

Fortran 統合開発環境

# Fortran Builder の紹介

日本ニューメリカルアルゴリズムズグループ

技術部 中野 剛

# Fortran Builder の概要

- Fortran 統合開発環境 – 学習用電子ブック搭載
- 世界で多くの実績、「NAGWare f95 コンパイラ」を使用
- 学習用電子ブック「Fortran 90/95 プログラミング入門」搭載
- LAPACK（線形代数パッケージ）利用支援ウィザード搭載
- プロットライブラリ利用支援ウィザード搭載
- 日本語メッセージ及び、日本語マニュアル完備
- デバッグ機能搭載
- Fortran77 から Fortran95 へのコンバート機能
- C/C++/Fortran の混在開発が可能



# 世界で実績ある、NAGWare f95 コンパイラ搭載

- ・ 世界の先端を行く機能を備えた Fortran コンパイラ
- ・ 世界に先駆けて Fortran2003 の機能を搭載
- ・ 分かり易いエラーメッセージ
- ・ C/C++のプログラムを簡単に混在利用
- ・ コンパイル時の厳密な文法エラーチェック

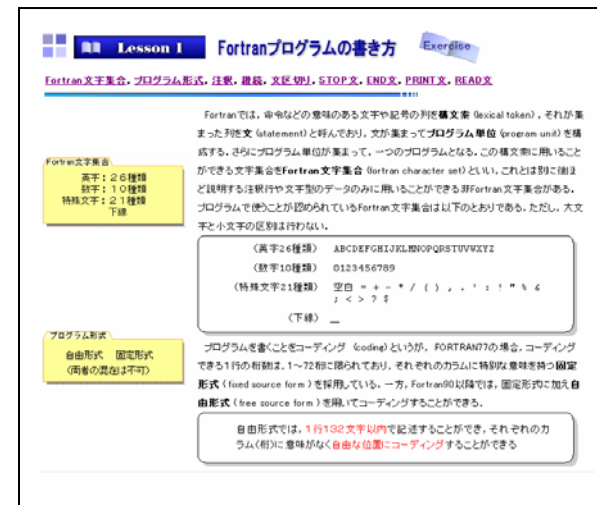
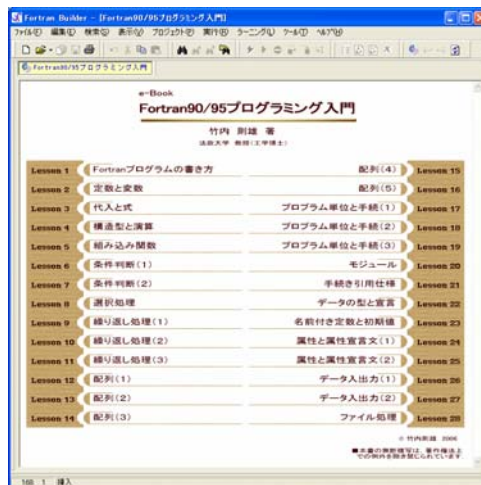
(比較表を参照)

エラーチェック比較表

項目	NAG	C社	I社
同じファイル内での引数タイプの違い	○	○	×
異なるファイル間での引数タイプの違い	○	×	×
定数引数に値を代入 (INTENT文なし)	○	×	×
サブルーチン内でDOループ変数への代入 (INTENT文なし)	○	×	×
配列領域外エラー (X(*)と指定)	○	×	×
配列領域外エラー (ダミー引数へ実際の大きさ以上の値を指定)	○	×	×
文字領域外エラー (ダミー引数へ実際の大きさ以上の値を指定)	○	×	×
ローカル変数の初期化忘れ	○	○	×
引数の初期化忘れ	○	○	×
COMMON変数の初期化忘れ	○	×	×
モジュール変数の初期化忘れ	○	×	×
配列要素の初期化忘れ	○	○	×
配列引数要素の初期化忘れ	○	○	×
COMMON配列要素の初期化忘れ	○	×	×
MODULE配列要素の初期化忘れ	○	×	×
SUBROUTINE内のローカル配列要素の初期化忘れ	○	×	×
自動配列要素の初期化忘れ	○	×	×
保存配列要素の初期化忘れ	○	○	×
INTENT(OUT)配列要素の初期化忘れ	○	○	×
DOループで増分0と指定	○	○	×
EQUIVALENCE経由でDOループ変数への代入	○	×	×
同じファイル内でSOBRUOTINEをFUNCTIONとして参照	○	○	×
別のファイル間でSOBRUOTINEをFUNCTIONとして参照	○	×	×
配列への規格外代入	○	×	×
領域が開放された後にその領域へポインタを通じて値を代入	○	×	×
サブルーチンから復帰後にグローバルポインタを通じてローカル配列へ値を代入	○	×	×

