NAG Fortran Library, Mark 24
FLMI624D9L - License Managed
Intel-based Mac, Mac OS X 64, NAG Fortran, Double Precision

ユーザーノート

内容

1. イントロダクション 1

2. リリース後の最新情報 1

3. 一般情報 2
   3.1. ライブラリのリンク方法 2
      3.1.1. C または C++ からライブラリを呼び出す場合 4
   3.2. インターフェースブロック 5
   3.3. Example プログラム 6
   3.4. Fortran 型と強調斜体文字の解釈 7
   3.5. NAG ルーチンからの出力 8

4. ルーチン固有の情報 9

5. ドキュメント 13

6. サポート 14

7. ユーザーフィードバック 15

追記 ー コンタクト先情報 15

1. イントロダクション

本ユーザーノートは NAG Fortran Library, Mark 24：FLMI624D9L（ライブラリ）を使用される方向けのドキュメントです．本ユーザーノートには NAG Library Manual, Mark 24（ライブラリマニュアル）に含まれない製品毎の情報が含まれます．ライブラリマニュアルに「ユーザーノート参照」などと書かれている場合は，本ユーザーノートをご参照ください．

ライブラリルーチンのご利用にあたり，以下のドキュメントを必ずお読みください．

(a) Essential Introduction （ライブラリについての基本的なドキュメント）
(b) Chapter Introduction （チャプター毎のドキュメント）
(c) Routine Document （ルーチン毎のドキュメント）

本ライブラリはマルチスレッド環境でご利用いただけます（スレッドセーフです）．

2. リリース後の最新情報

本ライブラリの動作環境やご利用方法についての最新の情報は，以下のウェブページをご確認ください．

http://www.nag.co.uk/doc/inun/fl24/mi6d9l/postrelease.html

3. 一般情報

パフォーマンスの面からは，Vector Library（vecLib）を利用するバージョンの NAG ライブラリ libnag_vl.a, libnag_vl.dylib のご利用を推奨します．もし本製品の製造に使用された vecLib とは異なるバージョンの vecLib を用いて問題が生じる場合は，vecLib を利用しないバージョンの NAG ライブラリ libnag_nag.a, libnag_nag.dylib をご利用ください．

3.1. ライブラリのリンク方法

本セクションでは［INSTALL_DIR］に本ライブラリがインストールされていることが前提となります．

デフォルトの［INSTALL_DIR］は /opt/NAG/flmi624d9l となります．
また，インストール時に［INSTALL_DIR］を指定することもできます．

- /opt/NAG/flmi624d9l が在れば，これが［INSTALL_DIR］になります．
- または，インストール時に指定したインストール先が［INSTALL_DIR］になります．

vecLib を利用するバージョンの NAG ライブラリを利用する場合は，以下のようにリンクを行ってください．（ここで driver.f90 がユーザープログラムです．）

スタティックライブラリを利用する場合：

```
nagfor -I[INSTALL_DIR]/nag_interface_blocks driver.f90 \
    [INSTALL_DIR]/lib/libnag_vl.a -framework vecLib \
    -framework IOKit -framework CoreFoundation
```

共有ライブラリを利用する場合：

```
nagfor -I[INSTALL_DIR]/nag_interface_blocks driver.f90 \
    [INSTALL_DIR]/lib/libnag_vl.dylib -framework vecLib \
    -framework IOKit -framework CoreFoundation
```

注意：本ライブラリ（libnag_vl.a と libnag_vl.dylib）は vecLib の version 3.6 には対応していません．

vecLib を利用しないバージョンの NAG ライブラリを利用する場合は，以下のようにリンクを行ってください.（ここで driver.f90 がユーザープログラムです.）

スタティックライブラリを利用する場合：

```
nagfor -I[INSTALL_DIR]/nag_interface_blocks driver.f90 \
    [INSTALL_DIR]/lib/libnag_nag.a \
    -framework IOKit -framework CoreFoundation
```

共有ライブラリを利用する場合：

```
nagfor -I[INSTALL_DIR]/nag_interface_blocks driver.f90 \
    [INSTALL_DIR]/lib/libnag_nag.dylib \
    -framework IOKit -framework CoreFoundation
```

共有ライブラリを利用する場合には，環境変数 DYLD_LIBRARY_PATH を正しく設定し，実行時のリンクが行えるようにしてください.

