User Info for LaMEM canonical

- please update this document periodically -

Thursday 26^{th} February, 2015

Contents

1 Installation and compilation

Before you proceed with installation or re-installation **remove/comment ALL PATH** modifications related to `valgrind/mpi/petsc` in `.bash_profile`. Also comment `PETSC_DIR`, `PETSC_ARCH` environmental variables.

Important: Current version of LaMEM is compiled with `petsc3.5`.

All installation instructions can be found in `LaMEM/doc/installation/`.

1.1 Installation on Mac

1.1.1 gcc compilers

1. Install `macports` from:

   ```
   http://www.macports.org/install.php
   ```

   Use Mac OS X Package (.pkg) Installer:

   ```
   sudo port -v selfupdate
   ```

2. Install compilers and debugger

   ```
   sudo port install gcc47
   sudo port install gdb
   ```

3. Create links

   ```
   sudo ln -s /opt/local/bin/gcc-mp-4.7 /opt/local/bin/gcc
   sudo ln -s /opt/local/bin/gfortran-mp-4.7 /opt/local/bin/gfortran
   sudo ln -s /opt/local/bin/g++-mp-4.7 /opt/local/bin/g++
   sudo ln -s /opt/local/bin/ggdb /opt/local/bin/gdb
   ```

   Your compiler commands will be:

   `gcc` - C compiler
   `g++` - C++ compiler
   `gfortran` - Fortran compiler
   `gdb` - GNU debugger

1.1.2 mpich2

1. Download:

   ```
   curl http://www.mpich.org/static/tarballs/1.4.1p1/mpich2-1.4.1p1.tar.gz -o mpich2-1.4.1p1.tar.gz
   ```

2. Unpack:

   ```
   tar xvzf mpich2-1.4.1p1.tar.gz
   cd mpich2-1.4.1p1
   ```

3. Configure and install

   ```
   ./configure \
   CC=gcc \
   CXX=g++ \
   F77=gfortran \
   FC=gfortran \
   --prefix=/opt/mpich2 \
   ```

```
--enable-f77 \
--enable-fc \
--enable-cxx \
--enable-threads=runtime \
--enable-g=none \
--enable-fast=O2 \
--with-thread-package=pthreads \
--with-pm=hydra

make
sudo make install
```

4. Set path in `.bash_profile`

```
export PATH=/opt/mpich2/bin:$PATH
```

1.1.3 petsc

Current version of LaMEM is compiled with `petsc3.5`.

1. Download:

```
curl http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.5.0.tar.gz -o petsc-3.5.0.tar.gz
```

2. Unpack:

```
tar xvzf petsc-3.5.0.tar.gz
cd petsc-3.5.0
```

(a) Debug version

```
./configure \
--prefix=/opt/petsc/petsc-3.5.0-int32-deb \
--download-fblaslapack=1 \
--with-debugging=1 \
--COPTFLAGS="-g -O0" \
--FOPTFLAGS="-g -O0" \
--CXXOPTFLAGS="-g -O0" \
--with-large-file-io=1 \
--with-cc=mpicc \
--with-cxx=mpicxx \
--with-fc=mpif90 \
--with-shared-libraries=0 \
--download-metis=1 \
--download-parmetis=1 \
--download-ml=1 \
--download-hypre=1 \
--download-scalapack=1 \
--download-mumps=1 \
--download-superlu_dist=1 \
--download-suitesparse=1

make all
sudo make install
```

(b) Optimized version

```
./configure \
--prefix=/opt/petsc/petsc-3.5.0-int32-opt \
--with-blas-lapack-lib="-Wl,-framework,vecLib" \
--with-debugging=0 \
--COPTFLAGS="-O3" \
--FOPTFLAGS="-O3" \
--CXXOPTFLAGS="-O3" \
--with-large-file-io=1 \
--with-cc=mpicc \
--with-cxx=mpicxx \
--with-fc=mpif90 \
--with-shared-libraries=0 \
--download-metis=1 \
--download-parmetis=1 \
--download-ml=1 \
--download-hypre=1 \
--download-scalapack=1 \
--download-mumps=1 \
--download-superlu_dist=1 \
--download-suitesparse=1

make all
sudo make install
```

3. Set paths in `.bash_profile`:

```
export PETSC_DEB=/opt/petsc/petsc-3.5.0-int32-deb
export PETSC_OPT=/opt/petsc/petsc-3.5.0-int32-opt
```

Note: No more `PETSC_DIR` and no more switching between the debug and optimized versions. Makefile is now supporting building both versions simultaneously.