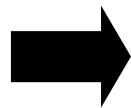
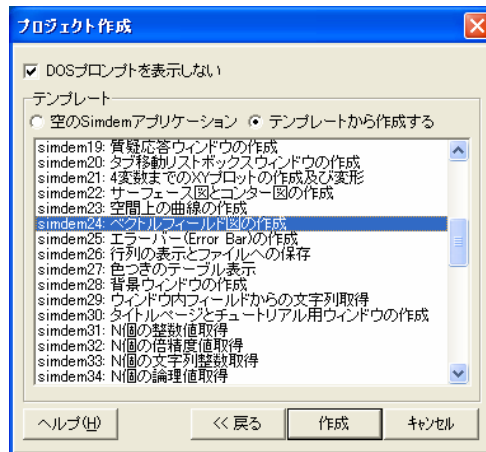
# Fortran 90/95 学習機能

- 法政大学 竹内則雄 教授 監修
- e-Book でのレッスン形式で、基礎から Fortran90/95 を体系的に習得できます
- 各レッスンのサンプルプログラムや演習問題の回答結果は Fortran Builder から実際に実行が可能であり、体験的な学習が可能です

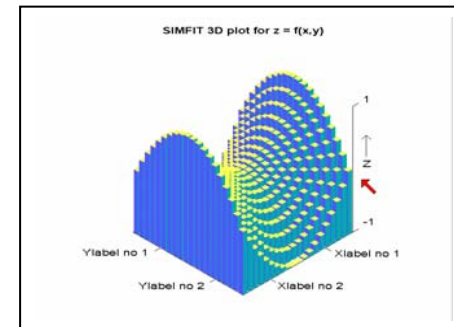


# プロットライブラリに「Simdem」を搭載

- ・ グラフィカルな表現を可能にするプロットライブラリ
- ・ **Fortran** のプログラムから簡単に GUI を実装できます
- ・ **Fortran** のプログラムから色々な種類のグラフが簡単に描けます

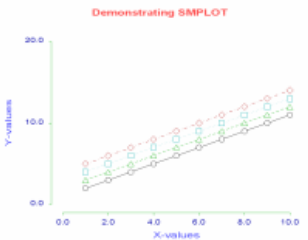
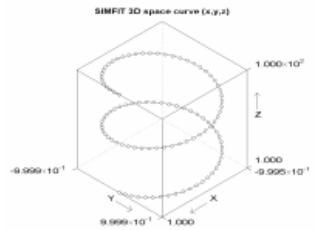
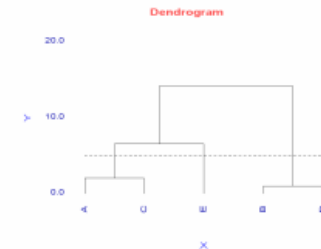
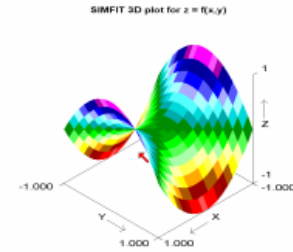
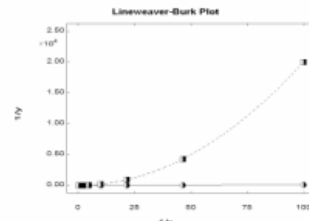
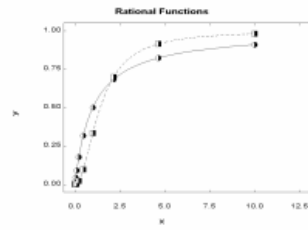
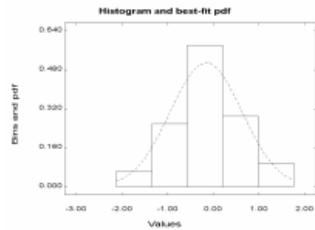


