

NAG Library for SMP and Multicore, Mark 24

FSW3224DCL - License Managed

Microsoft Windows, Intel Fortran

ユーザーノート

内容

1. イントロダクション	1
1.1. 制限事項	1
2. リリース後の最新情報	2
3. 一般情報	3
3.1. ライブラリのリンク方法	3
3.1.0. スレッド数の設定	6
3.1.1. Intel Fortran	6
3.1.1.1. コマンドウィンドウ	7
3.1.1.2. MS Visual Studio	9
3.1.2. モジュールファイルについて	14
3.1.3. NAG Fortran Builder	15
3.1.4. Silverfrost (Salford) FTN95	18
3.1.5. PGI Fortran	21
3.1.6. Microsoft Visual C++ / Visual Studio	22
3.1.7. Intel C/C++	24
3.1.8. Microsoft Visual Basic for Applications / Excel (32-bit)	25
3.1.9. Microsoft Visual Basic 6	27
3.1.10. Microsoft Visual Basic .NET	29
3.1.11. C#	31
3.1.12. Java	32
3.1.13. Python	32
3.1.14. R	32
3.1.15. Simfit (NAG Library edition)	32
3.1.16. その他の環境	33
3.1.17. アクセスチェック	33
3.2. インターフェースブロック	34
3.3. Example プログラム	36
3.4. Fortran 型と強調斜体文字の解釈	38
3.5. NAG ルーチンからの出力	39
4. ルーチン固有の情報	40
5. ドキュメント	46
6. サポート	48
7. ユーザーフィードバック	49
追記 - コンタクト先情報	49

## 1. イントロダクション

本ユーザーノートは NAG Library for SMP and Multicore, Mark 24 : FSW3224DCL (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でユーザーノートを参照するよう指示がある場合は、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、ライブラリマニュアルの以下のドキュメントを必ずお読みください。

- (a) Essential Introduction (ライブラリ一般に関する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)。

### 1.1. 制限事項

本ライブラリは Compaq Visual Fortran の呼び出し規約 (ifort /iface:cvf) を使います。これは本質的に引数が参照渡しであることを意味します。文字列はアドレスと文字長のペアで渡されます (/iface:cvf を指定すると、/iface:mixed\_str\_len\_arg が自動的に指定されます)。

本製品で提供されるモジュールファイル \*.mod (インターフェースブロック) は、Intel Fortran Compiler 13.0 で生成されたものです。異なるコンパイラ (Intel Fortran Compiler の異なるバージョンを含む) で利用する場合には再コンパイルが必要です。

Intel コンパイラ以外のコンパイラから本ライブラリの DLL 版を利用する場合は、二つの入出力システムが混在して使用されることに注意してください。ライブラリルーチンが使う Intel の入出力システムと、呼び出し元プログラムが使うコンパイラの入出力システムです。この場合、例えば E04UDF の Example プログラムのようなプログラムは単一のデータファイルから正しくデータを読み込むことができなくなります。何故なら、その Example プログラムでは、いくつかのデータをデータファイルから読み込み際にプログラム自体の入出力システムを用いる一方で、E04UDF (オプション設定ルーチン) が同じデータファイルからデータを読み込む際に Intel の入出力システムを用いるからです。こ

の二つの入出力システムは全く別々に動作するため、プログラム自体の入出力システムが到達したデータファイル上の位置を Intel の入出力システムは知ることができません。この問題はそれぞれの入出力システムに対応する二つのデータファイルを持つことによって回避できます。主に、チャプター H02 とチャプター E04 のオプション設定ルーチンがこの問題を持っています。

## 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/fs24/w32dcl/postrelease.html>

### 3. 一般情報

#### 3.1. ライブラリのリンク方法

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ー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。

C:\Program Files\NAG\FS24\fs3224dcl

もし、このフォルダー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した方）にお尋ねください。以降の説明ではこのフォルダーを `install_dir` として参照します。

また、以下の「スタート」メニューに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が前提となります。

すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
FSW3224DCL 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した方）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が前提となります。

(Windows 8 では、ショートカットはアプリケーションの全リストから利用することができます。「スタート」画面の背景部分を右クリックしてください。画面の右下隅に現れる「すべてのアプリ」ボタンを選択してください。「NAG」セクションからショートカットをご利用ください。)

本ライブラリの DLL 版 (FSW3224DC.dll) をご利用の場合は、実行時に DLL にアクセスできるように、`install_dir\bin` フォルダーと `install_dir\MKL_ia32_11.0\bin` 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。

この際、`install_dir\MKL_ia32_11.0\bin` は `install_dir\bin` よりも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 NAG 提供のもの (FSW3224DC.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)。

本ライブラリの DLL 版 (FSW3224DC.dll) へのアクセスをチェックするために、以下の「スタート」メニュー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。

すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
Check NAG DLL Accessibility for FSW3224DCL

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「4.2.3. アクセスチェック」をご参照ください。

コマンドプロンプトからの環境変数の設定については「3.1.1.1. コマンドウィンドウ」をご参照ください。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、必要な環境変数は既にシステム環境変数に設定されています。コントロールパネルから環境変数を設定する場合は、環境変数 PATH, LIB, INCLUDE に以下のパスを設定してください。

環境変数 PATH に以下のパスを設定してください。

```
C:\Program Files\NAG\FS24\fsw3224dc\batch;  
C:\Program Files\NAG\FS24\fsw3224dc\bin;  
C:\Program Files\NAG\FS24\fsw3224dc\MKL_ia32_11.0\bin;  
[既存のパス]
```

環境変数 LIB に以下のパスを設定してください。

```
C:\Program Files\NAG\FS24\fsw3224dc\lib;  
[既存のパス]
```

