FEAP - - A Finite Element Analysis Program

Version 8.7 FE^2 Multiscale User Manual

Robert L. Taylor & Sanjay Govindjee
Department of Civil and Environmental Engineering
University of California at Berkeley
Berkeley, California 94720-1710

January 2026

Contents

1 Introduction **1**

1.1 Solution process for FE^2 problems 1

1.2 Execution of problem 1

1.2.1 Input and solution files 2

1.3 Macro-scale input file 3

1.3.1 Macro-scale material models 3

1.3.2 Macro-scale global commands 5

1.3.3 Macro-scale solution commands 5

1.4 Micro-scale input file 6

1.4.1 RVE boundary conditions 6

1.4.2 Micro-scale solution commands 8

1.5 Solution steps by *FEAP* 9

2 Programmer developments **11**

2.1 User element for macro-scale 11

2.1.1 Data input: `isw=1` 11

2.1.2 Tangent, residual and stress computation 13

2.1.3 Defining MPI send/receive for RVE 14

2.1.4 Element output, projection and plotting 14

2.2 User element for micro-scale 15

Chapter 1

INTRODUCTION

The current release of the FE^2 multi-scale solution module for *FEAP* includes two types of formulations: A Hill-Mandel energy formulation as described in Reference,[12] or;[3] and an Irving-Kirkwood formulation as describe in Reference.[4] Currently, only formulations for solid stress analysis and for thermal analysis are implemented. However, multi-physics formulations may be added as user modules.

This manual serves as a supplement to the *FEAP* User manual[5] available at the web site `www.ce.berkeley.edu/feap`.

1.1 Solution process for FE^2 problems

The FE^2 solution is implemented as a parallel solution process. As such a processor is assigned for the macro-scale problem and the remaining specified number of processors are used for solution of the RVE's. This requires a special build of *FEAP* Thus,the multi-scale FE^2 module for *FEAP* requires the final build to be completed using access to MPI send and receive commands. This build is performed after first compiling a serial version of *FEAP*. For Unix based systems there is a `makefile` in directory `ver86` and in the subdirectory `fe2` to assist in the compilations.

1.2 Execution of problem

The execution of the program is initiated with an MPI command

```
mpirun -n np <pathname>/feap
```

where `np` denotes the total number of processors to use for the solution. This number includes the processor for the macro-scale problem, thus a minimum value for `np` is 2 and only one RVE mesh is permitted. There must be at least one processor available for each different RVE mesh. Usually, many elements in the macro-scale problem are associate with the same RVE. The current implementation is limited to 64 processors. This can be changed by modifying the parameter `mproc` in the `pfe2rve.f` file and the variables in the include file `oumatl.h`.

1.2.1 Input and solution files

The solution of multi-scale FE^2 problems by *FEAP* requires the user to prepare three types of data files:

1. An input file describing the macro-scale problem. This is a standard input file except for the specification of the material set data associated with an RVE describing the material constitution. Preparation of the macro-scale material set data is described in Section 1.3.1. In addition the first solution command *must be*

   ```
   BATCh
     RVE
   END
   ```

   Subsequent solutions can be given in either interactive or batch mode.

2. An input file describing each RVE is prepared also in a standard form except that the control record must have numerical values for all parameters (*e.g.*, `numnp, numel, nummat, ndm, ndf, nel`. In general an RVE is a rectangular parallelepiped and boundary conditions may be displacement, traction or periodic forms. Furthermore *no solution command statements are included in an RVE file*. See Section 1.4 for further information on preparing the input file and specifying boundary conditions.

3. The third required file is a file named

   ```
   solve_mpi
   ```

   that contains the solution commands for each RVE. This file is shared by all RVE's. A general form for the file is described in Section 1.4.2. Users should not change this file form except for possible adding of `print` or `echo` commands at the beginning.

