MYSTRAN FEA Program File Structure

1. Overview of the file structure

Prior to giving the file structure that exists on the MYSTRAN.zip file, a little overview of the way the MYSTRAN source code is laid out is in order. The main program is called MYSTRAN and it calls several major subroutines, called LINK's, to do the work involved in processing the input data, formulating the matrices, solving for the unknowns and processing the output requests. Below is a list of these LINK's with an overview of what each one does:

- **LINK0**:
  - Read input data file and check for errors and possible restart
  - Process grid and coordinate system input data
  - Process Case Control output requests
  - Forms degree of freedom (DOF) tables
  - Process concentrated mass input data
  - Calculate rigid body mass (Grid Point Weight Generator , or GPWG)
  - Process temperature and pressure load input data intp arrays needed for element load calculations
- **LINK1**:
  - Process MPC's and rigid elements into sparse G-set[1] array RMG
  - Process all applied forces (including grid forces and moments, gravity, pressure, thermal, centrifugal, scalar) into sparse G-set load array PG
  - Formulate the G-set sparse stiffness and mass arrays KGG and MGG
  - Formulate the G-set sparse differential stiffness array KGGD. Differential stiffness code is incomplete. The only element coded is the BAR.

[1] See Section 3.6 of the MYSTRAN Users Reference manual for a discussion of displacement set notation (which is the same as, or a subset of, that used in the NASTRAN program)

- **LINK2**:
  - Reduce the G-set stiffness, mass, load and constraint matrices to the L-set
- **LINK3**: (for statics problems only):
  - Solve for the L-set displacements
- **LINK4**: (for eigenvalue problems only):
  - Solve for the L-set eigenvalues and eigenvectors
- **LINK5**:
  - Build the A-set displacements back up to the G-set through use of the constraint matrices
- **LINK6**: (for Craig-Bampton model generation only):
  - Builds a Craig-Bampton model from the input physical model
- **LINK9**:
  - Use the G-set displacements from LINK5 to solve for the outputs requested in Case Control.

With the above as background Figure 1, below, shows the file structure for the MYSTRAN-FEA-Program.zip file.

Figure 1 MYSTRAN FEA program file structure

Fig 1-a
Overall view

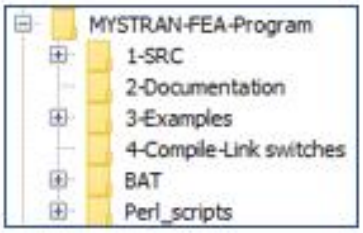
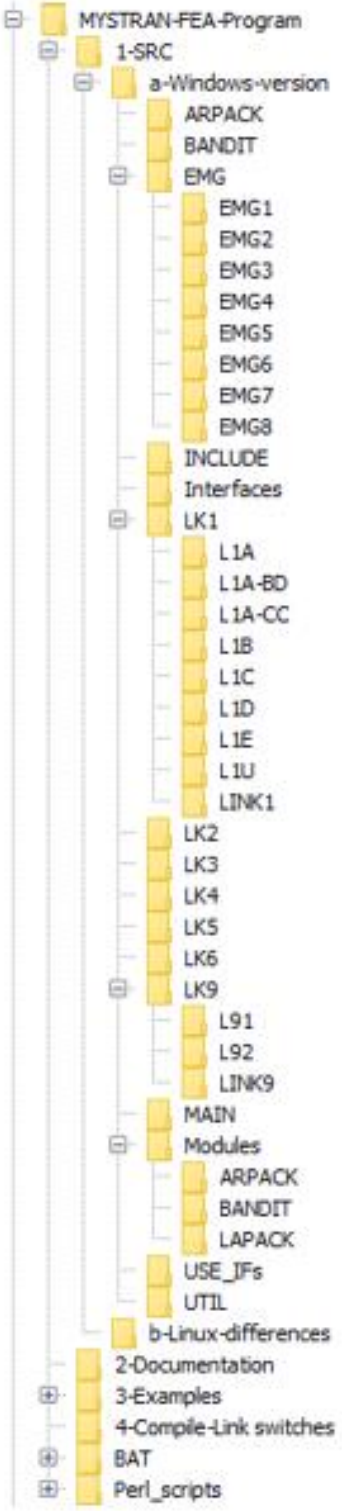


