magpar

version 0.9rc2 build 2916M

Generated by Doxygen 1.6.1

Tue Dec 8 15:52:48 2009

Contents

Chapter 1

magpar - Parallel Finite Element Micromagnetics Package

Author:

Werner Scholz
http://www.magpar.net/

Version:

```
#define MAGPAR_VERSION "0.9rc2"

#define MAGPAR_REVISION "2916M"
```

Warning:

This software is still in experimental stage. Use at your own risk.

1.1 Documentation

If you use magpar and publish results, which have been obtained/derived from its simulations, please cite the paper given on the Publications page and drop me a note at magpar(at)magpar.net.

This reference manual is also available in PDF format (without figures though).

1.2 License

magpar is distributed under the terms of the GNU General Public License

```
Copyright (C) 2002-2009 Werner Scholz

www:   http://www.magpar.net/
email: magpar(at)magpar.net

magpar is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.

magpar is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with magpar; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
```

1.3 Acknowledgment

I would like to gratefully acknowledge the encouragement and support of my supervisors Josef Fidler and Thomas Schrefl from the Advanced Magnetism Group at the Vienna University of Technology.

My friends and colleagues Dieter Suess, Rok Dittrich, Hermann Forster, Vassilios Tsiantos and Thorsten Matthias supported me with many tips and tricks.

In addition I would like to thank Roy Chantrell for his encouragement, which finally motivated me to implement this package.

Chapter 2

Introduction

The properties of modern magnetic materials are strongly influenced by their microstructure. The continuing improvement of the properties of SmCo based magnets has been made possible by additives like Cu and Zr and a refined processing route and heat treatment, which have a great impact on the resulting cellular precipitation structure. The typical size of the cells is in the order of 100 nm with an intercellular phase of around 10 nm. The particle size in magnetic recording tapes is of the same order of magnitude. The typical grain size in current hard disk storage media is about 8 nm with an intergranular region of about 2 nm for exchange decoupling of the grains.

These structures are so small that quantum mechanical effects like exchange have to be taken into account. However, they are too large for a pure quantum mechanical description, which would exceed the capabilities of today's ab-initio computational models. On this intermediate level between the macroscopic world and a description with atomic resolution, micromagnetic models have proved to be a useful tool. These computational models provide great freedom in the choice of experiment conditions and in the variation of material parameters. In addition to measurements of the remanent magnetization and the coercive field, it is possible to study the details of the magnetization distribution and the magnetization reversal processes, which are difficult to investigate experimentally.

In various fields of computer aided engineering like structural analysis, fluid dynamics, and electromagnetic field computation, as well as micromagnetics the finite element method has been successfully applied. Especially its flexibility in modeling arbitrary geometries has made it very popular. In the light of the importance of the microstructure of magnetic materials the finite element method has been chosen for the implementation of a micromagnetic model.

There are several commercial and open source micromagnetics packages available, however all of them use the finite difference method. In addition, static energy minimization methods for the study of SmCo permanent magnets as well as dynamic time integration methods for the investigation of the magnetization dynamics in magnetic nanoparticles are desirable.

Therefore, a finite element micromagnetics package has been implemented which combines several unique features:

- **Applicability** to a variety of static and dynamic micromagnetic problems including uniaxial and cubic anisotropy, exchange, magnetostatic interactions and external fields
- **Flexibility** of the finite element method concerning the geometry and accuracy by using unstructured graded meshes
- **Availability** due to its design based on free, open source software packages
- **Portability** to different hardware platforms, which range from simple PCs to massively parallel supercomputers
- **Scalability** due to its highly optimized design and efficient libraries
- **Versatility** by including static energy minimization and dynamic time integration methods

Chapter 3

Features

- Micromagnetics (see Input Files for Simulations)
  - Uniaxial and cubic anisotropy
  - Exchange
  - Magnetoelastic effects
  - Magnetostatic field (hybrid FEM/BEM method)
  - External field (quasistatic, sweeping, rotating)
- Dynamic integration of Landau-Lifshitz-Gilbert equation of motion using the SUNDIALS library
- Static energy minimization using the TAO library
- Data output (Output Files of Simulations)
  - Log files
  - PNG files
  - UCD files
  - "sampling line"
- Mesh import (see Preprocessing)
  - MSC.Patran neutral file
  - AVS project.inp, project.out: finite element mesh file
  - Gmsh , GiD meshes
- Mesh analysis (see Output Files of Simulations)
  - element and node volumes (max,min,avg)
  - edge lengths (max,min,avg)
  - element quality check
  - model bounding box
  - volume by property id
- Mesh distortion: shift, scale mesh; mimic surface/interface roughness
- Mesh refinement
  full regular refinement before partitioning:
  x8^n number of nodes and elements for n refinement iterations
- Problem independent parallelization (see Performance)
- Performance evaluation: timing, profiling (in every subroutine)
- Easy activation of optional components
- Consistency checking: assert statements
- Memory allocation tracking: PetscMalloc, PetscFree, memory usage statistics
- C++ compatible: required by TAO

Chapter 4

ChangeLog

- **version 0.9 (TBD) - TBD**
  - added Rok Dittrich's demag energy and demag factor calculator for regular prisms: s_demagcalc ({2586})
  - appended suffix to all scripts, e.g. .py, .sh, .awk - helps Windows users ({2618})
  - new command line option -condinp_file_t_ns (allopt.txt) implemented in checkiterationllg.c ({2535},{2544})
  - lock magnetization with alpha=999 in project.krn: material properties ({2540})
  - deactivated mesh partitioning, activated bandwidth optimization by default ({2401},{2471},{2475})
  - fixed problem where magpar exits too early when hstepfile is used - thanks to a bug report by Dr. Han ({2521})
  - added Stefan Tibus's new gmsh: msh2inp.awk (implemented in awk) ({2538})
  - replaced calls to Metis functions by new implementations: Mesh2Nodal(), Mesh2Dual() (this removes magpar's dependency on Metis - unless mesh partitioning is used) ({2674})
  - magpar runs in parallel also on Windows: Compiling magpar in Cygwin for Windows ({2696},{2712})
  - update and default to PETSc version 3.0.0 and TAO 1.10 ({2706},{2711})
  - removed support for PETSc versions older than 2.3.3 ({2400})
  - removed support for SUNDIALS versions older than 2.3.0 ({2400})
  - update for new default library versions: PETSc 3.0.0-p10 ({2890}), TAO 1.10-p1 ({2890}), LAPACK 3.2.1, MPICH2 1.1.1 ({2667}), libpng 1.2.40 ({2890}), ParMetis 3.1.1 ({2559}),
  - added options for downloading, compiling, activating hypre in PETSc-config-magpar.py (inactive by default) ({2519})
  - improvements to the initialization routines, which can reduce the setup time and memory requirements for large models significantly, especially matrix preallocation for the field matrices and in facnb.c::FacNB ({2562},{2564},{2571},{2583},{2585},{2697})
    - various improvements, fixes to documentation, e.g. "Physical Volumes" in Gmsh models (.geo files) have to start with 1 (Gmsh); reported by Rajmani Gaur - thanks! ({2633})
  - updated list of Publications ({2705},{2708})

- **version 0.8 ({2383}) - Dec. 14, 2008**
  - revised Makefile system ({1148},{1187},{1263}-{1267},{1839},{2002},{2004})
  - updated documentation, added call and caller graphs (e.g. see main.c), manual in PDF format, fixed and updated links ({1151},{1383},{1362},{1539},{1554})
  - updated list of Publications ({1689},{1980},{2314},{2360}), new sections in FAQ : Running magpar on Windows, How does boundary matrix size scale with mesh length?, Interpolating in a tetrahedral mesh with barycentric coordinates ({1617},{1814})
  - added section on MagParExt, the magpar Graphical User Interface for Windows developed by Tomasz Blachowicz and Bartlomiej Baron ({1595},{1597})
  - added section on Compiling magpar using the MinGW compilers in Cygwin ({2271},{2322},{2327},{2338})
  - fixed field sweep implementation (thanks to patches by Stefan Tibus, Univ. of Konstanz) ({1153}); use "-tol 0" with field sweep! ({1165})
  - new option -condinp_equil_j (allopt.txt) implemented by Stefan Tibus ({1690})
  - new option -hext_ho_hstepfile (allopt.txt) implemented by Stefan Tibus ({1691})

- various fixes and speedups for initialization of exchange and anisotropy field, calculate energy density, average magnetization based on magnetic volume so airboxes do not skew it ({1158},{1170},{1262},{2128},{2169},{2242})
- speed up boundary matrix calculation, separated out function bele.c to calculate contributions to individual matrix elements ({1203},{1940})
- set new defaults for TAO options ({2310},{2326}) - see allopt.txt ; implemented new convergence test for energy minimization ({2284})
- added inp2vt converters from Stefan Tibus (vtk tools) ({1430})
- fixed png output to honor option -res again ({1225})
- fixed png output to take cut at the correct position ({1226},{1233},{1246})
- added second png slice output (see writedatapng2.c and allopt.txt) ({1414})
- fixed problems with demag field calculation in certain geometries/FE meshes: fix suggested by Hiroki Kobayashi, Fujitsu ({1408})
- added General programming guidelines section to Programming, Debugging, Bug Reporting ({1407})
- update for latest library versions (libpng, petsc, mvapich2, parmetis) ({1405},{1424},{1428},{1739},{1933},{2220},{2311},{2367}); (override settings as shown in Makefile.in.host_oldlibs to continue using older library versions)
- fixed "-init_mag 12": head-to-head transverse domain wall with transition region ({1425})
- added "-init_mag 13": head-to-head vortex wall ({1569})
- updated Programming, Debugging, Bug Reporting page ({1454})
- fixed run modes 2,3 (allopt.txt) ({1459},{1461})
- changed default divtol for psolve KSP to 100 ({1503})
- checked magpar with `Valgrind` for memory corruption

- **version 0.7 - Nov. 2, 2007**
  - update for PETSc version 2.3.3 (requires TAO 1.9)
  - update for PETSc version 2.3.2 (requires TAO 1.8.2)
  - dropped support for PETSc version 2.1.6 and earlier (i.e. version 2.2.0 or later required)
  - update for SUNDIALS version 2.3.0
  - dropped support for SUNDIALS versions before 2.1.0
  - SUNDIALS/PVode saves its information in a separate log file project.log_pvode
  - improved Makefile system, updated Installation instructions
  - created new Makefile.in.tmpl for machine specific settings (cf. magpar)
  - created new Makefile.libs for automatic download, configuration, compilation, installation of all libraries
  - new import filter gmsh: gmshtoucd.py for Gmsh finite element meshes
  - new OpenDX tools: inp2dx converter and visual program
  - moved all tools to src/tools/
  - updated example sphere_larmor: Larmor precession and added example mumag3b: mumag standard problem #3 with 2 cubes with Gmsh geometry and mesh files
  - updated implementation of magnetoelastic effects and accompanying example stress: Magnetoelastic effects on domain structure (originally implemented by Ahmet Kaya from the research group of `Jim Bain` and `Jimmy Zhu` at the `Data Storage Systems Center (DSSC)` at `Carnegie Mellon University`)