簡単な Fortran  
のプログラム

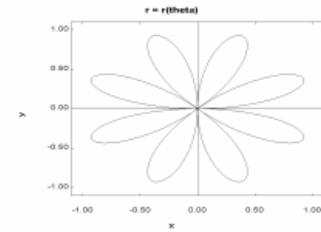
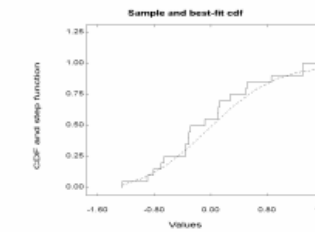
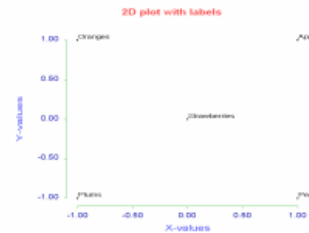
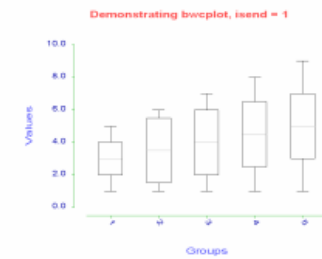
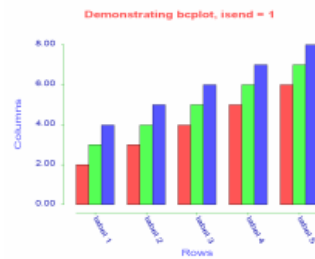
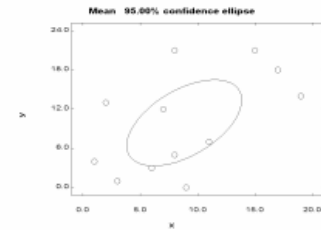
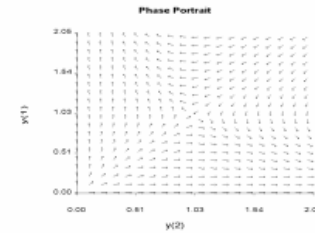


ROW/COLUMN	VARIABLE 1	VARIABLE 2
CASE 1	8.32042E+06	
CASE 2		

# 「Simdem」プロットイングイメージ

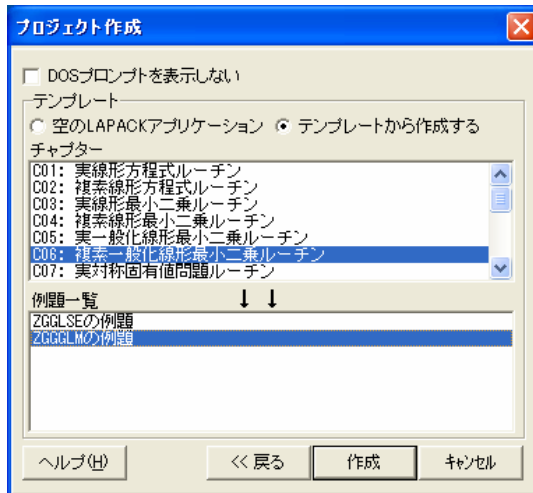


Demonstrating pcpplot, isend = 1



# 数値計算ライブラリ LAPACK がすぐ使える

- 数値計算ライブラリに **Lapack** を標準搭載
- **Fortran Builder** から簡単に利用可能です
- 各関数に対して日本語解説のヘルプを搭載