1.2 Installation on clusters

If modules and/or paths to compilers and petsc are not provided, please refer to installation instructions found in `LaMEM/doc/installation/`.

Currently, just add these modules or paths to .bashrc (Gaia, Sith and Mogon) and logout/login for them to take effect.

Gaia and Sith:

Add this to .bashrc:

```
# PETSC 3.5.0
export PETSC_DEB=/opt/petsc/petsc-3.5.0-int32-deb
export PETSC_OPT=/opt/petsc/petsc-3.5.0-int32-opt
```

Mogon:

Add this to .bashrc:

```
# MODULES
module load gcc/4.8.2
module load mpi/platform_mpi_ce/9.1.2
module load acml/5.3.0/gfortran/gfortran64_fma4
module load software/petsc/3.5.1/gcc_4.8.2_platformmpi_9.1.2
```

1.3 Get a local copy of LaMEM

To get access to LaMEM repository, you should have an account on Gaia (if it does not work, contact Boris for account info):

```
ssh USERNAME@gaia.geo.uni-mainz.de
```

1. If you want to checkout a new LaMEM repository, use this command:

   ```
   svn co svn+ssh://USERNAME@gaia.geo.uni-mainz.de/local/home/lkausb/svn/LaMEM/trunk ./LaMEM
   ```

2. If you have a version of LaMEM that you checked out previously already and are using, you don't have to delete that. Instead, go to the `./LaMEM` directory and type

   ```
   svn relocate svn+ssh://USERNAME@gaia.geo.uni-mainz.de/local/home/lkausb/svn/LaMEM/trunk
   ```

   If SVN complains that it doesn't know a 'relocate command', you have an older version of svn and we have to do two steps:

   (a) `svn switch --relocate svn+ssh://USERNAME@musashi.ethz.ch/var/svn svn+ssh://USERNAME@gaia.geo.uni-mainz.de/local/home/lkausb/svn`
   (b) `svn switch svn+ssh://lkausb@gaia.geo.uni-mainz.de/local/home/lkausb/svn/LaMEM/trunk`

3. If you are using the 'Versions' software, make a right-click on the 'bookmark' of your LaMEM directory, and select "Edit Bookmark". Then, change 'Location' to:
   `svn+ssh://USERNAME@gaia.geo.uni-mainz.de/local/home/lkausb/svn/LaMEM/trunk`
   and change your username and password to the ones on GAIA. After 'save', it should automatically relink it.

   You can verify that everything is correct by typing: `svn info` in the LaMEM directory. It should have as URL:
   `URL: svn+ssh://username@gaia.geo.uni-mainz.de/local/home/lkausb/svn/LaMEM/trunk`

1.4 Compilation of LaMEM

Once the compilers and pets have been installed, you can now compile LaMEM. Go to `./LaMEM/src` directory and do the following:

debug mode: `make mode=deb all`
or simply `make all`, `mode=deb` is the default to build everything in debug mode.

optimized mode: `make mode=opt all`
to build everything in optimized mode.

Note: No more `PETSC_DIR` and no more switching between the debug and optimized versions with `.bash_profile`. Makefile is now supporting building both versions simultaneously.

1. SVN
   Also it's not necessary to do svn updates explicitly.
   `make update` to update to latest version, and to write a new Version.h file.

2. VIEW ENVIRONMENT
   `make mode=deb print`
   or
   `make mode=opt print`
   to view your environmental variables.

3. TESTS (not yet in LaMEM canonical)

   Running tests, type in the `./LaMEM/tests` directory:
   `make mode=deb check` to perform tests in debug mode.
   `make mode=opt check` or (or simply `make check`, `mode=opt` is the default) to perform tests in optimized mode.

   Note: Take care that `mode=deb` is the default for LaMEM, and `mode=opt` is the default for tests.

1.5 Running LaMEM

The LaMEM executable is located in `/LaMEM/bin/deb/` for the debug version and `/LaMEM/bin/opt/` for the optimized version. You can run LaMEM by typing (give the path to the right LaMEM executable):

```
mpiexec -n <no_cpu> LaMEM -ParamFile <input_file> <command_line_options>
```

2 Physics

3 Input Parameters

Examples of input files can be found in `/LaMEM/input_models/PROJECTS_FDSTAG_CANONICAL/`. In this sections, all possible input parameters are explained.