Fig 1-b
Expanded view



2. MYSTRAN file structure

The major subfolders in the MYSTRAN FEA Program are shown in Figure 1-a:

1. **1-SRC**: containing all of the source code
2. **2-Documentation**: containing the program documentation
3. **3-Examples**: consisting of approximately 50 test problems (statics and eigenvalues)
4. **4-Compile-link switches:** the Lahey fortran switches used to compile and link MYSTRAN by the author

Figure 1-b shows the expanded 1-SRC subfolder which is the main focus of this document. This subfolder is discussed in detail below.

1-SRC

a. **Windows version**

- **ARPACK**: contains a subroutine called when the ARPACK eigenvalue routine writes messages
- **BANDIT**: contains some of the code used in the Bandit grid sequencing algorithm (see reference in the MYSTRAN User's Reference Manual)
- **EMG**: contains all of the element generation routines. Generally, element routines generate stiffness, mass, thermal and pressure load, stress/strain recovery matrices
  - **EMG1**: mostly this contains the top level routines used in the element matrix generation process.
  - **EMG2**: several routines used for element offset, element debug output and element coordinate transformations
  - **EMG3**: routines for 1-D elements
  - **EMG4**: routines for 2-D elements
  - **EMG5**: routines for 3-D elements
  - **EMG6**: routines used for isoparametric strain-displ matrices

- **EMG7**: isoparametric shape function routines and abscissa and weight coefficients for Gaussian integration
- **EMG8**: material matrices for 2 and 3-D stress-strain conditions, material coordinate transformation routine, Jacobian matrix routines

- **LK1**: contains main subroutines used in LINK1 (described in section 1)
  - **L1A**:
  - **L1A-BD**: routines that read specific Bulk Data entries
  - **L1A_CC**: routines that read specific Case Control entries
  - **L1B**: routines related to grid, grid sequencing coordinate system, and degree of freedom (DOF) processing
  - **L1C**: routines that process subcase requests, concentrated masses, GPWG, plus some element routines that are used to process properties for all elements (e.g. sorting element arrays)
  - **L1D**: routines that process element and grid forces, rigid elements and enforced displacements
  - **L1E**: routines that process the G-set matrices for stiffness, mass and load (i.e combine all loads processed in L1D into the G-set array)
  - **L1U**: utility routines used in LINK1
  - **LINK1**:
- **LK2**: contains main subroutines used in LINK2 (described in section 1)
- **LK3**: contains main subroutines used in LINK3 (described in section 1)
- **LK4**: contains main subroutines used in LINK4 (described in section 1)
- **LK5**: contains main subroutines used in LINK5 (described in section 1)
- **LK6**: contains main subroutines used in LINK6 (described in section 1)
- **LK9**: contains main subroutines used in LINK9 (described in section 1)
- **MAIN**: contains the main program and a few of the routines it calls

- **Modules**: contains module subroutines that are used in subroutines via the fortran "USE *module_name*" construct. The main folder contains module subroutines written specifically for MYSTRAN. There are 3 subfolders containing module subroutines that were obtained from references outlined in the MYSTRAN User's Reference Manual and are used in MYSTRAN for the:

    i. Bandit grid resequence algorithm
    ii. ARPACK Lanczos eigenvalue routine
    iii. LAPACK routines for solving linear equations

- **UTIL**: contains utility routines called by many of the above routines

b. **Linux differences**: contains 4 small subroutines that are different for the Linux version than for the Windows version:

    i. GET_INI_FILNAM
    ii. GET_MYSTRAN_DIR.f90
    iii. mkl_dtfi.f90
    iv. READ_CL.f90

c. **Interfaces**: Every subroutine has an interface and the code for each is a module subroutine and is contained here. These interface modules themselves do not need to be put into the subroutines that call them. Rather, the names of all the interface modules needed in any subroutine are contained in the USE_IFs subfolder below.

d. **USE_Ifs**: These are interface modules that are referenced in every subroutine that calls other subroutines. Each one is a collection of the interface module names that are to be used in that subroutine.