Next | Up | Previous

次へ: 複素一般化線形最小二乗 上へ: 実一般化線形最小二乗 前へ: CGGLSEの例題

### DGGGLMの例題

加重線形最小二乗問題を解きます。

$$\min_x \|B^{-1}(d - Ax)\|_2$$

ここで

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 \\ -1.93 & 1.08 & -0.31 \\ 2.30 & 0.24 & -0.40 \\ -0.02 & 1.03 & -1.43 \end{pmatrix}, B = \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 2.0 & 0 \\ 0 & 0 & 0 & 5.0 \end{pmatrix} \text{ and } d = \begin{pmatrix} 1.32 \\ -4.00 \\ 5.52 \\ 3.24 \end{pmatrix}$$

本例プログラム  
本例用データ  
例題概要

```

      J, M, N, RANK
      OUT=6)
      , NLVL, NMAX, SMLSIZ
      NB=64, NLVL=10, NMAX=16, SMLSIZ=25)
      ORK, LRWORK, LWORK
      X, LIWORK=3*MMAX*NLVL+11*MMAX,
      0*MMAX+2*MMAX*SMLSIZ+8*MMAX*NLVL+
      SMLSIZ+1)**2, LWORK=2*MMAX+NB*(MMA

```

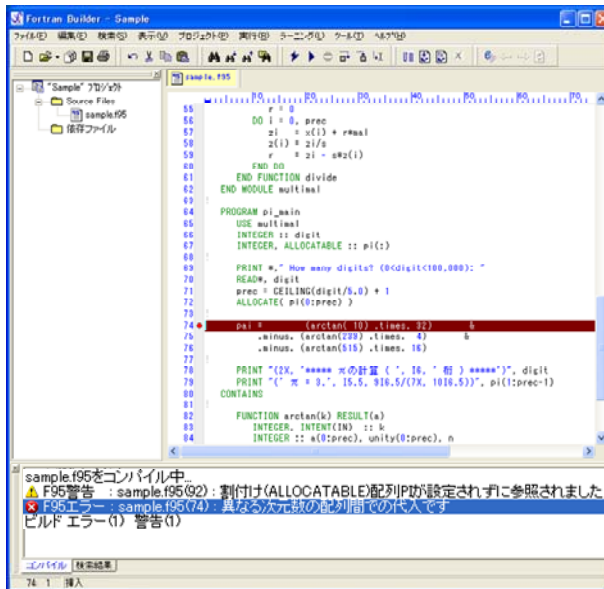
```

16 * .. Local Arrays ..
17 * COMPLEX *16 A(LDA,NMAX), B(NMAX), WORK(LWORK)
18 * DOUBLE PRECISION RWORK(LRWORK), S(NMAX)
19 * INTEGER IWORK(LIWORK)
20 * .. External Subroutines ..
21 * EXTERNAL ZGELSD
22 * .. Executable Statements ..
23 * WRITE (NOUT,*) 'ZGELSD Example Program Results'
24 * WRITE (NOUT,*)

```

# 日本語サポート

- 開発は日本で行われてます
- エラーメッセージ、マニュアル、メニューなど完全日本語対応





# デバッグ機能

- ・ プログラムの作成を支援するデバッグ機能を提供
- ・ ブレークポイントを設定した行で動作を停止できます
- ・ ステップ実行で一ステップずつ動作を確認できます
- ・ 変数値表示で変数の変化の状況を見ることができます

```
35 CALL DGBSV(N, KL, KU, 1, AB, LDAB, IPIV, B, N, INFO)
36 *
37 * IF (INFO.EQ.0) THEN
38 *
39 * Print so
40 *
41 * WRITE (N
42 * WRITE (N
43 *
44 * Print de
45 *
46 * WRITE (NOU1,*)
47 * IFAIL = 0
48 * CALL X04CEF(N, N, KL, KL+KU, AB, LDAB, 'Details of factorization',
49 * IFAIL)
```

変数ウォッチング	
Name	Value
Integer : INFO	
INFO	0
有効	表示番号 = 1

変数ウォッチング	
Name	Value
Integer : N	
N	4
有効	表示番号 = 2



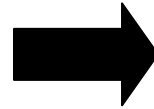
ブレークポイントの設定/解除(B)