  - all magpar options now have (reasonably sane) default values if not defined (complete list and defaults given in allopt.txt: simulation parameters)
  - all external fields are now added up in hexternal.c, so different implementations can be active at the same time
  - source code restructured and cleaned up: renamed files (e.g. hexternalcust.c is now hext_cu.c, moved code around, combined/removed some files; this wil likely break any third party extensions (e.g. hexternalcust.c, etc. - please contact me if you need help porting your extensions to this new magpar version).
  - reduced number of global variables (reduced size of struct GridData) by using more local and static variables
  - removed (incomplete) implementation of periodic boundary conditions
  - automated regression testing using standard examples
  - simplified input files for examples, added sample output files
  - added new simulation modes (2,3) (check allopt.txt: simulation parameters for details)
  - new options (check allopt.txt: simulation parameters for details) :
    * -mesh_scale
    * -shift
    * -optimizebw
    * -metispartition
    * -nsliceprop{ser,par}
    * -mode 99
    * -hdemag_u{1,2}_ksp_type
    * -helastic_propfile
    * -hext_cu
    * -logpid
    * -psolve_ksp_*
    * -ts_logsteps
    * -ts_nsteps
  - removed several options (check allopt_ret.txt)
  - renamed magpar executable magpar.exe (instead of magpar.{O,g}_c++)
  - new version of ngtoucd.py (which recognizes several top-level-objects and assigns them different property ids) contributed by Richard Boardman

- **version 0.6 - (private release)**

  - added magnetoelastic effects and accompanying example stress: Magnetoelastic effects on domain structure implemented by Ahmet Kaya

- **version 0.5 - Oct. 29, 2005**

  - update for PETSc version 2.3.0 (recommended but not required)
  - in turn, PETSc 2.3.0 requires the new TAO release 1.8 (cf. Installation)
  - speedup of about a factor of 2 (!) with PETSc 2.3.0 in parallel simulations
  - update for SUNDIALS version 2.1.0 (completely new API); magpar is NOT compatible with SUNDIALS version 2.0, but it is still backward compatible with SUNDIALS 1.0 (March 9, 2004 or earlier)!
  - added cubic anisotropy (implemented by Greg Parker)

  - expanded format of project.krn: material properties for third Euler angle for cubic anisotropy
  - added example for cubic anisotropy: sphere_cubic: Single domain particle with cubic anisotropy
  - added K_2 term to uniaxial anisotropy (implemented by Greg Parker)
  - storing data of the external field in project.INP.gz
  - magpar has been successfully compiled and run on Mac OS X: Apple Macintosh running Mac OS X (thanks to Richard Boardman and Greg Parker)
  - update for Single processor version without MPI
  - update for Compiling magpar in Cygwin for Windows
  - magpar executables for Windows are available on the magpar homepage.

- **version 0.4 - March 16, 2005**

  - update for PETSc version 2.2.1 (recommended but not required), patch by Richard Boardman
  - in turn, PETSc 2.2.1 requires the new TAO release 1.7 (cf. Installation)
  - please upgrade to PETSc version 2.2.1 and TAO 1.7: support for older versions of PETSc (<2.2.1) will be dropped in the next release (to make maintenance of the code easier)
  - use analytical formula for stiffness matrix elements of anisotropy field (cf. stiffmat.c)
  - changed format of project.krn: material properties file: added one column for damping constant, which can now be defined individually for each grain/part of the model (implemented upon request by Bill Bailey)
  - thus, removed option "-alpha" from allopt.txt: simulation parameters
  - symmetrized stiffness matrices for magnetostatic field
  - thus, CG solver with ICC preconditioning can be used (update options in your allopt.txt: simulation parameters files according to the template allopt.txt and the Examples!)
  - added new options for user defined selection of linear solver in psolve: "-psolve_*" (cf. allopt.txt)
  - update for 64-bit Linux (patch by Richard Boardman)
  - removed dependency on cblas, updated Installation instructions
  - added example thinfilm: Thin magnetic film
  - added magconv to manual
  - updated documentation (especially Installation, FAQ); thanks for many suggestions to Jehyun Lee
  - added ngtoucd.py tool by Richard Boardman and Hans Fangohr to contrib directory

- **version 0.3.1 - Aug. 30, 2004**

  one-line bugfix in src/init/serinit.c: uninitialized pointer

- **version 0.3 - July 21, 2004**

  - update for PETSc version 2.2.0 (not required)
    patches contributed by Richard Boardman - thanks!
  - in turn, PETSc 2.2.0 requires the new TAO release 1.6 (cf. Installation)
  - update to SUNDIALS version of March 9, 2004 (PVODE unchanged, update not required)
  - fixed preconditioning for LLG integration with PVODE
  - removed code for cubic anisotropy

  - project.krn file format simplified:
    any data after exchange constant are ignored (cf. project.krn: material properties)
  - removed parameters k2 and js from allopt.txt: simulation parameters :
    k2 has not been used
    js only used for internal scaling of magnetization and fields and 1 Tesla should do it anyway, therefore now hardcoded
  - more verbose output during initialization
  - updated examples
  - many code clean-ups
  - updated and improved documentation

- **Jan. 2004**

  Presented magpar at the 9th Joint MMM/Intermag Conference, January 5-9, 2004, Anaheim, CA, USA

  (cf. Publications)

- **version 0.2 - Nov. 13, 2003**

  - use ParMetis 3.1 instead of Metis 4.0
  - added magdist parameter (small random distortion of magnetization in equilibrium when using energy minimization)
  - created eminisolve.c to shrink solve.c
  - updated documentation: installation, added performance page
  - fixed several memory allocation problems and memory leaks
  - created local2global mappings in movedata.c (not while reading the mesh)
  - added some more output after mesh partitioning

- **version 0.1 - Sept. 12, 2003**

  first public release

  under the terms of the GNU General Public License

- **Aug. 2003**

  started documentation using Doxygen

- **July 27 - August 1, 2003**

  presentation of magpar at the "International Conference on Magnetism 2003" in Rome

  (cf. Publications)

- **March 2003**

  completed my dissertation "Scalable Parallel Micromagnetic Solvers for Magnetic Nanostructures"

  (cf. Publications)

- **Jan. 2003**

  major restructuring of the source code

- **Nov. 2002**

  implemented regular mesh refinement

- **Oct. 2002**
  implemented output of PNG graphics files (magnetization snapshots)
- **Sept. 2002**
  implemented static energy minimization
- **summer 2002**
  implemented exchange, anisotropy, external, demagnetizing field
  implemented preconditioned time integration of the dynamic Landau-Lifshitz-Gilbert equation
- **early 2002**
  evaluated parallel finite element, linear algebra software packages
  package selection and first implementation of FE solution of Poisson's equation

Chapter 5

Structure

Sections:

5.1 Diagrams

magpar is based on PETSc, which provides the required parallel data structures, linear algebra operations and solvers. PETSc in turn is based on MPI for message passing and the BLAS and LAPACK libraries (non-parallel) for linear algebra.

TAO and PVODE are used for energy minimization and dynamic time integration. Mesh partitioning, data compression, and graphics output are done by some more (non-parallel) libraries, Metis, zlib, and libpng, respectively.

The following figure gives a coarse flow chart of magpar (not quite up to date).

More detailed information about the libraries can be found on the page about Required Software.

5.2 Main program

main.c

main.c::main

- PetscInitialize
- TAOInitialize
- serinit.c::SerInit
- parinit.c::ParInit
- main.c::Solve
- PetscFinalize

5.3 Initialization

5.3.1 Serial part

serinit.c::SerInit

```
  ierr = InitInfo();CHKERRQ(ierr);
  ierr = ReadMesh(gdata);CHKERRQ(ierr);
  ierr = ReadKrn(gdata);CHKERRQ(ierr);
  ierr = ModifyPropSer(gdata);CHKERRQ(ierr);
  ierr = ModifyPropSerGrains(gdata);CHKERRQ(ierr);
  ierr = MagInit(gdata);CHKERRQ(ierr);
  ierr = Hext_in1_Init(gdata);CHKERRQ(ierr);
  ierr = Hext_in2_Init(gdata);CHKERRQ(ierr);
  ierr = FilterElements(&gdata->n_ele,&gdata->elevert,&gdata->eleprop,gdata->prop
      dat);CHKERRQ(ierr);
  ierr = MeshMirror(&gdata->n_vert,&gdata->n_ele,&gdata->vertxyz,&gdata->elevert,
      &gdata->eleprop);CHKERRQ(ierr);
  ierr = FilterNodes(&gdata->n_vert,&gdata->n_ele,&gdata->vertxyz,gdata->elevert,
      &gdata->M,&gdata->VH1,&gdata->VH2);CHKERRQ(ierr);
  ierr = FilterElements(&gdata->n_ele,&gdata->elevert,&gdata->eleprop,gdata->prop
      dat);CHKERRQ(ierr);
  ierr = RegularRefinement(gdata);CHKERRQ(ierr);
  ierr = DecoupleGrains(gdata);CHKERRQ(ierr);
  ierr = Reorder(gdata);CHKERRQ(ierr);
  ierr = FacNB(gdata);CHKERRQ(ierr);
  ierr = VertProp(gdata);CHKERRQ(ierr);
  ierr = DataPartitionSurfSer(gdata);CHKERRQ(ierr);
  ierr = DistortMesh(gdata);CHKERRQ(ierr);
  ierr = DataMoveData(gdata);CHKERRQ(ierr);
```

5.3.2 Parallel part

parinit.c::ParInit

```
  ierr = PetscOptionsGetInt(PETSC_NULL,"-mode",(PetscInt*)&gdata->mode,&flg);CHKE
      RRQ(ierr);
  ierr = ModifyPropPar(gdata);CHKERRQ(ierr);
  ierr = EleVertVol(gdata);CHKERRQ(ierr);
  ierr = PetscOptionsGetReal(PETSC_NULL,"-ts_init_time",&tstart,&flg);CHKERRQ(ier
      r);
  ierr = Htot(gdata);CHKERRQ(ierr);
  ierr = Htot_Energy(gdata);CHKERRQ(ierr);
      ierr = myTSCreatePVode(gdata);CHKERRQ(ierr);
      ierr = WriteFEMAVS(gdata);CHKERRQ(ierr);
      ierr = CheckIterationLLG(gdata);CHKERRQ(ierr);
      ierr = myTSCreateEmini(gdata);CHKERRQ(ierr);
      ierr = WriteFEMAVS(gdata);CHKERRQ(ierr);
      ierr = CheckIterationEmini(gdata);CHKERRQ(ierr);
  ierr = DataDestroyInit(gdata);CHKERRQ(ierr);
```

5.4 Solution loop

main.c::Solve

```
# pseudocode
while (keepsolving)
switch: -mode
```

(see allopt.txt)

main.c::keepsolving: check exit conditions, write project.9999.* (checkpoint) files, check after each iteration

5.4.1 Field and energy calculation

htot.c::Htot

```
ierr = Hcubic(gdata,VHtotsum);CHKERRQ(ierr);
ierr = Hdemag(gdata,VHtotsum);CHKERRQ(ierr);
ierr = Helastic(gdata,VHtotsum);CHKERRQ(ierr);
ierr = Hexchani(gdata,VHtotsum);CHKERRQ(ierr);
ierr = Hexternal(gdata,VHtotsum);CHKERRQ(ierr);
[...]
```

accordingly for energy (note that in general E!=M.H !)

5.4.2 Energy minimization

Initialization: eminisolve.c::myTSCreateEmini

Solution: eminisolve.c::EminiSolve : call TaoSolve to find energ minimum

5.4.3 LLG time integration

Initialization: mytscreatepvode.c::myTSCreatePVode

Solution: mytssteppvode.c::myTSStepPVode : call CVode to take on time step

LLG right hand side: rhsfunction.c::RHSfunction calls calc_dMdt::calc_dMdt.c

Preconditioning: precond.c::Precond, Jacobian: myllgjacobian.c::myLLGJacobian

5.5 Finalizing

Write final entry in log files and final set of output files

```
ierr = WriteLog(gdata);CHKERRQ(ierr);
ierr = WriteSet(gdata);CHKERRQ(ierr);
PetscFinalize();
```

At the end of the program we do not clean up carefully (i.e. we do not destroy all data structures or free dynamically allocated memory) because the OS will do it for us anyway.

