OOMMF Micromagnetics Public Code Project

http://math.nist.gov/oommf

National Institute of Standards and Technology
Gaithersburg, MD 20899, USA

Mike Donahue
Don Porter
Bob McMichael
Jason Eicke (GWU)
Jim Blue

Coming Attractions

1. Recap: Session CD-05 (Don Porter)
2. Demo: Near poster session Wed. afternoon
3. First official release:
   mailto:mumag-announce@cobalt.nist.gov

Outline

1. Project Objectives

2. Module Survey

3. Solver Details

4. Intermodule Communication (Don Porter)

Design Criteria

- Portable
  - Tcl/Tk
  - Windows NT & '95, Unix + X
- Open Design
  - Module level
  - Tcl/Tk script level
  - C++ source code level
- Flexible
  - Simplicity vs. functionality

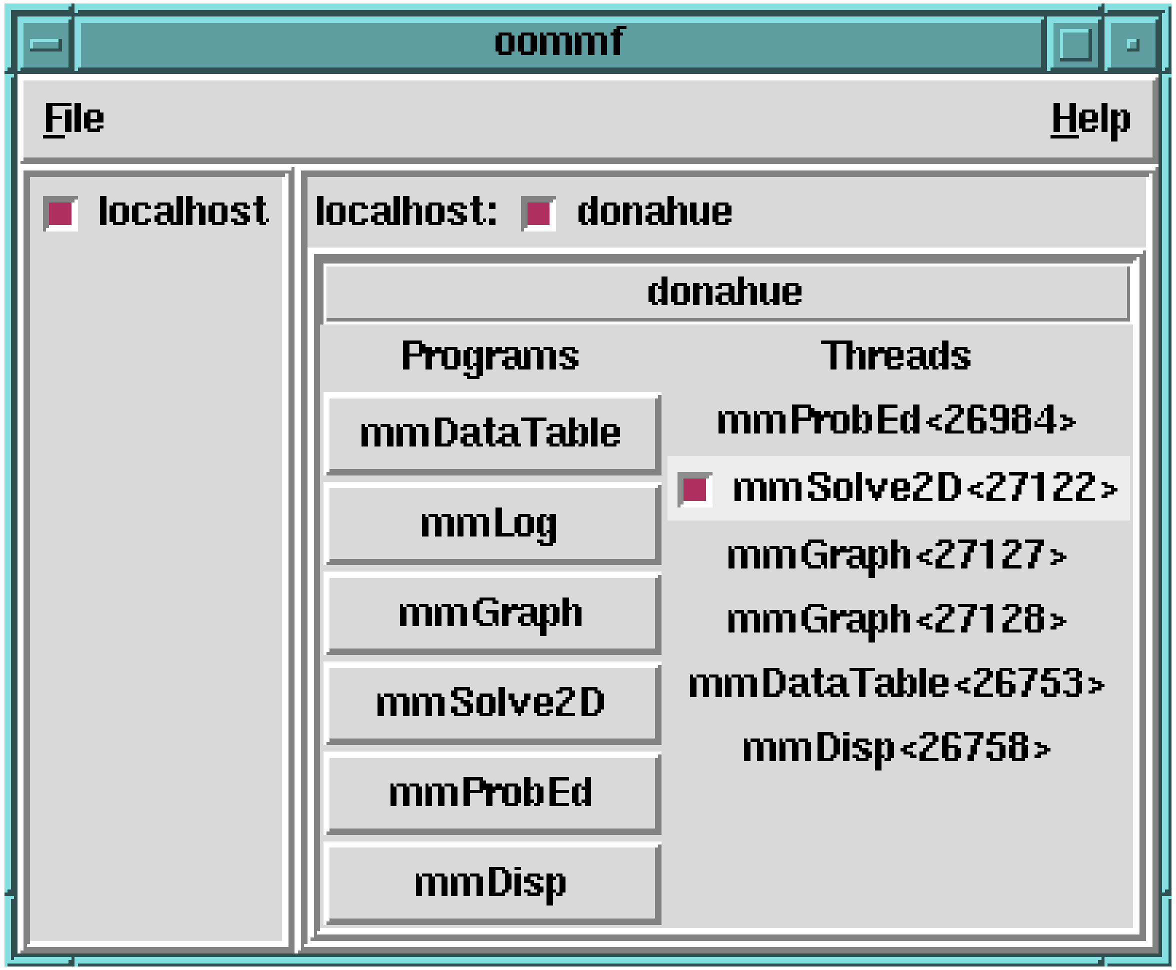
Requirements

- Tcl/Tk 7.5/4.1+ (8.0 recommended)
- Windows NT/95 or Unix+X (tested on SGI, Sun, Linux/Intel, Linux/Alpha)
- RAM: 32MB (?)
- Disk space: 5/25 MB
- C++ compiler to build from source

