

NAG Fortran Library, Mark 26
FLW6I26DEL - License Managed
Microsoft Windows, 64-bit, Intel Fortran, 32-bit integers

ユーザーノート

内容

1. イントロダクション	1
2. 追加情報	1
3. 一般情報	2
3.1. ライブラリのリンク方法	4
3.1.1. コマンドウィンドウ	6
3.1.2. Microsoft Visual Studio	8
3.1.3. モジュールファイルについて.....	12
3.1.4. Microsoft C/C++ または Intel C/C++	13
3.1.5. Fortran Builder	16
3.1.6. Microsoft C#	19
3.1.7. Microsoft Visual Basic for Application 7 / Excel	20
3.1.8. Microsoft Visual Basic .NET	22
3.1.9. その他の環境	25
3.2. インターフェースブロック	26
3.3. Example プログラム	28
3.4. Fortran 型と強調斜体文字の解釈	30
3.5. メンテナンスレベル	31
4. ルーチン固有の情報	32
5. ドキュメント	36
6. サポート	38
7. コンタクト情報	38

1. イントロダクション

本ユーザーノートは、NAG Fortran Library, Mark 26 – FLW6I26DEL (ライブラリ) のご利用方法 (リンク方法) を説明します。

本ユーザーノートには、NAG Library Manual, Mark 26 (ライブラリマニュアル) には含まれない製品毎の情報が含まれています。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際しては、ライブラリマニュアル (「5. ドキュメント」参照) の以下のドキュメントをお読みください。

- (a) How to Use the NAG Library and its Documentation
- (b) Chapter Introduction
- (c) Routine Document

2. 追加情報

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/fl26/w6idel/supplementary.html>

3. 一般情報

本ライブラリは、Intel® Math Kernel Library for Windows (MKL) が提供する BLAS/LAPACK ルーチンを利用するライブラリ (スタティック版と DLL 版) と、NAG が提供する BLAS/LAPACK ルーチンを利用するライブラリ (スタティック版と DLL 版) を提供します。

本ライブラリは、MKL version 11.3.3 を用いてテストされています。MKL version 11.3.3 は本製品の一部として提供されます。MKL の詳細については Intel 社のウェブサイト <https://software.intel.com/intel-mkl> をご参照ください。

パフォーマンスの面からは、MKL を利用するバージョンの NAG ライブラリ `nag_mkl_MT.lib`, `nag_mkl_MD.lib`, `FLW6I26DE_mkl.lib`/ `FLW6I26DE_mkl.dll` のご利用を推奨します。これらのライブラリは NAG が提供する BLAS/LAPACK ルーチンを含みません。

また、MKL を利用しないバージョンの NAG ライブラリ `nag_nag_MT.lib`, `nag_nag_MD.lib`, `FLW6I26DE_nag.lib`/ `FLW6I26DE_nag.dll` が提供されます。これらのライブラリは NAG が提供する BLAS/LAPACK ルーチンを含んでいます。

NAG ライブラリのスタティック版をご利用の場合は、共にリンクされる Microsoft ランタイムライブラリに従って、NAG ライブラリを選択する必要があります。マルチスレッドスタティックランタイムライブラリと共にリンクする場合は、`nag_mkl_MT.lib` または `nag_nag_MT.lib` をご利用ください。または、マルチスレッド DLL ランタイムライブラリと共にリンクする場合は、`nag_mkl_MD.lib` または `nag_nag_MD.lib` をご利用ください。

NAG ライブラリの DLL 版をご利用の場合は、インポートライブラリ `FLW6I26DE_mkl.lib` または `FLW6I26DE_nag.lib` をリンクしてください。実行時には、対応する DLL ファイル `FLW6I26DE_mkl.dll` または `FLW6I26DE_nag.dll` の格納フォルダーのパスが環境変数 `PATH` に設定されている必要があります。詳細は「3.1.1. コマンドウィンドウ」をご参照ください。

本製品で提供される MKL version 11.3.3 はマルチスレッド化されており、環境変数 `OMP_NUM_THREADS` が設定されていない場合、複数のプロセッサまたはマルチコアチップを持つシステムでは、計算速度の向上のためにマルチスレッドで計算を行います。もし、MKL に複数のプロセッサまたはコアを使わせたくない場合は、環境変数 `OMP_NUM_THREADS` に “1” (もしくは、必要なスレッド数) を設定してください。なお、X06 ルーチンは MKL のスレッドの振る舞いに影響を与えません。

MKLには、条件付きビット単位の再現性 (Bit-wise Reproducibility (BWR)) オプションがあります。一定の条件 (<https://software.intel.com/en-us/node/528579> 参照) をユーザーコードが満たしていれば、環境変数 MKL_CBWR を設定することにより BWR が有効になります。詳細は MKL のドキュメントをご参照ください。しかしながら、多くの NAG ルーチンはこれらの条件を満たしていません。従って、MKL を利用するバージョンの NAG ライブラリの全ルーチンに対して、異なる CPU アーキテクチャに渡り MKL_CBWR による BWR を保証することはできません。BWR に関するより一般的な情報は、“How to Use the NAG Library and its Documentation” ドキュメントの「3.11.1 Bit-wise Reproducibility (BWR)」をご参照ください。

本ライブラリは、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に適しています。

3.1. ライブラリのリンク方法

本ライブラリは、インストールノート「2.2. 開発環境」に記載されているバージョンの Intel Fortran コンパイラを用いてビルドされました。本ライブラリのビルドに用いた Intel Fortran コンパイラのランタイム DLL が `install_dir\rtl\bin` フォルダに提供されます。ご利用の Intel Fortran コンパイラのバージョンによっては、プログラムの実行時に、このフォルダにパスを通す必要があるかもしれません。

本セクションでは、以下のデフォルトのインストールフォルダに本ライブラリがインストールされていることを前提とします。

```
C:\Program Files\NAG\FL26\flw6i26del
```

