

## ディレクトリ-ファイル クロスリファレンス

### 1. ディレクトリ×ファイル

FrontISTR Ver.3.4 のディレクトリ構成を簡単に説明する. ファイル FrontISTR\_V42.tar.gz を解凍すると, ディレクトリ「FrontISTR」が作成される. ディレクトリ「FrontISTR」の下にはディレクトリ「fistr1」があり, ディレクトリ「fistr1」の下には FrontISTR Ver.3.4 のソースコードが含まれているディレクトリ「src」がある. ディレクトリ「src」の下には, 四

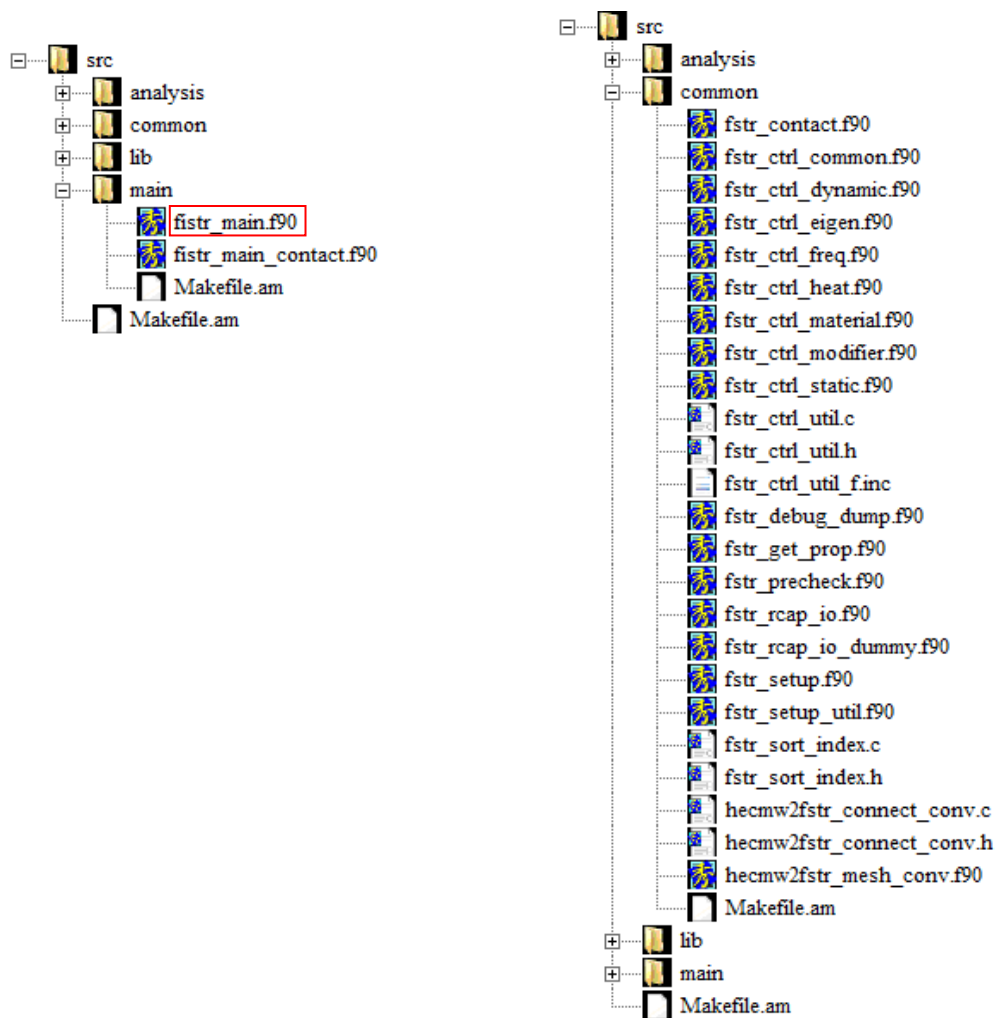


Fig.1 Structure of directories in FrontISTR (fistr1)