環境変数 INCLUDE に以下のパスを設定してください。

```
C:\Program Files\NAG\FS24\fsw3224dc\nag_interface_blocks;  
[既存のパス]
```

注意：上記のパス設定はデフォルトのインストール先にインストールを行った場合です。インストール先がデフォルトでない場合は適切なフォルダーで置き換えてください。

本ライブラリの DLL 版 (FSW3224DC.dll) では、NAG インポートライブラリ FSW3224DC.lib から MKL のシンボルが直接エクスポートされるため、リンク時には MKL インターフェースライブラリ mkl\_rt.lib を指定する必要はありません。しかし、もし MKL インターフェースライブラリの指定を行う場合は、NAG インポートライブラリ FSW3224DC.lib よりも後に指定してください。

例)

```
FSW3224DC.lib mkl_rt.lib
```

これは BLAS/LAPACK ルーチンのいくつかはベンダー提供のものとの問題を避けるために FSW3224DC.dll に含まれる NAG 提供のもの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)。

コールバック関数として NAG ルーチンに渡すユーザー作成の手続き (被積分関数など) をコンパイルする際には、手続き内のローカル変数が並列環境 (NAG ルーチンの並列領域) で安全に使用されるように注意を払う必要があります。特に、ローカル変数は静的にアロケートされてはいけません。ご利用のコンパイラによっては、これを実現するために、コンパイラオプションを付ける必要があります。また、これとは逆にローカル変数が静的にアロケートされる `-save` などのオプションは使用しないでください。

これから先のセクションでは、本ライブラリ (特に DLL 版) を様々な環境から利用する方法を説明します。また、以下の NAG ウェブサイトには NAG Fortran Library の DLL および NAG C Library の DLL のご利用に関する更に多くの情報が公開されています。

[http://www.nag.co.uk/numeric/Num\\_DLLhelp.asp](http://www.nag.co.uk/numeric/Num_DLLhelp.asp)

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/fs24/w32dcl/postrelease.html>

### 3.1.0. スレッド数の設定

環境変数 OMP\_NUM\_THREADS にご利用のスレッド数を設定してください。

例えば、コマンドウィンドウでは以下のように行います。

例)

```
set OMP_NUM_THREADS=N
```

N はご利用のスレッド数です。

OMP\_NUM\_THREADS はプログラムの実行毎に再設定することができます。

一般的に、推奨されるスレッドの最大数はご利用の SMP システムの物理コア数です。

### 3.1.1. Intel Fortran

本ライブラリはインストールノート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ています。それとは異なるバージョンの Intel Fortran コンパイラから本ライブラリを利用する場合は、正しい Intel Fortran コンパイラのランタイムライブラリがピックアップされるように、install\_dir¥bin フォルダに提供されている Intel Fortran コンパイラのランタイムライブラリ libifcoremd.dll, libmmd.dll, svml\_dispmd.dll の名前を変更する必要があります。これを簡単に行うためのバッチファイル hide\_ifort\_rtls.bat が同フォルダに提供されています。このバッチファイルは同フォルダにある libifcoremd.dll, libmmd.dll, svml\_dispmd.dll の名前を変更します。また、変更した名前を元に戻すためのバッチファイル expose\_ifort\_rtls.bat が同フォルダに併せて提供されます。

### 3.1.1.1. 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いただく場合には環境変数の設定が必要です。（通常、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設定されています。）

以下の「スタート」メニューのショートカットがご利用いただけます。

すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
FSW3224DCL Command Prompt

このショートカットは本ライブラリおよび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。

このショートカットを利用しない場合に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¥FS24¥fsw3224dcl¥batch¥envvars.bat
```

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。）