ステップ実行(S)	F8
トレース実行(T)	F7
カーソル位置まで実行	F4
変数ウォッチング	F9

# Fortran77 から Fortran95 への変換機能

- Fortran77 (固定形式) のソースを Fortran95 (自由形式) のソースに変換します

Fortran77



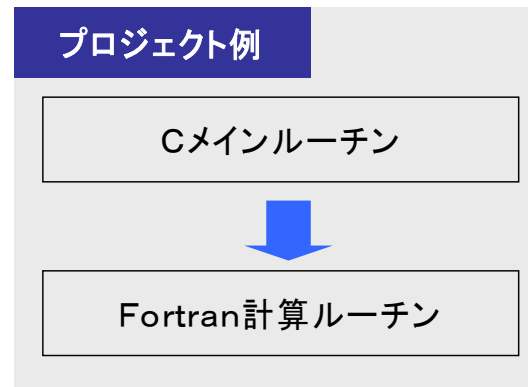
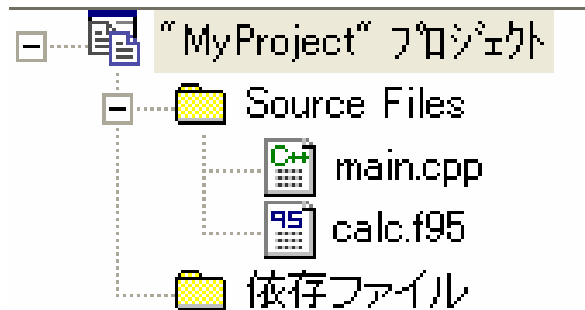
Fortran95

```
1 DGEEVX Example Program Text
2 NAG Copyright 2005.
3 .. Parameters ..
4 INTEGER NIN, NOUT
5 PARAMETER (NIN=5, NOUT=6)
6 INTEGER NB, NMAX
7 PARAMETER (NB=64, NMAX=10)
8 INTEGER LDA, LDVL, LDVR, LWORK
9 PARAMETER (LDA=NMAX, LDVL=NMAX, LDVR=NMAX, LWORK=(2+NB)*NMAX)
10 .. Local Scalars ..
11 COMPLEX *16 EIG
12 DOUBLE PRECISION ABNRM, EPS, ERBND, RCND, TOL
13 INTEGER I, IHI, ILO, INFO, J, LWKOPT, N
14 .. Local Arrays ..
15 DOUBLE PRECISION A(LDA,NMAX), RCONDE(NMAX), RCONDV(NMAX),
16 SCALE(NMAX), VL(LDVL,NMAX), VR(LDVR,NMAX),
17 WI(NMAX), WORK(LWORK), WR(NMAX)
18 INTEGER IWORK(2*NMAX-2)
19 .. External Functions ..
20 DOUBLE PRECISION DLAMCH
21 EXTERNAL DLAMCH
22 .. External Subroutines ..
23 EXTERNAL DGEEVX
24 .. Intrinsic Functions ..
25 INTRINSIC DCMPX
26 .. Executable Statements ..
27 WRITE (NOUT,*) 'DGEEVX Example Program Results'
28 Skip heading in data file
29 READ (NIN,*)
30 READ (NIN,*) N
31 IF (N.LE.NMAX) THEN
```

```
1 DGEEVX Example Program Text
2 NAG Copyright 2005.
3 .. Parameters ..
4 INTEGER nin, nout
5 PARAMETER (nin=5, nout=6)
6 INTEGER nb, nmax
7 PARAMETER (nb=64, nmax=10)
8 INTEGER lda, ldvl, ldvr, lwork
9 PARAMETER (lda=nmax, ldvl=nmax, ldvr=nmax, lwork=(2+nb)*nmax)
10 .. Local Scalars ..
11 COMPLEX*16 eig
12 DOUBLE PRECISION abnrm, eps, erbnd, rcnd, tol
13 INTEGER i, ihi, ilo, info, j, lwkopt, n
14 .. Local Arrays ..
15 DOUBLE PRECISION a(lda,nmax), rconde(nmax), rcondv(nmax), scale(nmax) &
16 vl(ldvl,nmax), vr(ldvr,nmax), wi(nmax), work(lwork), wr(nmax)
17 INTEGER iwork(2*nmax-2)
18 .. External Functions ..
19 DOUBLE PRECISION dlamch
20 EXTERNAL dlamch
21 .. External Subroutines ..
22 EXTERNAL dgeevx
23 .. Intrinsic Functions ..
24 INTRINSIC dcmplx
25 .. Executable Statements ..
26 WRITE (nout,*) 'DGEEVX Example Program Results'
27 Skip heading in data file
28 READ (nin,*)
29 READ (nin,*) n
30 IF (n<=nmax) THEN
```