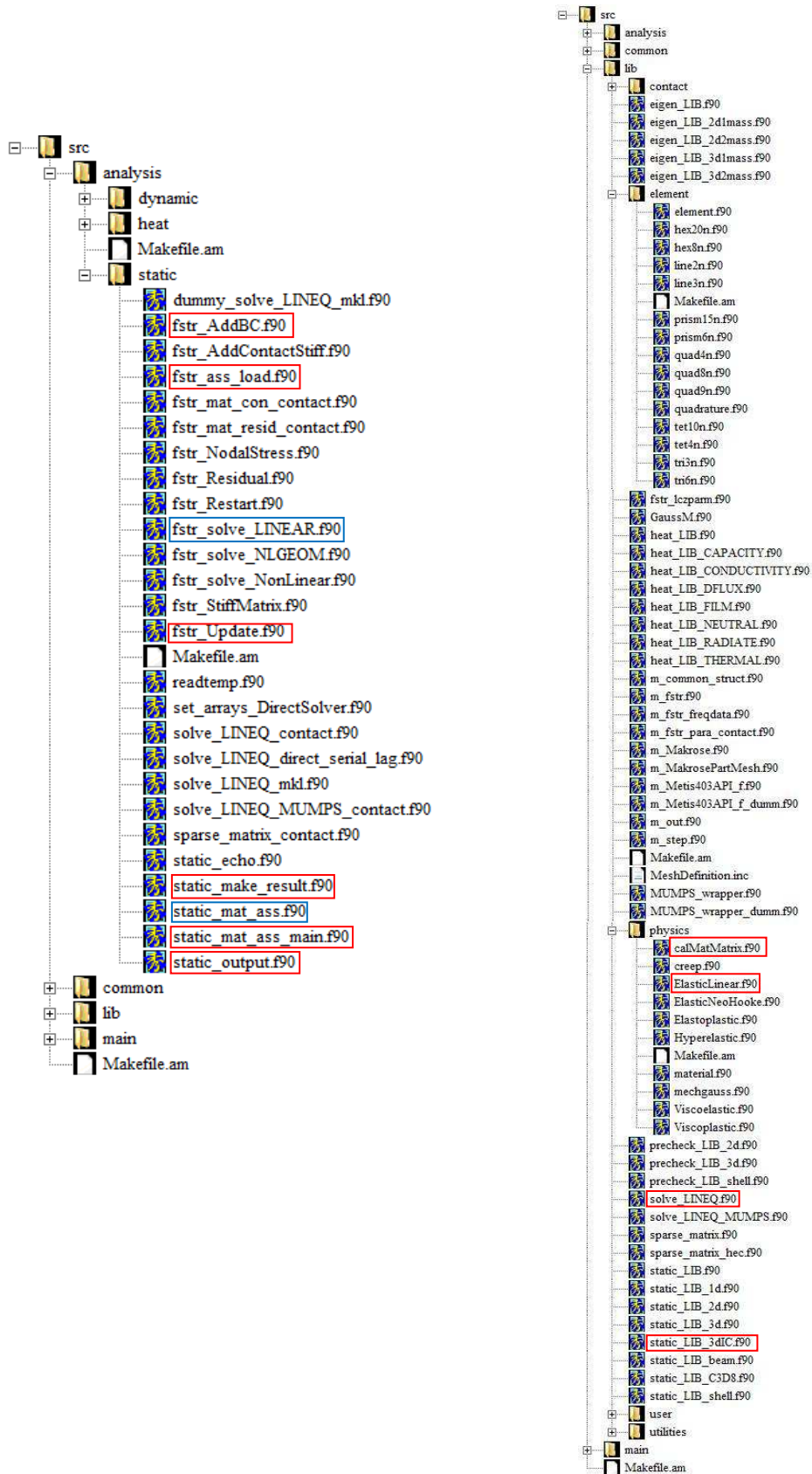


Fig.1 (continued)

つのディレクトリ, すなわち

ディレクトリ「main」,  
ディレクトリ「common」,  
ディレクトリ「analysis」,  
ディレクトリ「lib」

がある (Fig.1 参照).

ディレクトリ「main」はメインプログラム「fistr\_main.f90」を含み, 「common」はファイルの読み込み関係のプログラム, FrontISTR 内のグローバルな変数を管理するプログラムなどを含む. ディレクトリ「analysis」には,

静解析用プログラムが入ったディレクトリ「static」,  
動解析用プログラムが入ったディレクトリ「dynamic」,  
伝熱解析用プログラムが入ったディレクトリ「heat」

が含まれる. ディレクトリ「dynamic」の下には,

固有値解析用プログラムが入ったディレクトリ「mode」,  
周波数応答用プログラムが入ったディレクトリ「freq」,  
時間積分による解析用プログラムが入ったディレクトリ「transit」

がある.