```
ifort /iface:cvf /MD /Qopenmp driver.f90 FSW3224DC.lib
```

```
ifort /iface:cvf /MD /Qopenmp driver.f90 FSW3224DC_static.lib mkl_rt.lib  
user32.lib
```

1 番目のコマンドは、本ライブラリの DLL 版 (FSW3224DC.dll) を利用する場合です。ここでは、NAG インポートライブラリ FSW3224DC.lib から MKL DLL の 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。

2 番目のコマンドは、本ライブラリのスタティック版 (FSW3224DC\_static.lib) を利用する場合です。ここでは、MKL インポートライブラリ mkl\_rt.lib と Microsoft ランタイムライブラリ user32.lib が必要です。

どちらのコマンドにも `/iface:cvf` オプションと `/MD` オプションが付いていることに注意してください。 `/iface:cvf` オプションは CVF 呼び出し規約を使用することをコンパイラに指示します。 `/MD` オプションはコンパイラのランタイムライブラリに関してマルチスレッド DLL 版を使用することをコンパイラに指示します。本ライブラリとの互換性のために、これらのコンパイラオプションは重要です。

`/Qopenmp` オプションによって、コンパイラが OpenMP コードをサポートするようになり、リンカーがコンパイラのスレッドライブラリ `libiomp5md.lib` をリンクするようになります。

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。

また、`/Qopenmp` オプションは `/Qauto` オプションを含んでいるため、ローカル変数は静的に割り当てられません（「3.1. ライブラリのリンク方法」参照）。

詳細はコンパイラの User's Guide をご参照ください。

### 3.1.1.2. MS Visual Studio

本セクションの説明は Visual Studio 2005/2008/2010/2012 および Intel Fortran Compiler 13.0 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。

実行時に NAG DLL (FSW3224DC.dll) と MKL DLL にアクセスできるように、NAG DLL のフォルダー `install_dir¥bin` と MKL DLL のフォルダー `install_dir¥MKL_ia32_11.0¥bin` が環境変数 PATH に設定されている必要があります。ただし、パス設定の順番に関して、`install_dir¥MKL_ia32_11.0¥bin` は `install_dir¥bin` よりも後でなければいけません。

Visual Studio を起動してください。以下の手順に従って Intel Fortran コンパイラで利用するフォルダーの設定を行ってください。以下の設定は Intel Fortran コンパイラを使うプロジェクト (Intel Fortran プロジェクト) すべてに適用されます。

1. メニュー「ツール > オプション」をクリックしてください。
2. 「オプション」ウィンドウで「インテル(R) Fortran」(または「インテル(R) Visual Fortran」) をクリックし「コンパイラー」を選択してください。(Visual Studio のバージョンによっては Intel コンパイラのオプションを見るために「すべての設定を表示」をクリックする必要があるかもしれません。) Intel コンパイラの新しいバージョンでは、「インテル(R) Composer XE > インテル(R) Visual Fortran > コンパイラー」を選択してください。
3. 右側のパネルの「ライブラリー」の右端の「...」ボタンをクリックしてください。
4. 「ディレクトリー・リストの設定」ウィンドウで NAG インポートライブラリの格納フォルダーのパスを追加してください。デフォルトは以下のようになります。

`C:¥Program Files¥NAG¥FS24¥fsw3224dcl¥lib`

5. MKL インポートライブラリの格納フォルダーのパスを追加する必要はありません。BLAS/LAPACK のシンボルは NAG インポートライブラリ (FSW3224DC.lib) からエクスポートされるからです。(もし「ライブラリー」パスに MKL ライブラリフォルダーを追加する場合は、「3.1. ライブラリーのリンク方法」で説明されている通り、NAG ライブラリフォルダーの後に追加する必要があります。)

6. 「ディレクトリー・リストの設定」ウィンドウで OK ボタンをクリックしてください。
7. 右側のパネルの「インクルード」の右端の「...」ボタンをクリックしてください。
8. 「ディレクトリー・リストの設定」ウィンドウで NAG インターフェースブロックの格納フォルダーのパスを追加してください。デフォルトは以下のようになります。

C:\Program Files\NAG\FS24\fsw3224dc\ nag\_interface\_blocks

9. 「ディレクトリー・リストの設定」ウィンドウで OK ボタンをクリックしてください。
10. 「オプション」ウィンドウで OK ボタンをクリックしてください。

上記の設定を行うことにより、Intel Fortran プロジェクトでコンパイル／リンクを行う際に、ライブラリおよび NAG インターフェースブロックをフルパスで指定する必要がなくなります。

上記の設定は、全ての Intel Fortran プロジェクトに適用されます。

下記の設定は、個々の Intel Fortran プロジェクトに対して行う必要があります。

本ライブラリはフル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 モードへの設定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。

以下の方法で NAG ライブラリをプロジェクトに追加することができます。

1. プロジェクトのプロパティページを開いてください。
  - ソリューションエクスプローラーでグループプロジェクト（一行目）が選択されていないことを確認して、メニューから「プロジェクト > \*\*\* のプロパティ」を選択する。
  - ソリューションエクスプローラーで特定のプロジェクトを右クリックして「プロパティ」を選択する。
  - ソリューションエクスプローラーで特定の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」ボタンを選択する。出現する「プロパティ」ウィンドウの「プロパティページ」ボタンを選択する。
2. プロパティの左側のパネルの「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」に必要に応じて適切なライブラリを追加します。本ライブラリのスタティック版をご利用になる場合は、「追加の依存ファイル」に FSW3224DC\_static.lib mkl\_rt.lib user32.lib を追加してください。これらのライブラリはそれぞれスペースで区切ることに注意してください。また、FSW3224DC\_static.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。同様に、本ライブラリの DLL 版をご利用になる場合は、「追加の依存ファイル」に FSW3224DC.lib を追加してください。
3. 呼び出し規約を CVF に設定する必要があります。プロパティの左側のパネルの「Fortran > 外部プロシージャ」を選択してください。右側のパネルの「呼び出し規約」をクリックし、ドロップダウンリストから「CVF (/iface:cvf)」を選択してください。この時、「文字長引数渡し」の設定が自動的に「個々の文字列引数の後」に変更されます。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。
4. 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。プロパティの左側のパネルの「Fortran > ライブラリー」を選択してください。右側のパネルの「ランタイム・ライブラリー」において、「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。

5. コンパイラが OpenMP コードをサポートするように、コンパイラオプション /Qopenmp を指定する必要があります。このオプションを有効にするには、プロパティの左側のパネルの「Fortran > 言語」を選択してください。右側のパネルの「OpenMP 宣言子の処理」において「並列コードの生成 (/Qopenmp)」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。
6. OK ボタンをクリックして、「プロパティページ」ウィンドウを閉じてください。

以上の設定を以下の表に要約します。プロジェクトの「追加の依存ファイル」と「ランタイム・ライブラリー」および環境変数 PATH は一貫していなければなりません。

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
FSW3224DC_static.lib mkl_rt.lib user32.lib	マルチスレッド DLL	install_dir¥ MKL_ia32_11.0¥bin
FSW3224DC.lib	マルチスレッド DLL	install_dir¥bin install_dir¥ MKL_ia32_11.0¥bin

本ライブラリは 32-bit ライブラリです。64-bit システムで実行する場合は、「構成マネージャー」の「アクティブソリューションプラットフォーム」を“Win32”に設定してください。

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。

「デバッグ」メニュー（例えば「デバックなしで開始」など）から、Visual Studio 上でプログラムを実行することができます。プログラムの実行に標準入出力のリダイレクションを伴う場合は、(Visual Studio 2005 以降では)「プロパティページ」の「デバック > コマンド引数」に必要なコマンド（実行時引数）を設定することができます。（注意：Visual Studio 2008 では、入出力リダイレクションが動作しない場合があります。）

例)