1.3 Macro-scale input file

1.3.1 Macro-scale material models

Multi-scale analysis is implemented as a two-scale method in which the main problem is described according to the descriptions in the *FEAP* User Manual. The only exception is for material behavior to be described by a finer scale analysis using a *representative volume* (RVE) formulation.[3] For these materials the macro-scale material data input consists of

```
MATErial ma
  SOLId
    FRVE,<filename>
    TYPE <HILL,KIRK>
```

for finite deformation problems or

```
MATErial ma
  SOLId
    SRVE,<filename>
    TYPE <HILL,KIRK>
```

for small deformation problems. In the finite deformation problem the FE^2 is implemented to send displacement gradients from the macro-scale problem to the micro-scale as described in Chapter 7 of.[3] That is the deformation gradient is expressed as:

$$\mathbf{F}^M = \mathbf{I} + \frac{\partial \mathbf{u}^M}{\partial \mathbf{X}} = \mathbf{I} + \mathbf{G}^M$$

where superscript M denotes macro-scale. For small strain the macro-scale is the symmetric displacement gradient. At the micro-scale the deformation gradient is expressed in the usual manner as

$$\mathbf{F}^m = \mathbf{I} + \frac{\partial \mathbf{u}^m}{\partial \mathbf{X}} = \mathbf{I} + \mathbf{G}^m$$

and for small strain

$$\boldsymbol{\epsilon}^m = \tfrac{1}{2}\left(\mathbf{G}^m + (\mathbf{G}^m)^T\right)$$

where superscript m denotes micro-scale.

An alternative representation is also available in which the micro-scale deformation gradient is expressed as

$$\mathbf{F}^m = \mathbf{I} + \mathbf{G}^M + \frac{\partial \widehat{\mathbf{u}}^m}{\partial \mathbf{X}} = \mathbf{I} + \mathbf{G}^M + \widehat{\mathbf{G}}^m$$

in which $\widehat{\mathbf{G}}^m$ represents a fluctuation about the macro-scale displacement gradient. To activate this option the input forms are

```
MATErial ma
  SOLId
    IFRVe,<filename>
    TYPE <HILL,KIRK>
```

for finite deformation problems or

```
MATErial ma
  SOLId
    ISRVe,<filename>
    TYPE <HILL,KIRK>
```

for small deformation problems.

For thermal problems the commands are given as

```
MATErial ma
  THERmal
    TRVE,<filename>
    TYPE <HILL,KIRK>
```

or for the fluctuation form as

```
MATErial ma
  THERmal
    ITRVe,<filename>
    TYPE <HILL,KIRK>
```

Finally, the framework for coupled problems is available using

```
MATErial ma
  THERmal
    CRVE,<filename>
    TYPE <HILL,KIRK>
```

however, at present no elements are provided to carry out an analysis.

Multiple RVE types may be described by specifying different `filename` parameters to the `FRVE/SRVE/TRVE` record. The number of different RVE meshes is restricted to one

less than the number of processors used to solve the problem. Thus, if 4 processors are used to run the problem 1 is for the macro-scale analysis and the other 3 may be used for 1 to 3 different RVE meshes.

N.B. Problem solutions tested to date include: Plane strain; Axisymmetric; Three dimensional solids.

The main problem input file must start with `I` or `i` followed by a name. For example, the main file `Iblk3` describes a main problem to be solve. Each micro-scale RVE will be associated with a file `Iblk3_001`, `Iblk3_002`, etc. and these files are automatically created based on the number of processors specified for the problem solutions (see below for specifying the number of processors).

1.3.2 Macro-scale global commands

If all the material sets in the model are associated with an RVE, the type of formulation may be set for all materials using a global command

```
GLOBal
  RVE=<HILL,KIRK>
```

In this case it is not necessary to specify the `TYPE` command as part of any material set. However, a specified material set `TYPE` command always is used instead of the global command.

