OOMMF

Workshop 2000

Mike Donahue   Don Porter

Installation & Architecture Overview

http://math.nist.gov/oommf/

Orientation

- Pick a machine.
- Login as `guest3`
- Each of you has a working directory

  ```
  $ cd porter
  ```

Tcl/Tk: Shells and Scripts

- A *shell* is a program that interprets and executes commands.
- A *script* is a text file containing commands to be interpreted by a shell.
- Tcl (Tool Command Language) is a C library which interprets a simple command language, also known as Tcl.
- Tcl can be *extended* – that is, it can be taught how to interpret new commands.
- Tk is an extension of Tcl that adds commands for creating and controlling graphical widgets.
- Tcl/Tk provide two shells, `tclsh` and `wish`

Using wish

- `wish` can interpret commands interactively...

```
$ wish
% pack [button .b -text exit -command exit]
```

- ...or from a file

```
$ cat demo.tcl
pack [button .b -text exit -command exit]
$ wish demo.tcl
```

Shells and Scripts in OOMMF

- OOMMF is written in C++ and Tcl/Tk.
- Tk adds new Tcl commands in `wish`
- OOMMF C++ code adds new Tcl commands to make its own shells.
  - `omfsh`
  - `mmdispsh`
  - `mmsolve`
  - `oxs`
- Most applications are scripts.
- Aids portability. Scripts are plain text files. Same on all platforms. Once shells are ported, OOMMF is ported.

OOMMF Installation: Instructions

- Start with the `README`
- Start `mmhelp` for instructions

```
$ wish app/mmhelp/mmhelp.tcl
```

- Too sluggish? Use web browser.

```
$ netscape doc/userguide/userguide.html
```

- **Problem:** Where is everything?

OOMMF Installation: Platform

- Check platform configuration

```
$ tclsh oommf.tcl +platform
<1945> oommf.tcl 1.2.0.0  info:
OOMMF release 1.2.0.0
Platform Name:          lintel
C++ compiler:           /usr/bin/g++
Tcl configuration file: /usr/local/lib/tclConfig.sh
tclsh:                  /usr/local/bin/tclsh8.3
Tcl release:            8.3.1 (config)  8.3.1 (running)
Tk configuration file:  /usr/local/lib/tkConfig.sh
wish:                   /usr/local/bin/wish8.3
Tk release:             8.3.1 (config)  8.3.1 (running)
```

OOMMF Installation: Platform

- Note: `oommf.tcl` can find its own shell.

```
$ oommf.tcl +platform
...
```

- Note the platform name.
- Is your compiler ready to go? *No!*

OOMMF Installation: The “Bootstrap”

- Two problems:
  - Where are the application scripts?
  - What shells do they need?
- **Solution:** Let a program find out for you.

```
$ oommf.tcl foo
<1950> oommf.tcl 1.2.0.0  panic:
Can’t find application satisfying ’foo’
for platform ’lintel’:
        No matching application in the search path.

Available applications for platform ’lintel’:
        pimake 1.2.0.0
```

OOMMF Installation: The “Bootstrap”

- The bootstrap script, `oommf.tcl`, knows what programs are available, and how to launch them.

  ```
  $ oommf.tcl +command pimake
  /.../tclsh8.3 /.../oommf/app/pimake/pimake.tcl
  ```

- The script file and the needed interpreting shell are found.
- Leave out `+command` to launch the application.

OOMMF Installation: PIMake

- `pimake` – Platform-independent make
- Why? `make` varies too much over platforms.

```
$ oommf.tcl pimake -help
<1973> pimake 1.2.0.0  info:
Usage: pimake [options] [target]
where
  target  Target to build (..."help" prints choices)
Available options:
  -cwd         directory    Change working directory
  ...
```

OOMMF Installation: PIMake

- PIMake help

```
$ oommf.tcl pimake help
Default targets:
  help                print this message
  all                 build all executables
  configure           platform-specific preparations
  objclean            remove intermediate object files
  clean               remove executables and libraries
  distclean           remove all files not in source distribu
  maintainer-clean            remove all generated files
  upgrade             remove all files not in latest release
```

- Default target is `all`

OOMMF Installation: PIMake

- PIMake finds dependencies automatically
- PIMake can be run in subdirectories

```
$ touch ext/oc/oc.cc
$ oommf.tcl -cwd app/omfsh pimake
$ cd app/mmdisp
$ ../../oommf.tcl pimake
$ cd ../..
$ oommf.tcl pimake
```

OOMMF Installation: Bootstrap Options

- Already have seen
  - `+platform`
  - `+command`
- Other useful options
  - `+help`
  - `+cwd`
  - `+bg`, `+fg`
- `+` for bootstrap options
- `-` for application options

```
$ oommf.tcl +command pimake -help
```

OOMMF: Common Options

- `-help`
- `-cwd` *directory*
- `-tk <0|1>`
- `-console`

OOMMF: Local Customizations

- Create a local configuration directory

  ```
  $ mkdir config/local
  ```

- Copy and edit `options.tcl`

  ```
  $ cp config/options.tcl config/local
  ```

OOMMF Simple Demo

- Setting up a problem with **mmProbEd**
- Solving with **mmSolve2D**
- Monitoring with display apps
- Post-processing solver results
  - Examining log file
  - Animation
- Complex geometry input