```
< input_file > output_file
```

入力ファイルがアプリケーションの作業ディレクトリ以外にある場合や、出力ファイルを作業ディレクトリ以外に出力したい場合は、フルパスまたは（作業ディレクトリからの）相対パスでファイルを指定してください。なお、オプションファイル (\*.opt) を伴う NAG ライブラリ Example をご利用の場合は、オプションファイル (\*.opt) は必ず作業ディレクトリに置いてください。なお、作業ディレクトリは「プロパティページ」の「デバック > 作業ディレクトリ」で変更することができます。

### 3.1.2. 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.

### 3.1.3. NAG Fortran Builder

本ライブラリの DLL 版（以下 NAG DLL と呼ぶ）は、NAG Fortran Builder（NAG Fortran コンパイラ）でもご利用いただけます。NAG Fortran Builder のバージョン 5.3 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。

コマンドプロンプトからご利用になる場合は、PATH 環境変数が正しく設定されていることを確認してください。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。）

```
nagfor -thread_safe -compatible -I"install_dir¥nag_interface_blocks_nagfor"  
-o driver.exe driver.f90 "install_dir¥lib¥FSW3224DC.lib"
```

NAG インポートライブラリ FSW3224DC.lib はフルパスで指定してください。パスがスペースを含む場合は、クォテーションで括弧する必要があります。

注意：-thread\_safe オプションは、ローカル変数が静的に割り当てられないようにします（「3.1. ライブラリのリンク方法」参照）。なお、これとは逆にローカル変数が静的に割り当てられるようになる -save オプションは使用しないでください。

注意：64-bit Windows でご利用の場合は、上記のコマンドに -abi32 オプションを加える必要があります。

統合開発環境（NAG Fortran Builder）からご利用になる場合は、次のように設定を行なってください。

Fortran Builder 6.0 以降をご利用の場合：

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. 「ビットモード > 32-bit」を選択する.
5. 「DLL 互換 (-compatible)」にチェックを入れる.  
(これにより、コンパイラオプション `-compatible` が有効になり、本ライブラリと同じ呼び出し規約 `stdcall` をコンパイラが使うようになります.)
6. 「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 インターフェー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、NAG インポートライブラリ `FSW3224DC.lib` が自動的にリンクされます.)
7. 「Fortran コンパイラ > 詳細設定(1)」タブを開く.
8. 「スレッドセーフなコードを生成する (-thread\_safe)」にチェックを入れる.

以上で NAG DLL を利用する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。

Fortran Builder 5.3.2 以前をご利用の場合 :

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. 「ビットモード > 32-bit」を選択する.
5. 「DLL 互換 (-compatible)」にチェックを入れる.  
(これにより、コンパイラオプション `-compatible` が有効になり、本ライブラリと同じ呼び出し規約 `stdcall` をコンパイラが使うようになります.)
6. 「ディレクトリ > インクルード」タブを開く.
7. 「インクルード」に  
`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.)
8. 「リンク > 基本設定」タブを開く.
9. 「リンクするライブラリ」に  
`install_dir¥lib¥FSW3224DC.lib`  
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.)
10. 「Fortran コンパイラ > 詳細設定(1)」タブを開く.
11. 「スレッドセーフなコードを生成する (-thread\_safe)」にチェックを入れる.

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.

### 3.1.4. Silverfrost (Salford) FTN95

NAG Library インターフェースブロックを利用する場合は（例えば、NAG ライブラリの Example プログラムを利用する場合など）、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（「3.2. インターフェースブロック」参照）。FTN95 でインターフェースブロックをコンパイルする際には、`/f_stdcall` オプションを付けて `STDCALL` 呼び出し規約が使われるようにしてください。

注意：インターフェースブロックのソースファイル `nag_precisions.f90` は、FTN95 でコンパイルする前に修正する必要があります。Fortran の `kind` 関数は、FTN95 では組込み関数として認識されません。 `kind` と `selected_int_kind` の `intrinsic` 宣言を削除し、他のパラメーターも次のように変更してください。

```
INTEGER, PARAMETER      :: HP = 2
INTEGER, PARAMETER      :: I4B = 3
INTEGER, PARAMETER      :: RP = 1
INTEGER, PARAMETER      :: WP = 2
```

更に、インターフェースブロックのソースファイル `nag_e_ib.f90` は、組込み関数 `max` の宣言のために、そのままでは FTN95 でコンパイルに通りません。 `max` の `intrinsic` 宣言を削除してください。

FTN95 は `cdecl` 呼び出し規約の一種をデフォルトで使っているため、`CVF` 呼び出し規約を使用する旨を明示的に指定する必要があります（`/import_lib` オプション）。以下のよう  
にコンパイル／リンクを行ってください。

```
ftn95 /f_stdcall /mod_path nag_interface_blocks_ftn95 driver.f90
      /import_lib "install_dir¥bin¥FSW3224DC.dll" /link
```

（ここでは、FTN95 で生成したインターフェースブロックのモジュールファイルが、`nag_interface_blocks_ftn95` フォルダーに置かれていると仮定しています。）  
`FSW3224DC.dll` はフルパスで指定する必要があります。パスにスペースが含まれる場合は、パスをクォーテーションで括ってください。

これらのオプションによって、DLL の全てのエクスポートされた名前は CVF STDCALL となり、それらを利用する場合には CVF STDCALL 呼び出し規約を用いなくてはならないことが明示されます。また、引数リストを通して NAG DLL ルーチンに渡される外部名は、同じソースファイルにあるかどうか自動的に判別されます。

以下のようにコンパイル／リンクすることもできます。