1.3.3 Macro-scale solution commands

The solution steps for the macro-scale problem are given in a standard manner (batch execution is recommended for the main solution steps) except that the first command must always be

```
RVE <POINt num>
```

to allocate the files to each processor. The option `POINt` specifies a quadrature point in the main problem for which specified deformation gradients (or strains) along with returned RVE stress and moduli are written to files (currently unit 90, 91, 92, respectively). These may be used after completion of the multi-scale problem to run a serial *FEAP* analysis using the `PERIodic` mesh command along with the appropriate `HILL` or `KIRK` solutions options. The results for the local stress and displacement contours may then be displayed for this single point.

1.4 Micro-scale input file

The RVE input file (name given to `filename` in macro-scale problem) is also a standard *FEAP* input file except that all values on the control record must be specified (*i.e.*, values for nodes, elements, material sets, mesh dimension, dofs/node, and nodes/element). Thus the start of the file becomes:

```
FEAP * * RVE mesh
numnp numel nummat ndm ndf nen ! with numbers

GLOBal
  RVE = <HILL,KIRK>
  <QBAR,EBAR,FBAR>

MATErial 1
  <SOLId,THERmal> ...
  etc.

END ! mesh
ELINk
   ...
```

in which numerical values must be given for `numnp`, etc. The RVE input file does not have solution commands and the `STOP` command is optional.

The formulation used for the solution of the RVE must match the method used at the macro-scale and is specified by the global command `RVE` as shown above. Alternatively, the formulation type may be specified as part of the material data set as

```
MATErial ma
  <SOLId,THERmal,....>
     TYPE = <HILL,KIRK>
```

however, if multiple material sets are used for the RVE, the command must be in every set.

1.4.1 RVE boundary conditions

Generally, the RVE is a rectangular parallelepiped in which all boundaries coincide with a constant coordinate value. It is possible to specify four types of boundary conditions:

- The boundaries of the RVE may be fully restrained using `EBOUndary` commands. It is not necessary for the boundary nodes to match on opposite faces.

- A full Taylor condition in which all nodes of the RVE are fixed may be specified as

  ```
  TBOUndary
  ```

  In this case there are no equations to solve on the RVE.

- The boundaries of the RVE may be fully restrained using `EBOUndary` commands. It is not necessary for the boundary nodes to match on opposite faces or for the RVE to be a parallelepiped.

- Periodic boundary conditions may be given by linking the edges in all directions of the RVE using `ELINk` commands (after the `END` of mesh). For this case nodes must match on opposite linked edges.

- Traction conditions may be imposed using a Lagrange multiplier constraint of the form

  $$\Pi_\sigma = \sigma_{ij}^M \left[\epsilon_{ij}^M \, V - \int_V \epsilon_{ij}^m \, \mathrm{d}V \right]$$

  where superscript M denotes macro-scale and m the micro-scale; V is the volume of the RVE. The condition is included as an element computation and requires the global command

  ```
  GLOBal
    QBAR
          ...
  ```

  for thermal problems,

  ```
  GLOBal
    EBAR
          ...
  ```

  for small deformation solids, and

  ```
  GLOBal
    FBAR
          ...
  ```

  for finite deformation solids.

  Alternatively, the traction condition may be included as part of a material set using

```
MATErial ma
  <SOLId,THERmal,....>
       ...
    <QBAR,EBAR,FBAR>
       ...
```

Once execution of the macro-scale problem begins *FEAP* will create a file `Iblk3.001`, `Iblk3.002`, etc. for each of the `np-1` RVE solutions. Each of these files has the simple structure:

```
NOCOunt
UFEAP <* * optional title>

INCLUde solve_mpi

STOP
```

In addition a computation of the total number of calls for each RVE along with a proportion to the different processors to achieve a balance in time will be made. It is assumed each RVE will require about the same compute time to converge each solution step.