C シェルの場合：

```
setenv DYLD_LIBRARY_PATH [INSTALL_DIR]/lib
```

または，既存の設定がある場合には次のように拡張します.

```
setenv DYLD_LIBRARY_PATH [INSTALL_DIR]/lib:${DYLD_LIBRARY_PATH}
```

Bourne シェルの場合：

```
DYLD_LIBRARY_PATH=[INSTALL_DIR]/lib
export DYLD_LIBRARY_PATH
```

または，既存の設定がある場合には次のように拡張します.

```
DYLD_LIBRARY_PATH=[INSTALL_DIR]/lib:${DYLD_LIBRARY_PATH}
export DYLD_LIBRARY_PATH
```

注意：ご利用の環境に依っては，その他のパスも DYLD_LIBRARY_PATH に設定する必要があるかもしれません.（例えば，より新しいバージョンのコンパイラでは，コンパイラの実行時ライブラリのパスが必要となるかもしれません.）

3.1.1. C または C++ からライブラリを呼び出す場合

本ライブラリは C または C++ 環境からもご利用いただけます.
ご利用の支援として Fortran と C の間の型マッピング情報を持った C/C++ ヘッダーファイル [INSTALL_DIR]/c_headers/nagmk24.h が提供されます. ヘッダーファイルから必要な部分だけを（ファイルの先頭にある #defines なども忘れずに）自身のプログラムにコピー＆ペーストするか，もしくはヘッダーファイルを単純にインクルードしてご利用ください.

C または C++ から NAG Fortran Library を呼び出す際のアドバイスは，ドキュメント [INSTALL_DIR]/c_headers/techdoc.html をご参照ください.

3.2. インターフェースブロック

NAG Fortran Library インターフェースブロック（引用仕様宣言）はライブラリルーチンの型と引数を定義します. Fortran プログラムからライブラリルーチンを呼び出す際に必ず必要という性質のものではありませんが（ただし本製品で提供される Example を利用する際には必要となります），これを用いることでライブラリルーチンが正しく呼び出されているかどうかのチェックを Fortran コンパイラに任せることができます．具体的にはコンパイラが以下のチェックを行うことを可能とします.

(a) サブルーチン呼び出しの整合性
(b) 関数宣言の型
(c) 引数の数
(d) 引数の型

NAG Fortran Library インターフェースブロックファイルはチャプター毎のモジュールとして提供されますが，これらをまとめて一つにしたモジュールが提供されます.

nag_library

モジュールファイルはプリコンパイルされた形式（.mod ファイル）で提供されます．コンパイル時に -I"pathname" オプションを用いて，モジュールファイルが置かれているディレクトリのパス（[INSTALL_DIR]/nag_interface_blocks）を指定してください.

提供されるモジュールファイル（.mod ファイル）は，インストールノートの「2.1. 動作環境」に記載されているコンパイラを用いて生成されています．モジュールファイルはコンパイラ依存のファイルであるため，ご利用のコンパイラと互換性がない場合は，ご利用のコンパイラでモジュールファイルを生成する必要があります.（自身のプログラムでインターフェースブロックをご利用にならないのであれば，この必要はありません. ただし，Example プログラムはインターフェースブロックを利用しますので，Example プログラムをご利用になる場合は必要です.）

更なる情報は，以下のウェブページ（リリース後の最新情報）をご確認ください.

http://www.nag.co.uk/doc/inun/fl24/mi6d9l/postrelease.html

3.3. Example プログラム

提供される Example 結果は，インストールノートの「2.2. 開発環境」に記載されている環境で生成されています．Example プログラムの実行結果は，異なる環境下（例えば，異なる Fortran コンパイラ，異なるコンパイラライブラリ，異なる BLAS または LAPACK ルーチンなど）で若干異なる場合があります．そのような違いが顕著な計算結果としては，固有ベクトル（スカラー（多くの場合 -1 ）倍の違い），反復回数や関数評価，残差（その他マシン精度と同じくらい小さい量）などがあげられます．

提供される Example 結果は NAG スタティックライブラリ libnag_vl.a（vecLib の BLAS／LAPACK ルーチンを使用）を用いて算出されています．

Example プログラムは本ライブラリが想定する動作環境に適した状態で提供されます．そのため，ライブラリマニュアルに記載／提供されている Example プログラムに比べて，その内容が若干異なる場合があります．

[INSTALL_DIR]/scripts には nag_example_vl，nag_example_shar_vl，nag_example，nag_example_shar の４つのスクリプトが提供されます．