もし、このフォルダが存在しない場合は、システム管理者（本ライブラリをインストールされた方）にお尋ねください。以降の説明ではこのフォルダを `install_dir` として参照します。

また、スタートメニューの NAG Fortran Library (FLW6I26DEL) に以下のライブラリコマンドプロンプトのショートカットが置かれていることを前提とします。

```
NAG FLW6I26DEL Command Prompt
```

もし、このショートカットが存在しない場合は、システム管理者（本ライブラリをインストールされた方）にお尋ねください。また、本ライブラリのインストール時に作成される他のショートカットも同じ場所に置かれていることを前提とします。

NAG DLL (`FLW6I26DE_mkl.dll` / `FLW6I26DE_nag.dll`) をご利用の場合は、実行時に NAG DLL にアクセスできるように `install_dir\bin` フォルダにパスを通してください。また、適切な Intel ランタイムライブラリにパスが通っていない場合は、`install_dir\rtl\bin` フォルダにパスを通してください。また、MKL を利用する NAG DLL (`FLW6I26DE_mkl.dll`) をご利用の場合は、`install_dir\mkl_intel64_11.3.3\bin` フォルダにパスを通してください。この時、`install_dir\mkl_intel64_11.3.3\bin` は `install_dir\bin` の後ろに設定してください。これは BLAS/LAPACK ルーチンのいくつかは、ベンダーバージョンとの問題を避けるために、NAG バージョン (`FLW6I26DE_mkl.dll` に含まれる) を使用する必要があるからです。（「4. ルーチン固有の情報」参照）

NAG DLL へのアクセスをチェックするために、スタートメニューの
NAG Fortran Library (FLW6I26DEL) にある以下のショートカットから診断プログラム
NAG_Fortran_DLL_info.exe を実行してください。

Check NAG FLW6I26DEL DLL Accessibility

この診断プログラムの詳細については、インストールノートの「4.2.2. アクセスチェッ
ク」をご参照ください。

3.1.1. コマンドウィンドウ

本ライブラリをコマンドウィンドウからご利用になる場合は、環境変数の設定が必要です。(なお、インストール時に環境変数の自動設定を選択された場合は、必要な環境変数はシステム環境変数に既に設定されています。)、スタートメニューの NAG Fortran Library (FLW6I26DEL) にある以下のショートカットがご利用いただけます。

NAG FLW6I26DEL Command Prompt

このショートカットは、本ライブラリと本製品で提供される MKL に対して必要な環境変数 INCLUDE, LIB, PATH を正しく設定した上でコマンドプロンプトを開きます。また、バッチファイル nag_example_*.bat が必要とする環境変数 NAG_FLW6I26DEL も設定します。

このショートカットを利用しない場合は、環境変数の設定を手動で行う必要があります。環境変数の設定はバッチファイル envvars.bat を用いて行うことができます。このバッチファイルのデフォルトの格納位置を以下に示します。

```
C:\Program Files\NAG\FL26\flw6i26del\batch\envvars.bat
```

その後、以下に示すコマンドの一つでコンパイル／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。いくつかのコマンドは紙面の都合により二行で書かれていますが、実際は一行で打ち込んでください。

```
ifort /MT driver.f90 nag_mkl_MT.lib mkl_intel_lp64.lib  
      mkl_intel_thread.lib mkl_core.lib libiomp5md.lib user32.lib
```

```
ifort /MT driver.f90 nag_nag_MT.lib user32.lib
```

```
ifort /MD driver.f90 nag_mkl_MD.lib mkl_intel_lp64.lib  
      mkl_intel_thread.lib mkl_core.lib libiomp5md.lib user32.lib
```

```
ifort /MD driver.f90 nag_nag_MD.lib user32.lib
```

```
ifort /MD driver.f90 FLW6I26DE_mkl.lib
```

```
ifort /MD driver.f90 FLW6I26DE_nag.lib
```

指定するライブラリの順番は重要です。これは BLAS/LAPACK ルーチンのいくつかはベンダーバージョンとの問題を避けるために NAG バージョンを使用する必要があるからです。
(「4. ルーチン固有の情報」参照)

最初の2つのコマンドは /MT オプションでコンパイルされた NAG スタティックライブラリを利用する場合です。1番目のコマンドは、NAG の BLAS/LAPACK を含まない (MKL の BLAS/LAPACK を利用する) NAG スタティックライブラリ nag_mkl_MT.lib を利用する場合です。(注: Intel コンパイラの新しいバージョンには、OMP ランタイムライブラリの MT バージョン libiomp5mt.lib はありません。代わりに、libiomp5md.lib をリンクする必要があります。) 2番目のコマンドは、NAG の BLAS/LAPACK を含む NAG スタティックライブラリ nag_nag_MT.lib を利用する場合です。/MT オプションは正しいランタイムライブラリ (マルチスレッドスタティックランタイムライブラリ) とのリンクを指示するために必要です。

3番目と4番目のコマンドは /MD オプションでコンパイルされた NAG スタティックライブラリを利用する場合です。3番目のコマンドは、NAG の BLAS/LAPACK を含まない (MKL の BLAS/LAPACK を利用する) NAG スタティックライブラリ nag_mkl_MD.lib を利用する場合です。4番目のコマンドは、NAG の BLAS/LAPACK を含む NAG スタティックライブラリ nag_nag_MD.lib を利用する場合です。/MD オプションは正しいランタイムライブラリ (マルチスレッド DLL ランタイムライブラリ) とのリンクを指示するために必要です。

最後の2つのコマンドは /MD オプションでコンパイルされた NAG DLL インポートライブラリを利用する場合です。5番目のコマンドは、NAG の BLAS/LAPACK を含まない (MKL の BLAS/LAPACK を利用する) NAG DLL (FLW6I26DE_mkl.dll) を利用する場合です。NAG DLL インポートライブラリ FLW6I26DE_mkl.lib から BLAS/LAPACK シンボルがエクスポートされるため、MKL インポートライブラリのパスは必要ありません。最後のコマンドは、NAG の BLAS/LAPACK を含む NAG DLL (FLW6I26DE_nag.dll) を利用する場合です。/MD オプションは正しいランタイムライブラリ (マルチスレッド DLL ランタイムライブラリ) とのリンクを指示するために必要です。