5.6 GridData: Global data structure

GridData

(see HTML documentation)

Chapter 6

Performance

The performance (the speedup in particular) of magpar version 0.1 has been measured on a Compaq SC45 cluster consisting of 11 nodes Alpha Server ES45 with 4 Alpha processors (EV68 @ 1 GHz, 8 MB Cache/CPU) and 16 GB of shared memory each. The nodes are interconnected with a Quadrics switch, which provides a maximum MPI bandwidth of 600 MB/s. Since this machine has been shared with several other users, up to 24 processors have been available for speedup measurements.

The speedup has been measured as $S_P = t_1 / t_P$, where t_1 is the execution time of the program for a given problem on a single processor and t_P is the execution time for the same problem on P processors.

The energy minimization method, which uses the LMVM method of the TAO package, has been applied to calculate the nucleation field of FePt nanoparticles. The timing results are summarized in the following figure:

processors	CPU time (h)	speedup
initialization		
1	0.202	1.00
4	0.080	2.52
8	0.046	4.38
16	0.032	6.26
20	0.027	7.33
24	0.025	7.86
solution		
1	5.047	1.00
4	1.500	3.36
8	0.568	8.87
16	0.307	16.41
20	0.233	21.57
24	0.210	23.97
total		
1	5.249	1.00
4	1.581	3.32
8	0.615	8.53
16	0.339	15.44
20	0.261	20.06
24	0.236	22.20

On 8 and 16 processors we find a "superlinear" behavior of the solution part of the application. This is a well known phenomenon in parallel computing and can be attributed to caching effects. As the same total amount of data is distributed over more processors, the relative amount decreases and may reach a size, where it fits into the fast cache memory of modern computer architectures. As a result, the data need not be fetched from the main memory (which is a lot slower than the cache memory) and the calculations are completed a lot faster.

The parallel time integration using PVODE is not as efficiently parallelized as the TAO package, which is shown in the following figure:

processors	CPU time (h)	speedup
initialization		
1	0.255	1.00
2	0.196	1.30
3	0.141	1.81
6	0.080	3.19
8	0.062	4.07
16	0.037	6.73
20	0.032	7.96
solution		
1	6.309	1.00
2	3.379	1.86
3	2.416	2.61
6	1.120	5.63
8	0.913	6.91
16	0.451	13.98
20	0.393	16.03
total		
1	6.565	1.00
2	3.576	1.83
3	2.557	2.56
6	1.200	5.46
8	0.975	6.72
16	0.489	13.42
20	0.425	15.42

For comparison, the next figure shows the speedup obtained on a Beowulf type cluster of 900 MHz AMD PCs running Linux (for a different problem). These machines are linked with a standard switched 100 MBit Ethernet network.

processors	CPU time (h)	speedup
initialization		
1	0.075	1.00
3	0.083	0.91
5	0.057	1.32
solution		
1	24.334	1.00
3	8.5059	2.86
5	5.8314	4.17
total		
1	24.41	1.00
3	8.589	2.84
5	5.889	4.15

Chapter 7

Supported Machines

magpar should run on all platforms, which are supported by PETSc:

- Linux on Intel IA32 and compatibles (i386, Pentium, AMD Athlon, etc.)
- Linux on AMD64 (AMD Opteron, AMD Athlon64) and x64 (Intel EMT64) compatibles
- Linux on Intel IA64 (Itanium, Itanium2)
- Linux on Alpha
- MS Windows
- Mac OS X
- DEC/Compaq/hp Alpha running OSF/Tru64
- IBM RS6000 (including SP)
- SGI Workstations (IRIX), SGI Origin (IRIX64)
- Sun Sparcstations running Solaris
- Cray T3E
- FreeBSD on Intel
- and more (cf. http://www.mcs.anl.gov/petsc/petsc-as/)

All other required packages (cf. Required Software) are also available for these platforms.

See FAQ for successful installations of magpar!

Chapter 8

Required Software

8.1 BLAS/LAPACK

BLAS and LAPACK are required for PETSc. It is highly recommended to use any vendor specific (and therefore highly optimized) BLAS and LAPACK libraries (cf. FAQ : Optimized BLAS libraries). The generic implementation is available from netlib:

http://www.netlib.org/blas/

http://www.netlib.org/lapack/

Optimized BLAS libraries can be found on the website of the ATLAS project:

http://sourceforge.net/projects/math-atlas/

8.2 MPI

If possible it is again recommended to use any vendor specific MPI libraries, which are highly optimized for your specific hardware platform. If you do not have any MPI implementation readily available, you can use MPICH, OpenMPI, or LAM/MPI.

http://www.mcs.anl.gov/research/projects/mpich2/

http://www.open-mpi.org/

http://www.lam-mpi.org/

However, magpar/PETSc can also be compiled without MPI support: Single processor version without MPI

8.3 SUNDIALS

The parallel time integrator PVODE from the SUNDIALS package is used for the dynamic time integration of the Landau-Lifshitz-Gilbert equation, which describes the time evolution of the magnetization distribution.

http://www.llnl.gov/CASC/sundials/

http://acts.nersc.gov/sundials/

mapgar can also be compiled without SUNDIALS and just with TAO for energy minimization.

8.4 PETSc

PETSc is the core package for magpar. It provides the parallel data types, matrix-vector operations, data input and output, many utility functions, etc.

http://www.mcs.anl.gov/petsc/petsc-as/

8.5 Optional Packages

8.6 ParMETIS

The Metis library can be used for mesh partitioning - to split the problem (i.e. the finite element mesh) into a number of submeshes, which are then dealt with by different processors. Metis uses a very efficient multilevel *k*-way partitioning algorithm, which gives high quality partitionings.

http://glaros.dtc.umn.edu/gkhome/views/metis/

http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview

http://www-users.cs.umn.edu/~karypis/metis/

http://www-users.cs.umn.edu/~karypis/metis/parmetis/index.html

Without ParMETIS magpar can still perform bandwidth optimization or just work with the natural ordering (see options "-metispartition" and "-optimizebw" in allopt.txt).

8.6.1 TAO

"The Advanced Optimization" library is used for the static energy minimzation method.

http://www.mcs.anl.gov/research/projects/tao/

8.6.2 zlib

This compression library is used to reduce the size of the output files. Especially, the inp-files for AVS, which store snapshots of the magnetization distribution, are compressed very effectively.

http://www.zlib.net

8.6.3 libpng

The libpng library allows magpar to save snapshots of the magnetization distribution as PNG graphics files.

http://www.libpng.org/pub/png/libpng.html

8.6.4 BlockSolve95

The BlockSolve95 package contains routines for solving large sparse symmetric systems of linear equations on massively parallel distributed memory systems and networks of workstations.

http://info.mcs.anl.gov/pub/BlockSolve95/

8.6.5 hypre

High performance preconditioners for solving large, sparse linear systems of equations on massively parallel computers

https://computation.llnl.gov/casc/linear_solvers/sls_hypre.html

8.6.6 SuperLU

SuperLU is a general purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations on high performance machines.

http://crd.lbl.gov/~xiaoye/SuperLU/

8.7 Copyright

Each external software package has its own licensing conditions and copyright. However, they are all freely available as open source and most of them published under GNU Public or other OSI approved licenses. Please refer to the list of Licenses.

Chapter 9

Licenses

License terms for magpar and various libraries

9.1 magpar

GNU General Public License

9.2 Atlas

BSD-style license

http://math-atlas.sourceforge.net/faq.html#license

9.3 LAPACK

freely-available even for commercial use

http://www.netlib.org/lapack/faq.html#1.2

9.4 MPI

MPICH: free

http://www.mcs.anl.gov/research/projects/mpich2/downloads/license.txt

Open MPI: New BSD license

http://www.open-mpi.org/community/license.php

LAM/MPI: free (BSD style license)

http://www.lam-mpi.org/community/license.php

9.5 ParMetis

free with limitations

http://www-users.cs.umn.edu/~karypis/metis/metis/faq.html#distribute

http://www-users.cs.umn.edu/~karypis/.discus/messages/16/122.html?1126805053

```
Copyright Notice
----------------

The ParMETIS/METIS package is copyrighted by the Regents of the
University of Minnesota. It can be freely used for educational and
research purposes by non-profit institutions and US government agencies
only. Other organizations are allowed to use ParMETIS/METIS only for
evaluation purposes, and any further uses will require prior approval.
The software may not be sold or redistributed without prior approval.
One may make copies of the software for their use provided that the
copies, are not sold or distributed, are used under the same terms
and conditions.
```

```
As unestablished research software, this code is provided on an
``as is'' basis without warranty of any kind, either expressed or
implied. The downloading, or executing any part of this software
constitutes an implicit agreement to these terms. These terms and
conditions are subject to change at any time without prior notice.
```

9.6 SUNDIALS

free (BSD style license)

http://www.llnl.gov/CASC/sundials/download/license.html

```
Copyright (c) 2002, The Regents of the University of California.
Produced at the Lawrence Livermore National Laboratory.
Written by S.D. Cohen, A.C. Hindmarsh, R. Serban,
           D. Shumaker, and A.G. Taylor.
UCRL-CODE-155951    (CVODE)
UCRL-CODE-155950    (CVODES)
UCRL-CODE-155952    (IDA)
UCRL-CODE-155953    (KINSOL)
All rights reserved.

This file is part of SUNDIALS.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the disclaimer below.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the disclaimer (as noted below)
in the documentation and/or other materials provided with the
distribution.

3. Neither the name of the UC/LLNL nor the names of its contributors
may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
REGENTS OF THE UNIVERSITY OF CALIFORNIA, THE U.S. DEPARTMENT OF ENERGY
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional BSD Notice
---------------------
1. This notice is required to be provided under our contract with
the U.S. Department of Energy (DOE). This work was produced at the
University of California, Lawrence Livermore National Laboratory
under Contract No. W-7405-ENG-48 with the DOE.

2. Neither the United States Government nor the University of
California nor any of their employees, makes any warranty, express
or implied, or assumes any liability or responsibility for the
```

```
accuracy, completeness, or usefulness of any information, apparatus,
product, or process disclosed, or represents that its use would not
infringe privately-owned rights.

3. Also, reference herein to any specific commercial products,
process, or services by trade name, trademark, manufacturer or
otherwise does not necessarily constitute or imply its endorsement,
recommendation, or favoring by the United States Government or the
University of California. The views and opinions of authors expressed
herein do not necessarily state or reflect those of the United States
Government or the University of California, and shall not be used for
advertising or product endorsement purposes.
```