3.1 General input parameters

Setting Input Parameters in input file and/or from command line

The general command to run LaMEM is:
`mpiexec -n <no_cpu> LaMEM -ParamFile <input_file> <extra command line options>`

NOTE:

In general, parameters can be specified in the input file or from the command line. The command line options need to have a "-" prefix (i.e. `-L 10 -W 100 -H 90`).

If multiple definitions of the same parameter, command line options overwrite input file options! Some input parameters are not needed anymore, such as: `num_phases`.

Table 1 lists all major parameters. Please add new LaMEM parameters or options to this table.

3.2 Variable grid spacing

3.3 PETSc and solver options

3.4 Boundary Conditions

3.5 Pushing BC

3.6 Output

Variable	Name	Type	Comments
`nel_x`	# of cells in x-dir	int	Negative number implies corresponding number of mesh segments for variable grid spacing
`nel_y`	# of cells in y-dir	int	same as for `nel_x`
`nel_z`	# of cells in z-dir	int	same as for `nel_x`
`seg_x`		scalar array	Array includes: 1. coordinates of the delimiters between the segments (n-1 points); 2. number of cells (n points); 3. bias coefficients (n points). See Variable grid spacing
`seg_y`		scalar array	same as for `seg_x`
`seg_z`		scalar array	same as for `seg_x`
`DimensionalUnits`	dimensional indicator	int	0 - non-dimensional, 1 - dimensional; See Scaling
`L`	length (y-dir)	scalar	domain size in y-dir
`W`	width (x-dir)	scalar	domain size in x-dir
`H`	height (z-dir)	scalar	domain size in z-dir
`x_left`	coord of left corner	scalar	the x-coords: `(x_left:nel_x:x_left+W)`
`y_front`	coord of front corner	scalar	the y-coords: `(y_front:nel_y:y_front+L)`
`z_bot`	coord of bottom corner	scalar	the z-coords: `(z_bottom:nel_z:z_bottom+H)`
`msetup`	model setup	string	See Model Setup
`ParticleFilename`	name of markers file	string	See (parallel and redundant input)
`LoadInitialParticlesDirectory`	name of directory	string	directory where the marker files are
`NumPartX`	# of markers/cell (x-dir)	int	min = 2
`NumPartY`	# of markers/cell (y-dir)	int	min = 2
`NumPartZ`	# of markers/cell (z-dir)	int	min = 2
`OutputFile`	name of output file	string	
`save_timesteps`	save every # timesteps	int	example: `save_timesteps = 10` will save output every 10 time steps
`time_end`	# of timesteps	int	the simulation will finish after (0+`time_end`)
`save_breakpoints`	save every # breakpoints	int	needed to restart simulation; example: `save_breakpoints = 10` will save breakpoint files every 10 time steps
`SaveParticles`	save markers indicator	int	1 - save markers to file after every time step
`-AddRandomNoiseParticles`	flag for random noise	int	if `TRUE`, will add some random noise to marker distribution
`-restart`	restart flag	int	0-no restart; 1-restart simulation from the last saved breakpoint. If `restart = 1`, but no breakpoint files were found, it will start the simulation from beginning.
`-SavePartitioning`	proc partitioning flag	bool	if `TRUE`, the domain partitioning between processors will be printed to file. Warning: simulation will be stopped after the file was created.
`CFL`	Courant criterion	scalar	should be ¡ 1 (generally taken as 0.5)
`dt_max`	max. time step	scalar	time step is calculated internally with the CFL, but this is the maximum admisable
`DII_ref`	initial reference strain rate	scalar	needed for initial guess in non-linear solve
`LowerViscosityCutoff`	lower viscosity cutoff	scalar	min viscosity allowed in the simulation
`UpperViscosityCutoff`	upper viscosity cutoff	scalar	max viscosity allowed in the simulation
`InitViscosity`	initial reference viscosity	scalar	needed for initial guess in non-linear solve
`<Characteristic Values>`	Scaling	struct	See Scaling
`<BoundaryConditions>`	Boundary conditions	struct	See Boundary Conditions
`<Output>`	Output	struct	See Output
`<Pushing>`	Pushing BC	struct	See Pushing BC
`<Softening Laws>`	Softening Laws	struct	See Softening Laws
`<Material Properties>`	Material parameters	struct	See Material Properties
`<PetscOptions>`	Petsc and solver options	struct	See Petsc and Solvers
`-use_quasi_harmonic_viscosity`	flag	bool	flag to compute quasi-harmonic viscosities

Table 1: Major parameters to be set in input file/command line. Note, if you set them from the command line, every parameter name should have a "–" prefix (i.e. `-L 10 -W 100 -H 90`)

3.7 Scaling, characteristic values and units

The units in the input file should be consistent with the type of units requested.