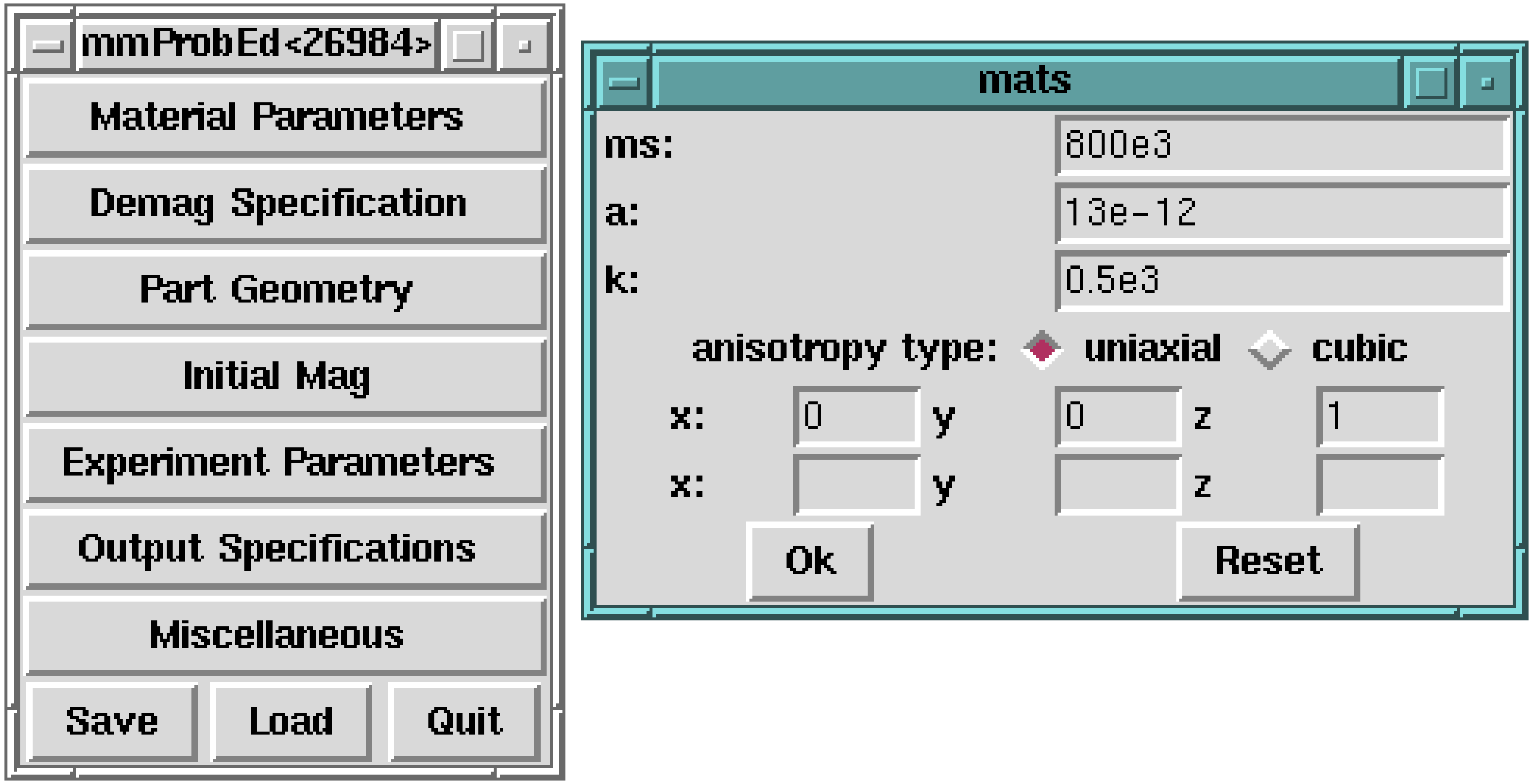
OOMMF Program Modules

- OOMMF Control/Module Launcher
- Problem Editor
- 2D Solver
- Data Table Display
- Graph Display
- Magnetization/Field Display