/MT オプションは /libs:static /threads オプションと等価です。

/MD オプションは /libs:dll /threads オプションと等価です。

Intel Visual Fortran コンパイラの環境変数の設定にもご注意ください。

詳細はコンパイラの Users' Guide をご参照ください。

3.1.2. Microsoft Visual Studio

本セクションの説明は Visual Studio 2015 および Intel Fortran Compiler 16.0 を想定しています。他のバージョンでは詳細が異なるかもしれません。

Visual Studio がビルド時と実行時に NAG インターフェースブロックファイルとライブラリにアクセスできるように、環境変数 PATH, LIB, INCLUDE に適切な値が設定されている必要があります。NAG ライブラリのインストール時に、これらの環境変数の更新を許可した場合は、必要な設定は既に行われています。そうでない場合は、実行時に NAG DLL (FLW6I26DE_mkl.dll/FLW6I26DE_nag.dll) にアクセスできるように、NAG DLL の格納フォルダー `install_dir¥bin` を環境変数 PATH に設定してください。更に、MKL を利用する NAG DLL (FLW6I26DE_mkl.dll) の使う場合は、実行時に MKL DLL にアクセスできるように、MKL DLL の格納フォルダー `install_dir¥mkl_intel64_11.3.3¥bin` を環境変数 PATH に設定してください。この時、`install_dir¥mkl_intel64_11.3.3¥bin` は `install_dir¥bin` の後ろに設定してください。

本ライブラリは最大最適化されています。そのため Debug モードだと C ランタイムライブラリについての警告メッセージが表示されますが、通常これは無視して構いません。Release モードではこの警告メッセージは出力されません。Release モードへの変更は、ツールバーもしくはメニューの「ビルド > 構成マネージャー」から行うことができます。

Intel Fortran プロジェクトに NAG ライブラリを追加する手順を以下に示します。

1. プロジェクトのプロパティページを開いてください。
プロパティページは次のいずれかの操作で開くことができます。
 - ソリューションエクスプローラーでプロジェクトを選択して、メニュー「プロジェクト > プロパティ」を選択してください。
 - ソリューションエクスプローラーでプロジェクトを右クリックして、「プロパティ」を選択してください。
 - ソリューションエクスプローラーでプロジェクトを選択して、ツールバーの「プロパティウィンドウ」ボタンを選択してください。
「プロパティ」ウィンドウの「プロパティページ」アイコンを選択してください。

2. 本ライブラリは 64-bit ライブラリです。64-bit コードをビルドするためには、プラットフォームを「x64」に設定する必要があります。「x64」オプションがリストに無い場合は、「構成マネージャー」を開き、「アクティブソリューションプラットフォーム」から「新規作成」を選択してください。そして、ドロップダウンリストから「x64」を選択し、「設定のコピー元」は「Win32」もしくは「x86」を選択してください。必要に応じて、「新しいプロジェクトプラットフォームを作成する」にチェックを入れてください。その後、「プロジェクトのコンテキスト」テーブルの「プラットフォーム」に「x64」が表示されていることを確認してください。

3. プロパティページの左パネルから「構成プロパティ > Fortran > 全般」を選択してください。右パネルの「追加のインクルード・ディレクトリ」に、フォルダー

```
install_dir¥nag_interface_blocks
```

を設定してください。デフォルトでは、

```
C:¥Program Files¥NAG¥FL26¥flw6i26del¥nag_interface_blocks
```

4. プロパティページの左パネルから「構成プロパティ > リンカー > 全般」を選択してください。右パネルの「追加のライブラリディレクトリ」に、3つのフォルダー

```
install_dir¥lib
```

```
install_dir¥mkl_intel64_11.3.3¥lib
```

```
install_dir¥rtlib¥lib
```

を設定してください。デフォルトでは、

```
C:¥Program Files¥NAG¥FL26¥flw6i26del¥lib
```

```
C:¥Program Files¥NAG¥FL26¥flw6i26del¥mkl_intel64_11.3.3¥lib
```

```
C:¥Program Files¥NAG¥FL26¥flw6i26del¥rtlib¥lib
```

また、libucrt.lib ファイルの格納場所を加える必要があるかもしれません。

(例えば、Windows 10 では、一般的には以下のような場所にあります)

```
C:¥Program Files (x86)¥Windows Kits¥10¥Lib¥10.0.10240.0¥ucrt¥x64
```

これは、ご利用のシステムによって変わります。また、Windows の古いバージョンでは、適切な Windows アップデートをインストールする必要があるかもしれません。)

5. プロパティページの左パネルから「構成プロパティ > リンカー > 入力」を選択してください。右側のパネルの「追加の依存ファイル」に、例えば MKL を利用する NAG スタティックライブラリ nag_mkl_MT.lib をご利用になる場合は、6つのライブラリ nag_mkl_MT.lib mkl_intel_lp64.lib mkl_intel_thread.lib mkl_core.lib libiomp5md.lib user32.lib を追加してください。各ライブラリの名前をスペースで区切ることに注意してください。また、nag_mkl_MT.lib を最初に置く必要があります。変更を有効にするために「適用」ボタンをクリックしてください。もし、MKL を利用しない NAG スタティックライブラリ nag_nag_MT.lib をご利用になる場合は、上記の6つのライブラリの代わりに、2つのライブラリ nag_nag_MT.lib user32.lib を追加してください。同様に、ご利用の NAG ライブラリに依って、適切なライブラリを「追加の依存ファイル」に追加してください。（以下の表をご参照ください。）