9.7 PETSc

free

http://www.mcs.anl.gov/petsc/petsc-as/documentation/copyright.html

9.8 TAO

```
COPYRIGHT NOTIFICATION

TAO: Toolkit for Advanced Optimization

In addition to the following copyright, we also request that you give this
citation in any publications or software that utilize TAO.

Copyright 2009 , UChicago Argonne, LLC Operator of Argonne National Laboratory

All rights reserved.

Toolkit for Advanced Optimization (TAO), Version 1.10

OPEN SOURCE LICENSE

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

   1. Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer. Software changes,
      modifications, or derivative works, should be noted with comments and
      the author and organization's name.

   2. Redistributions in binary form must reproduce the above copyright notice,
      this list of conditions and the following disclaimer in the documentation
      and/or other materials provided with the distribution.

   3. Neither the names of UChicago Argonne, LLC or the Department of Energy
      nor the names of its contributors may be used to endorse or promote
      products derived from this software without specific prior written
      permission.

   4. The software and the end-user documentation included with the
      redistribution, if any, must include the following acknowledgment: "This
      product includes software produced by UChicago Argonne, LLC under
      Contract No. DE-AC02-06CH11357 with the Department of Energy."

*****************************************************************************
DISCLAIMER

THE SOFTWARE IS SUPPLIED "AS IS" WITHOUT WARRANTY OF ANY KIND.
```

```
Neither the United States GOVERNMENT, nor the United States Department of
Energy, NOR uchicago argonne, LLC, nor any of their employees, makes any
warranty, express or implied, or assumes any legal liability or responsibility
for the accuracy, completeness, or usefulness of any information, data,
apparatus, product, or process disclosed, or represents that its use would not
infringe privately owned rights.
*******************************************************************************
```

9.9 zlib

free (OSI approved license)

http://www.zlib.net/zlib_license.html

9.10 libpng

free (OSI approved license)

http://www.libpng.org/pub/png/src/libpng-LICENSE.txt

Chapter 10

GNU General Public License

magpar is distributed under the terms of the GNU General Public License:

```
    GNU GENERAL PUBLIC LICENSE
       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
                       59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

    Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
patents.  We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary.  To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

  The precise terms and conditions for copying, distribution and
modification follow.

    GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License.  The "Program", below,
refers to any such program or work, and a "work based on the Program"
```

means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language. (Hereinafter, translation is included without limitation in
the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope. The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
    these conditions, and telling the user how to view a copy of this
    License. (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or

otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE

PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

see also: GNU General Public License

Chapter 11

Installation

Sections:

11.1 Configuration

Create a directory, which will contain all libraries, source code, tools, and documentation for magpar, and set the environment variable MAGPAR_HOME:

```
# change into any directory, where you want to install magpar
# for example in your $HOME/work directory
cd $HOME; mkdir work; cd work
# download the magpar source archive by hand or using wget:
lib=magpar
wget http://www.magpar.net/$lib/download/$lib.tar.gz
# unpack the archive
tar xzvf $lib.tar.gz
cd $lib
MAGPAR_HOME=$PWD; export MAGPAR_HOME   # sh/bash syntax (use "setenv" for csh)
PD=$MAGPAR_HOME/libs; export PD
```

If you are upgrading from a previous version of magpar, you can usually reuse the libraries, which you have already compiled and installed. However, please note the ChangeLog and upgrade libraries as required or recommended.

The current version of magpar has been developed and tested with the configuration and library versions defined in Makefile.in.defaults .

Links to the websites of the libraries can be found in the list of Required Software.

Create Makefile.in.$HOSTNAME using Makefile.in.defaults (or one of the other Makefile.in.host_∗):

```
cp Makefile.in.defaults Makefile.in.$HOSTNAME
```

and edit it:

- set paths
- (de)activate libraries, set/override library versions (see Makefile.in.host_oldlibs)
- set compiler flags for optimization

It should not be necessary to modify Makefile or Makefile.in at all any more!

Please refer to the FAQ for tips and suggestions for the installation of magpar on specific systems and software environments.

11.2 Automated Installation

The (manual) installation procedures described below are now conveniently combined in Makefile.libs .

Just simply do

```
cd $MAGPAR_HOME/src
make -f Makefile.libs
```

All libraries will be downloaded automatically using "wget", configured, compiled, and installed in $PD in the following order:

```
atlas lapack mpi parmetis sundials petsc tao zlib libpng
```

It is also possible to install the libraries one at a time like this (e.g. PETSc)

```
cd $MAGPAR_HOME/src
make -f Makefile.libs petsc
```

using the names listed above. This makes it easier to use precompiled packages (e.g. Precompiled packages on Ubuntu/Debian) and then just install the remaining ones with the convenience of using Makefile.libs .

If this does not work, then please follow the manual installation instructions below.

Once all libraries are compiled and installed, compile magpar as described below.

11.3 Manual Installation

Check for Required Software which is already preinstalled on your machine. For example, there are Precompiled packages on Ubuntu/Debian available. Download, unpack, compile and install all other required libraries in the order given below in the directory $PD. Some libraries are optional,and only required if you want to try different linear solvers (e.g. BlockSolve95, hypre, SuperLU). It is highly recommended to use any machine specific (vendor provided and highly optimized) libraries. On most high performance machines there are optimized BLAS, LAPACK, and MPI libraries available (cf. FAQ : Optimized BLAS libraries). In this case you just have to set the paths to your libraries properly, when you configure and compile various packages.

If you have trouble installing any of the required libraries, please check their respective installation guides/-documentation/FAQs/website first. The URLs of their websites can be found on the Required Software page.

11.4 ATLAS

For your convenience get one of the binary packages for your hardware platform from the stable branch of ATLAS (unless you do already have Optimized BLAS libraries).

```
cd $PD
# set lib with the name of your ATLAS library
lib=atlas3.6.0_Linux_PIIISSE1.tar.gz
tar xzvf $lib
# create a symbolic link to the directory with the ATLAS binaries:
ln -s Linux_* atlas
lapacklib=$PD/atlas/lib/liblapack.a
# rename (incomplete) lapack library provided by ATLAS (cf. LAPACK below)
mv $lapacklib $lapacklib.atlas
```

11.5 LAPACK

Debian, RedHat and other distributors provide precompiled binaries of LAPACK. Try the Precompiled packages on Ubuntu/Debian or check the web for availablility.

http://www.debian.org/distrib/packages

http://rpmfind.net/

http://www.redhat.com/

You may also recompile it from source:

```
# set Fortran compiler
# GNU GCC >=4.0 Fortran 77/95: gfortran
FC=gfortran; TIMER=INT_ETIME
#
# GNU GCC < 4.0 Fortran 77:    g77
#FC=g77; TIMER=EXT_ETIME
#
# check that Fortran compiler works
$FC --version
#
cd $PD
lib=lapack.tgz
wget http://www.netlib.org/lapack/$lib
tar xzvf $lib
cd lapack-*
cp INSTALL/make.inc.LINUX make.inc
#
# add CPU specific options to OPTS, e.g. -march=pentium4 -msse2 (cf. man gcc)
# set correct Fortran compiler (check additional options in make.inc!):
make "FORTRAN=$FC" "LOADER=$FC" \
"TIMER=$TIMER" \
"BLASLIB=$PD/atlas/lib/libf77blas.a $PD/atlas/lib/libatlas.a" \
"OPTS=-funroll-all-loops -O3 $OOPTS" \
lapacklib
#
# run tests (optional)
make lapack_testing
```

Now we have to add the missing LAPACK functions to the ATLAS library:

(cf. $PD/atlas/README, Building a complete LAPACK library)

```
cd $PD/atlas/lib
```

```
cp $lapacklib.atlas $lapacklib
mkdir tmp; cd tmp
ar x $PD/lapack-*/lapack_LINUX.a
ar r ../liblapack.a *.o
cd ..; rm -rf tmp
```

11.6 MPI

MPICH, OpenMPI, LAM/MPI, or any other MPI library, which implements the MPI standard (version 1 or 2) may be used.

You need rsh (recommended) or ssh to be installed and configured properly! Don't forget to create a ".rhosts" file with the names of all machines (also your local machine!) in your home directory (cf. "man rhosts"). The configuration can be tested with "rsh $HOSTNAME uname -a". You may also use ssh for encrypted communication between processors.

The directory $PD/mpi/bin should be added to the $PATH variable. Update your login scripts, e.g. .bashrc, .login, .profile, to make this permanent by appending the following code snippet:

```
PATH=$PD/mpi/bin:$PATH
export PATH
```

or the programs installed in $PD/mpi/bin should be copied to $HOME/bin or any other directory within your $PATH, so that mpirun and other MPI tools can be called from the command line.

11.6.1 MPICH2

If ssh should be used instead of rsh for login on remote machines use "./configure -rsh=ssh" when compiling MPICH. In this case public key authentication should be configured for ssh to enable login without passwords (cf. "man ssh").

The configure-script of MPICH will try both, rsh and ssh, and print a warning if neither service is configured properly.

```
cd $PD
lib=mpich2.tar.gz
wget -N --retr-symlinks ftp://ftp.mcs.anl.gov/pub/mpi/$lib
# better download the latest version from
# http://www.mcs.anl.gov/research/projects/mpich2/
tar xzvf $lib
#
# change into mpich2 subdirectory (adjust to the downloaded version)
cd mpich2-*
#
# use "ssm" (sockets and shared memory) for use on clusters of SMPs
# (communication on the same machine goes through shared memory;
# communication between different machines goes over sockets)
# instead of default "sock"
./configure --prefix=$PD/mpich2 --with-device=ch3:ssm 2>&1 | tee configure.log
#
make -j 1 2>&1 | tee make.log
make install 2>&1 | tee install.log
#
# set symbolic link to MPICH installation directory
ln -s mpich2 $PD/mpi
#
# Please refer to $PD/mpi/README or
# $PD/mpi/doc on how to use MPICH2 and
# start a ring of MPI's process managers mpd!
```

Installation instructions for MPICH1 and LAM/MPI have been moved to the FAQ.

11.7 ParMetis

```
cd $PD
lib=ParMetis-3.1.1
wget -N http://glaros.dtc.umn.edu/gkhome/fetch/sw/parmetis/$lib.tar.gz
tar xzvf $lib
cd $lib
make "CC=$PD/mpi/bin/mpicc" "LD=$PD/mpi/bin/mpicc"
#
# run tests (optional)
cd Graphs
$PD/mpi/bin/mpirun -np 4 ptest rotor.graph
# more tests in ParMetis-3.1.1/INSTALL
```

11.8 SUNDIALS

(SUNDIALS version 2.3.0)

Download the library from the SUNDIALS website (registration required).

```
cd $PD
lib=sundials-2.3.0
tar xzvf $lib.tar.gz
cd $(PD)/$lib
# set compiler options (modify for your setup!)
# add CPU specific options, e.g. -march=pentium4 -msse2 (cf. man gcc)
CFLAGS="-O3"
export CFLAGS
./configure --prefix=$PD/$lib --with-mpi-root=$PD/mpi
make && make -i install
# (generates static libraries and installs libraries and include files
# in $PD/$lib/libs and $PD/$lib/include)
```

11.9 PETSc

(PETSc version 2.3.0 and later)

Starting with PETSc version 2.3.0 you have to use the automatic Python-based configure system, which requires Python 2.2 or later. Please refer to the FAQ Installing Python if you need to install Python by hand.

```
cd $PD
lib=petsc-2.3.3-p15
wget ftp://ftp.mcs.anl.gov/pub/petsc/release-snapshots/$lib.tar.gz
tar xzvf $lib.tar.gz
cd $lib
#
# set environment variables
# (here: bash style - use "setenv" in sh/csh)
#
PETSC_DIR=$PD/$lib
export PETSC_DIR
PETSC_ARCH=PETSc-config-magpar
export PETSC_ARCH
PRECISION=double
export PRECISION
#
```

```
# edit PETSc-config-magpar.py
# (select MPI, optional libraries, optimization options, etc.)
# use the templates in $PETSC_DIR/config/ for platforms other than Linux
# copy PETSc configuration script
# (needs to be a copy - must not be a symbolic link!)
#
cp $MAGPAR_HOME/src/PETSc-config-magpar.py $PETSC_DIR/config/
#
# Run
# ./config/configure.py --help
# to see all command line options for configure.py.
#
# for static binaries edit
# $PETSC_DIR/bmake/PETSc-config-magpar/petscconf (recommended):
# remove all occurences of "-lgcc_s" and add "-static" to the linker flags:
# CC_LINKER_FLAGS =   -Wall -O3 -static
#
./config/PETSc-config-magpar.py
make all
#
# run tests (optional)
make test
```

Also refer to the installation instructions on the PETSc homepage!