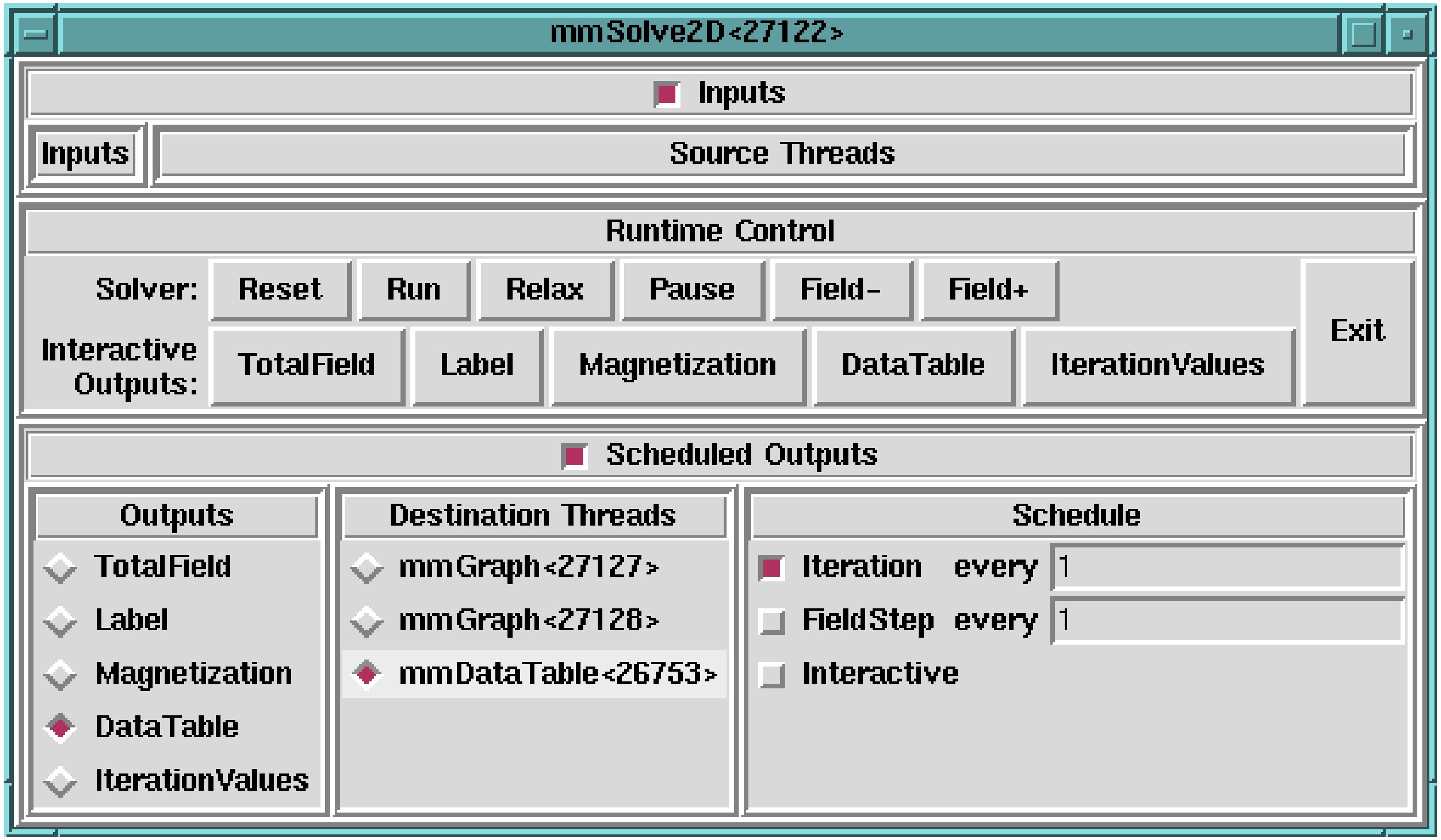
OOMMF Main Control/Module Launcher



Problem Editor

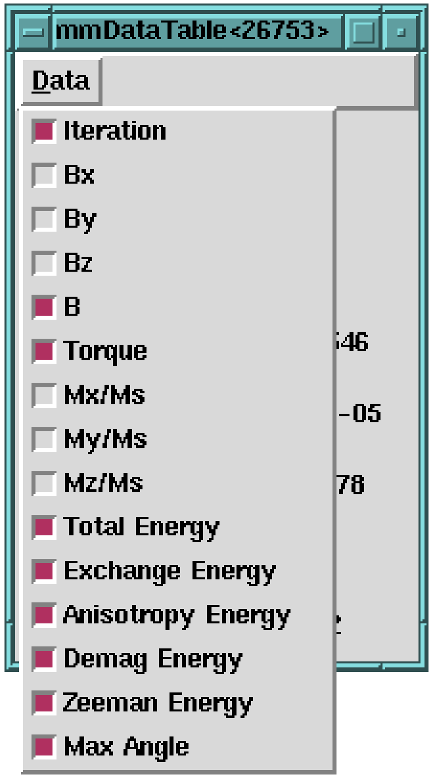


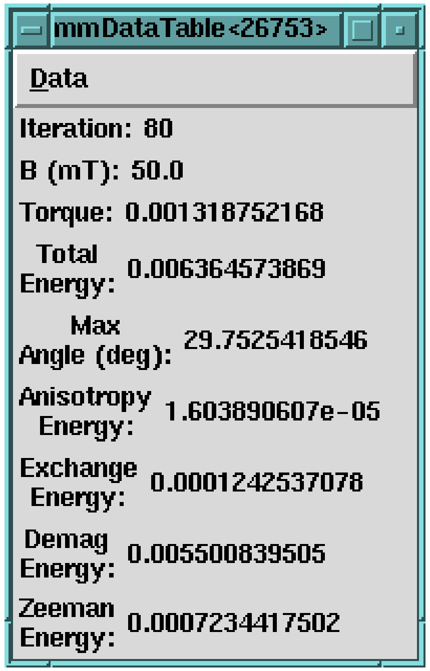
2D Solver



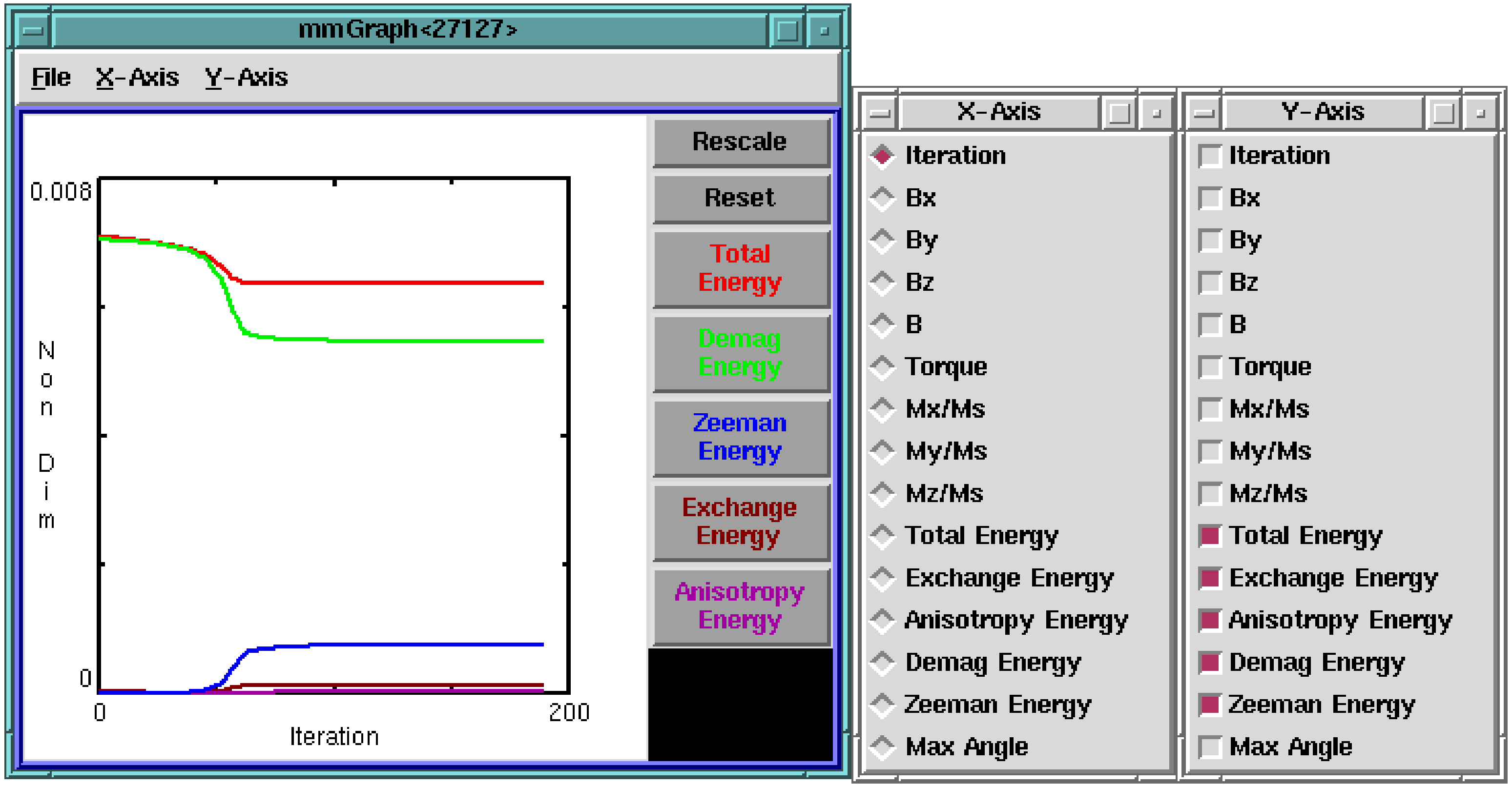
mmSolve2D<27122>
Inputs
Inputs
Source Threads
Runtime Control
Solver:
Reset
Run
Relax
Pause
Field-
Field+
Exit
Interactive Outputs:
TotalField
Label
Magnetization
DataTable
IterationValues
Scheduled Outputs
Outputs
TotalField
Label
Magnetization
DataTable
IterationValues
Destination Threads
mmGraph<27127>
mmGraph<27128>
mmDataTable<26753>
Schedule
Iteration every 1
FieldStep every 1
Interactive