To run a **non-dimensional setup** set these parameters in the input file:

```
DimensionalUnits  = 0
units             = none
```

To run a **dimensional setup**, set:

```
DimensionalUnits  = 1
units             = si # all input parameters are given in SI units
```

OR

```
DimensionalUnits  = 1
units             = geo # all input parameters are given in geological units
```

If `DimensionalUnits = 1`, the following parameters (characteristic values) **must** also be specified in input:

```
Characteristic.Length       = 1 # depends on domain size
Characteristic.Viscosity    = 1 #
Characteristic.Temperature  = 1 #
Characteristic.Stress       = 1 #
```

Characteristic values are dependent on model setup (check how they are set in the examples in `/LaMEM/input_models/`). They are needed to perform the non-dimensionalization (scaling) of the model. For this, all input parameters, characteristic values and material parameters should have consistent units!

Table 2 shows in which units parameters should be given. Note: degrees for angles are the preferred units, also for `units = si`.

	`units`		
Parameter	`none`	`si`	`geo`
mass	-	kg	kg
time	-	s	Myr
length	-	m	km
temperature	-	K	C
force	-	N	N
angle	-	deg	deg
velocity	-	m/s	cm/yr
stress	-	Pa	MPa
strain rate	-	1/s	1/s
heat flux	-	W/m^2	mW/m^2
dissipation rate	-	W/m^3	W/m^3
angular velocity	-	deg/s	deg/Myr
density	-	kg/m^3	kg/m^3
viscosity	-	Pa.s	Pa.s
	Derived units		
gravity strength	= force/mass		
energy	= force*length		
power	= energy/time		
specific heat	= energy/mass/temperature		
conductivity	= power/length/temperature		
heat production	= power/mass		
expansivity	= 1/temperature		
pressure sensitivity	= temperature/stress		

Table 2: Setting dimensional units in input file.

3.8 Softening laws

Every softening law should be defined in the input file in a structure beginning with `<SofteningStart>` and ending with `<SofteningEnd>`, and should have a unique phase identifier (`softID`), a softening ratio (`A`), a start of softening (`APS1`) and an end of softening (`APS2`) such as below.

Each parameter is explained in Table 3. Maximum number of softening laws: 10 (this number can be changes in LaMEM.h - max_num_soft).

```
<SofteningStart>
   softID = 1
   A      = 0.5
   APS1   = 1
   APS2   = 3
<SofteningEnd>
```

Parameters to define	Definition	Dimen. values	Comments
`softID`	soft law id	0 - 9	always set this parameter
`A`	softening ratio	[0,1]	`A = 1 - a/a0`, where `a` - is cohesion(or friction angle) after softening, and `a0` is cohesion (or friction angle) before softening and specific for each material
`APS1`	weakening plastic strain (begin)		
`APS2`	weakening plastic strain (end)		

Table 3: List of parameters needed to define a softening law.

3.9 Material parameters

Setting Material Parameters in input file

Every material phase should be defined in the input file in a structure beginning with `<MaterialStart>` and ending with `<MaterialEnd>`, and should have a unique phase identifier (`ID`), such as below. The principle of setting material parameters, is that, if you specify it, it will be active; you don't specify it, it's not going to be active. All the material parameters and cases are listed in Table 4. Maximum number of phases: 32 (this number can be changes in LaMEM.h - max_num_phases).

```
<MaterialStart>
   ID = 1
   ...
<MaterialEnd>
```

Examples:

1) Viscous falling block with 2 material phases (non-dimensional)

```
<MaterialStart>
   ID = 0
   rho0 = 1
   eta = 1
<MaterialEnd>
```

```
<MaterialStart>
   ID = 1
   rho0 = 2
   eta = 1e3
<MaterialEnd>
```

2) Power-law material (non-dimensional)

```
<MaterialStart>
   ID = 0
   rho0 = 1
   Bn = 0.5
   n = 3
<MaterialEnd>
```

OR

```
<MaterialStart>
   ID = 0
   rho0 = 1
   eta0 = 1e3
   e0 = 1e-5
   n = 3
<MaterialEnd>
```

