
 VIEJZT : 第 5 分册, 196  
 VIENMQ : 第 5 分册, 190  
 VIEPAI : 第 5 分册, 202  
 VIERFC : 第 5 分册, 226  
 VIERRF : 第 5 分册, 224  
 VIETHE : 第 5 分册, 193  
 VIGAMX : 第 5 分册, 159  
 VIGBET : 第 5 分册, 174  
 VIGDIG : 第 5 分册, 172  
 VIGLGX : 第 5 分册, 162  
 VIICNC : 第 5 分册, 222  
 VIICND : 第 5 分册, 220  
 VIIDAW : 第 5 分册, 218  
 VIIEXP : 第 5 分册, 205  
 WIIFCO : 第 5 分册, 216  
 WIIFSI : 第 5 分册, 214  
 WIIOLOG : 第 5 分册, 208  
 VINPLG : 第 5 分册, 263  
 VIXSLA : 第 5 分册, 266  
 VIXSPS : 第 5 分册, 258  
 VIXZTA : 第 5 分册, 268  
  
 WBTCLS : 第 2 分册, 252  
 WBTCSL : 第 2 分册, 249  
 WBTDLs : 第 2 分册, 246  
 WBTDSL : 第 2 分册, 243  
 WIBHOX : 第 5 分册, 143  
 WIBH1X : 第 5 分册, 145  
 WIBHYO : 第 5 分册, 147  
 WIBHY1 : 第 5 分册, 149  
 WIBIOX : 第 5 分册, 100  
 WIBI1X : 第 5 分册, 104  
 WIBJOX : 第 5 分册, 67  
 WIBJ1X : 第 5 分册, 71  
 WIBKOX : 第 5 分册, 102  
 WIBK1X : 第 5 分册, 106  
 WIBY0X : 第 5 分册, 69  
  
 WIBY1X : 第 5 分册, 73  
 WIDBEY : 第 5 分册, 261  
 WIECI1 : 第 5 分册, 176  
 WIECI2 : 第 5 分册, 178  
 WIEJAC : 第 5 分册, 188  
 WIEJEP : 第 5 分册, 198  
 WIEJTE : 第 5 分册, 200  
 WIEJZT : 第 5 分册, 196  
 WIENMQ : 第 5 分册, 190  
 WIEPAI : 第 5 分册, 202  
 WIERFC : 第 5 分册, 226  
 WIERRF : 第 5 分册, 224  
 WIETHE : 第 5 分册, 193  
 WIGAMX : 第 5 分册, 159  
 WIGBET : 第 5 分册, 174  
 WIGDIG : 第 5 分册, 172  
 WIGLGX : 第 5 分册, 162  
 WIICNC : 第 5 分册, 222  
 WIICND : 第 5 分册, 220  
 WIIDAW : 第 5 分册, 218  
 WIIEXP : 第 5 分册, 205  
 WIIFCO : 第 5 分册, 216  
 WIIFSI : 第 5 分册, 214  
 WIIOLOG : 第 5 分册, 208  
 WINPLG : 第 5 分册, 263  
 WIXSLA : 第 5 分册, 266  
 WIXSPS : 第 5 分册, 258  
 WIXZTA : 第 5 分册, 268  
  
 ZAM1HH : 第 1 分册, 83  
 ZAM1HM : 第 1 分册, 80  
 ZAM1MH : 第 1 分册, 77  
 ZAM1MM : 第 1 分册, 74  
 ZAN1HH : 第 1 分册, 95  
 ZAN1HM : 第 1 分册, 92  
 ZAN1MH : 第 1 分册, 89  
 ZAN1MM : 第 1 分册, 86  
 ZANVJ1 : 第 1 分册, 123  
 ZARGJM : 第 1 分册, 36  
 ZARSJD : 第 1 分册, 31  
 ZBGMDI : 第 2 分册, 71  
 ZBGMLC : 第 2 分册, 64  
 ZBGMLS : 第 2 分册, 66  
 ZBGMLU : 第 2 分册, 62  
 ZBGMLX : 第 2 分册, 73