6. 最後に、正しいランタイムライブラリを指定する必要があります。プロパティページの左パネルから「構成プロパティ > Fortran > ライブラリー」を選択してください。右パネルの「ランタイム・ライブラリー」において、nag_nag_MT.lib または nag_mkl_MT.lib をご利用の場合は「マルチスレッド」を選択し、その他の NAG ライブラリをご利用の場合は「マルチスレッド DLL」を選択してください。変更を有効にするために「適用」ボタンをクリックしてください。

7. 「OK」ボタンをクリックして、プロパティページを閉じてください。

以下の表に設定を要約します。

追加の依存ファイル	ランタイム・ライブラリー	環境変数 PATH
nag_mkl_MT.lib mkl_intel_lp64.lib mkl_intel_thread.lib mkl_core.lib libiomp5md.lib user32.lib	マルチスレッド	-
nag_nag_MT.lib user32.lib	マルチスレッド	-
nag_mkl_MD.lib mkl_intel_lp64.lib mkl_intel_thread.lib mkl_core.lib libiomp5md.lib user32.lib	マルチスレッド DLL	-
nag_nag_MD.lib user32.lib	マルチスレッド DLL	-
FLW6126DE_mkl.lib	マルチスレッド DLL	install_dir¥bin install_dir¥ mkl_intel64_11.3.3¥bin
FLW6126DE_nag.lib	マルチスレッド DLL	install_dir¥bin

以上で、プロジェクトのビルド（コンパイル／リンク）を行うことができます。

Microsoft Development Environment 上でのプログラムの実行は、「デバッグ」メニュー（例えば、「デバッグなしで開始 (Ctrl+F5)」など）から行うことができます。

プログラムの実行に入出力ダイレクションが伴う場合は、プロパティページの「構成プロパティ > デバッグ」から「コマンド引数」に適切なコマンドを指定してください。例えば、

< input_file > output_file

アプリケーションの作業フォルダー以外で入出力を行う場合は、フルパスもしくは相対パスでファイルを指定する必要があります。作業フォルダーの設定は、プロパティページの「構成プロパティ > デバッグ」から「作業ディレクトリ」で行うことができます。

3.1.3. モジュールファイルについて

Install_dir¥nag_interface_blocks フォルダに提供されるモジュールファイル (*.mod) は, Intel コンパイラ (ifort) 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, 他のコンパイラではご利用いただけません. 他のコンパイラでご利用の場合は, 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必要ありません. 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.) 詳細は「3.2. インターフェースブロック」をご参照ください.

3.1.4. Microsoft C/C++ または Intel C/C++

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 `nagmk26.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++ から本ライブラリの DLL 版（以下 NAG DLL と呼ぶ）を利用する Example が以下のフォルダーに提供されます。

```
install_dir\samples\c_examples
```

および、

```
install_dir\samples\cpp_examples
```

C または C++ から NAG DLL を呼び出す際のより詳細なアドバイスは、ドキュメント `install_dir/c_headers/techdoc.html` をご参照ください。なお、このドキュメントのショートカットがスタートメニューの NAG Fortran Library (FLW6I26DEL) に提供されます。

Calling NAG FLW6I26DEL from C & C++

キーとなる情報：

- 配列のアクセス順序が異なる。
C は行優先 (Row Major), Fortran は列優先 (Column Major) である。
- 提供されるヘッダーファイルを利用する。
- Fortran の文字列は二つのパラメーターとして扱われる (文字列と文字列長)。
- C から NAG DLL を利用する Example が提供される。

```
install_dir\samples\c_examples
```

- C++ から NAG DLL を利用する Example が提供される.

```
install_dir¥samples¥cpp_examples
```

- C プログラムは .c 拡張子, C++ プログラムは .cpp 拡張子を用いる.

C プログラムから NAG DLL をご利用になる場合は, NAG DLL インポートライブラリの格納フォルダーのパスが環境変数 LIB に設定されていることが前提となります. 以下のようにコンパイル/リンクを行ってください. ここで driver.c がユーザープログラムです. (以下のコマンドは Microsoft C コンパイラ (cl) を使用しています. Microsoft C コンパイラは, 例えば, VS2015 x64 Native Tools Command Prompt から利用できます.)

```
cl driver.c FLW6I26DE_mkl.lib
```

または,

```
cl driver.c FLW6I26DE_nag.lib
```

上記のコマンドはヘッダーファイルの格納フォルダーのパスが環境変数 INCLUDE に設定されていることを前提としています. このパスが設定されていない場合は, 以下のようにコンパイル/リンクを行ってください.

```
cl /I"install_dir¥c_headers" driver.c FLW6I26DE_mkl.lib
```

または,

```
cl /I"install_dir¥c_headers" driver.c FLW6I26DE_nag.lib
```

NAG DLL の代わりに NAG スタティックライブラリをリンクする場合は, コンパイラのランタイムライブラリの格納フォルダー install_dir¥rtl¥lib へのアクセスが必要となります. このフォルダーのパスが環境変数 LIB に設定されていることを前提として, 以下のようにコンパイル/リンクを行ってください.

```
cl /MD /I"install_dir%c_headers" driver.c nag_nag_MD.lib \  
libifcoremd.lib user32.lib
```

または,

```
cl /MT /I"install_dir%c_headers" driver.c nag_nag_MT.lib \  
libifcoremt.lib user32.lib
```

ここでは、実行時の問題を回避するために、Intel コンパイラのランタイムライブラリ libifcoremd.lib および libifcoremt.lib が指定されていることに注意してください。これらのランタイムライブラリは、nag_nag_MD.lib または nag_nag_MT.lib を Microsoft C コンパイラで使用するときが必要です。

MKL を利用するバージョンの NAG スタティックライブラリ nag_mkl_MD.lib または nag_mkl_MT.lib をリンクする場合は、いくつかのライブラリを追加する必要があります。詳細は「3.1.1. コマンドウィンドウ」をご参照ください。

Intel C コンパイラ (icl) でのご利用方法は、上記の Microsoft C コンパイラ (cl) でのご利用方法と同じです。ただし、環境変数 LIB に install_dir%rtlib フォルダのパスを設定する必要はありません。