ディレクトリ「lib」には, 要素の B マトリックス, D マトリックス, 剛性マトリックス (または, 接線剛性マトリックス), 内力を計算するプログラムが含まれる. ここで, B マトリックスの計算に対して, ディレクトリ「element」に含まれる有限要素の幾何情報を計算するプログラムが使用される. また, D マトリックスの計算に対して, ディレクトリ「physics」に含まれる材料情報を計算するプログラムが使用される.

## 2. ファイル×サブルーチン

FrontISTR の機能を拡張していくためには, 各プログラムの内容を理解することが重要である. FrontISTR Ver.3.4 のプログラムは, Fortran 90 (一部 C 言語) で書かれている. FrontISTR Ver.3.4 の Fortran 90 プログラムはモジュール単位で作成されていて, 各モジュールにはメンバ関数として複数のサブルーチンや関数が含まれる.

以下では, 微小変形静弾性解析 (解析制御ファイル\*.cnt 内で TYPE=STATIC のように設定) で使用する主なプログラムについて説明する. ただし, 3次元六面体8節点要素を使用する場合を考える.

- main/fistr\_main.f90 : メインプログラム

プログラム名は「fistr\_main」である. データの初期化と削除, データの入力, 線形解析

/非線形解析/固有値解析/伝熱解析の条件分岐などを行う。

- **analysis/static/static\_mat\_ass\_main.f90** : モジュールプログラム  
モジュール名は、「m\_static\_mat\_ass\_main」である。全体剛性マトリックスを作成する。
- **lib/solve\_LINEQ.f90** : モジュールプログラム  
モジュール名は、「m\_solve\_LINEQ」である。連立一次方程式を直接法または反復法で計算するサブルーチンと呼ぶ。
- **analysis/static/fstr\_Update.f90** : モジュールプログラム  
モジュール名は、「m\_fstr\_Update」である。応力とひずみを計算するサブルーチンと呼ぶ。微小変形の場合、**lib/static\_LIB\_3dIC.f90** のサブルーチン `m_static_LIB_3dIC :: UpdateST_C3D8IC()` を呼ぶ。
- **analysis/static/static\_output.f90** : モジュールプログラム  
モジュール名は、「m\_static\_output」である。ログファイルに、節点での変位、反力、応力を出力する。
- **analysis/static/static\_make\_result.f90** : モジュールプログラム  
モジュール名は、「m\_static\_make\_result」である。結果ファイル用の変位、応力、ひずみのデータを作成する。
- **analysis/static/fstr\_AddBC.f90** : モジュールプログラム  
モジュール名は、「m\_fstr\_AddBC」である。変位境界条件を与える。
- **analysis/static/fstr\_ass\_load.f90** : モジュールプログラム  
モジュール名は、「m\_fstr\_ass\_load」である。節点集中荷重、面分布荷重、温度による荷重を与える。
- **lib/static\_LIB\_3dIC.f90** : モジュールプログラム  
モジュール名は、「m\_static\_LIB\_3dIC」である。微小変形の場合、3次元六面体8節点要素(非適合要素)に関するBマトリックスおよび要素剛性マトリックスの計算をサブルーチン `m_static_LIB_3dIC :: STF_C3D8IC()` で行う。`m_static_LIB_3dIC :: STF_C3D8IC()` では、**lib/physics/calMatMatrix.f90** のサブルーチン `MatlMatrix()` を呼ぶ。
- **lib/physics/calMatMatrix.f90** : モジュールプログラム  
モジュール名は、「m\_MatMatrix」である。各材料のDマトリックスを計算するサブルーチンと呼ぶ。サブルーチン `m_MatMatrix :: MatlMatrix()` では、材料の種類で条件分岐がある。線形弾性体の場合には、**lib/physics/ElasticLinear.f90** のサブルーチン `m_ElasticLinear :: calElasticMatrix()` を呼ぶ。
- **lib/physics/ElasticLinear.f90** : モジュールプログラム  
モジュール名は、「m\_ElasticLinear」である。サブルーチン `m_ElasticLinear :: calElasticMatrix()` では、剛性マトリックスの計算に必要となるDマトリックスの計算を行う。
- **analysis/static/fstr\_solve\_LINEAR.f90** : モジュールプログラム