11.10 TAO

magpar requires

(PETSc version 2.3.3 and TAO 1.9) (highly recommended) or

(PETSc version 2.3.2 and TAO 1.8.2) or

(PETSc version 2.3.0 and TAO 1.8) or

(PETSc version 2.2.1 and TAO 1.7) or

(PETSc version 2.2.0 and TAO 1.6)

```
lib=tao-1.9
wget -N http://www.mcs.anl.gov/research/projects/tao/download/$lib
tar xzvf $lib
TAO_DIR=$PD/$lib; export TAO_DIR
cd $TAO_DIR
make
```

11.11 zlib

```
cd $PD
lib=zlib-1.2.3
wget -N http://downloads.sourceforge.net/libpng/$lib.tar.gz
tar xzvf $lib.tar.gz
ln -s $lib zlib
cd $lib
make CFLAGS="-O -fPIC" && make test
```

11.12 libpng

```
cd $PD
```

```
lib=libpng-1.2.33
wget -N http://downloads.sourceforge.net/libpng/$lib.tar.gz?download
tar xzvf $lib.tar.gz
ln -s $lib libpng
cd $lib
instdir=$(PD)/$lib
./configure --prefix=$instdir --enable-shared=no 2>&1 | tee configure.log
CFLAGS="-I$PD/zlib"; export CFLAGS
LDFLAGS="-L$PD/zlib"; export LDFLAGS
make 2>&1 | tee make.log
make install 2>&1 | tee makeinst.log
make check 2>&1 | tee makecheck.log
# alternatively use the old method with a static Makefile:
cp scripts/makefile.linux Makefile
make ZLIBLIB=../zlib ZLIBINC=../zlib && make test
```

11.13 magpar

Once all libraries are compiled and installed, compile magpar with

```
cd $MAGPAR_HOME/src
make
```

If everything compiled (hopefully) ok, you should get the executable magpar.exe.

Chapter 12

Makefile system

Sections:

12.1 Makefile

```
##############################################################################
# $Id: Makefile 2909 2009-12-08 20:01:05Z scholz $
##############################################################################
#
# This file is part of magpar.
#
# Copyright (C) 2002-2009 Werner Scholz
#
# www:   http://www.magpar.net/
# email: magpar(at)magpar.net
#
# magpar is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# magpar is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with magpar; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#
##############################################################################

magpar_exe  = magpar.exe
magpar_revh = magpar_revision.h
magpar_verh = magpar_version.h

all: $(magpar_exe)

# (de)activate libraries, library versions, set paths, compiler flags, etc.
# imported from Makefile.in and Makefile.in.$HOSTNAME
include Makefile.in

# read list of files and other settings
include Makefile.files

magpar_rev=$(shell awk '{print $$3}' $(magpar_revh)|sed  "s/\"//"g)
magpar_ver=$(shell awk '{print $$3}' $(magpar_verh)|sed  "s/\"//"g)

$(magpar_revh): FORCE
@r_old=$(magpar_rev); \
if [ -d .svn ] && which svnversion >& /dev/null && svn -u status >& /dev/null; then \
```

```
  r_new=`svnversion`; \
else \
  r_old2=`echo $$r_old|sed  "s/_mod//"g`; \
  r_new=""$$r_old2"_mod"; \
fi; \
if [ "$$r_new" == "exported" ]; then \
  r_old2=`echo $$r_old|sed  "s/_mod//"g`; \
  r_new=""$$r_old2"_mod"; \
fi; \
if [ "$$r_new" != "$$r_old" ]; then \
  echo '#define MAGPAR_REVISION "'$$r_new'"' > $@ ; \
  echo "Recreated $@ for revision $$r_new"; \
fi

# all *.c depend on grid*.h and the Makefile (due to #define) !
$(magparobj): %.o: %.c $(griddata) Makefile Makefile.in

$(magpar_exe): $(magparobj)
-$(CLINKER) $(magparobj) -o $(magpar_exe) $(TAO_LIB) $(PETSC_TS_LIB)
@echo ""; echo "Built magpar version $(magpar_ver) revision $(magpar_rev)"
# strip $(magpar_exe)

tags: FORCE
ctags `find . -name "*.c" -o -name "*.h"`

doxygen: $(magpar_revh) FORCE
-rm -f ../doc/html/*
ENABLED_SECTIONS="$(ENABLED_SECTIONS) changelog"; export ENABLED_SECTIONS; \
PROJECT_NUMBER="Version $(magpar_ver) Build $(magpar_rev)";   export PROJECT_NUMBER; \
GENERATE_HTML="YES";    export GENERATE_HTML; \
GENERATE_LATEX="NO";    export GENERATE_LATEX; \
EXTRACT_ALL="YES";      export EXTRACT_ALL; \
SOURCE_BROWSER="YES";   export SOURCE_BROWSER; \
VERBATIM_HEADERS="YES"; export VERBATIM_HEADERS; \
HAVE_DOT="YES";         export HAVE_DOT; \
CALL_GRAPH="YES";       export CALL_GRAPH; \
CALLER_GRAPH="YES";     export CALLER_GRAPH; \
MAGPAR_HOME=$(MAGPAR_HOME); export MAGPAR_HOME; \
ALIASES=$(DOXYGEN_ALIASES); export ALIASES; \
(set; echo $$MAGPAR_HOME; doxygen --version; doxygen Doxyfile; echo ""; echo "Documentation complete!") \
>& make_doxygen.log

doxygen_ng: $(magpar_revh) FORCE
-rm -f ../doc/html/*
ENABLED_SECTIONS="$(ENABLED_SECTIONS) changelog"; export ENABLED_SECTIONS; \
PROJECT_NUMBER="version $(magpar_ver) build $(magpar_rev)";   export PROJECT_NUMBER; \
GENERATE_HTML="YES";    export GENERATE_HTML; \
GENERATE_LATEX="NO";    export GENERATE_LATEX; \
EXTRACT_ALL="YES";      export EXTRACT_ALL; \
SOURCE_BROWSER="YES";   export SOURCE_BROWSER; \
VERBATIM_HEADERS="YES"; export VERBATIM_HEADERS; \
HAVE_DOT="NO";          export HAVE_DOT; \
CALL_GRAPH="NO";        export CALL_GRAPH; \
CALLER_GRAPH="NO";      export CALLER_GRAPH; \
MAGPAR_HOME=$(MAGPAR_HOME); export MAGPAR_HOME; \
ALIASES=$(DOXYGEN_ALIASES); export ALIASES; \
(doxygen --version; doxygen Doxyfile; echo ""; echo "Documentation complete!") \
>& make_doxygen.log

# cannot use $(MAKE) here: otherwise current make waits for submake to finish
doxygenb: FORCE
make doxygen >& make_doxygenb.log &

doxygen_latex: $(magpar_revh) FORCE
-rm -f ../doc/latex/* AAA
ln -s doc AAA
ENABLED_SECTIONS="$(ENABLED_SECTIONS)"; export ENABLED_SECTIONS; \
```

```
PROJECT_NUMBER="version $(magpar_ver) build $(magpar_rev)";   export PROJECT_NUMBER; \
GENERATE_HTML="NO";    export GENERATE_HTML; \
GENERATE_LATEX="YES";  export GENERATE_LATEX; \
EXTRACT_ALL="NO";      export EXTRACT_ALL; \
SOURCE_BROWSER="NO";   export SOURCE_BROWSER; \
VERBATIM_HEADERS="NO"; export VERBATIM_HEADERS; \
HAVE_DOT="NO";         export HAVE_DOT; \
CALL_GRAPH="NO";       export CALL_GRAPH; \
CALLER_GRAPH="NO";     export CALLER_GRAPH; \
ALIASES=$(DOXYGEN_ALIASES); export ALIASES; \
doxygen Doxyfile >& make_doxygenlatex.log
rm AAA
(cd ../doc/latex; $(MAKE)) >> make_doxygenlatex.log; true
mv ../doc/latex/refman.pdf ../doc/magpar.pdf; rm -r -f ../doc/latex/

linkcheck: FORCE
-cd ../doc/html/; linkchecker index.html >& ../../src/make_linkchecker.log
(echo ""; echo "===== Errors and Warnings ====="; \
  grep Warning make_linkchecker.log; grep Error make_linkchecker.log \
) > make_linkchecker2.log

alloptcheck: FORCE
grep "PetscOption" `find . -name "*.c"`|awk -F "\"" '{print $$2}'|sort|uniq > make_options_code.log
(for i in `find . -name "allopt_*" | grep -v "\.svn"`; do grep "^-" $$i; done)|awk '{print $$1}' >> make_c
sort make_options_code.log|uniq > x_t; mv x_t make_options_code.log
grep "PetscOption" `find . -name "*.c"`|grep "-"|sed 's/:[^\"]*\"/\t\t/'|sed 's/\".*//'|sort|uniq > make_c
(for i in `find . -name "allopt*" | grep -v "\.svn"`; do grep "^-" $$i; done)|awk '{print $$1}'|sort|uniq

bakclean: FORCE
-rm -f -v `find . -name "~*"`
-rm -f -v `find . -name ".#*"`

oclean: FORCE
-rm -f `find . -name "*.o"` $(magpar_exe)

distclean: bakclean oclean
-rm -f *.exe *.log

varinfo: $(magpar_revh) FORCE
@echo "HOSTNAME         = $(HOSTNAME)"
@echo "MAGPAR_HOME      = $(MAGPAR_HOME)"
@echo "UNIPROC          = $(UNIPROC)"
@echo "uni              = $(uni)"
@echo "EXCH             = $(EXCH)"
@echo "PRECISION        = $(PRECISION)"
@echo "CC               = $(CC)"
@echo "CXX              = $(CXX)"
@echo "F77              = $(F77)"
@echo "FC               = $(FC)"
@echo "F90              = $(F90)"
@echo "CFLAGS           = $(CFLAGS)"
@echo "OPTFLAGS         = $(OPTFLAGS)"
@echo "C_CLINKER_SLFLAG = $(C_CLINKER_SLFLAG)"
@echo "EXTERNAL_LIB     = $(EXTERNAL_LIB)"
@echo "ENABLED_SECTIONS = $(ENABLED_SECTIONS)"
@echo "dyn              = $(dyn)"
@echo "SVNSERVER        = $(SVNSERVER)"
@echo "magpar_ver       = $(magpar_ver)"
@echo "magpar_rev       = $(magpar_rev)"
@echo
@echo "Libraries:"
@echo "PD               = $(PD)"
@echo "PETSC_VERSION    = $(PETSC_VERSION)"
@echo "PETSC_ARCH       = $(PETSC_ARCH)"
@echo "PETSC_DIR        = $(PETSC_DIR)"
@echo "TAO_DIR          = $(TAO_DIR)"
@echo "METIS_DIR        = $(METIS_DIR)"
```

```
@echo "SUNDIALS_VERSION = $(SUNDIALS_VERSION)"
@echo "SUNDIALS_DIR     = $(SUNDIALS_DIR)"
@echo "ATLAS_DIR        = $(ATLAS_DIR)"
@echo "MPI_DIR          = $(MPI_DIR)"
@echo "PNG_DIR          = $(PNG_DIR)"
@echo "ZLIB_DIR         = $(ZLIB_DIR)"

testall: FORCE
@echo "--------------------------------------------------------------------"
$(MAKE) distclean
@echo "--------------------------------------------------------------------"
for i in 3.0.0 2.3.3; do \
  for j in 230; do \
    $(MAKE) PETSC_VERSION=$$i SUNDIALS_VERSION=$$j test; \
  done; \
done

test: distclean FORCE
@echo "--------------------------------------------------------------------"
for i in test1 test2 test3 test4 test5 test6; do \
  $(MAKE) $$i ; \
done; \
$(MAKE)

test1: $(magpar_revh) FORCE
@echo "--------------------------------------------------------------------"
-$(MAKE) oclean
-$(MAKE) SUNDIALS_VERSION=FALSE PNG_DIR=FALSE varinfo $(magpar_exe) >& make_$(magpar_rev)_$@.log
-mv $(magpar_exe) magpar_$(magpar_rev)_$@.exe
test2: $(magpar_revh) FORCE
@echo "--------------------------------------------------------------------"
-$(MAKE) oclean
-$(MAKE) TAO_DIR=FALSE ZLIB_DIR=FALSE PNG_DIR=FALSE varinfo $(magpar_exe) >& make_$(magpar_rev)_$@.log
-mv $(magpar_exe) magpar_$(magpar_rev)_$@.exe
test3: $(magpar_revh) FORCE
@echo "--------------------------------------------------------------------"
-$(MAKE) oclean
-$(MAKE) ADDONS=FALSE varinfo $(magpar_exe) >& make_$(magpar_rev)_$@.log
-mv $(magpar_exe) magpar_$(magpar_rev)_$@.exe
test4: $(magpar_revh) FORCE
@echo "--------------------------------------------------------------------"
-$(MAKE) oclean
-$(MAKE) ADDONS=FALSE TAO_DIR=FALSE ZLIB_DIR=FALSE PNG_DIR=FALSE varinfo $(magpar_exe) >& make_$(magpar_re
-mv $(magpar_exe) magpar_$(magpar_rev)_$@.exe
test5: $(magpar_revh) FORCE
@echo "--------------------------------------------------------------------"
-$(MAKE) oclean
-$(MAKE) ADDONS=TRUE MFP=TRUE varinfo $(magpar_exe) >& make_$(magpar_rev)_$@.log
-mv $(magpar_exe) magpar_$(magpar_rev)_$@.exe
test6: $(magpar_revh) FORCE
@echo "--------------------------------------------------------------------"
-$(MAKE) oclean
-$(MAKE) varinfo $(magpar_exe) >& make_$(magpar_rev)_$@.log
-mv $(magpar_exe) magpar_$(magpar_rev)_$@.exe

FORCE:
```