```
ftn95 /f_stdcall /mod_path nag_interface_blocks_ftn95 driver.f90  
slink driver.obj "install_dir¥bin¥FSW3224DC.dll"
```

パスにスペースが含まれる場合は、パスをクォーテーションで括ってください。  
リンカーは、\*.lib ファイルではなく DLL を直接リンクすることに注意してください。

FTN95 コンパイラの制限として、NAG ルーチンの引数として渡すルーチンが driver.f90 の内部でモジュールとして定義されている場合は STDCALL 属性が与えられません（リンク時または実行時にエラーとなります）。この場合 NAG ルーチンの引数として渡すルーチンを内部モジュールではなく外部ルーチンとして宣言してください。この制限は FTN95 version 6.00.0 で存在するようです。なお、NAG ライブラリの Example プログラムの多くは、このような内部モジュールを持っていることに注意してください。

また、別の制限として、FTN95 は FLUSH 文（Fortran 2003 から導入された機能）をサポートしていません。NAG ライブラリの Example プログラムのいくつかは FLUSH 文を使用しています。これらの Example プログラムをコンパイルに通すためには、FLUSH 文をコメントアウトする必要があります。

Plato は FTN95 と共に提供される統合開発環境（IDE）です。Plato から NAG DLL を利用する場合は、以下のように行ってください。

1. File メニューから New Project を選択してください。
2. Fortran Application を選択して、プロジェクト名と格納場所を設定してください。
3. Project Explorer ウィンドウでソースファイルを右クリックして、プロジェクトを構成するファイルを入力してください（例えば、NAG ライブラリの Example プログラム e01bafe.f90）。

4. Project Explorer ウィンドウの References を右クリックして、NAG DLL の名前を入力してください（例えば、FSW3224DC.dll）。
5. FTN95 version 5.2 以前のバージョンで NAG STDCALL DLL をご利用になる場合は、Checkmate オプションを使わないでください（Checkmate オプションを付けると正常に動作しません）。ツールバーのドロップダウンメニューから、例えば Release Win32 を選択してください。
6. Project > Properties > Compiler Options > Miscellaneous > Extra compiler options プロパティで /F\_STDCALL を指定してください。または、Project Explorer ウィンドウの References の NAG DLL を右クリックして、Properties の STDCALL にチェックを入れてください。
7. NAG インターフェースブロックを利用する場合は、Project > Properties > Compiler Options > Source > Specify MODULE paths にモジュールファイルの格納場所を入力してください。（モジュールファイルの生成については、本セクションの最初をご参照ください。）
8. Plato は標準入出力のリダイレクションをサポートしていません。従って、入出力の際には、メインプログラムの中で明示的にファイルをオープンする必要があります。例えば、計算結果を c:\test.res ファイルに書き出す場合（装置番号 6 で write 文を使用する前に）、以下のように open 文でファイルをオープンしてください。

```
open(6, file=' c:\test.res')
```

### 3.1.5. PGI Fortran

NAG Fortran インターフェースブロックを利用する場合は（例えば、NAG ライブラリの Example プログラムを利用する場合など）、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（「3.2. インターフェースブロック」参照）。

pgf90 から NAG DLL を利用する場合は、以下のようにコンパイル／リンクを行ってください。ただし、本ライブラリのインストール先を考慮して環境変数 LIB と PATH が正しく設定されていることが前提となります。

```
pgf90 -Mrecursive driver.f90 -module nag_interface_blocks_pgi FSW3224DC.lib  
-o driver.exe
```

ここで、nag\_interface\_blocks\_pgi はユーザーがコンパイルしたモジュールファイルの格納フォルダーを表しています。-Mrecursive オプションは、ローカル変数が静的にアロケートされないようにします（「3.1. ライブラリのリンク方法」参照）。従って、これとは逆にローカル変数が静的にアロケートされるようになる -Msave オプションは使用しないでください。

上記のコマンドは PGI pgf90 コンパイラの version 13.4 で検証されています。

### 3.1.6. Microsoft Visual C++ / Visual Studio

NAG DLL は Microsoft Visual Studio .NET 2003 以降の 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 `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 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¥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に提供されます。

すべてのプログラム | NAG | FL24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
Calling FSW3224DCL from C & C++

キーとなる情報：

- STDCALL 呼び出し規約を指定する。
- 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, 以下のようにコンパイル/リンクを行ってください. 下記のコマンドは NAG 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を前提としています.

ここで driver.c はユーザープログラムです.

```
cl driver.c FSW3224DC.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 FSW3224DC.lib
```

以下の説明は Visual Studio .NET 2003, Visual Studio 2005, Visual Studio 2008 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 IDE (統合開発環境) をご利用の場合は, リンクが行えるようにするために以下の設定を行ってください. プロジェクトのプロパティから「構成プロパティ > リンカー > 入力 > 追加の依存ファイル」に FSW3224DC.lib を追加してください. もし環境変数 LIB が他で設定されていなければ, プロジェクトのプロパティから「構成プロパティ > リンカー > 全般 > 追加のライブラリディレクトリ」に install\_dir¥lib を追加してください. もし nagmk24.h のローカルコピーを使用しないなら, プロジェクトのプロパティから「構成プロパティ > C/C++ > 全般 > 追加のインスクリードディレクトリ」に install\_dir¥c\_headers を追加してください.

### 3.1.7. Intel C/C++

上記「3.1.6. Microsoft Visual C++ / Visual Studio」の情報（ヘッダーファイルなど）は Intel C についても同様に当てはまります。

C プログラムから NAG DLL をご利用になる場合は、以下のようにコンパイル／リンクを行ってください。下記のコマンドは NAG 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を前提としています。