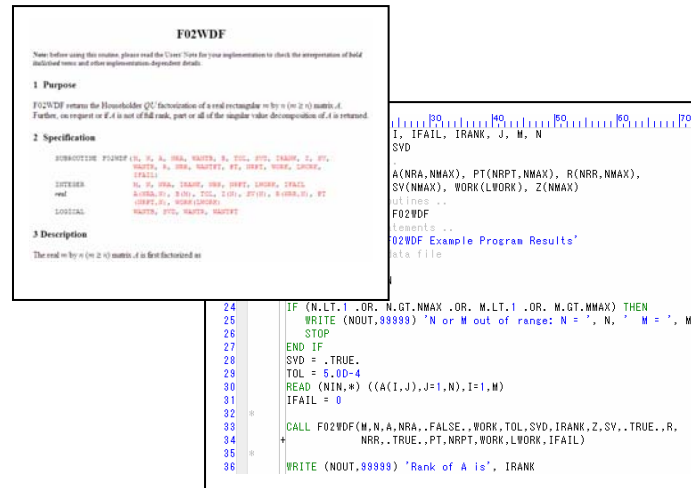
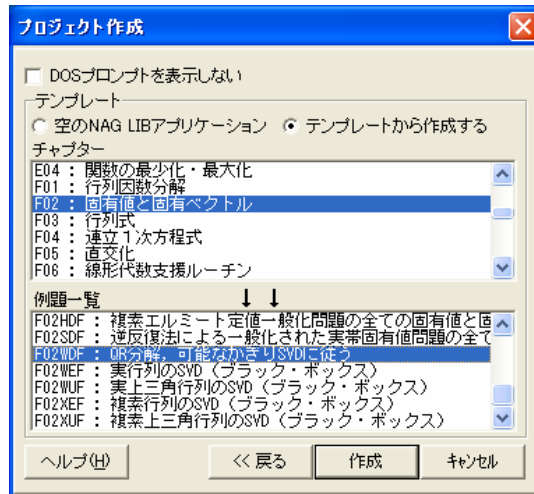
# Fortran77/90/95/C/C++ 混在開発環境

- C/C++/Fortran を一つのプロジェクトの中に混在させた形で開発が可能です
- C/C++/Fortran それぞれの言語の得意の分野を組み合わせることでプログラムをすることができます



# 「NAG Fortran Library」 (体験版) 付属

- 高い信頼性、計算精度、スピードを持つ、世界標準ともいえる数値計算ライブラリです
- Fortran Builder から簡単に利用可能です
- ライブラリの全ての機能を、30日間ご試用頂けます