モジュール名は、「m\_fstr\_solve\_LINEAR」である。全体剛性マトリックスの作成/荷重の設定/境界条件の設定を行うサブルーチン, 連立一次方程式を計算するサブルーチン, 計算結果を出力するサブルーチンを呼ぶ。

- [analysis/static/static\\_mat\\_ass.f90](#) : モジュールプログラム

モジュール名は、「m\_static\_mat\_ass」である。全体剛性マトリックスを作成するサブルーチン, 荷重を設定するサブルーチン, 境界条件を設定するサブルーチンを行う。

### 3. 静弾性解析に関するサブルーチンの流れ

微小変形静弾性解析 (解析制御ファイル\*.cnt内でTYPE=STATICと設定) の流れについて Fig.2 に示す。ただし, 3次元六面体8節点要素を使用する場合である。

[fistr\_main.f90] fstr\_main  
(a) hecmw :: hecmw\_init()  
(b) hecmw :: hecmw\_get\_mesh()  
(1.1) [fistr\_main.f90] fstr\_main :: fstr\_init()  
(c) hecmw :: hecmw\_nullify\_matrix()  
(d) hecmw :: hecmw\_nullify\_result\_data()  
(2.1) [fistr\_main.f90] fstr\_main :: fstr\_init\_file()  
(e) hecmw :: hecmw\_mat\_con()  
(2.2) [fistr\_main.f90] fstr\_main :: fstr\_condition()  
(f) hecmw :: hecmw\_ctrl\_get\_control\_file()  
(1.2) [fistr\_main.f90] fstr\_main :: fstr\_linear\_static\_analysis()  
(2.3) [fstr\_solve\_LINEAR.f90] m\_fstr\_solve\_LINEAR :: fstr\_solve\_LINEAR()  
(3.1) [static\_mat\_ass.f90] m\_static\_mat\_ass :: fstr\_mat\_ass()  
(4.1) [static\_mat\_ass\_main.f90] m\_static\_mat\_ass\_main::fstr\_mat\_ass\_main()  
(g) hecmw :: hecmw\_mat\_clear()  
(5.1) [static\_mat\_ass\_main.f90] m\_static\_mat\_ass\_main :: fstr\_local\_stf\_create()  
(6.1) [static\_LIB\_3dIC.f90] m\_static\_LIB\_3dIC :: STF\_C3D8IC()  
(7.1) [calMatMatrix.f90] m\_MatMatrix :: MatlMatrix()  
(8.1) [ElasticLinear.f90] m\_ElasticLinear :: calElasticMatrix()  
(h) hecmw :: hecmw\_mat\_ass\_elem()  
(4.2) [fstr\_ass\_load.f90] m\_fstr\_ass\_load :: fstr\_ass\_load()  
(4.3) [fstr\_AddBC.f90] m\_fstr\_AddBC :: fstr\_AddBC()  
(i) hecmw :: hecmw\_allREDUCE\_R1()  
(3.2) [solve\_LINEQ.f90] m\_solve\_LINEQ :: solve\_LINEQ()  
(j) hecmw :: hecmw\_solve\_33()  
(k) hecmw :: hecmw\_update\_3\_R()  
(3.3) [fstr\_Update.f90] m\_fstr\_Update :: fstr\_Update3D()  
(4.5) [static\_LIB\_3dIC.f90] m\_static\_LIB\_3dIC :: UpdateST\_C3D8IC()  
(4.6) [static\_LIB\_3dIC.f90] m\_static\_LIB\_3dIC :: STF\_C3D8IC()  
(3.4) [static\_output.f90] m\_static\_output :: fstr\_static\_Output()  
(4.7) [static\_make\_result.f90] m\_static\_make\_result :: fstr\_write\_static\_result()  
(1.3) [fistr\_main.f90] fstr\_main :: fstr\_finalize()  
(l) hecmw :: hecmw\_finalize()

**Fig.2** Flowchart of programs for linear static analysis (C3D8IC) in FrontISTR