1.4.2 Micro-scale solution commands

The `INCLude solve_mpi` in each RVE file describes the solution steps that can be performed. At this time it is recommended that the solution for an RVE only be given as

```
BATCh
  FE2  STARt sw its     ! Start RVE solution (sw = if parameter)
  FE2  PLOT  pl pt rk   ! Plot microscale. One plot/rank allowed
  LOOP,INFInite         ! Permits infinite solution iterations
    FE2  GET            ! Get F from macro solution: Input U, H
    IF sw-12            ! Test for time step (isw = 12)
      NOPRint           ! Do nothing statement needed
    ELSE sw-3           ! For all 3 < isw < 12
      LOOP              ! Iterate local solution (no number)
        TANG,,1
      next              ! End local unit problem
      IF pl
        PLOT  ....      ! Plot requests for micro-scale point
```

```
      ENDIf
    ENDIf
    FE2  SEND          ! Send stresses to macro solution: Out U,H
  NEXT                 ! Next macro solution
END                    ! End of batch execution file
```

The parameter `its` specifies the maximum number of iterations for the `TANG` command. At the start of each time step only one iteration is performed, thus, any inelastic material should return an elastic predictor in the first iteration. In subsequent iterations up to `its` iterations are performed for each RVE.

Optional plot for a point of each rank in the micro-scale may be performed for a `pt` of rank `rk`. The actual quadrature point may be determined by output of the array `RVESD` in the macro-scale using the command

```
SHOW RVESD    .
```

If the micro-scale RVE has a poor solution structure the profile may be optimized by modifying the above to:

```
OPTImize
FE2  STARt sw its ! Start RVE solution (sw = if parameter)
 .....
```

However, it is always best to test the RVE in a serial execution prior to performing the FE^2 solution and performming the optimmization followed by an `OUTMesh` command.

1.5 Solution steps by *FEAP*

The solution of FE^2 problems by *FEAP* is accomplished in a number of steps on the macro-scale problem. These may be summarized as:

1. During input of the material set data for the problem (`ISW=1` in an element) the name of the input file containing the mesh, material data and boundary conditions of the RVE are given along with the type of solution. For the solid elements this might be the set of commands

   ```
   MATErial 1
     SOLId
       FRVE Irve_file
   ```

   for a finite deformation problem.

2. After input of the mesh data `FEAP` loops through all element to assign initial data to history variables (in each element this is performed in an `isw=14` part of the element module). For each integration point in an element an array `RVEMA` [`mr(np(269))`] has entries of the material set number [`ma` is the material set number]. This array is used subsequently to allocate RVE's for each material to a processor. If more processors exist than RVE's a distribution to balance computation is made.

3. For each solution step [*i.e.*, `tang,,1`] a loop through all elements is made in module `pforma.f` to only compute the appropriate function gradient (*i.e.*, $\mathbf{G}^M$ or ***epsilon***). The latter are saved in an array `FRVEL` [`mr(np(260)`] which are later sent by an MPI command to the individual RVE's.

4. After the first pass through all elements each entry in `FRVEL` is sent to a processor [this is done using module `rvesr.f` called from the `pform.f` module in the `fe2` directory. The RVE module receives the data (by an MPI receive command) and each *FEAP* RVE process solves the micro-scale problem as described in Chapter 7 of reference.[3] After solution the RVE performs an MPI send of the computed RVE stress and its associated tangent matrix back to the macro-scale process.

5. The macro-scale problem collects the received RVE solution values and saves in an array `SRVEL` [`mr(np(261)`] in module `rvesr.f`.

6. After all RVE solutions are received a second pass through the macro-scale elements is made. In this process the residual and tangent matrix for all elements is computed. For RVE's the required stress and tangent are obtained from the `SRVEL` array, for elements that have a normal constitutive equation the stress and tangent moduli are computed as usual.

7. At this point the full macro problem tangent and residual are formed and may be solved as a Newton iteration step.

The required constitutive module for an RVE may be found in the `fe2` subdirectory in the `srvemat.f`, `frvemat.f` and `trvemat.f` for a small strain, finite deformation and thermal problem, respectively.