これらのスクリプトを用いて Example プログラムを簡単に利用することができます．

- nag_example_vl
  NAG スタティックライブラリ libnag_vl.a および vecLib ライブラリをリンクします．
- nag_example_shar_vl
  NAG 共有ライブラリ libnag_vl.dylib および vecLib ライブラリをリンクします．
- nag_example
  NAG スタティックライブラリ libnag_nag.a をリンクします．
- nag_example_shar
  NAG 共有ライブラリ libnag_nag.dylib をリンクします．

これらのスクリプトは，Example プログラムのソースファイル（必要に応じて，データファイル，オプションファイルその他）をカレントディレクトリにコピーして，コンパイル／リンク／実行を行います．

ご利用の NAG ライブラリルーチンの名前をスクリプトの引数に指定してください.
例）
nag_example_vl e04nrf

この例では，e04nrfe.f（ソースファイル），e04nrfe.d（データファイル），e04nrfe.opt（オプションファイル）をカレントディレクトリにコピーして，コンパイル／リンク／実行を行い e04nrfe.r（結果ファイル）を生成します.

注意：本ライブラリ（libnag_vl.a と libnag_vl.dylib）は vecLib の version 3.6 には対応していません．従って，nag_example_vl と nag_example_shar_vl は，vecLib の version 3.6 と共にご利用いただけません．ただし，nag_example と nag_example_shar は vecLib の有無に関係なくご利用いただけます.

3.4. Fortran 型と強調斜体文字の解釈

ライブラリとライブラリマニュアルでは浮動小数点変数を以下のようにパラメーター化された型を用いて記述しています.

REAL(KIND=nag_wp)

ここで nag_wp は Fortran の種別パラメーターを表しています.
nag_wp の値は製品毎に異なり，その値は nag_library モジュールに定義されています.

これに加え，いくつかのルーチンで以下の型が使用されます.

REAL(KIND=nag_rp)

これらの型の使用例については各種 Example プログラムをご参照ください.

本ライブラリでは，これらの型は次のような意味を持っています.

REAL (kind=nag_rp)　　- REAL（単精度実数）
REAL (kind=nag_wp)　　- DOUBLE PRECISION
COMPLEX (kind=nag_rp) - COMPLEX（単精度複素数）
COMPLEX (kind=nag_wp) - 倍精度複素数（例えば COMPLEX*16）

上記に加え，ライブラリマニュアルでは強調斜体文字を用いていくつかの用語を表現しています.

一つ重要なものは ***machine precision*** という表現で，これは DOUBLE PRECISION 浮動小数が計算機内で格納されている相対精度を意味します．例えば 10 進で約 16 桁の実装であれば ***machine precision*** は 1.0D-16 に近い値を持ちます.

machine precision の正確な値はルーチン X02AJF を使って確認できます.
チャプター X02 のその他のルーチンを使うと，オーバーフロー用の閾値や表現可能な最大整数といった実装異存の定数値を求めることができます.
詳細については X02 Chapter Introduction をご参照ください.

brock size という表現はチャプター F07 と F08 で用いられます．これは，ブロックアルゴリズムで用いられるブロックサイズを表すものです．用意すべき作業エリアの量に影響が及ぶ場合にのみ，この値に留意する必要があります．関係する Routine Document と Chapter Introduction に記述されているパラメーター WORK と LWORK についてご参照ください.

3.5. NAG ルーチンからの出力

いくつかのルーチンはエラーメッセージやアドバイスメッセージを出力します．出力装置番号は X04AAF（エラーメッセージの場合）または X04ABF（アドバイスメッセージの場合）で再設定することが可能です．デフォルト値は「4. ルーチン固有の情報」をご参照ください.

4. ルーチン固有の情報

本ライブラリルーチン固有の情報を（チャプター毎に）以下に示します.

a. F06, F07, F08, F16

チャプター F06, F07, F08, F16 においては BLAS／LAPACK 由来のルーチンに対して別個のルーチン名が用意されています.
これらのルーチン名については，関係する Chapter Introduction をご参照ください.
パフォーマンスの面からは，NAG スタイルの名前よりも BLAS／LAPACK スタイルの名前でルーチンを使用してください.

