Building a reusable LAMMPS script library

Craig M. Tenney

University of Notre Dame

February 2010

overview

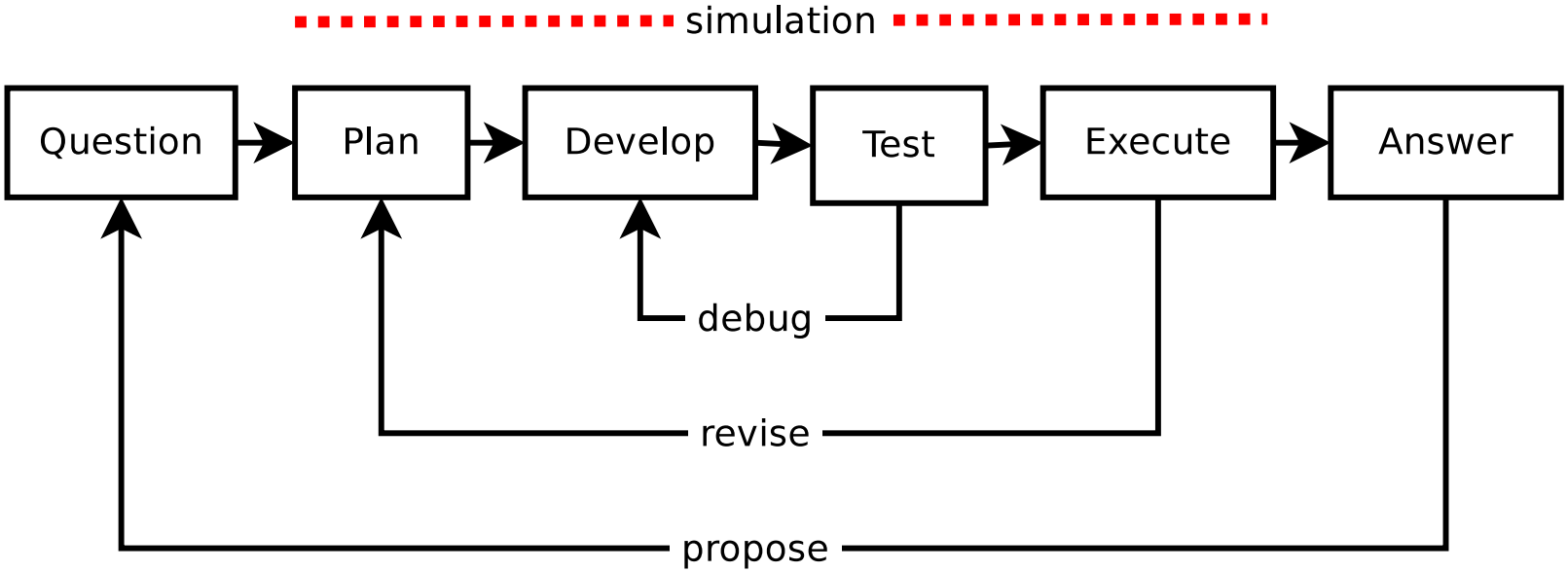
motivation

Something done once will likely need doing again...
after I've forgotten how to do it.

motivation

Something done once will likely need doing again...
after I've forgotten how to do it.

Research workflow:



Goal: loop more efficiently

- code re-use
- friendly documentation

simulation strategy v0.1

Model: one script per job

- create new script when needed
- copy and tweak existing scripts when possible

simulation strategy v0.1

Model: one script per job

- create new script when needed
- copy and tweak existing scripts when possible

Pros:

- easiest implemention
- clear connection between job and script

Cons:

- scattered script collection
- no clear evolutionary record
- frequent reinvention and regression

simulation strategy v0.2

Model: one script for many jobs

- simulation parameters are script variables
- variables are passed from command line
  (e.g. “lammps -var name value -in myscript”)

simulation strategy v0.2

Model: one script for many jobs

- simulation parameters are script variables
- variables are passed from command line (e.g. “lammps -var name value -in myscript”)

Pros:

- improved code re-use
- fairly easy implementation

Cons:

- weaker connection between job and script
- interactive and batch jobs handled differently

simulation strategy v0.2.1

Model: one script per job *type*

- general-purpose scripts stored in central location

simulation strategy v0.2.1

Model: one script per job *type*

- general-purpose scripts stored in central location

Pros:

- script evolution tracked via version control
- much improved code re-use

Cons:

- much weaker connection between job and script

current simulation strategy v0.3

Model: one script *template* per job type

- create custom script from general-purpose template (similar to building web pages dynamically)
- run LAMMPS with custom script

current simulation strategy v0.3

Model: one script *template* per job type

- create custom script from general-purpose template (similar to building web pages dynamically)
- run LAMMPS with custom script

Pros:

- clear connection between job and script

Cons:

- requires specialized front-end

implementation details

two primary components

- library of LAMMPS template scripts
  (simulation setup, modification, and production tasks)
- front-end for script manipulation and submission
  (bourne shell script)

implementation details

template = script with parameters set via "index" variables

- variables should have safe-ish default values
- runtime script is template copy with *new* variable values
  (e.g. "variable myvar index 0" in template
  becomes "variable myvar index 100")

implementation details

template = script with parameters set via "index" variables

- variables should have safe-ish default values
- runtime script is template copy with *new* variable values
  (e.g. "variable myvar index 0" in template
  becomes "variable myvar index 100")

template library

- for generality, most templates expect restart file as input
- forcefield-specific setup templates use data files
- template can include special "help" comment section

implementation details

front-end

- provide “help” framework
- collect parameter variables from command line
- create custom script from specified template
- configure runtime environment
- start interactive (test) or batch (production) job

help example 1

```
$ sub.lammps
```

help example 1

```
$ sub.lammps

Available scripts:
in.lammps.alter
in.lammps.data2restart
in.lammps.npt
in.lammps.nv
in.lammps.rnemd
in.lammps.setup-bulk
in.lammps.setup-droplet
in.lammps.sllod
in.lammps.therm_cond

Usage: sub.lammps lammps_script [--outname=NAME] [--variab
```

help example 2

```
$ sub.lammps in.lammps.npt
```

help example 2

```
$ sub.lammps in.lammps.npt

# NPT simulation of periodic (bulk) system

# these variables must be set from the command line:
variable outname index basename_of_all_output_files
variable infile index full_name_of_read_data_or_restart_fil
variable numsteps index 0
# initial and final thermostat setpoints
variable beg_temp index 300
variable end_temp index $beg_temp
# initial and final barostat setpoints
variable beg_press index 1
variable end_press index $beg_press

# options:
# snap_freq: output thermo snapshots at specified interval
variable snap_freq index 10000
```

help example 2 cont'd

```
# dump_freq: output basic dump snapshots at specified inter
variable dump_freq index 100000
# traj_freq: output detailed trajectory info at specified i
variable traj_freq index 0

# these variables have generally conservative defaults:
# kspace != 0: turn on kspace_style
variable kspace index 1
# restart_freq: output restart files at specified interval
variable restart_freq index 1000000
# new_step >= 0: reset current step to new_step
variable new_step index -1

# ___end command line variable section___

Usage: sub.lammps lammps_script [--outname=NAME] [--variabl
```

job submission example

```
$ sub.lammps in.lammps.npt --outname=mysimulation \
--infile=myrestart.0 --numsteps=10000 --beg_temp=300
```

job submission example

```
$ sub.lammps in.lammps.npt --outname=mysimulation \
--infile=myrestart.0 --numsteps=10000 --beg_temp=300

Building custom LAMMPS input script from provided variables
template: /home/ctenney/Private/src/lammps/scripts/in.lammp
replacing variable outname in mysimulation.in
replacing variable infile in mysimulation.in
replacing variable numsteps in mysimulation.in
replacing variable beg_temp in mysimulation.in

LAMMPS command:
/home/ctenney/Private/bin/lmp_serial -in mysimulation.in

LAMMPS (15 Jan 2010)
Reading restart file ...
WARNING: Restart file version does not match LAMMPS version
  restart file = 7 Jul 2009, LAMMPS = 15 Jan 2010
  orthogonal box = (-11.2178 -11.2178 -11.2178) to (100.96
```

closing

LAMMPS script library implementation

- template scripts
  - bulk and droplet simulation setup and modification
  - NPT, NVT, NVE, and various flavors of NEMD
- front-end
  - provides pseudo “man” pages
  - creates archivable LAMMPS scripts from templates
  - handles environment setup and job submission

more info on Maginn group wiki:

`http://puccini.cheg.nd.edu`