Chapter 2

Programmer developments

Users may extend the capability of the FE^2 module by adding their own element at either the macro-scale or the micro-scale. Generally, elements at the micro-scale need no special capability other than being limited to either thermal or stress analysis options. At the macro-scale a user element needs to interface to the MPI send-receive options of the FE^2 solution process. This requires special coding in addition to the usual steps enumerated in the *FEAP* Programer Manual.[6]

2.1 User element for macro-scale

Developing a user element for use at the macro-scale requires careful attention to details at the basic steps of input and computation of tangents and residuals.

2.1.1 Data input: `isw=1`

During the input of the material set it is necessary to:

1. Set the analysis type as thermal or mechanical
2. Allocate history variable to store flux or stress
3. Set the RVE filename.

The first two items may be defined using the user module `urveinp`:

```
call urveinp(utype, utext)
```

where `utype(1)=1` for thermal analysis and `utype(1)=2` for stress analysis; `utype(2)=1` for small deformation and `utype(1)=-1` for finite deformation; `utext` is a `character (len=15)` array. Prior to the call to `urveinp` it is necessary to perform a data input using

```
setvar = tinput(utext,2,td,*) ! specify value for * as necessary
```

A example of a code fragment that performs input of data is:

```
utext(1) = 'start'
do while( .nor.pcomp(utext(1),'    ',4))
  setvar = tinput(utext,2,td,2)
  if(pcomp(utext(1),'urve',4)) then
    urvefl = .true.
    utype  =  2
    call urveinp(utype, utext)
      ...
  elseif(pcomp(utext(1),'quad',4)) then
    d(5) = td(1)
  elseif ....

  endif
end do ! while
```

The above sets the data for stress analysis.

The last step in this phase is setting storage for the flux or stress. For a thermal analysis three (3) values for each element quadrature point are used; for stress analysis six (6) values are used. Thus, the required assignment for the stress analysis defined above is

```
nh1 = nh1 + 6*nint(d(5))    ! Stress analysis
```

For a thermal analysis the statement would be

```
nh1 = nh1 + 3*nint(d(5))    ! Thermal analysis
```

Using the above code form a typical material set data input for the user element would be:

```
MATErial ma
```

```
USER n
  URVE Irve_file
  QUADrature,,nq
    !  Blank termination record
```

where `ma` is the material set number, `n` the user element number (*i.e.*, `elmt0n`), and `nq` the number of quadrature points for the evaluation of element arrays.

2.1.2 Tangent, residual and stress computation

During the computation of the tangent and/or residual it is necessary to set a parameter for the type of multi-scale analysis to be performed. At present there are two options: (1) Hill-Mandel or (2) Irving-Kirkwood. In general it is recommended to first use a Hill-Mandel method since it does not require extra coding in the micro-scale element. The parameter is set by using the include statement

```
include   'elmate.h'
```

and setting the parameter `typrve` to 1 or 2.

There are three modules available to perform the constitutive evaluations for an FE^2 analysis:

Thermal analysis

For a two or three dimensional thermal analysis the constitutive evaluation is performed using:

```
call trvemat(ta,tgrad, hn1,nh, tflux,kt,rhoc, isw)
```

where `ta` is the temperature, `tgrad(3)` is the temperature gradient, `hn1(*)` is the history variable array ta time t_{n+1}, `hn` is the number of components used in the analysis, `tflux(3)` is the thermal flux, `kt(3,3)` is the thermal conductivity, `rhoc` is the ρc for transient analysis at the macro-scale, and `isw` is the element switch parameter. Note that two dimensional analyses still require all three components to be dimensioned, however, only the first two are needed in the analysis.