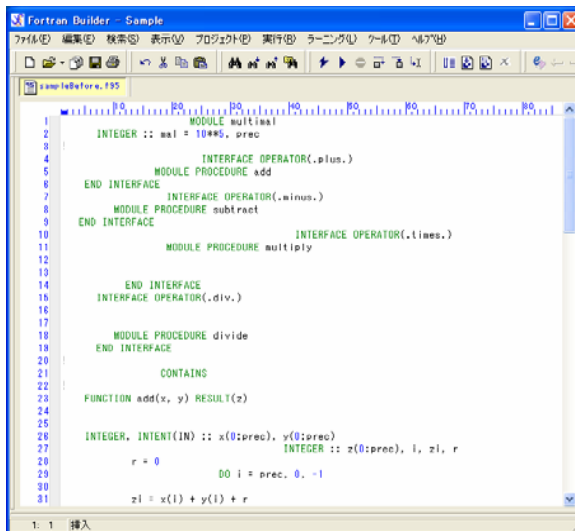
# 「NAG Fortran Library」 ご提供機能

- ・ **特殊関数**  
双曲線関数 1、ガンマ関数、誤差関数、ベッセル関数、フレネル関数、楕円積分、楕円関数、エアリー関数、ケルビン関数、エラー関数、Hankel 関数
- ・ **行列、ベクトル操作**  
逆行列、疎行列ユーティリティ
- ・ **線形方程式**  
一般連立線形方程式、対称連立方程式、三角連立方程式、一般帯連立方程式、対象帯連立方程式、LU 分解、コレスキー分解、疎行列連立方程式
- ・ **固有値問題**  
固有値、固有ベクトル、シュール分解
- ・ **特異値分解 (SVD)**
- ・ **最小二乗問題**
- ・ **FFT**
- ・ **畳み込み**
- ・ **曲線、曲面フィテティング、補間**  
エルミート補間、1次元スプラインフィット、2次元スプラインフィット、修正シェパード法、チェビシェフ級数
- ・ **最適化**  
線形計画法(LP)、2次計画法(QP)、非線形最小二乗法、非線形計画法、1変量最小化
- ・ **非線形方程式**  
多項式の根、非線形方程式の根、連立方程式の根
- ・ **求積**  
有限区間の数値積分、無限区間の数値積分、多次元積分
- ・ **積分方程式**  
線形フレッドホルム積分方程式、非線形ヴォルテラ畳み込み方程式、アーベル型方程式
- ・ **常微分方程式**  
ルンゲクッタ、初期値問題、アダムス、BDF、境界値問題
- ・ **偏微分方程式**  
ヘルムホルツ方程式(Helmholtz)、マルチグリッド、楕円微分方程式、放物型偏微分方程式、ブラックショールズ(Black Scholes)、Bond
- ・ **メッシュ生成**  
反復法、Delaunay、Advancing-Front
- ・ **オペレーションズリサーチ(OR)**  
整数計画、最短経路問題
- ・ **統計分散関数 (偏差、確率)**  
正規分布、学生T分布、 $\chi^2$  二乗分布 (カイ二乗分布)、F 分布、ベータ分布、ガンマ分布、離散分布
- ・ **乱数発生**  
準乱数、一様分布、正規分布、多変量正規分布、ベータ分布、指数分布、ガンマ分布、2項分布、超幾何分布、フォン・ミゼス分布、離散分布
- ・ **1変量推定**  
2項分布信頼区間、ポアソン分布信頼区間、ワイブル分布信頼区間、ロバスト推定
- ・ **回帰分析**  
線形回帰分析、多重線形回帰分析
- ・ **相関分析**  
ピアソン積率相関係数、共分散行列、偏相関行列、偏共分散行列
- ・ **多変量解析**  
因子分析、主成分分析、正準分析、クラスタ分析、判別分析
- ・ **一般化線形モデル (GLM)**
- ・ **分散分析(ANOVA)**
- ・ **時系列分析**  
ARIMA モデルフィット、ARMA モデルフィット、予測、伝達関数、スペクトル解析、ACF、PACF
- ・ **生存解析**  
 Kaplan-Meier 推定値、Cox のハザード・モデル、危険集合
- ・ **ノンパラメトリック統計**  
コックススチュアート検定、ウィルコクソン検定、ラン検定、マクネマー検定、マンホイットニー検定、フリードマン検定、クラスカルウォリス検定、コ克蘭 Q 検定、コルモゴロフスミルノフ検定、ケンドールの合致係数、ケンドールの階数相関