Data Table

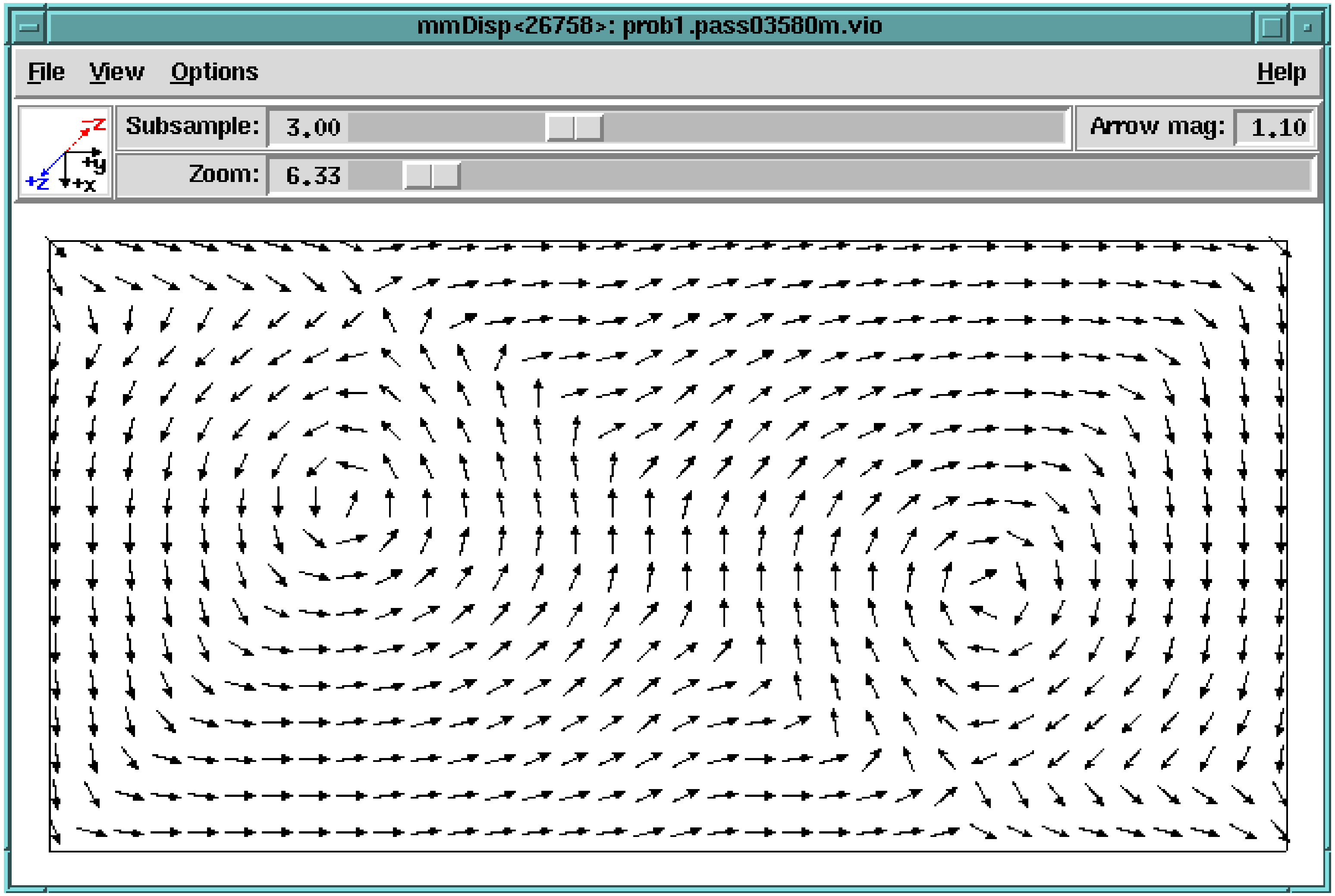

mmDataTable<26753>
Data
Iteration
Bx
By
Bz
B
Torque
Mx/Ms
My/Ms
Mz/Ms
Total Energy
Exchange Energy
Anisotropy Energy
Demag Energy
Zeeman Energy
Max Angle


mmDataTable<26753>
Data
Iteration: 80
B (mT): 50.0
Torque: 0.001318752168
Total Energy: 0.006364573869
Max Angle (deg): 29.7525418546
Anisotropy Energy: 1.603890607e-05
Exchange Energy: 0.0001242537078
Demag Energy: 0.005500839505
Zeeman Energy: 0.0007234417502