```
icl /I"install_dir%c_headers" driver.c FSW3224DC.lib
```

### 3.1.8. Microsoft Visual Basic for Applications / Excel (32-bit)

NAG DLL 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 (VBA) コードから呼び出すことができます。(VBA と Visual Basic 6 (VB 6) は類似しており、NAG ルーチンに対して両方で同じ Declare 文が使われます。) 本セクションの情報は Excel の 32-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¥excel_examples
```

install\_dir¥samples¥excel\_examples¥linear\_algebra¥xls\_demo.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。また、VB 6 / VBA からの NAG DLL 呼び出しの具体例として VB 6 の Example もご参照ください。

キーとなる情報：

- install\_dir¥vb6\_headers フォルダの flvb6-〈チャプター名〉.bas (例えば flvb6-a.bas) ファイルには VB 6 / VBA 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 flvb6-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 flvb6-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。
- 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 VBA 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 flvb6-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。
- 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、VBA 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。文字長引数は対応する文字引数の直後に置く必要があります。

以上の情報は Microsoft Office Excel 2003, 2007, 2010 で検証されています。

### 3.1.9. Microsoft Visual Basic 6

Visual Basic 6 (VB 6) と Visual Basic for Applications (VBA) は類似しているため、上記の VBA についての情報のほとんどは VB 6 にも当てはまります。特に、配列と文字列の取り扱いにご注意ください。

NAG DLL を Visual Basic 6 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¥vb6_examples
```

また、VB 6 / VBA からの NAG DLL 呼び出しの具体例として Excel の Example の VBA コードもご参照ください。

キーとなる情報：

- `install_dir¥vb6_headers` フォルダの `flvb6-〈チャプター名〉.bas` (例えば `flvb6-a.bas`) ファイルには VB 6 / VBA 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 `flvb6-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 `flvb6-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。
- `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 6 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 `flvb6-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。
- 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、VB 6 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。文字長引数は対応する文字引数の直後に置く必要があります。

以上の情報は Microsoft Visual Basic 6.0 で検証されています。

### 3.1.10. 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.net_examples
```

注意：これらの Example は Visual Studio .NET 2003 で生成されています。Visual Studio 2005 以降でロードした場合は、ソリューションとプロジェクトファイルは Visual Studio 変換ウィザードでコンバートされます。

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.net_headers¥flvbdnet.vb
```

- 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。
- 型マッピングは VB 6 の場合とは異なります。  
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 (Mark 22 や VB 6 のように配列の最初の要素ではありません)。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 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。具体例として, d02ejf\_example のコードをご参照ください。
- VB.NET の配列は行優先です。一方で VB 6 と 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, 文字列引数の直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, 文字列引数の直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 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, d01bdf\_example のコード, または various\_routines 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 .NET 2003, 2005, 2008, 2010, 2012 で検証されています。

もし、64-bit システムで実行する場合は、BadImageFormatException を避けるために、ターゲット CPU を x86 に設定してください。

### 3.1.11. C#

C# からの NAG DLL のご利用方法については、以下のウェブサイトをご参照ください。

<http://www.nag.co.uk/numeric/csharpinfo.asp>

C# から NAG DLL 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 C# コンパイラを用いて以下のようなコマンドでご利用いただけます。

```
csc driver.cs
```

(DLL の名前は Example ファイルの中に書き込まれています。)

64-bit Windows マシンでは、`/platform:x86` フラグを加える必要があります。

もしくは、NAG Library for .NET (.NET 用の NAG ライブラリ) もございます。

詳細は以下のウェブサイトをご覧ください。

[http://www.nag.co.uk/microsoft\\_dotnet.asp](http://www.nag.co.uk/microsoft_dotnet.asp)

### 3.1.12. Java

Java からの NAG DLL のご利用方法については、以下のウェブサイトをご参照ください。

[http://www.nag.co.uk/doc/TechRep/html/Tr2\\_09/Tr2\\_09.asp](http://www.nag.co.uk/doc/TechRep/html/Tr2_09/Tr2_09.asp)

もしくは、NAG Library for Java (NAG ライブラリの Java ラッパー) をご利用いただく方がより簡単です。詳細は以下のウェブサイトをご覧ください。

<http://www.nag.co.uk/nag-library-for-java>

### 3.1.13. Python

F2PY を使用した Python からの NAG DLL のご利用方法については、以下の PDF ファイルをご参照ください。

[http://www.nag.co.uk/doc/TechRep/pdf/TR1\\_08.pdf](http://www.nag.co.uk/doc/TechRep/pdf/TR1_08.pdf)

### 3.1.14. R

R からの NAG DLL のご利用方法については、以下のウェブサイトをご参照ください。

<http://www.nag.co.uk/numeric/RunderWindows.asp>

### 3.1.15. Simfit (NAG Library editon)

Simfit (NAG Library edition) は NAG DLL をアルゴリズムソースとして使用しています。詳細は以下のウェブサイトをご覧ください。

<http://www.nag.co.uk/educationuk/simfit.asp>

### 3.1.16. 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方法については、以下のリリース後の最新情報をご参照ください。

<http://www.nag.co.uk/doc/inun/fs24/w32dcl/postrelease.html>

もしくは、日本 NAG までお問い合わせください。

### 3.1.17. アクセスチェック

診断プログラム NAG\_Fortran\_DLL\_info.exe を用いて、ご利用のマシン環境から本ライブラリの DLL 版 (FSW3224DC.dll) にアクセスできるかどうかを確認することができます。診断プログラムは以下の「スタート」メニューのショートカットから実行することができます。

すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
Check NAG DLL Accessibility for FSW3224DCL

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「4.2.3. アクセスチェック」をご参照ください。

## 3.2. インターフェースブロック

NAG 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 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) 用に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、「3.1.1.1. 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。）

まずは、オリジナルのモジュールファイルのバックアップを取ってください。例えば、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。この例では、Intel コンパイラ ifort を使用しています。他のコンパイラをご利用の場合は、ifort /iface:cvf の部分をご利用のコンパイラの名前と必要なコンパイラオプションで置き換えてください。