# Fortran95 ソース整形機能

- Fortran95 ソースを読みやすい形に整形してファイル出力します
- Fortran77 で作成した過去の資産を Fortran95 で活用できます

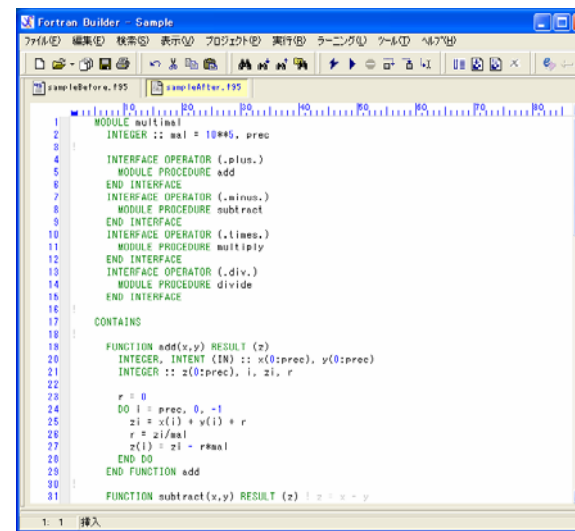
整形前



```
1  MODULE multimal
2
3  INTEGER :: mal = 10**5, prec
4
5  INTERFACE OPERATOR(.plus.)
6  MODULE PROCEDURE add
7  END INTERFACE
8  INTERFACE OPERATOR(.minus.)
9  MODULE PROCEDURE subtract
10 END INTERFACE
11 INTERFACE OPERATOR(.times.)
12 MODULE PROCEDURE multiply
13 END INTERFACE
14 INTERFACE OPERATOR(.div.)
15 MODULE PROCEDURE divide
16 END INTERFACE
17 CONTAINS
18 FUNCTION add(x, y) RESULT(z)
19
20 INTEGER, INTENT(IN) :: x(0:prec), y(0:prec)
21 INTEGER :: z(0:prec), i, zi, r
22
23 r = 0
24 DO i = prec, 0, -1
25   zi = x(i) + y(i) + r
26   r = zi/mal
27   z(i) = zi - r*mal
28 END DO
29 END FUNCTION add
30
31 FUNCTION subtract(x,y) RESULT(z) ! z = x - y
```



整形後



```
1  MODULE multimal
2
3  INTEGER :: mal = 10**5, prec
4
5  INTERFACE OPERATOR (.plus.)
6  MODULE PROCEDURE add
7  END INTERFACE
8  INTERFACE OPERATOR (.minus.)
9  MODULE PROCEDURE subtract
10 END INTERFACE
11 INTERFACE OPERATOR (.times.)
12 MODULE PROCEDURE multiply
13 END INTERFACE
14 INTERFACE OPERATOR (.div.)
15 MODULE PROCEDURE divide
16 END INTERFACE
17 CONTAINS
18 FUNCTION add(x,y) RESULT (z)
19   INTEGER, INTENT (IN) :: x(0:prec), y(0:prec)
20   INTEGER :: z(0:prec), i, zi, r
21
22   r = 0
23   DO i = prec, 0, -1
24     zi = x(i) + y(i) + r
25     r = zi/mal
26     z(i) = zi - r*mal
27   END DO
28 END FUNCTION add
29
30 FUNCTION subtract(x,y) RESULT (z) ! z = x - y
```

## その他の機能

- DLL 作成機能
  - ⇒ 簡単に DLL を作成でき、C/C++, JAVA などから呼び出す事が可能になります
- 入力補完機能
  - ⇒ Fortran のキーワードの補完、使用変数や構造体のメンバーの補完などが行えます
- コールグラフ機能
  - ⇒ F95 ソースのルーチン間の呼び出し関係を解析し、ファイル出力します
- 壁紙機能
  - ⇒ エディタ編集画面に好きな画像を背景に配置する事ができます

# ライセンス形態

Fortran Builder につきましては、以下の3種類のライセンス形態を用意しています

- ノードロックライセンス  
⇒ ライセンス登録をした PC（一台のみ）で Fortran Builder をご利用頂けます
- USB ユーザロックライセンス  
⇒ Fortran Builder をインストールした PC（複数可）に USB キーを差し込むことで Fortran Builder をご利用頂けます
- フローティングライセンス  
⇒ 同一ネットワーク内の PC に対して、ご購入頂いたライセンス数の方が、同時にご利用頂けます



## まとめ

- Fortran 言語が分かりやすい形で提供される事によって、Fortran 言語の利用者が増える事を期待して本製品は製作されています
- 学術分野、ソフトウェア開発の現場や情報処理教育において、本製品の活躍の場が広がると考えています