Small strain stress analysis

For two or three dimensional small strain stress analysis the constitutive evaluation is performed using:

```
call srvemat(eps,ta,hn1,nh, sig,dd,rho, isw)
```

where `eps(6)` is strain in Voigt notation, `ta` is temperature, `hn1(*)` is the history variable array ta time t_{n+1}, `hn` is the number of components used in the analysis, `sig(6)` is the Cauchy stress in Voigt notation, `dd(6,6)` is the tangent moduli in Voigt form, `rho` is the density, and `isw` the element switch parameter.

Finite deformation stress analysis

For two or three dimensional finite deformation stress analysis the constitutive evaluation is performed using:

```
call frvemat(g,detf,ta,hn1,nh, sig,dd,rho, isw)
```

where `g(3,3)` is the displacement gradient (*i.e.*, $\mathbf{G}^M = \mathbf{F}^M - \mathbf{I}$), `detf` is the determinant of the deformation gradient, `ta` is temperature, `hn1(*)` is the history variable array ta time t_{n+1}, `hn` is the number of components used in the analysis, `sig(6)` is the Cauchy stress in Voigt notation, `dd(6,6)` is the tangent moduli in Voigt notation, `rho` is the density, and `isw` the element switch parameter.

2.1.3 Defining MPI send/receive for RVE

After mesh input and before problem solution steps *FEAP* passes through all elements with `isw=14` to initialize history variables, count the number of RVE problems to solve, and set lengths for the send/receive records. In each element module in the `isw=14` section the necessary setting of data may be accomplished by calling the routines described in Section 2.1.2 for *the same number of times used in the stiffness computation.* Thus, a typical evaluation for small strain stress analysis is:

```
do l = 1,lint
  call srvemat(eps,ta,hn1,nh, sig,dd,rho, isw)
end do ! l
```

where `lint` is set to the number of quadrature points used for stiffness evaluation. In this step the only argument variable used is `isw`.

2.1.4 Element output, projection and plotting

The routines described in Section 2.1.2 may be used for all steps in the solution process and replace the direct evaluation of material behavior. During multi-scale analysis to

compute the element tangent matrix and residual the element module is called twice with `isw=3`. In the first call the strain, displacement gradient or temperature gradient is computed and the stress and tangent moduli are set to zero. These measures are sent to the RVE for evaluation of the stress or flux and the associated tangent which are used in a second pass to compute the actual element residual and tangent matrices. This involves considerable computation for large problems and thus constitutes most of the solution time. For output of results and plotting of element variables the stress or thermal flux is recovered from a *history variable* so that there is no need for further multi-scale computation.

2.2 User element for micro-scale

For solutions using the Hill-Mandel algorithm there are no modifications needed for user elements used at the micro-scale. All multi-scale solution steps on the RVE use only the operations for switch values `isw = 1, 3, 6, 12, 14`. At the micro-scale the *FEAP* program running on the RVE processor makes direct calls to the elements.

Bibliography

[1] V. Kouznetsova, W.A.M. Brekelmans, and F.P.T. Baaijens. An approach to micro-macro modelling of heterogeneous materials. *Computational Mechanics*, 27:37–48, 2001.

[2] V.G. Kouznetsova. *Computational homogenization for the multi-scale analysis of multi-phase materials*. Phd. thesis, Eindhoven University of Technology, Eindhoven, Netherlands, 2002.

[3] O.C. Zienkiewicz, R.L. Taylor, and D. Fox. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier, Oxford, 7^{th} edition, 2013.

[4] B.S. Mercer, K.K. Mandadapu, and P. Papadopoulos. Novel formulations of microscopic boundary-value problems in continuous multiscale finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 286:268–292, 2015.

[5] R.L. Taylor and S. Govindjee. *FEAP - A Finite Element Analysis Program, User Manual*. University of California, Berkeley.

[6] R.L. Taylor and S. Govindjee. *FEAP - A Finite Element Analysis Program, Programmer Manual*. University of California, Berkeley.