多くの LAPACK ルーチンは“workspace query”メカニズムを利用します．ルーチン呼び出し側にどれだけのワークスペースが必要であるかを問い合わせるメカニズムですが，NAG 提供の LAPACK と vecLib 提供の LAPACK ではこのワークスペースサイズが異なる場合がありますのでご注意ください.

libnag_vl.a と libnag_vl.dylib では，ベンダー提供の BLAS／LAPACK の問題を回避するために，以下の BLAS／LAPACK ルーチンについては NAG 提供のものが含まれています（呼び出されます）.

daxpyi	ddoti	dgthr	dgthrz	droti	dsctr
zaxpyi	zdotci	zdotui	zgthr	zgthrz	zsctr
dnrm2	dtpsv	dcopy	zdotc	zdotu	sasum
sdot	scasum	dgeesx	dgges	dggesx	dgerfs
dtrsyl	dgeqrf	dsgesv	zcgesv	dgees	dgesdd
dgesv	dsyevr	zheevr	dstevr	dstegr	zstegr

b. G02

このチャプターで出てくる ACC の値（マシン依存の定数）は 1.0D-13 です.

c. S07 - S21

これらのチャプターの関数の動作は，ライブラリ実装毎に異なります.

一般的な詳細はライブラリマニュアルをご参照ください.
本ライブラリ固有の値を以下に示します.

```
S07AAF  F_1 = 1.0E+13
        F_2 = 1.0E-14

S10AAF  E_1 = 1.8715E+1
S10ABF  E_1 = 7.080E+2
S10ACF  E_1 = 7.080E+2

S13AAF  x_hi = 7.083E+2
S13ACF  x_hi = 1.0E+16
S13ADF  x_hi = 1.0E+17

S14AAF  IFAIL = 1 if X > 1.70E+2
        IFAIL = 2 if X < -1.70E+2
        IFAIL = 3 if abs(X) < 2.23E-308
S14ABF  IFAIL = 2 if X > x_big = 2.55E+305

S15ADF  x_hi = 2.65E+1
S15AEF  x_hi = 2.65E+1
S15AFF  underflow trap was necessary
S15AGF  IFAIL = 1 if X >= 2.53E+307
        IFAIL = 2 if 4.74E+7 <= X < 2.53E+307
        IFAIL = 3 if X < -2.66E+1

S17ACF  IFAIL = 1 if X > 1.0E+16
S17ADF  IFAIL = 1 if X > 1.0E+16
        IFAIL = 3 if 0 < X <= 2.23E-308
S17AEF  IFAIL = 1 if abs(X) > 1.0E+16
S17AFF  IFAIL = 1 if abs(X) > 1.0E+16
S17AGF  IFAIL = 1 if X > 1.038E+2
        IFAIL = 2 if X < -5.7E+10
S17AHF  IFAIL = 1 if X > 1.041E+2
        IFAIL = 2 if X < -5.7E+10
S17AJF  IFAIL = 1 if X > 1.041E+2
```

```
        IFAIL = 2 if X < -1.9E+9
S17AKF  IFAIL = 1 if X > 1.041E+2
        IFAIL = 2 if X < -1.9E+9
S17DCF  IFAIL = 2 if abs(Z) < 3.92223E-305
        IFAIL = 4 if abs(Z) or FNU+N-1 > 3.27679E+4
        IFAIL = 5 if abs(Z) or FNU+N-1 > 1.07374E+9
S17DEF  IFAIL = 2 if AIMAG(Z) > 7.00921E+2
        IFAIL = 3 if abs(Z) or FNU+N-1 > 3.27679E+4
        IFAIL = 4 if abs(Z) or FNU+N-1 > 1.07374E+9
S17DGF  IFAIL = 3 if abs(Z) > 1.02399E+3
        IFAIL = 4 if abs(Z) > 1.04857E+6
S17DHF  IFAIL = 3 if abs(Z) > 1.02399E+3
        IFAIL = 4 if abs(Z) > 1.04857E+6
S17DLF  IFAIL = 2 if abs(Z) < 3.92223E-305
        IFAIL = 4 if abs(Z) or FNU+N-1 > 3.27679E+4
        IFAIL = 5 if abs(Z) or FNU+N-1 > 1.07374E+9

S18ADF  IFAIL = 2 if 0 < X <= 2.23E-308
S18AEF  IFAIL = 1 if abs(X) > 7.116E+2
S18AFF  IFAIL = 1 if abs(X) > 7.116E+2
S18DCF  IFAIL = 2 if abs(Z) < 3.92223E-305
        IFAIL = 4 if abs(Z) or FNU+N-1 > 3.27679E+4
        IFAIL = 5 if abs(Z) or FNU+N-1 > 1.07374E+9
S18DEF  IFAIL = 2 if REAL(Z) > 7.00921E+2
        IFAIL = 3 if abs(Z) or FNU+N-1 > 3.27679E+4
        IFAIL = 4 if abs(Z) or FNU+N-1 > 1.07374E+9

S19AAF  IFAIL = 1 if abs(X) >= 5.04818E+1
S19ABF  IFAIL = 1 if abs(X) >= 5.04818E+1
S19ACF  IFAIL = 1 if X > 9.9726E+2
S19ADF  IFAIL = 1 if X > 9.9726E+2

S21BCF  IFAIL = 3 if an argument < 1.583E-205
        IFAIL = 4 if an argument >= 3.765E+202
S21BDF  IFAIL = 3 if an argument < 2.813E-103
        IFAIL = 4 if an argument >= 1.407E+102
```