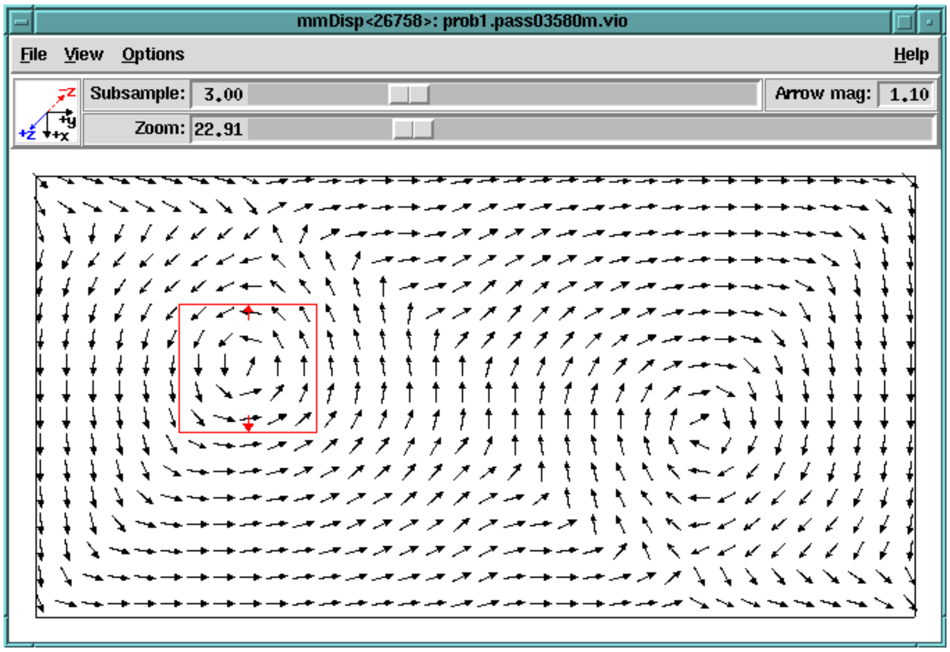
Graph Display

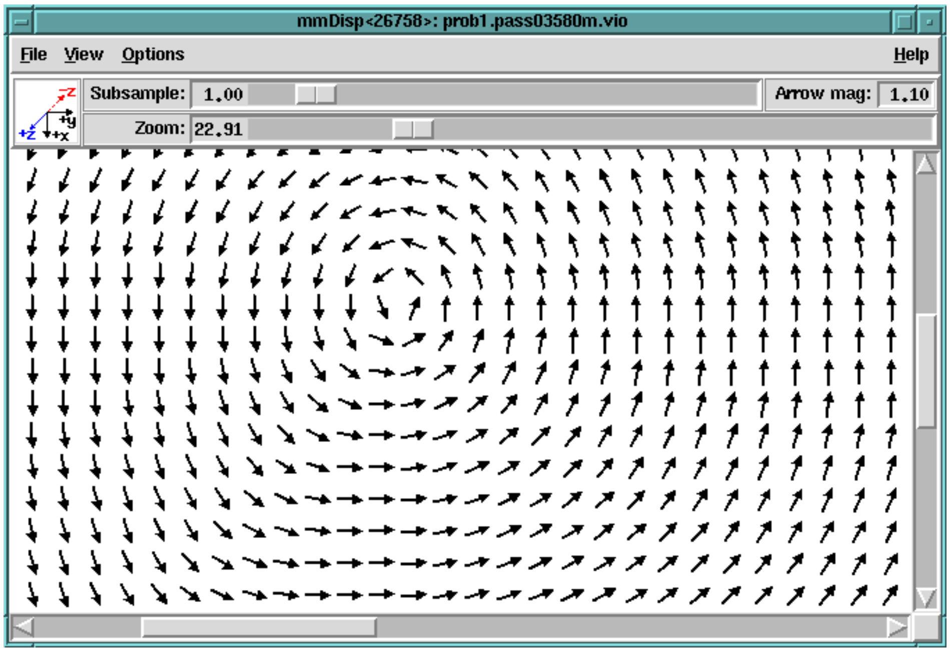


Magnetization/Field Display

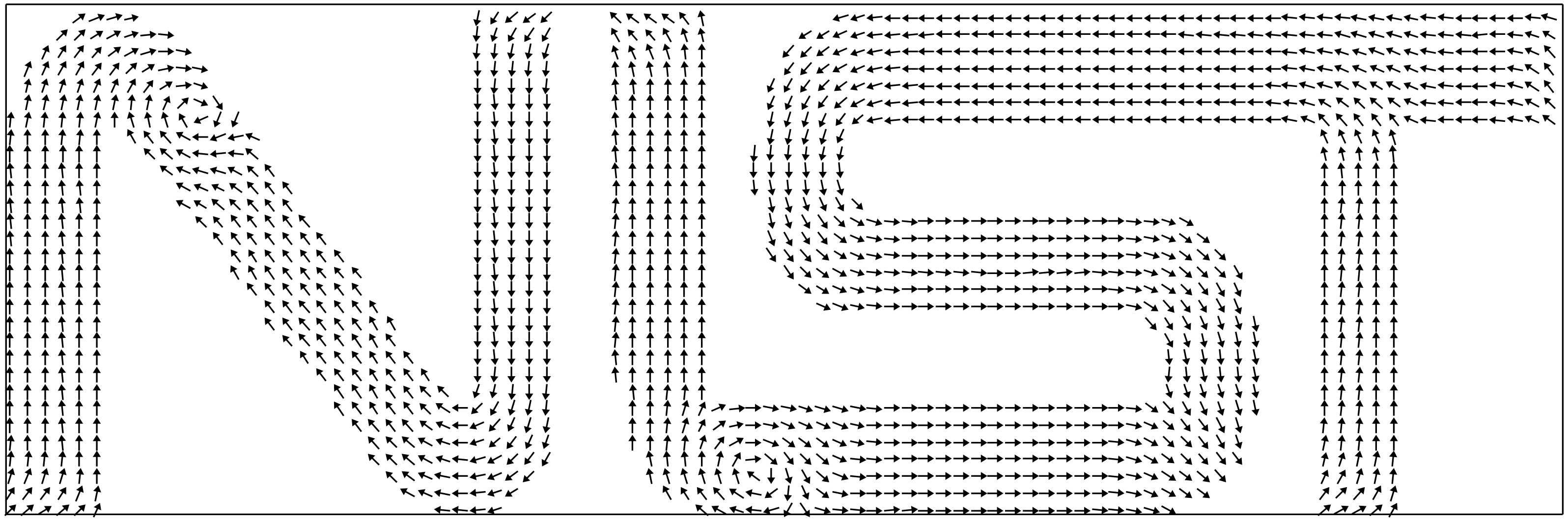


Magnetization/Field Display

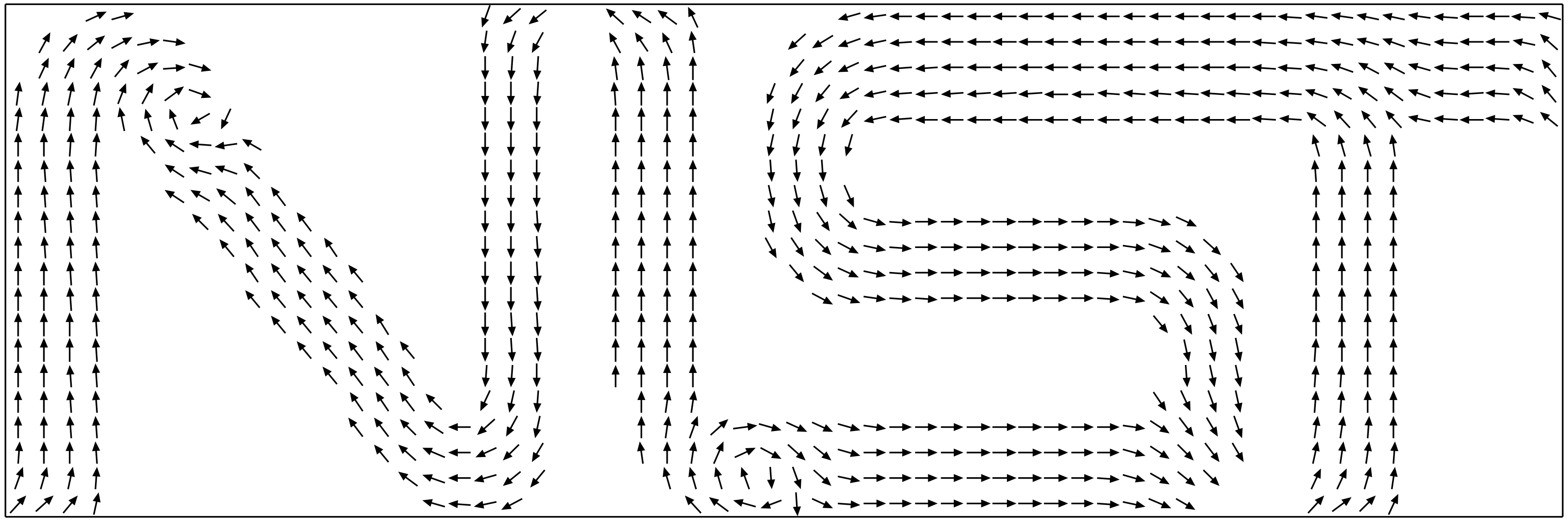




2D Solver



2D Solver



Solver Module Features

- 3D spins on a square 2D grid
- LLG ODE solver
- Demag calculated via FFT
- Anisotropy, applied field, and initial magnetization ptwise selectable
- Shaped elements

Interactive Solver Control

```
[asynca058:~/mag/OOMMF/mmsolve]$ bin-i386/mmsolve -tk
% set solver [omf_solver new -mif_filename prob1.mif]
_omf_solver0
% $solver GetEnergies
{Total 0.007102167589} {Exchange 3.826565412e-05}
{Anisotropy 4.040824693e-07} {Demag 0.007054657329}
{Zeeman 8.840523243e-06}
% $solver GetMaxTorque
0.1001536264
% $solver StepSolver
0.09989174052
% for {set i 0} {$i<10} {incr i} {$solver StepSolver}
% $solver GetMaxTorque
0.0954368046
```

```
% $solver help

'<omf_solver instance>' usage:
        $instance <instance_method> ?args?

Invalid call:
        $instance help

Invalid instance method: help

Valid instance methods:
        GetAveMag GetEnergies GetFieldStepCount
GetIterationValueLabels GetIterationValues GetMaxAngle
GetMaxTorque GetNomAppliedField GetODEStepCount
GetOutputBasename GuessAppliedFieldAxis HystLoop
IncrementFieldStep RelaxState RelaxStopCheck Reset
StepSolver WriteFieldFile WriteMagFile cget class