- ZBGMMS : 第 2 分册, 68  
ZBGMSL : 第 2 分册, 58  
ZBGMSM : 第 2 分册, 54  
ZBGNDI : 第 2 分册, 90  
ZBGNLC : 第 2 分册, 83  
ZBGNLS : 第 2 分册, 85  
ZBGNLU : 第 2 分册, 81  
ZBGNLX : 第 2 分册, 92  
ZBGNMS : 第 2 分册, 87  
ZBGNSL : 第 2 分册, 78  
ZBGNSM : 第 2 分册, 75  
ZBHEDI : 第 2 分册, 208  
ZBHEL5 : 第 2 分册, 203  
ZBHELX : 第 2 分册, 210  
ZBHEMS : 第 2 分册, 205  
ZBHESL : 第 2 分册, 196  
ZBHEUC : 第 2 分册, 201  
ZBHEUD : 第 2 分册, 199  
ZBHFDI : 第 2 分册, 192  
ZBHFLS : 第 2 分册, 187  
ZBHFLX : 第 2 分册, 194  
ZBHFMS : 第 2 分册, 189  
ZBHFSL : 第 2 分册, 179  
ZBHFUC : 第 2 分册, 185  
ZBHFUD : 第 2 分册, 183  
ZBHPTDI : 第 2 分册, 158  
ZBHPLS : 第 2 分册, 153  
ZBHPLX : 第 2 分册, 160  
ZBHPMS : 第 2 分册, 155  
ZBHPSL : 第 2 分册, 145  
ZBHPUC : 第 2 分册, 151  
ZBHPUD : 第 2 分册, 149  
ZBHRDI : 第 2 分册, 175  
ZBHRLS : 第 2 分册, 170  
ZBHRLX : 第 2 分册, 177  
ZBHRMS : 第 2 分册, 172  
ZBHRSL : 第 2 分册, 162  
ZBHRUC : 第 2 分册, 168  
ZBHRUD : 第 2 分册, 166  
ZCGEAA : 第 1 分册, 155  
ZCGEAN : 第 1 分册, 158  
ZCGHAA : 第 1 分册, 306  
ZCGHAN : 第 1 分册, 310  
ZCGJAA : 第 1 分册, 312  
ZCGJAN : 第 1 分册, 316  
ZCGKAA : 第 1 分册, 318  
ZCGKAN : 第 1 分册, 322  
ZCGNAA : 第 1 分册, 160  
ZCGNAN : 第 1 分册, 163  
ZCGRAA : 第 1 分册, 300  
ZCGRAN : 第 1 分册, 304  
ZCHEAA : 第 1 分册, 197  
ZCHEAN : 第 1 分册, 200  
ZCHEEE : 第 1 分册, 208  
ZCHEEN : 第 1 分册, 212  
ZCHESN : 第 1 分册, 206  
ZCHESS : 第 1 分册, 202  
ZCHJSS : 第 1 分册, 258  
ZCHRAA : 第 1 分册, 179  
ZCHRAN : 第 1 分册, 182  
ZCHREE : 第 1 分册, 190  
ZCHREN : 第 1 分册, 195  
ZCHRSN : 第 1 分册, 188  
ZCHRSS : 第 1 分册, 184  
ZFC1BF : 第 3 分册, 53  
ZFC1FB : 第 3 分册, 50  
ZFC2BF : 第 3 分册, 103  
ZFC2FB : 第 3 分册, 100  
ZFC3BF : 第 3 分册, 128  
ZFC3FB : 第 3 分册, 125  
ZFCMBF : 第 3 分册, 79  
ZFCMFB : 第 3 分册, 76  
ZIBH1N : 第 5 分册, 131  
ZIBH2N : 第 5 分册, 133  
ZIBINZ : 第 5 分册, 118  
ZIBJNZ : 第 5 分册, 85  
ZIBKNZ : 第 5 分册, 120  
ZIBYNZ : 第 5 分册, 87  
ZIGAMZ : 第 5 分册, 168  
ZIGLGZ : 第 5 分册, 170  
ZLACHA : 第 5 分册, 327  
ZLNCIS : 第 5 分册, 342

アプリケーションシステム  
科学技術計算ライブラリ  
ASL ユーザーズガイド

〈 基本機能編 第 3 分冊 〉

2023 年 3 月 ASL (1.1)

付属説明書 3.0.0-230301

日本電気株式会社

© NEC Corporation 2023

日本電気株式会社の許可なく複製・改変などを行うことはできません。

本書の内容に関しては将来予告なしに変更することがあります。