d. X01

数学定数は以下のとおりです.

X01AAF (pi)　　= 3.1415926535897932
X01ABF (gamma) = 0.5772156649015328

e. X02

マシン定数は以下のとおりです.

浮動小数点演算の基本的なパラメーター:

X02BHF = 2
X02BJF = 53
X02BKF = -1021
X02BLF = 1024

浮動小数点演算の派生的なパラメーター:

X02AJF = 1.11022302462516E-16
X02AKF = 2.22507385850721E-308
X02ALF = 1.79769313486231E+308
X02AMF = 2.22507385850721E-308
X02ANF = 2.22507385850721E-308

コンピューター環境のその他のパラメーター:

X02AHF = 1.42724769270596E+45
X02BBF = 2147483647
X02BEF = 15

f. X04

エラーメッセージおよびアドバイスメッセージのデフォルトの出力先装置番号は 6 番となります.

5. ドキュメント

ライブラリマニュアルは本製品の一部として提供されます.
また NAG のウェブサイトからダウンロードすることもできます.
ライブラリマニュアルの最新版は以下のウェブサイトをご参照ください.

http://www.nag.co.uk/numeric/FL/FLdocumentation.asp

ライブラリマニュアルは以下の形式で提供されます.

- HTML5 - HTML／MathML マニュアル（各ドキュメントの PDF 版へのリンクを含む）
- PDF - PDF マニュアル（PDF のしおり，または HTML 目次ファイルから閲覧する）

これらの形式に対して，以下の目次ファイルが提供されます.

nagdoc_fl24/html/FRONTMATTER/manconts.html
nagdoc_fl24/pdf/FRONTMATTER/manconts.pdf
nagdoc_fl24/pdf/FRONTMATTER/manconts.html

また，便利のために，これらの目次ファイルへのリンクをまとめたマスター目次ファイルが提供されます.

nagdoc_fl24/index.html

各形式の閲覧方法についての更なる詳細は“Online Documentation”ドキュメントをご参照ください.

加えて，以下のドキュメントが提供されます.

- in.html - インストールノート（英語版）
- un.html - ユーザーノート（英語版）

6. サポート

(a) ご質問等

保守サービスにご加入いただいているお客様は,
電子メールにて「日本 NAG ヘルプデスク」までお問い合わせください.
その際, ご利用の製品の製品コード（FLMI624D9L）および保守 ID をご明記いただきますようお願い致します. ご返答は, 平日 9：30～12:00, 13:00～17:30 に行わせていただきます. 何卒よろしくお願い致します.

日本 NAG ヘルプデスク

email: naghelp@nag-j.co.jp

(b) NAG のウェブサイト

NAG のウェブサイトでは製品およびサービスに関する情報を定期的に更新しています.

http://www.nag-j.co.jp/ （日本）
http://www.nag.co.uk/ （英国本社）
http://www.nag.com/ （米国）

7. ユーザーフィードバック

NAG ではユーザー様からのフィードバックをバージョンアップなどに活かして行きたいと考えています．フィードバックにご協力いただける場合は，下記のコンタクト先にご連絡ください．

コンタクト先情報

日本ニューメリカルアルゴリズムズグループ株式会社
（略称：日本 NAG）

〒104-0032
東京都中央区八丁堀 4-9-9 八丁堀フロンティアビル 2F

email: sales@nag-j.co.jp
Tel: 03-5542-6311
Fax: 03-5542-6312

※ 日本ニューメリカルアルゴリズムズグループ株式会社から提供されるサービス内容は（お問い合わせ先など）日本国内ユーザー様向けに独自のものとなっています．