configure delete
```

70 Line Micromagnetic Program

```
set mif_file prob1.mif
if {$argc>0} { set mif_file [lindex $argv 0]}

# Text output subroutines
proc PrintState { solver } {
    set stepcount [$solver GetODEStepCount]
    set torque [$solver GetMaxTorque]
    set maxangle [$solver GetMaxAngle]
    set energies [join [$solver GetEnergies]]
    puts [format "%5d %10.7f  %5.1f  %12.8f %12.8f %12.8f %12.8f %12.8f" \
            $stepcount $torque $maxangle [lindex $energies 1] \
            [lindex $energies 3] [lindex $energies 5] \
            [lindex $energies 7] [lindex $energies 9]]
}

proc PrintLabels {} {
puts "  ODE     Max       Max                              ENERGIES"
puts " Step    Torque    Angle     Total       Exchange    Anisotropy    \
       Demag        Zeeman"
}

# Graph update subroutine
proc InspectLoop { solver graph } {
    global last_step
    set this_step [$solver GetODEStepCount]
    if {![info exists last_step] || $last_step!=$this_step} {
        set torque [$solver GetMaxTorque]
        set energies [join [$solver GetEnergies]]
        if {1 || $this_step % 10 == 0} {
            puts "Step [format "%4d" $this_step]: \
                    Max Torque=[format "%7.5f" $torque]"
        }
        $graph AddPoints "Total Energy" $this_step [lindex $energies 1]
        $graph AddPoints "Exchange Energy" $this_step [lindex $energies 3]
```