3) Defining a visco-elasto-plastic material phase with T-dependence (no strain weakening or radiogenic heat production)

```
<MaterialStart>
   ID             = 0      # phase id
   rho0           = 3000   # density [kg/m3]

   # creep
   eta            = 1e20   # ref.  viscosity [Pa.s]
   eta0           = 1e18   # ref.  viscosity [Pa.s]
   e0             = 1e-15  # ref.  strain rate [1/s]
   n              = 3      # power-law exponent
   Ed             = 3e5    # activation energy [J/mol]
   En             = 5.4e5  # activation energy [J/mol]

   # elasticity
   shear          = 5e10   # elastic shear modulus [Pa]

   # plasticity
   cohesion       = 4e6    # cohesion [Pa]
   friction       = 30     # friction angle [deg]

   # temp
   alpha          = 1e-5   # coeff.  of thermal expansion [1/K]
   cp             = 1.2e3  # heat capacity [J/kg/K]
   k              = 2.5    # thermal conductivity [W/m/K]
<MaterialEnd>
```

Errors related to material properties input:

1. Phase `ID` is not specified
2. If `rho0` is not specified - this is an essential parameter

3. If no creep law is specified (i.e. `Bd, Bn, and Bp` are all 0). Make sure you set parameters for at least one creep law!
4. Power-law:
   (a) If specified (`eta0, e0`) or (`Bn`), but not `n`.
   (b) If specified `n`, but not all/any (`eta0, e0`) or (`Bn`).
5. Peierls creep: if `Bp` is not defined, but any of the others Peierls parameters is.
6. Plasticity: if `cohesion` and `friction` are not defined together.
7. Temperature: if `cp` and `k` are not defined together.

Case	Parameters to define	Definition	Dimen. values	Comments
constant density	`rho0`	reference density	0 - 5000 kg/m^3	
linear viscous	`eta`	reference viscosity	1e-10 - 1e28 Pa.s	either define (`eta`) or (`Bd`);
(diffusion creep)	**OR (not both)**			if both `eta` and `Bd` are defined, `Bd` has priority
	`Bd`	diff. creep coeff.		`Bd = 1/(2*eta)`
	`Ed`	diff. creep activation energy	3.0e5 J/mol	(*T)
	`Vd`	diff. creep activation volume	cm^3/mol	(*T)
power-law	`eta0`	ref. viscosity	1e-10 - 1e28 Pa.s	either define (`eta0, e0, n`) or
(dislocation creep)	`e0`	ref. strain rate	1e-15 s^{-1}	(`Bn, n`); if all are defined, `Bn` has priority;
	`n`	power-law exponent	1 - 10	`e0` and `n` must be defined together
	OR (not both)			
	`Bn`	disl. creep coeff.		`Bn = (2*eta0)`$^{-n}$ `e0`$^{(1-n)}$
	`En`	disl. creep activation energy	5.4e5 J/mol	(*T)
	`Vn`	disl. creep activation volume	cm^3/mol	(*T)
Peierls creep	`Bp`	pre-exponential constant	~5e11 s^{-1}	
	`Ep`	disl. creep activation energy	5.4e5 J/mol	(*T)
	`Vp`	disl. creep activation volume	cm^3/mol	(*T)
	`taup`	scaling stress	~ 1e9-1e10 Pa	more info - Kameyama et al. 1999
	`gamma`	approx. parameter	0.1	
	`q`	stress-dep. param.	2	
elasticity	`shear`	elastic shear modulus	~500 kbar	if not specified, no elasticity
	`bulk`	elastic bulk modulus	~500 kbar	
	`Kp`	pressure-dep. param		
plasticity	`cohesion`	cohesion	10 MPa	`cohesion` and `friction` must be
(Drucker-Prager)	`friction`	friction angle	0 - 45[30°]	defined together; if not specified, no plasticity
	`chSoftID`	ID for softening law	< numSoft	Strain weakening parameter
	`frSoftID`	ID for softening law	< numSoft	Strain weakening parameter
temperature (*T)	`alpha`	coefficient of thermal expansion	10^{-5}K^{-1}	`rho=rho0*(1-alpha(T-T0));` where `T0` = [0°C, 273.15K]
	`cp`	specific heat (capacity at ct. pressure)	J/kg/K	
	`k`	thermal conductivity	W/m/K	
	`A`	radiogenic heat production	10^{-8}-10^{-8} W/m^3	

Table 4: List of material parameters that can be specified for a material phase. **Notes: 1.** (*T) Temperature-dependent constitutive laws are active only when 'energy' equation is coupled. **2.** Activation energy and volume can be specified for every creep law.

4 Model Setup

5 Visualization