以下の説明は Visual Studio 2015 を想定しています。他のバージョンでは詳細が異なるかもしれません。

Visual Studio から NAG ライブラリを利用する場合は、以下のようにリンカーの設定を行ってください。まず、ソリューションプラットフォームを x64 に設定してください。次に、プロジェクトのプロパティを開き、「構成プロパティ > リンカー > 入力」の「追加の依存ファイル」に FLW6I26DE_mkl.lib または FLW6I26DE_nag.lib を追加してください。次に、「構成プロパティ > リンカー > 全般」の「追加のライブラリディレクトリ」に install_dir%rtlib を追加してください。最後に、「C/C++ > 全般」の「追加のインクルードディレクトリ」に install_dir%c_headers を追加するか、もしくは、ソリューションエクスプローラーの「ヘッダーファイル」フォルダーに nagmk26.h を追加してください。

注意 : C++ Example (install_dir%samples%c++_examples) を Microsoft C++ で利用する場合は、コンパイラオプション /EHsc が必要です。

3.1.5. Fortran Builder

本ライブラリの DLL 版（以下 NAG DLL と呼ぶ）は、Fortran Builder（NAG Fortran コンパイラ）から以下の 3 つの方法でご利用いただけます。

Fortran Builder 自動リンク：

ご利用の Fortran Builder が本ライブラリ（FLW6I26DEL）を自動リンクできるバージョンの場合、以下の方法で本ライブラリをご利用ください。

1. 「コンソールアプリケーション」プロジェクトを新規作成する。
2. メニューバーから「プロジェクト > プロジェクトの設定」を開く。
3. 「基本設定」タブを開く。
4. 「追加ライブラリ > NAG Fortran Library を利用する」にチェックを入れる。
（これにより、ビルド時に、NAG インターフェイスブロックの格納フォルダーが自動的にインクルードされ、MKL を利用する NAG DLL（FLW6I26DE_mkl.dll）が自動的にリンクされます。）
5. 「OK」ボタンを押し、プロジェクトの設定を閉じる。

以上で、MKL を利用する NAG DLL（FLW6I26DE_mkl.dll）を利用したプロジェクトをビルド／実行することができます。

MKL を利用しない NAG DLL（FLW6I26DE_nag.dll）をご利用になる場合は、手動でリンク設定を行なってください。設定方法については、次ページをご参照ください。

注意：「Fortran コンパイラ > 実行時診断」タブの「未定義の変数 (=undefined)」オプションは、本ライブラリと互換性がありません。もし、このオプションにチェックを入れてビルドすると、コンパイルエラーとなります。

Fortran Builder 手動リンク :

ご利用の Fortran Builder が本ライブラリ (FLW6I26DEL) を自動リンクできないバージョンの場合、以下の方法で本ライブラリをご利用ください。

1. 「コンソールアプリケーション」プロジェクトを新規作成する。
2. メニューバーから「プロジェクト > プロジェクトの設定」を開く。
3. 「ディレクトリ > インクルード」タブを開く。
4. 「インクルード」にフォルダー,
install_dir¥nag_interface_blocks_nagfor
を追加する。

注意 : パスにスペースが含まれていても、クォテーションで括らないでください。

5. 「リンク > 基本設定」タブを開く。
6. 「リンクするライブラリ」に DLL ファイル,
install_dir¥bin¥FLW6I26DE_mkl.dll
もしくは,
install_dir¥bin¥FLW6I26DE_nag.dll
を追加する。

注意 : DLL インポートライブラリではなく、DLL 本体を指定してください。

7. 「OK」ボタンを押し、プロジェクトの設定を閉じる。

以上で NAG DLL を利用したプロジェクトをビルド/実行することができます。

注意 : 「Fortran コンパイラ > 実行時診断」タブの「未定義の変数 (=undefined)」オプションは、本ライブラリと互換性がありません。もし、このオプションにチェックを入れてビルドすると、コンパイルエラーとなります。

コマンドウィンドウ：

コマンドウィンドウからご利用になる場合は、「3.1.1. コマンドウィンドウ」と同じく環境変数 PATH が正しく設定されていることを確認してください。

NAG Fortran コンパイラを用いて生成されたインターフェースブロックのモジュールファイル (*.mod) が、install_dir¥nag_interface_blocks_nagfor フォルダに提供されます。もし、異なるバージョンの NAG Fortran コンパイラでご利用になる場合は、モジュールファイルを再生成する必要があります(「3.2. インターフェースブロック」参照)。

NAG Fortran コンパイラでは、DLL インポートライブラリではなく、DLL 本体に直接リンクしなくてはなりません。

以下に示すコマンドでコンパイル／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。)

MKL を利用する NAG DLL を利用する場合：

```
nagfor -ieee=full -I"install_dir¥nag_interface_blocks_nagfor" driver.f90  
"install_dir¥bin¥FLW6I26DE_mkl.dll" -o driver.exe
```

MKL を利用しない NAG DLL を利用する場合：

```
nagfor -ieee=full -I"install_dir¥nag_interface_blocks_nagfor" driver.f90  
"install_dir¥bin¥FLW6I26DE_nag.dll" -o driver.exe
```

FLW6I26DE_mkl.dll または FLW6I26DE_nag.dll ファイルはフルパスで指定してください。また、パスがスペースを含む場合は、クォテーションで括る必要があります。

3.1.6. Microsoft C#

本ライブラリは C# 環境からで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# ヘッダーファイル flcsdnet64.cs が提供されます。このヘッダーファイルから必要な部分だけを自身のプログラムにコピー&ペーストしてご利用ください。

C# から本ライブラリの DLL 版を利用する Example が install_dir¥samples¥cs_examples フォルダに提供されます。これらの Example は、コマンドプロンプトから C# コンパイラ csc を用いて以下のように簡単にご利用いただけます。