```
ifort /iface:cvf -c nag_precisions.f90
ifort /iface:cvf -c nag_a_ib.f90
ifort /iface:cvf -c nag_blast_ib.f90
ifort /iface:cvf -c nag_blas_consts.f90
ifort /iface:cvf -c nag_blas_ib.f90
ifort /iface:cvf -c nag_c_ib.f90
ifort /iface:cvf -c nag_d_ib.f90
ifort /iface:cvf -c nag_e_ib.f90
ifort /iface:cvf -c nag_f_ib.f90
ifort /iface:cvf -c nag_g_ib.f90
ifort /iface:cvf -c nag_h_ib.f90
ifort /iface:cvf -c nag_lapack_ib.f90
ifort /iface:cvf -c nag_m_ib.f90
ifort /iface:cvf -c nag_omp_ib.f90
ifort /iface:cvf -c nag_s_ib.f90
ifort /iface:cvf -c nag_w_ib.f90
ifort /iface:cvf -c nag_x_ib.f90
ifort /iface:cvf -c nag_long_names.f90
ifort /iface:cvf -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、本ライブラリの DLL 版 FSW3224DC.dll を用いて、および、OMP\_NUM\_THREADS を 1 に設定して、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。

以下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。

- nagsmp\_example\_dll.bat  
本ライブラリの DLL インポートライブラリ FSW3224DC.lib をリンクします。  
実行は、本ライブラリの DLL 版 FSW3224DC.dll を利用します。
- nagsmp\_example\_static.bat  
本ライブラリのスタティック版 FSW3224DC\_static.lib をリンクします。

これらのバッチファイルは環境変数 NAG\_FSW3224DCL を参照します。

インストーラーは「スタート」メニューに以下のショートカットを作成します。

```
すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
FSW3224DCL Command Prompt
```

このショートカットは必要な環境変数（NAG\_FSW3224DCL を含む）を設定した上でコマンドプロンプトを開きます。

このショートカットを利用しない場合には、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場所を以下に示します。

```
C:\Program Files\NAG\FSW3224DC\batch\envvars.bat
```

バッチファイル nagsmp\_example\*.bat 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ーにコピーして、コンパイル／リンク／実行を行います。バッチファイルの引数に、ご利用の NAG ライブラリルーチンの名前と OpenMP スレッド数を指定してください。

本ライブラリのスタティック版 FSW3224DC\_static.lib（および MKL のスタティック版）をリンクする場合は、nagsmp\_example\_static.bat をご利用ください。

例)

```
nagsmp_example_static e04ucf 4
```

この例では、e04ucfe.f（ソースファイル）と e04ucfe.d（データファイル）をカレントフォルダーにコピーして、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。

同様に nagsmp\_example\_dll.bat を利用することができます。

例)

```
nagsmp_example_dll e04ucf 4
```

nagsmp\_example\_dll.bat は、本ライブラリの DLL インポートライブラリ FSW3224DC.lib をリンクします。

### 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。

CHAPTER 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依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。

*block 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 に記載されているパラメーター WORK と LWORK についての説明をご参照ください。

### 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. C06

以下の NAG ルーチンは可能な限り本製品で提供される MKL ライブラリから Intel Discrete Fourier Transforms Interface (DFTI) ルーチンを呼び出して使います。

C06PAF C06PCF C06PFF C06PJF C06PKF C06PPF C06PQF C06PRF  
C06PSF C06PUF C06PVF C06PWF C06PXF C06PYF C06PZF C06RAF  
C06RBF C06RCF C06RDF

Intel DFTI ルーチンは必要なワークスペースを自身で内部的に割り当てます。従って、上記の NAG C06 ルーチンの引数 WORK（ワークスペース配列）のサイズは、それぞれの Routine Document に示されている値で十分です（変更の必要はありません）。