12.2 Makefile.in

```
#############################################################################
# $Id: Makefile.in 2890 2009-12-03 20:48:54Z scholz $
#############################################################################

# include default settings ########################################################
include Makefile.in.defaults
```

```
# include Makefile.in.$HOSTNAME ##############################################

# set hostname if undefined
ifeq ($(HOSTNAME),)
  HOSTNAME=$(shell hostname)
endif

# possibly override various options (above) with those
# from machine specific Makefiles.in.$HOSTNAME

# check if file exists
exist := $(wildcard Makefile.in.$(HOSTNAME))
ifneq ($(strip $(exist)),)
  # include if it does
  include Makefile.in.$(HOSTNAME)
else
  exist := $(wildcard addons/Makefile.in.$(HOSTNAME))
  ifneq ($(strip $(exist)),)
    # include if it does
    include addons/Makefile.in.$(HOSTNAME)
  endif
endif

# PETSC ######################################################################

ifeq ($(PETSC_VERSION),3.0.0)
  PETSC_ARCH  = PETSc-config-magpar
  PETSC_DIR   = $(PD)/petsc-3.0.0-p10
  TAO_DIR     = $(PD)/tao-1.10-p1

  # check if we compile for single processor only
  uni=$(shell grep "define  *PETSC_HAVE_MPIUNI  *1" $(PETSC_DIR)/$(PETSC_ARCH)/include/petscconf.h)
endif

ifeq ($(PETSC_VERSION),2.3.3)
  PETSC_ARCH  = PETSc-config-magpar
  PETSC_DIR   = $(PD)/petsc-2.3.3-p15
  TAO_DIR     = $(PD)/tao-1.9

  # check if we compile for single processor only
  uni=$(shell grep "define  *PETSC_HAVE_MPIUNI  *1" $(PETSC_DIR)/bmake/$(PETSC_ARCH)/petscconf.h)
endif

ifneq ($(TAO_DIR),FALSE)
  file := $(TAO_DIR)/bmake/tao_common
  exist := $(wildcard $(file))
  ifneq ($(strip $(exist)),)
    include $(file)
    CFLAGS    += -DTAO
  endif
else
  ifeq ($(PETSC_VERSION),3.0.0)
    file := $(PETSC_DIR)/conf/base
  else
    file := $(PETSC_DIR)/bmake/common/base
  endif
  exist := $(wildcard $(file))
  ifneq ($(strip $(exist)),)
    include $(file)
  endif
endif

# SUNDIALS ###################################################################

SUNDIALS_LIBS  = -L$(SUNDIALS_DIR)/lib -lsundials_cvode
```

```
ifeq ($(uni),)
  UNIPROC = FALSE
else
  UNIPROC = TRUE
  CFLAGS      += -DUNIPROC
endif

ifeq ($(UNIPROC),TRUE)
  SUNDIALS_LIBS  += -lsundials_nvecserial
else
  SUNDIALS_LIBS  += -lsundials_nvecparallel
endif

ifeq ($(SUNDIALS_VERSION),240)
  # SUNDIALS version 2.4.0
  SUNDIALS_DIR   = $(PD)/sundials-2.4.0
  CFLAGS         += -DSUNDIALS_VERSION=$(SUNDIALS_VERSION) -I$(SUNDIALS_DIR)/include
endif

ifeq ($(SUNDIALS_VERSION),230)
  # SUNDIALS version 2.3.0
  SUNDIALS_DIR   = $(PD)/sundials-2.3.0
  CFLAGS         += -DSUNDIALS_VERSION=$(SUNDIALS_VERSION) -I$(SUNDIALS_DIR)/include
endif

ifneq ($(SUNDIALS_VERSION),FALSE)
  EXTERNAL_LIB += $(SUNDIALS_LIBS)
endif

# other libraries ############################################################

ifneq ($(METIS_DIR),FALSE)
  EXTERNAL_LIB += -L$(METIS_DIR) -lmetis
  CFLAGS       += -DMETIS -I$(METIS_DIR)/METISLib
endif

ifneq ($(PNG_DIR),FALSE)
  EXTERNAL_LIB += -L$(PNG_DIR) -L$(PNG_DIR)/lib -lpng
  CFLAGS       += -DPNG -I$(PNG_DIR)
endif

ifneq ($(ZLIB_DIR),FALSE)
  EXTERNAL_LIB += -L$(ZLIB_DIR) -lz
  CFLAGS       += -DZLIB -I$(ZLIB_DIR)
endif

ifneq ($(PYTHON),FALSE)
  EXTERNAL_LIB += -L$(PYTHON)/config/ -l$(PYTHON)
  CFLAGS       += -DPYTHON -I/usr/include/$(PYTHON)
endif
```

12.3 Makefile.in.defaults

```
###############################################################################
# $Id: Makefile.in.defaults 2891 2009-12-03 20:50:13Z scholz $
###############################################################################

# default settings ############################################################

# magpar home directory
MAGPAR_HOME = $(HOME)/work/magpar

# directory where the manually compiled libraries are installed
PD = $(MAGPAR_HOME)/libs
```

```
# directory of magpar source code
MAGPAR_SRC = $(MAGPAR_HOME)/src

# set floating point precision (single/double)
PRECISION=double

# select your PETSc version (suitable TAO required!)
# check Makefile.in for valid values
PETSC_VERSION = 3.0.0

# set to FALSE to disable
#TAO_DIR =

# set dummy
PETSC_XTRALIBS=

# select your SUNDIALS version
#
# check Makefile.in for valid values
# and update the path to your SUNDIALS installation directory;
# set to FALSE to disable
SUNDIALS_VERSION = 230

# select BLAS/ATLAS version
# below in the hardware specific section!

# select LAPACK version
liblapack=lapack-3.2.1

# additional libraries ###################################################
#
# activate by setting correct path
# deactivate by setting to FALSE
# NB: libpng requires zlib!

# ParMetis is required!
METIS_DIR   = $(PD)/ParMetis-3.1.1
PNG_DIR     = $(PD)/libpng-1.2.40
ZLIB_DIR    = $(PD)/zlib-1.2.3

#DRL: added these, will be used by PETSc,
#rather than assuming they are under $(PD)
ATLAS_DIR   = $(PD)/atlas/lib
MPI_DIR     = $(PD)/mpi

##### use SUPERLU for A*u1=divM (experimental)
#CFLAGS      += -DSUPERLU
# XOR
##### BLOCKSOLVE for A*u1=divM (experimental)
#CFLAGS      += -DBLOCKSOLVE

# magpar specific flags ##################################################

##### calculate exchange energy separately?
#
# set define to have exchange field and energy calculated separately from
# anisotropy; then the exchange energy is also slightly more accurate and
# the exchange energy is stored in a separate column in the log file;
# calculation of the anisotropy energy for cubic anisotropy is done
# properly including K1 and K2 terms for both uniaxial and cubic anisotropy
# (important especially for large K2!)
#
# undefine (comment out) to have anisotropy and exchange field and energy
# combined (slightly faster: saves two matrix-vector multiplications)
# anisotropy energy might be very inaccurate (for large K2)
# K2 term of uniaxial anisotropy is ignored!
```

```
#
# disable
#EXCH = FALSE
# enable
EXCH = TRUE

ADDONS   = FALSE
SPINTORQ = FALSE
EBM      = FALSE
PYTHON   = FALSE

SVNSERVER = 300.300.300.300 # dummy

# compiler flags ############################################################
CFLAGS      += -Wunused

# Adjust the following variables according to your hardware #################

# set filename of requested ATLAS library (precompiled binary)
# for your hardware check ATLAS download page for available versions:
# http://sourceforge.net/project/showfiles.php?group_id=23725
#
libatlas = atlas3.6.0_Linux_PIIISSE1.tar.gz

# general optimization options
#
# used during PETSc compilation and then immutable
# use CFLAGS to change compile options later
#
OPTFLAGS += -O2
#OPTFLAGS += -O3

# CPU specific compiler options for GCC >= 3.0
#
#OPTFLAGS += -march=pentium4  -msse -msse2  -mfpmath=sse
#OPTFLAGS += -march=pentium-m -msse -msse2  -mfpmath=sse
#OPTFLAGS += -march=athlon    -m3dnow       -mfpmath=sse
#OPTFLAGS += -march=athlon-xp -m3dnow       -mfpmath=sse
#OPTFLAGS += -march=opteron   -msse -msse2  -mfpmath=sse -m64
#OPTFLAGS += -march=nocona    -msse2        -mfpmath=sse -m64


# for Mac OS X on an iBook G4 (cf. FAQ in documentation)
#
#OPTFLAGS += -mcpu=7450 -O2 -arch=G4 -faltivec

# more (possibly unsafe!) optimizations (for newer gcc versions)
# (inspired by Acovea: http://www.coyotegulch.com/products/acovea/ )
#
#OPTFLAGS += -ffast-math
# --fast-math implies:
# -fno-math-errno -funsafe-math-optimizations -ffinite-math-only -fno-trapping-math
#OPTFLAGS += -funroll-loops -ftree-loop-linear
#OPTFLAGS += -ftree-vectorize -ftracer -fvariable-expansion-in-unroller
#OPTFLAGS += -funsafe-loop-optimizations -Wunsafe-loop-optimizations

# compiler options for development/debugging
#
#OPTFLAGS += -pedantic -msg_enable noansi -msg_enable obsolescent
#OPTFLAGS += -msg_enable performance -msg_enable portable
#OPTFLAGS += -msg_enable overflow -msg_enable questcode
#OPTFLAGS += -msg_enable unused #-msg_enable returnchecks
#OPTFLAGS += -fbounds-check # currently only supported by gcj and gfortran

# set Fortran compiler
# default: gfortran
# use g77 on old Linux distributions
```

```
# g77 usually requires libg2c (uncomment PETSC_XTRALIBS below)
#
#FC = g77
#PETSC_XTRALIBS=/usr/lib/gcc-lib/i386-redhat-linux/2.96/libg2c.a

# link static magpar binary
#STATIC   = TRUE
```