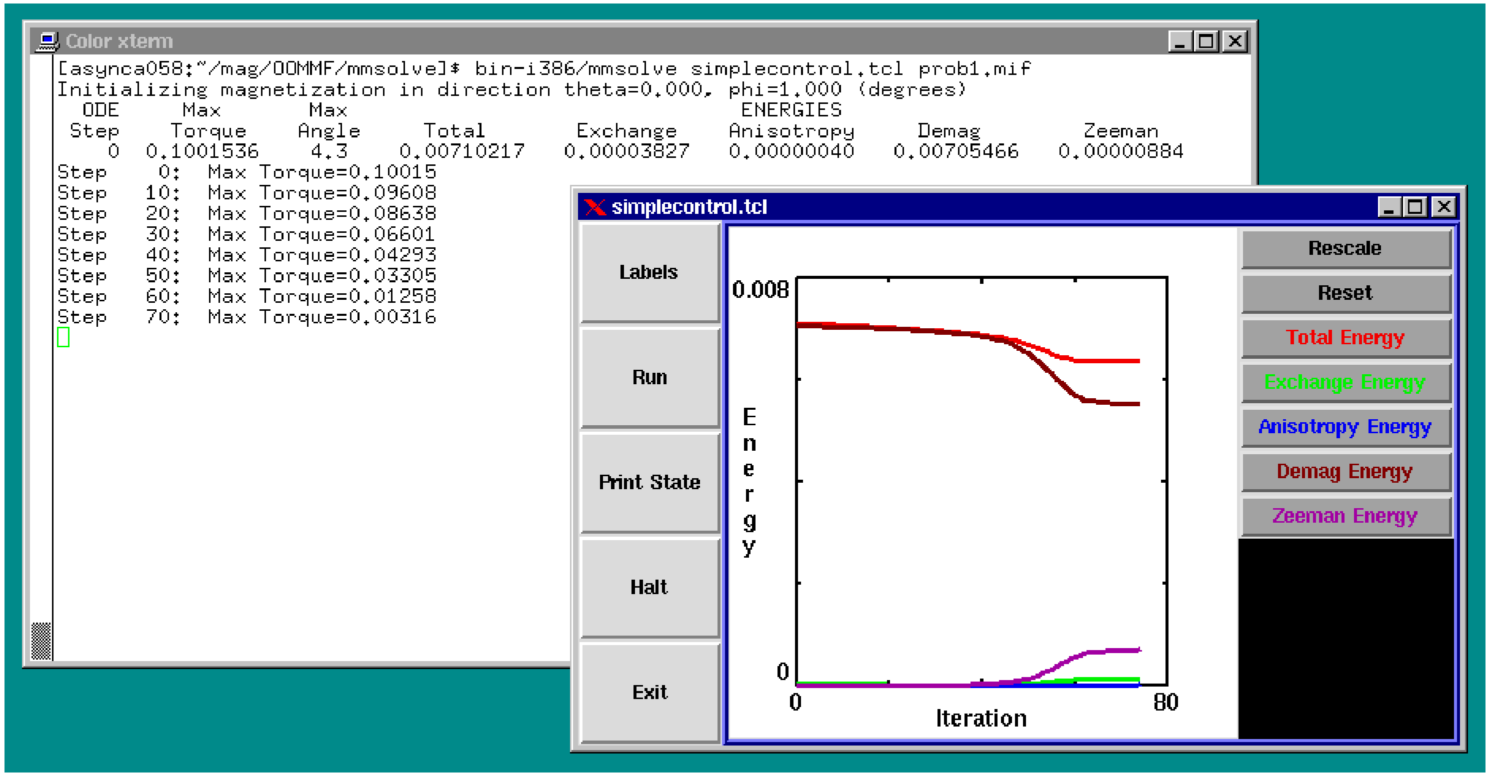
```
        $graph AddPoints "Anisotropy Energy" $this_step [lindex $energies 5]
        $graph AddPoints "Demag Energy" $this_step [lindex $energies 7]
        $graph AddPoints "Zeeman Energy" $this_step [lindex $energies 9]
    }
    set last_step $this_step
    after 20 InspectLoop $solver $graph
}

# Initialize solver and display graph
set solver [omf_solver new -mif_filename $mif_file]
$solver configure -relax_stop_torque 1e-5

# Initialize display graph
set graph [omf_graphbox new . -exit_button 0 -autodraw 1 -curve_width 3]
$graph SetLabels "Iteration" "Energy"
$graph NewCurve "Total Energy"
$graph NewCurve "Exchange Energy"
$graph NewCurve "Anisotropy Energy"
$graph NewCurve "Demag Energy"
$graph NewCurve "Zeeman Energy"
pack [$graph cget -winpath] -side right -fill both -expand 1

# Build control buttons
set btn_frame [frame .f]
set b0 [button .b0 -text Labels -command PrintLabels]
set b1 [button .b1 -text "Run" \
        -command "InspectLoop $solver $graph ; $solver RelaxState"]
set b2 [button .b2 -text "Print State" \
        -command "PrintState $solver"]
set b3 [button .b3 -text "Halt" \
        -command "$solver configure -stop_request_flag USER ; \
                  after cancel InspectLoop $solver $graph"]
set b4 [button .b4 -text "Exit" \
        -command "$solver delete ; exit"]
pack $b0 $b1 $b2 $b3 $b4 -in $btn_frame -side top -fill both -expand 1
pack $btn_frame -side left -fill y
```

70 Line Micromagnetic Program



- Controller code is written in Tcl
- Tcl code is plain ASCII, interpreted at runtime

$\Longrightarrow$ Controller code is DATA!

Module Architecture

- Benefits
  - Networked Operation: Multiple Hosts
  - Layer of Open Design
  - Interactive Response

Module Architecture

- Objectives
  - Portable
  - Multiple Simultaneous Copies
  - Module Independence
  - Open Extensible Protocol

```
# Support libraries
package require Tk
package require em
package require net

set log [em_widgetlog new -dateformat "" -title ""]
set protocol [net_protocol new -name "OOMMF log protocol 0.0"]
$protocol Wrap $log 1
set server [net_server new -protocol $protocol -alias mmLog \
             -log em_nop]
$server Start 0
wm title . mmLog<[pid]>
em_eventHandler new $log delete { after 2000 exit }
vwait omf_forever
```