##### b. C09

Intel コンパイラの現行バージョンの制限により、以下のルーチンは本ライブラリではシリアルです。

C09FAF C09FBF C09FCF C09FDF

##### c. 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。

本ライブラリでは、BLAS/LAPACK ルーチンは MKL 提供のものが使われます。

ただし、以下のルーチンは NAG 提供のものが使われます。

DBDSDC DGEES DGEESX DGERFS DGGES DGGESX  
DGGEVX DSBEV DSBEVX ZGEES ZGEESX ZGGES  
ZGGESX ZHBEV ZHBEVX ZTRSEN

以下の NAG 名のルーチンは MKL から LAPACK ルーチンを呼び出すためのラッパーです。

F07ADF/DGETRF	F07AEF/DGETRS	F07ARF/ZGETRF	F07ASF/ZGETRS
F07AVF/ZGERFS	F07BDF/DGBTRF	F07BEF/DGBTRS	F07BHF/DGBRFS
F07BRF/ZGBTRF	F07BSF/ZGBTRS	F07BVF/ZGBRFS	F07CHF/DGTRFS
F07CVF/ZGTRFS	F07FDF/DPOTRF	F07FEF/DPOTRS	F07FHF/DPORFS
F07FJF/DPOTRI	F07FRF/ZPOTRF	F07FSF/ZPOTRS	F07FVF/ZPORFS
F07GEF/DPPTRS	F07GHF/DPPRFS	F07GSF/ZPPTRS	F07GVF/ZPPRFS
F07HEF/DPBTRS	F07HHF/DPBRFS	F07HSF/ZPBTRS	F07HVF/ZPBRFS
F07JHF/DPTRFS	F07JVF/ZPTRFS	F07MHF/DSYRFS	F07MVF/ZHERFS
F07NVF/ZSYRFS	F07PHF/DSPRFS	F07PVF/ZHPRFS	F07QVF/ZSPRFS
F07THF/DTRRFS	F07TVF/ZTRRFS	F07UEF/DTPTRS	F07UHF/DTPRFS
F07USF/ZTPTRS	F07UVF/ZTPRFS	F07VEF/DTBTRS	F07VHF/DTBRFS
F07VSF/ZTBTRS	F07VVF/ZTBRFS	F08AEF/DGEQRF	F08AFF/DORGQR
F08AGF/DORMQR	F08ASF/ZGEQRF	F08ATF/ZUNGQR	F08AUF/ZUNMQR
F08FEF/DSYTRD	F08FFF/DORGTR	F08FSF/ZHETRD	F08FTF/ZUNGTR
F08GFF/DOPGTR	F08GTF/ZUPGTR	F08JEF/DSTEQR	F08JJF/DSTEBZ
F08JKF/DSTEIN	F08JSF/ZSTEQR	F08JXF/ZSTEIN	F08KEF/DGEBRD
F08KSF/ZGEBRD	F08MEF/DBDSQR	F08MSF/ZBDSQR	F08NEF/DGEHRD
F08NGF/DORMHR	F08NSF/ZGEHRD	F08PEF/DHSEQR	F08PKF/DHSEIN
F08PSF/ZHSEQR	F08PXF/ZHSEIN	F08TAF/DSPGV	F08TBF/DSPGVX
F08TCF/DSPGVD	F08TNF/ZHPGV	F08TPF/ZHPGVX	F08TQF/ZHPGVD

d. 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。

e. S07 - S21

これらのチャプターの関数は、不正な引数で呼び出された場合にエラーメッセージを出力します。

ライブラリマニュアルに記載されている定数は、本ライブラリでは以下に示す値を持ちます。

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  $\text{abs}(X) < 2.23E-308$   
 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  $X \geq 2.53E+307$   
 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  $\text{abs}(Z) < 3.92223\text{E}-305$   
IFAIL = 4 if  $\text{abs}(Z)$  or  $\text{FNU}+\text{N}-1 > 3.27679\text{E}+4$   
IFAIL = 5 if  $\text{abs}(Z)$  or  $\text{FNU}+\text{N}-1 > 1.07374\text{E}+9$

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

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

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

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

S18ADF IFAIL = 2 if  $0 < X \leq 2.23\text{E}-308$

S18AEF IFAIL = 1 if  $\text{abs}(X) > 7.116\text{E}+2$

S18AFF IFAIL = 1 if  $\text{abs}(X) > 7.116\text{E}+2$

S18DCF IFAIL = 2 if  $\text{abs}(Z) < 3.92223\text{E}-305$   
IFAIL = 4 if  $\text{abs}(Z)$  or  $\text{FNU}+\text{N}-1 > 3.27679\text{E}+4$   
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  $\geq 1.407\text{E}+102$

f. X01

数学定数は以下のとおりです.

X01AAF (pi) = 3.1415926535897932

X01ABF (gamma) = 0.5772156649015328

g. 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

h. 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.

i. OpenMP 並列領域からユーザー関数を呼び出すルーチン

本ライブラリでは、以下の NAG ルーチンはルーチン内の OpenMP 並列領域からユーザー関数を呼び出します。

D03RAF

D03RBF

E05SAF

E05SBF

E05UCF

E05USF

F01ELF

F01EMF

F01FLF

F01FMF

F01JBF

F01JCF

F01KBF

F01KCF

従って、本ライブラリの製造に使用されたものと同じ OpenMP ランタイムライブラリ（これは通常、同じコンパイラを意味します）を使用していない場合は、ユーザー関数内で OpenMP プログラム（指示文）を用いるべきではありません。また、ユーザー用のワークスペース配列 IUSER と RUSER もスレッドセーフである必要があります。これらの配列は読み取り専用のデータをユーザー関数に与えるためにだけ使用するのがベストです。

## 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/FSdocumentation.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

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」メニューから開くことができます。

すべてのプログラム | NAG | Mark 24 Manual |

NAG Library Manual Mk24 (HTML5)

NAG Library Manual Mk24 (PDF)

NAG Library Manual Mk24 (PDF + HTML Index)

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。

また、HTML ヘルプ形式のライブラリマニュアル (nagdoc\_f124.chm) が提供されます。数式を正しく表示するために Design Science 社の MathPlayer プラグインが必要です。ライブラリマニュアルのインストールの最後で MathPlayer をインストールするかどうかを選択することができます。また、MathPlayer は Design Science 社のウェブサイトからダウンロードすることもできます。

<http://www.dessci.com/en/products/mathplayer/download.htm>

HTML ヘルプ形式のライブラリマニュアル (nagdoc\_f124.chm) は「スタート」メニューから開くことができます。

すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
NAG Library Manual Mk24 HTML Help

HTML ヘルプ形式のライブラリマニュアル (nagdoc\_f124.chm) をネットワークドライブからローカルドライブにコピーした場合などに、

“Web ページへのナビゲーションは取り消されました”

というメッセージが表示される場合は、Windows または Internet Explorer のセキュリティアップデートによってファイルがブロックされている状態です。これを回避するには、nagdoc\_f124.chm を右クリックして表示されるポップアップメニューからプロパティを開いてください。次に、プロパティの下方にある「ブロックの解除」ボタンをクリックしてください。最後に、OK ボタンをクリックしてプロパティを閉じてください。

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)

ユーザーノート (英語版) は「スタート」メニューから開くことができます。

すべてのプログラム | NAG | FS24 | NAG Library for SMP and Multicore (FSW3224DCL) |  
FSW3224DCL Users' Note

## 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（FSW3224DCL）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](mailto: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 CONTACT 先にご連絡ください。

### CONTACT 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](mailto: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。