12.4 Makefile.in.host_debian

```
# example Makefile.in.$HOST for a machine running Debian testing (lenny/sid)
#
# using Debian packages:
#   libatlas-$ATLAS_ARCH-dev (ATLAS_ARCH=base, 3dnow, sse, or sse2) - see below
#   libmpich1.0-dev
#   mpichbin
#   libparmetis-dev
#   libpng12-dev
#   zlib1g-dev
#
# Debian packages need to be installed first:
# apt-get install libmpich1.0-dev libparmetis-dev libpng12-dev zlib1g-dev

METIS_DIR     =
EXTERNAL_LIB  += -lmetis
CFLAGS        += -DMETIS -I/usr/include/metis

PNG_DIR     = /usr/lib
ZLIB_DIR    = /usr/lib
MPI_DIR     = /usr/lib/mpich

# ATLAS packages: includes full BLAS and LAPACK
#
# Debian has different ATLAS packages optimized for different optimized
# instruction sets.  Check the processor flags in /proc/cpuinfo to find out
# which extensions your processor supports, and install the appropriate
# package.
#
# apt-get install libatlas3-$ATLAS_ARCH-dev
#
# ATLAS_ARCH should be set to the highest-performing vectorized math implementation
# that your system is capable of, in decreasing order of preference:
#
# optimized packages:
#
# ATLAS_ARCH  comment
# ---------------------------------------------------------------
#   sse2      AMD/Intel processors with SSE2 extensions (32-bit mode)
#   sse       AMD/Intel processors with SSE extensions (32-bit mode)
#   3dnow     AMD processors with 3dnow extensions (32-bit mode)
#   altivec   PowerPC processors
#   ev6       Alpha processors
#   v9        Sparc processors
#
#ATLAS_ARCH    = sse2
#ATLAS_DIR     = /usr/lib/$ATLAS_ARCH/atlas
#EXTERNAL_LIB  += -L/usr/lib/$ATLAS_ARCH -latlas
#
# generic package:
#
# ATLAS_ARCH  comment
# ---------------------------------------------------------------
#   base      generic for i386 compatible processors and
#             for AMD/Intel 64-bit processors in 64-bit mode
```

```
#
ATLAS_ARCH    = generic
ATLAS_DIR     = /usr/lib/atlas
EXTERNAL_LIB  += -latlas
```

12.5 Makefile.in.host_oldlibs

```
# example Makefile.in.$HOST using older libraries

# override settings in Makefile.in.defaults
libatlas = atlas3.6.0_Linux_HAMMER64SSE2.tar.gz
liblapack=lapack-3.1

METIS_DIR    = $(PD)/ParMetis-3.1
PNG_DIR     = $(PD)/libpng-1.2.22
ZLIB_DIR    = $(PD)/zlib-1.2.3

# override settings in Makefile.in
PETSC_VERSION = 2.3.3-p4
PETSC_ARCH  = PETSc-config-magpar
PETSC_DIR   = $(PD)/petsc-$(PETSC_VERSION)
TAO_DIR     = $(PD)/tao-1.9

# set Fortran compiler
# use g77 on old RedHat
FC = g77
PETSC_XTRALIBS=/usr/lib/gcc-lib/i386-redhat-linux/2.96/libg2c.a
```

12.6 Makefile.files

```
###############################################################################
# $Id: Makefile.files 2746 2009-08-26 21:49:39Z scholz $
###############################################################################

griddata  = griddata.h

magparobj = main.o
main.o: $(magpar_verh) $(magpar_revh)

initdir   = init
initobj   = $(initdir)/destroyinit.o \
            $(initdir)/distortmesh.o \
            $(initdir)/elevertvol.o \
            $(initdir)/facnb.o \
            $(initdir)/filterelements.o \
            $(initdir)/filternodes.o \
            $(initdir)/initinfo.o \
            $(initdir)/maginit.o \
            $(initdir)/magset.o \
            $(initdir)/modifyprop_par.o \
            $(initdir)/modifyprop_ser.o \
            $(initdir)/movedata.o \
            $(initdir)/parinit.o \
            $(initdir)/parteleser.o \
            $(initdir)/regrefine.o \
            $(initdir)/reorder.o \
            $(initdir)/serinit.o \
            $(initdir)/vertprop.o
magparobj += $(initobj)

$(initdir)/initinfo.o: $(magpar_verh)
```

```
fielddir  = field
fieldobj  = $(fielddir)/bele.o \
            $(fielddir)/hdemag.o \
            $(fielddir)/hexch_ani.o \
            $(fielddir)/hcubic.o \
            $(fielddir)/helastic.o \
            $(fielddir)/hexternal.o \
            $(fielddir)/hext_cu.o \
            $(fielddir)/hext_ho.o \
            $(fielddir)/hext_py.o \
            $(fielddir)/htot.o \
            $(fielddir)/hstep_file.o
magparobj += $(fieldobj)

$(fielddir)/hdemag.o: $(fielddir)/bmatrix.c

iodir     = io
ioobj     = $(iodir)/readinp.o \
            $(iodir)/readmesh.o \
            $(iodir)/readkrn.o \
            $(iodir)/readpatran.o \
            $(iodir)/writedata.o \
            $(iodir)/writedataavs.o \
            $(iodir)/writedatadat.o \
            $(iodir)/writefemavs.o \
            $(iodir)/writelog.o \
            $(iodir)/writelog_pid.o
magparobj += $(ioobj)

utildir   = util
utilobj   = $(utildir)/area.o \
            $(utildir)/ascat.o \
            $(utildir)/axesrot.o \
            $(utildir)/barycent.o \
            $(utildir)/bbox2.o \
            $(utildir)/calAfe2fe.o \
            $(utildir)/calAfe2sq.o \
            $(utildir)/calAsq2fe.o \
            $(utildir)/calcbbox.o \
            $(utildir)/cart2sphere.o \
            $(utildir)/distint.o \
            $(utildir)/distortvec.o \
            $(utildir)/distpointline.o \
            $(utildir)/ipol.o \
            $(utildir)/matviewstruct.o \
            $(utildir)/matcreateseqadj.o \
            $(utildir)/mesh2dual.o \
            $(utildir)/printmatinfo.o \
            $(utildir)/progressbar.o \
            $(utildir)/renormvec.o \
            $(utildir)/solidangle.o \
            $(utildir)/syncffprintf.o \
            $(utildir)/tettri.o \
            $(utildir)/vecsetvec.o
magparobj += $(utilobj)

llgdir    = llg
llgobj    = $(llgdir)/myllgjacobian.o \
            $(llgdir)/calc_dMdt.o

ifneq ($(SUNDIALS_VERSION),FALSE)
llgobj    +=$(llgdir)/checkiterationllg.o \
            $(llgdir)/mytscreatepvode.o \
            $(llgdir)/mytssteppvode.o \
            $(llgdir)/precond.o \
            $(llgdir)/rhsfunction.o \
            $(llgdir)/writelog_pvode.o
```

```
endif
magparobj += $(llgobj)

ifneq ($(TAO_DIR),FALSE)
eminidir  = emini
eminiobj  = $(eminidir)/checkiterationemini.o \
            $(eminidir)/eminisolve.o
magparobj += $(eminiobj)
endif

ifneq ($(PNG_DIR),FALSE)
pngdir    = png
pngobj    = $(pngdir)/writedatapng.o \
            $(pngdir)/writedatapng2.o \
            $(pngdir)/writepng.o
magparobj += $(pngobj)
endif

ifeq ($(EBM),TRUE)
ifneq ($(SUNDIALS_VERSION),FALSE)
  CFLAGS += -DEBM
  include ebm/Makefile.in
endif
endif

ifeq ($(ADDONS),TRUE)
  ENABLED_SECTIONS = addons
  CFLAGS += -DADDONS
  addonsdir = addons
  include $(addonsdir)/Makefile.in

  llg2dir   = llg2
  llg2obj   = $(llg2dir)/myts2.o \
              $(llg2dir)/checkiteration2.o
  magparobj += $(llg2obj)

  llgtsdir  = llgts
  llgtsobj  = $(llgtsdir)/mytscreate.o \
              $(llgtsdir)/checkiterationllgts.o
  magparobj += $(llgtsobj)

  browndir  = brown
  brownobj  = $(browndir)/brown.o
  magparobj += $(brownobj)
endif

ifeq ($(SPINTORQ),TRUE)
  CFLAGS += -DSPINTORQ
  spintorqdir = spintorq
  include $(spintorqdir)/Makefile.in
endif

# other settings ##############################################################

ifeq ($(EXCH),TRUE)
  CFLAGS += -DEXCH
endif

# append default CFLAGS (maybe set to "CFLAGS += -I. -o $@")
CFLAGS += -I.

# configure for linking static executable if requested
ifeq ($(STATIC),TRUE)
  EXTERNAL_LIB   += -static
  SL_LINKER_LIBS := $(SL_LINKER_LIBS:-lgcc_s=)
  BLASLAPACK_LIB := $(BLASLAPACK_LIB:-lgcc_s=)
  PCC_LINKER_LIBS:= $(PCC_LINKER_LIBS:-lgcc_s=)
```

```
  FC_LINKER_LIBS := $(FC_LINKER_LIBS:-lgcc_s=)
endif
```