(ここで driver.cs が任意の Example ソースファイルです。)

```
csc /platform:x64 driver.cs
```

更なる情報は以下のウェブページをご参照ください。

<http://www.nag.co.uk/content/calling-nag-fortran-library-dll-csharp>

また、別途 .NET 環境に対応した NAG ライブラリ製品 NAG Library for .NET がご利用いただけます。詳細は以下のウェブページをご参照ください。

<http://www.nag.co.uk/content/nag-library-net>

3.1.7. Microsoft Visual Basic for Applications 7 / Excel

本ライブラリの DLL 版（以下 NAG DLL と呼ぶ）は Excel スプレッドシートでもご利用いただけます。NAG ライブラリルーチンは、Visual Basic for Applications 7.0 (VBA7) コードから呼び出すことができます。本セクションの情報は Excel の 64-bit 版に関するものです。

NAG DLL を Excel から利用する Example が以下のフォルダーに提供されます。

```
install_dir¥samples¥excel64_examples
```

install_dir¥samples¥excel64_examples¥linear_algebra¥xls_demo_64.html ファイルには、Excel スプレッドシートから NAG DLL を利用する際のヒントが記載されています。

キーとなる情報：

- Install_dir¥vba7-64_headers フォルダの flvba764-〈チャプター名〉.bas（例えば flvba764-a.bas）ファイルには、VBA7 で利用できる Declare 文がチャプター毎に定義されています。また、flvba764-types.bas ファイルには、これらのファイルで利用される定数やユーザー定義型が定義されています。また、flvba764-f-blaslapack.bas ファイルにはチャプター F のルーチンが（NAG 名ではなく）BLAS/LAPACK 名で定義されています。
- MKL の BLAS/LAPACK を利用する NAG DLL をご利用になる場合は、宣言文の DLL 名を FLW6I26DE_nag.dll から FLW6I26DE_mkl.dll に変更してください。
- Declare 文のご利用は、ファイルから必要な部分だけをご自身のモジュールにコピー & ペーストするか、もしくはファイルをモジュールとして VBA7 プロジェクトにインポートしてください。場合によっては、上述の flvba764-types.bas も合わせてインポートする必要があります。
- Fortran の配列は 1 から始まるので、Option Base 1 の設定を推奨します。
- 実際の引数として Variant 型は使用できません。Long, Double, String（および、ごく稀に Single）が必要です。Option Explicit を使用してください。

- Long は Fortran の INTEGER に、Double は Fortran の DOUBLE PRECISION に、Single は Fortran の REAL にそれぞれ対応します。
- Long は Fortran の LOGICAL に対応します。NAGTRUE と NAGFALSE がそれぞれ -1 と 0 に対応します。
- LongPtr はコールバック関数へのポインターに使用されます。実際の関数へのポインターは VB の AddressOf 演算子を使用して渡されます。
- 構造体 Complex と ComplexSimple は Fortran の COMPLEX*16 と COMPLEX にそれぞれ対応します。
- Fortran の配列引数に対しては、VBA7 配列の最初の要素を指定します。
例えば、A(1, 1)。
- 数式は ByVal 引数に渡されます。その他の引数はデフォルトでは ByRef です。この点が明確になるように、ByRef と ByVal は Declare 文の全体を通して明示的に指定されています。
- Fortran の文字引数に対しては、2つの VBA 引数が必要となります。ByVal 文字引数と ByVal 文字長引数 (Long 型) です。文字長引数は引数リストの最後に置く必要があります。

以上の情報は Microsoft Office Excel 2010, 2013, 2016 で検証されています。

3.1.8. Microsoft Visual Basic .NET

NAG ライブラリルーチンの多くは Visual Basic .NET (VB.NET) から呼び出すことができます。VB.NET から NAG DLL を利用する Example が以下のフォルダーに提供されます。

```
install_dir\samples\vb.net64_examples
```

これらの Example は Visual Studio 2010 で生成されています。

キーとなる情報：

- 以下のファイルに VB.NET で利用できる Declare 文が定義されています。

```
Install_dir\vb.net64_headers\flvbdnet64.vb
```

- MKL の BLAS/LAPACK を利用する NAG DLL をご利用になる場合は、宣言文の DLL 名を FLW6I26DE_nag.dll から FLW6I26DE_mkl.dll に変更してください。
- Declare 文のご利用は、ファイルから必要な部分だけをご自身のモジュールにコピー & ペーストするか、もしくはファイルをモジュールとして VB.NET プロジェクトにインポートしてください。
- Fortran 配列は 1 から始まりますが、VB.NET 配列は 0 から始まります。
- 次の型マッピングが使われます。
Integer は Fortran の INTEGER に、Double は Fortran の DOUBLE PRECISION に、Single は Fortran の REAL にそれぞれ対応します。
- Integer は Fortran の LOGICAL に対応します。NAGTRUE と NAGFALSE がそれぞれ -1 と 0 に対応します。
- 構造体 Complex と ComplexSimple は Fortran の COMPLEX*16 と COMPLEX にそれぞれ対応します。

- 全てのスカラー値は参照渡し (ByRef) です。VB.NET はデフォルトでは値渡し (ByVal) なので、参照渡し (ByRef) を明示的に指定する必要があります。この点が明確になるように、ByRef と ByVal は Declare 文の全体を通して明示的に指定されています。
- 配列引数には配列名を渡してください。全ての配列は値渡し (ByVal) です。また宣言には Fortran 側の用途 (入力, 出力, 入出力) によって <[In] ()>, <Out ()>, <[In] (), Out ()> のいずれかの decoration が付加されています。具体例として、various_routines_64 Example の G02EEFE () Sub プロシージャをご参照ください。
- VB.NET ではコールバック関数における配列は値渡しされた IntPtr によって表現されます。
- VB.NET の配列は行優先です。一方で Fortran の配列は列優先です。このため Fortran ルーチンが正しく配列を解釈するためには配列の転置が必要です。
- 配列の格納形式が異なるため、Fortran ルーチンの Leading Dimension は VB.NET の配列の 2 次元目に対応します。例えば、VB.NET の A(2, 3) では Leading Dimension として 4 (配列は 0 から始まるため) を渡します。
- Fortran 側で CHARACTER* 型 (例えば、CHARACTER*(*) または CHARACTER*1) のスカラー引数が求められる場合は、文字列を VB.NET の String で値渡ししてください。そして、引数リストの最後に文字列の長さを Integer で値渡ししてください。
- Fortran 側で CHARACTER* 型の配列引数が求められる場合は、VB.NET の一つの String に全ての配列要素を結合したものを渡してください。そして、引数リストの最後に配列の一つの要素の長さを Integer で値渡ししてください。具体例として、various_routine_64 Example の M01CCFE () Sub プロシージャをご参照ください。
- Fortran 側でコールバック関数が求められる場合は、VB.NET で interface の宣言として Delegate function を定義する必要があります。引数はその Delegate function 型で値渡ししてください。Delegate function の実装を引数として渡す際には、キーワード AddressOf を利用してください。具体例として、various_routines_64 Example の D01BDFE () Sub プロシージャをご参照ください。

