FE-BUI 0.7
User Manual

21-04-05

FE-BUI is written in FORTRAN 77 using double precision memory address for real arrays and scalars. Its installation requires the freely available libraries:
BLAS(http://www.netlib.org/blas),
LAPACK (http://www.netlib.org/Lapack),
METIS (http://www-users.cs.umn.edu/~karypis/metis) and
MPI (http://www-unix.mcs.anl.gov/mpi)

FE-BUI has been tested on a ROCKS 2.3 (http://www.rocksclusters.org) Beowulf cluster with:
ATLAS 3.4.1 (http://math-atlas.sourceforge.net), LAPACK 3.0, LAM/MPI 6.5.9 (http://www.lam-mpi.org), MPICH 1.2.6 and Intel FORTRAN Compiler 7.0 (http://www.intel.com/software/products/compilers/linux)

As a first step with the FE-BUI package, the new user can simply call from his/her serial code the driver subroutine FEBUIdrv replacing the serial solver call. This driver takes care of the solution of the linear system by calling the default partitioning and solver subroutines.

Using the FEBUIdrv in 5 steps

Step 1

At the main program include the file mpif.h:

```
      include 'mpif.h'
```

and after the array and scalar declarations initialize an MPI group with the three commands:

```
ccccccc INITIALIZE MPI ccccccc
      call MPI_INIT(ierror)
ccccccccccccccccccccccccccccc
ccccccc Find whoami and how many other processor are around cccccccc
      call MPI_COMM_RANK(MPI_COMM_WORLD,MyID,ierror)
      call MPI_COMM_SIZE(MPI_COMM_WORLD,Nprocs,ierror)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

Arguments

ierror	OUTPUT – INTEGER error printing message number

MyID OUTPUT – INTEGER
the process ID

Nprocs OUTPUT/INTEGER
the total number of processes

The integer MPI_COMM_WORLD is defined in the file mpif.h.

Step 2

Replace the serial solver call with:

```
   call FEBUIdrv
$(MyID, Nprocs, NN_global, NE_global, NOP_global, NNperElem,
$ NCOD_global, BC_global, GMRESIter, GMRESKrylov, x_global, u_global,
$ initFEBUI, user_sub, MATVECform)
```

Arguments

MyID INPUT – INTEGER
the output of MPI_COMM_RANK

Nprocs INPUT – INTEGER
the output of MPI_COMM_SIZE

NN_global INPUT – INTEGER
the total number of the nodes of the mesh

NE_global INPUT – INTEGER
the total number of the elements of the mesh

NOP_global INPUT – INTEGER ARRAY of size [NE_global,max(NNperElem)]
this array associates the local (element level) and global (mesh level) numbering of nodes

NNperElem INPUT – INTEGER ARRAY of size [NE_global]
the number of nodes that each element has

NCOD_global INPUT – INTEGER ARRAY of size [NN_global]
if NCOD(i) = 1 then to the i node is imposed the Dirichlet boundary condition BC_global(i), otherwise NCOD(i) should be set equal to 0

BC_global INPUT – DOUBLE PRECISION ARRAY of size [NN_global]

x_global INPUT/OUTPUT – DOUBLE PRECISION ARRAY of size [NN_global]
on INPUT an initial guess of the solution of the linear system

on OUTPUT the approximate solution of the linear system

u_global	INPUT – DOUBLE PRECISION ARRAY of size [NN_global] the current solution of the non linear system
initFEBUI	INPUT – INTEGER initFEBUI = 1 for the first call of the FEBUIdrv, otherwise initFEBUI ≠ 1
user_sub	INPUT the name of the user supplied subroutine that computes the element contributions to the matrix and the right hand side of the linear system. It should be declared as external.
MATVECform	INPUT – INTEGER MATVECform = 1 for Matrix-Vector product in CSR format MATVECform = 2 for Matrix-Vector product in Local format MATVECform = 3 for Matrix-Vector product in Difference format (nonlinear case only)

Step 3

Replace all KEYBOARD READs [read(*,*)] with FILE READs [read(1,*)]

Put all write commands of your code in a BLOCK IF statement. For example if your original code is:

```
do i=1,NN_global
write(1,*) i, u_global(i)
end do
```

then replace it with:

```
if(MyID.eq.0) then
  do i=1,NN_global
  write(1,*) i, u_global(i)
  end do
endif
```

Step 4

Before stop your code (before the END statement), write a call:

```
call MPI_FINALIZE(ierror)
```

Step 5

The user supplied subroutine named <user_sub> must have the following form:

subroutine <user_sub> (i, estifm, b, u)

Arguments

i	INPUT – INTEGER the number of the element
estifm	OUTPUT – DOUBLE PRECISION ARRAY of size [max(NNperElem), max(NNperElem)] the contribution of element i to the matrix of linear system
b	OUTPUT – DOUBLE PRECISION ARRAY of size [NN_global] the right hand side of the linear system
u	INPUT – DOUBLE PRECISION ARRAY of size [NN_global] the current solution of the non linear system