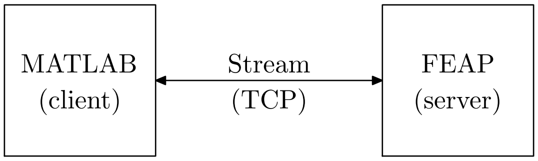
MATFEAP user documentation

David Bindel

April 11, 2008

1 Introduction

MATFEAP is an interface between MATLAB and the finite element analysis program FEAP. Using either the Java Virtual Machine installed with MATLAB or a C MEX file, MATFEAP communicates with a FEAP server via either a socket or a pipe:



The FEAP server may run on the same machine as the MATLAB client, or on a different machine altogether. The server machine does not need to have MATLAB installed.

2 Simple example

To give the flavor for how MATFEAP works in practice, we give a simple example that uses several of MATFEAP's capabilities.

3 Building MATFEAP

To build the FEAP server, edit `makefile.in` in the MATFEAP home directory and run `make`. This will build the FEAP pipe server (`srv/feapp`) and the FEAP socket server (`srv/feaps`).

If you wish to use the (non-required) C MEX interface to the network socket functions, you will also need to type `make cclient`. You need not do this if you plan to use the Java interface. If you wish to use MATFEAP with Octave – which does not have a JVM attached – you can do so with the C MEX interface

4 Running MATFEAP

There are three modes for running MATFEAP: pipe, TCP, or UNIX. The default Java client can communicate with FEAP using a pipe or a TCP socket; the C MEX client can communicate with FEAP using a TCP socket or a UNIX-domain socket. For single-system use, TCP sockets are relatively slow, so the default behavior is to use the pipe interface (when running with Java) or the UNIX-domain socket interface (when running with C). Of these mechanisms, the UNIX-domain sockets are fastest on the limited number of systems where I have done timings.

Before running any simulations, you will need to run the `matfeap_init` script to set the system paths and select the default communication mode. This initialization routine should work fine with MATLAB versions 7+, including the student edition. In MATLAB 6.5, the class path used by MATLAB's JVM cannot be dynamically modified, so one has to set that variable in another way – see the Mathworks site for details.

4.1 Running in pipe mode

If you wish to use pipe mode (the default mode for the Java client), you don't need to do anything special. Just run the desired script:

```
>> matfeap_init
Using Java socket bindings
Using FEAP in pipe mode: /home/dbindel/work/feap/matfeap/srv/feapp
>> cd example
>> blocktest1
norm(K*u+R) = 2.01376e-15
>>
```

4.2 Running with UNIX-domain sockets

If you wish to use UNIX sockets (the default mode for the C client), you will first need to start the FEAP server. From the shell prompt, run `srv/feapu`. Then open a MATLAB session, and you should be able to run the desired script.

```
>> matfeap_init
Using C socket bindings
Using FEAP on UNIX socket: /tmp/feaps-dbindel
>> cd example
>> blocktest1
norm(K*u+R) = 2.01376e-15
>>
```

If you are testing new elements, you may want to use the `feapu-vg` script, which runs the FEAP socket server (for a UNIX domain socket) in `valgrind`.

4.3 Running with TCP sockets

If you wish to use TCP sockets, you will first need to start the FEAP server. From the shell prompt on the machine where you wish to run FEAP, run `srv/feaps`. Then open a MATLAB session, and after running `matfeap_init`, use the `feaps_tcp` command to specify that you are using a TCP socket server.

```
>> matfeap_init
Using C socket bindings
Using FEAP on UNIX socket: /tmp/feaps-dbindel
>> feaps_tcp
Using FEAP in TCP mode: 127.0.0.1:3490
>> cd example
>> blocktest1
norm(K*u+R) = 2.01376e-15
>>
```

If you are running the client and the server on different machines, you will need to specify the server hostname as an argument to `feaps_tcp`.

```
>> matfeap_init
Using C socket bindings
Using FEAP on UNIX socket: /tmp/feaps-dbindel
>> feaps_tcp('box173.cims.nyu.edu')
Using FEAP in TCP mode: box173.cims.nyu.edu:3490
>> cd example
>> blocktest1
norm(K*u+R) = 2.01376e-15
>>
```