- これらの宣言を VB.NET に認識させるために、VB.NET ソースコードの一番上には次の一行が必要です。

```
Imports System.Runtime.InteropServices
```

以上の情報は Visual Studio 2010, 2012, 2013, 2015 で検証されています。

3.1.9. その他の環境

その他の環境からの本ライブラリのご利用については、以下の追加情報ページをご参照ください。

<http://www.nag.co.uk/doc/inun/fl26/w6idel/supplementary.html>

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事ができます。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

これらのモジュールは Intel Fortran コンパイラ (ifort) を用いてプリコンパイルされた形式 (*.mod ファイル) で提供されます。

本ライブラリのコマンドプロンプト（スタートメニューのショートカットとして提供される）を利用する場合、もしくはバッチファイル envvars.bat を実行して環境変数の設定を行った場合は、環境変数 INCLUDE があらかじめ設定されるため、「3.1.1. コマンドウィンドウ」で示されるコマンドでこれらのモジュールにアクセスすることができます。

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「2.2. 開発環境」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の間に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以下のような方法で再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、この限りではありません。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。）

オリジナルのモジュールファイルのバックアップのために、任意の場所に任意の名前で（例えば、nag_interface_blocks_original）フォルダーを作成し、nag_interface_blocks フォルダーの内容物をそのフォルダーにコピーしてください。

そして、nag_interface_blocks フォルダーにおいて、すべての *.f90 ファイルをご利用の Fortran コンパイラでコンパイルしてください。その際、インターフェースブロックには依存関係があるため、コンパイルの順番が重要となります。以下に示す順番でコンパイルを行ってください。（FCOMP をご利用のコンパイラ名で置き換えてください。）

```
FCOMP -c nag_precisions.f90
FCOMP -c nag_a_ib.f90
FCOMP -c nag_blast_ib.f90
FCOMP -c nag_blas_consts.f90
FCOMP -c nag_blas_ib.f90
FCOMP -c nag_c_ib.f90
FCOMP -c nag_d_ib.f90
FCOMP -c nag_e_ib.f90
FCOMP -c nag_f_ib.f90
FCOMP -c nag_g_ib.f90
FCOMP -c nag_h_ib.f90
FCOMP -c nag_lapack_ib.f90
FCOMP -c nag_m_ib.f90
FCOMP -c nag_s_ib.f90
FCOMP -c nag_x_ib.f90
FCOMP -c nag_long_names.f90
FCOMP -c nag_library.f90
```

コンパイルによって生成されるオブジェクトファイルは必要ありません。
モジュールファイル (*.mod ファイル) だけをご利用ください。

3.3. Example プログラム

提供される Example 結果は、nag_mkl_MD.lib (MKL BLAS/LAPACK ルーチンを利用する NAG スタティックライブラリ) を用いて、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。

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¥batch フォルダに3つのバッチファイル
nag_example_DLL.bat, nag_example_static_MT.bat, nag_example_static_MD.bat
が提供されます。

これらのバッチファイルを用いて Example プログラムを簡単に利用することができます。これらのバッチファイルは、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）をカレントフォルダにコピーして、コンパイル／リンク／実行を行います。

これらのバッチファイルをご利用の際には、Fortran コンパイラと NAG ライブラリに対して必要な環境変数が設定されていなければなりません。特に、環境変数 NAG_FLW6126DEL に本ライブラリのインストール先（例えば、C:¥Program Files¥NAG¥FL26¥flw6i26del）が設定されている必要があります。

ご利用の NAG ライブラリルーチンの名前をバッチの引数に指定してください。

例)

```
nag_example_DLL e04ucf
```

この例では、e04ucfe.f90（ソースファイル）、e04ucfe.d（データファイル）、e04ucfe.opt（オプションファイル）をカレントフォルダにコピーして、コンパイル／リンク／実行を行い e04ucfe.r（結果ファイル）を生成します。

- `nag_example_DLL.bat`

`FLW6I26DE_nag.dll` (NAG BLAS/LAPACK を利用する NAG DLL ライブラリ) をリンクします。

例)

```
nag_example_DLL e04ucf
```

`FLW6I26DE_mkl.dll` (MKL BLAS/LAPACK を利用する NAG DLL ライブラリ) をリンクする場合は `-mkl` オプションを付けてください。

例)

```
nag_example_DLL -mkl e04ucf
```

- `nag_example_static_MD.bat`

`nag_nag_MD.lib` (NAG BLAS/LAPACK を利用する NAG スタティックライブラリ (/MD)) をリンクします。

例)

```
nag_example_static_MD e04ucf
```

`nag_mkl_MD.lib` (MKL BLAS/LAPACK を利用する NAG スタティックライブラリ (/MD)) をリンクする場合は `-mkl` オプションを付けてください。

例)

```
nag_example_static_MD -mkl e04ucf
```

- `nag_example_static_MT.bat`

`nag_nag_MT.lib` (NAG BLAS/LAPACK を利用する NAG スタティックライブラリ (/MT)) をリンクします。

例)

```
nag_example_static_MT e04ucf
```

`nag_mkl_MT.lib` (MKL BLAS/LAPACK を利用する NAG スタティックライブラリ (/MT)) をリンクする場合は `-mkl` オプションを付けてください。

例)

```
nag_example_static_MT -mkl e04ucf
```

3.4. Fortran 型と強調斜体文字の解釈

本ライブラリは 32-bit 整数を使用します。

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 (倍精度実数)
COMPLEX (kind=nag_rp) - COMPLEX (単精度複素数)
COMPLEX (kind=nag_wp) - 倍精度複素数 (e. g. 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。詳細は “How to Use the NAG Library and its Documentation” ドキュメントの「4.4 実装依存情報」をご参照ください。

3.5. メンテナンスレベル

ライブラリのメンテナンスレベルは、ライブラリルーチン A00AAF の Example プログラムをコンパイル／リンク／実行することにより確認することができます。この時、バッチファイル nag_example_DLL.bat を引数 a00aaf と共に用いれば、Example プログラムのコンパイル／リンク／実行を容易に行うことができます(「3.3. Example プログラム」参照)。ライブラリルーチン A00AAF はライブラリの詳細(タイトル、製品コード、使用されるコンパイラおよび精度、バージョン (Mark) など) を出力します。

または、診断プログラム NAG_Fortran_DLL_info.exe を利用することもできます。

診断プログラムはその中で A00AAF を呼び出します。

(インストールノートの「4.2.2. アクセスチェック」参照)

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

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 MKL 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注意してください.

b. 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 $x_{hi} = 2.65E+1$

S15AGF IFAIL = 1 if $X \geq 2.53E+307$
IFAIL = 2 if $4.74E+7 \leq X < 2.53E+307$
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 $0 < X \leq 2.23E-308$

S17AEF IFAIL = 1 if $\text{abs}(X) > 1.0E+16$
S17AFF IFAIL = 1 if $\text{abs}(X) > 1.0E+16$

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 $\text{abs}(Z) < 3.92223E-305$
IFAIL = 4 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 3.27679E+4$
IFAIL = 5 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 1.07374E+9$

S17DEF IFAIL = 2 if $\text{AIMAG}(Z) > 7.00921E+2$
IFAIL = 3 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 3.27679E+4$
IFAIL = 4 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 1.07374E+9$

S17DGF IFAIL = 3 if $\text{abs}(Z) > 1.02399E+3$
IFAIL = 4 if $\text{abs}(Z) > 1.04857E+6$

S17DHF IFAIL = 3 if $\text{abs}(Z) > 1.02399E+3$
IFAIL = 4 if $\text{abs}(Z) > 1.04857E+6$

S17DLF IFAIL = 2 if $\text{abs}(Z) < 3.92223E-305$
IFAIL = 4 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 3.27679E+4$
IFAIL = 5 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 1.07374E+9$

S18ADF IFAIL = 2 if $0 < X \leq 2.23E-308$
S18AEF IFAIL = 1 if $\text{abs}(X) > 7.116E+2$
S18AFF IFAIL = 1 if $\text{abs}(X) > 7.116E+2$
S18DCF IFAIL = 2 if $\text{abs}(Z) < 3.92223E-305$
IFAIL = 4 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 3.27679E+4$

IFAIL = 5 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 1.07374\text{E}+9$
S18DEF IFAIL = 2 if $\text{REAL}(Z) > 7.00921\text{E}+2$
IFAIL = 3 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 3.27679\text{E}+4$
IFAIL = 4 if $\text{abs}(Z)$ or $\text{FNU}+\text{N}-1 > 1.07374\text{E}+9$

S19AAF IFAIL = 1 if $\text{abs}(X) \geq 5.04818\text{E}+1$
S19ABF IFAIL = 1 if $\text{abs}(X) \geq 5.04818\text{E}+1$
S19ACF IFAIL = 1 if $X > 9.9726\text{E}+2$
S19ADF IFAIL = 1 if $X > 9.9726\text{E}+2$

S21BCF IFAIL = 3 if an argument $< 1.583\text{E}-205$
IFAIL = 4 if an argument $\geq 3.765\text{E}+202$
S21BDF IFAIL = 3 if an argument $< 2.813\text{E}-103$
IFAIL = 4 if an argument $\geq 1.407\text{E}+102$

c. X01

数学定数は以下のとおりです.

X01AAF (pi) = 3.1415926535897932

X01ABF (gamma) = 0.5772156649015328

d. 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

e. 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/content/nag-fortran-library-manual>

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_fl26¥html¥frontmatter¥manconts.html

nagdoc_fl26¥pdf¥frontmatter¥manconts.pdf

nagdoc_fl26¥pdf¥frontmatter¥manconts.html

ライブラリマニュアルをインストールした場合, これらの目次ファイルはスタートメニューの NAG Mark 26 Manual にある以下のショートカットから開くことができます。

NAG Library Manual Mark 26 (HTML5)

NAG Library Manual Mark 26 (PDF)

NAG Library Manual Mark 26 (PDF + HTML Index)

また, 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_fl26¥index.html

各形式の閲覧方法および操作方法については, 以下のドキュメントをご参照ください。

http://www.nag.co.uk/numeric/fl/nagdoc_fl26/html/genint/essint.html

また、以下のウェブサイトから、HTML ヘルプ形式のライブラリマニュアルをダウンロードすることができます。

<http://www.nag.co.uk/content/nag-fortran-library-manual>

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）

ユーザーノート（英語版）は、スタートメニューの NAG Fortran Library (FLW6I26DEL) にある以下のショートカットから開くことができます。

NAG FLW6I26DEL Users' Note

6. サポート

製品のご利用に関してご質問等がございましたら、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（FLW6I26DEL）並びに、お客様の User 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。

日本 NAG ヘルプデスク

Email: naghelp@nag-j.co.jp

7. コンタクト情報

日本ニューメリカルアルゴリズムズグループ株式会社（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

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）