12.7 Makefile.libs

```
################################################################################
# $Id: Makefile.libs 2908 2009-12-08 19:54:00Z scholz $
################################################################################
#
# This file is part of magpar.
#
# Copyright (C) 2002-2009 Werner Scholz
#
# www:   http://www.magpar.net/
# email: magpar(at)magpar.net
#
# magpar is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# magpar is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with magpar; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
#
################################################################################

# set Fortran compiler
# GNU GCC >=4.0 Fortran 77/95: gfortran
FC=gfortran
TIMER=INT_ETIME
#
# GNU GCC < 4.0 Fortran 77: g77
#FC=g77
#TIMER=EXT_ETIME

# Modify Makefile.in, Makefile.in.$HOSTNAME:
# Update variables, paths to required libraries, compiler options, etc.

# get desired configuration
include Makefile.in

# Tell make to export all variables to child processes by default.
# http://www.cs.utah.edu/dept/old/texinfo/make/make.html#SEC62
export

all: atlas lapack mpi parmetis sundials petsc tao zlib libpng
others: atlas_compile gmp mpfr gcc mgridgen python numpy matplotlib scotch

############################
# GMP
############################

# http://gmplib.org/

gmplib=$(PD)/gmplib/

# latest version is: 4.3.1
gmp: $(gmplib)
```

```
$(gmplib):
gmprel=4.2.2; \
cd $(PD); \
lib=gmp-$$gmprel; \
if [ ! -e $$lib.tar.gz ]; then wget -N ftp://ftp.gnu.org/gnu/gmp/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
instdir=$(PD)/gmp_$$gmprel; \
./configure --prefix=$$instdir 2>&1 | tee configure.log; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) check 2>&1 | tee makecheck.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
ln -s $$instdir $(PD)/gmp

############################
# MPFR
############################

# http://www.mpfr.org/

mpfrlib=$(PD)/mpfrlib/

# latest version is: 2.4.2
mpfr: $(mpfrlib)
$(mpfrlib):
mpfrrel=2.3.1; \
cd $(PD); \
lib=mpfr-$$mpfrrel; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://www.mpfr.org/mpfr-current/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
instdir=$(PD)/mpfr_$$mpfrrel; \
./configure --with-gmp=$(PD)/gmp --prefix=$$instdir 2>&1 | tee configure.log; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) check 2>&1 | tee makecheck.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
ln -s $$instdir $(PD)/mpfr

############################
# GCC
############################

# http://gcc.gnu.org/mirrors.html
# ftp://mirrors.laffeycomputer.com/pub/gcc.gnu.org/pub/gcc/releases/

gccbin=$(PD)/gcc/bin/gcc

# latest version is: 4.4.2
gcc: $(gccbin)
$(gccbin):
gccrel=4.3.0; \
cd $(PD); \
srcdir=$(PD)/gccsrc-$$gccrel; \
mkdir $$srcdir; \
cd $$srcdir; \
ftpserv=http://mirrors.usc.edu/pub/gnu/gcc/; \
for i in core fortran g++ ; do \
  lib=gcc-$$i-$$gccrel.tar.bz2; \
  if [ ! -e $$lib ]; then wget -N $$ftpserv/gcc-$$gccrel/$$lib; fi; \
  bunzip2 -c $$lib | tar xvf -; \
done; \
instdir=$(PD)/gcc-$$gccrel; \
cd gcc-$$gccrel;\
./configure --with-gmp=$(PD)/gmp --with-mpfr=$(PD)/mpfr --prefix=$$instdir 2>&1 | tee configure.log; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
ln -s $$instdir $(PD)/gcc
```

```
############################
# ATLAS
############################

atlaslib = $(PD)/atlas/lib/libatlas.a
atlas: $(atlaslib)
$(atlaslib):
cd $(PD); \
if [ -n "$$libatlas" ]; then lib=$(libatlas); \
else lib=atlas3.6.0_Linux_PIIISSE1.tar.gz; \
fi; \
if [ ! -e $$lib ]; then wget -N http://downloads.sourceforge.net/math-atlas/$$lib; fi; \
arctype=`file $$lib | awk '{print $$2}'` ; \
echo $$lib; \
if [ ! -d $$lib ] ; then $$arctype -d -c $$lib | tar xvf - ; fi ; \
atlasdir=`$$arctype -d -c $$lib | tar tf - |grep libatlas.a` ; \
atlasdir=`dirname $$atlasdir`; atlasdir=`dirname $$atlasdir`; \
rm -f atlas; ln -s -f $$atlasdir atlas; \
mv $(lapacklib) $(lapacklib).atlas

atlas_compile:
instdir=$(PD)/atlas; \
cd $(PD); \
mkdir -p $$instdir; \
lib=atlas3.9.17.tar.bz2; \
if [ ! -e $$lib ]; then wget -N http://downloads.sourceforge.net/math-atlas/$$lib; fi; \
bunzip2 -c $$lib | tar xvf -; \
cd $(PD)/atlas; \
$(PD)/ATLAS/configure $(atlasgcc) 2>&1 | tee configure.log; \
$(MAKE) -j 1 2>&1 | tee make.log; \
$(MAKE) check 2>&1 | tee make_check.log; \
$(MAKE) ptcheck 2>&1 | tee make_ptcheck.log; \
$(MAKE) test 2>&1 | tee make_test.log; \
$(MAKE) time 2>&1 | tee make_time.log; \
ln -s -f Linux_* atlas; \
mv $(lapacklib) $(lapacklib).atlas

############################
# LAPACK
############################

lapacklib2 = $(PD)/$(liblapack)/lapack_LINUX.a
$(lapacklib2): $(atlaslib)
$(FC) --version
cd $(PD); \
lib=$(liblapack); \
if [ ! -e $$lib.tgz ]; then wget -N http://www.netlib.org/lapack/$$lib.tgz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tgz|tar xv ; fi ; \
cd $(liblapack); \
mtmpl=INSTALL/make.inc.gfortran; \
cp $$mtmpl make.inc; \
$(MAKE) "FORTRAN=$(FC)" "LOADER=$(FC)" \
  "TIMER=$(TIMER)" \
  "BLASLIB=$(PD)/atlas/lib/libf77blas.a $(PD)/atlas/lib/libatlas.a" \
  "OPTS=-funroll-all-loops $(OPTFLAGS)" \
  "NOOPT=$(NOOPT)" \
  lapacklib 2>&1 | tee make.log; \

lapacklib = $(PD)/atlas/lib/liblapack.a
lapack: $(lapacklib2) $(lapacklib)
$(lapacklib): $(atlaslib) FORCE
cd $(PD)/atlas/lib; \
cp $(lapacklib).atlas $(lapacklib); \
mkdir tmp; cd tmp; \
ar x $(PD)/$(liblapack)/lapack_LINUX.a; \
ar r ../liblapack.a *.o; \
```

```
cd ..; rm -rf tmp

############################
# MPI
############################

# select requested mpi library here (choose from targets below)
mpiimp=mpich2

mpilib=$($(mpiimp)lib)
mpi: $(mpiimp)

############################
# MPICH2
############################

mpich2lib = $(PD)/mpich2/lib/libmpich.a
mpich2: $(mpich2lib)
$(mpich2lib):
cd $(PD); \
lib=1.2.1; \
if [ ! -e mpich2-$$lib.tar.gz ]; then wget -N --retr-symlinks http://www.mcs.anl.gov/research/projects/mpi
if [ ! -e mpich2-$$lib.tar.gz ]; then \
  lib=1.2.1; \
  echo "Automatic fallback to older version mpich2-$$lib"; \
  wget -N --retr-symlinks ftp://ftp.mcs.anl.gov/pub/mpi/mpich2-$$lib.tar.gz; \
fi; \
if [ ! -d mpich2-$$lib ] ; then gunzip -c mpich2-$$lib.tar.gz|tar xv ; fi ; \
cd mpich2-$$lib; \
instdir=$(PD)/$(mpiimp); \
./configure --prefix=$$instdir --with-device=ch3:ssm 2>&1 | tee configure.log; \
echo "Enforcing '-j 1' because of problems"; \
echo "with simultaneous/parallel make jobs"; \
$(MAKE) -j 1 2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
cd ..; ln -s $(mpiimp) mpi

############################
# MVAPICH2 1.0.X
############################

mvapich2olib = $(PD)/mvapich2o/lib/libmpich.a
mvapich2o: $(mvapich2olib)
$(mvapich2olib):
cd $(PD); \
lib=mvapich2-1.1;  ext=tar.gz; \
pl= ; \
if [ ! -e $$lib$$pl.tar.gz ]; then wget -N --retr-symlinks http://mvapich.cse.ohio-state.edu/download/mvap
if [ ! -d $$lib ] ; then gunzip -c $$lib$$pl.$$ext|tar xv ; fi ; \
cd $$lib; \
instdir=$(PD)/$(mpiimp); \
unset CFLAGS; export PREFIX=$$instdir; export OPEN_IB_HOME=/usr; export F77=$(FC); export F90=$(FC); \
./make.mvapich2.ofa | tee configure.log; \
cd ..; ln -s $(mpiimp) mpi

# ./configure --prefix=$$instdir --with-device=osu_ch3:mrail:ssm --with-rdma=gen2 --with-pm=mpd --disable-
# $(MAKE) 2>&1 | tee make.log; \
# $(MAKE) install 2>&1 | tee makeinst.log

############################
# MVAPICH2 1.2 and later
############################

mvapich2lib = $(PD)/mvapich2/lib/libmpich.a
mvapich2: $(mvapich2lib)
$(mvapich2lib):
cd $(PD); \
```

```
lib=mvapich2-1.4; ext=tgz; \
pl= ; \
if [ ! -e $$lib$$pl.tar.gz ]; then wget -N --retr-symlinks http://mvapich.cse.ohio-state.edu/download/mvap
if [ ! -d $$lib ] ; then gunzip -c $$lib$$pl.$$ext|tar xv ; fi ; \
cd $$lib; \
instdir=$(PD)/$(mpiimp); \
./configure --prefix=$$instdir 2>&1 | tee configure.log; \
$(MAKE)  2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
cd ..; ln -s $(mpiimp) mpi

############################
# MPICH1
############################

mpich1lib = $(PD)/mpich1/lib/libmpich.a
mpich1: $(mpich1lib)
$(mpich1lib):
cd $(PD); \
lib=mpich-1.2.7p1; \
if [ ! -e $$lib.tar.gz ]; then wget -N --retr-symlinks ftp://ftp.mcs.anl.gov/pub/mpi/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
instdir=$(PD)/$(mpiimp); \
./configure | tee configure.log; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) testing 2>&1 | tee makeinst.log; \
./bin/mpiinstall --prefix=$$instdir; \
cd ..; ln -s $(mpiimp) mpi

############################
# OpenMPI
############################

openmpilib = $(PD)/openmpi/lib/libopenmpi.a
openmpi: $(openmpilib)
$(openmpilib):
cd $(PD); \
lib=openmpi-1.3.4; \
if [ ! -e $$lib.tar.gz ]; then wget -N --retr-symlinks http://www.open-mpi.org/software/ompi/v1.2/downloac
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
instdir=$(PD)/$(mpiimp); \
./configure --prefix=$$instdir | tee configure.log; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
cd ..; ln -s $(mpiimp) mpi

############################
# ParMETIS
############################

parmetislib = $(PD)/ParMetis-3.1.1/libmetis.a
parmetis: $(parmetislib)
$(parmetislib):
cd $(PD); \
lib=ParMetis-3.1.1; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://glaros.dtc.umn.edu/gkhome/fetch/sw/parmetis/$$lib.tar.gz; fi
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib/METISLib; \
$(MAKE) "CC=$(MPI_DIR)/bin/mpicc" "LD=$(MPI_DIR)/bin/mpicc" 2>&1 | tee make.log

############################
# MGridGen
############################

mgridgenlib = $(PD)/ParMGridGen-1.0/libmgrid.a
```

```
mgridgen: $(mgridgenlib)
$(mgridgenlib):
cd $(PD); \
lib=ParMGridGen-1.0; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://www-users.cs.umn.edu/~moulitsa/download/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib/; \
$(MAKE) "make=make" 2>&1 | tee make.log

############################
# SCOTCH
############################

scotchlib = $(PD)/scotch_5.0/libscotch.a
scotch: $(scotchlib)
$(scotchlib):
cd $(PD); \
lib=scotch_5.0; \
if [ ! -e $$lib.6.tgz ]; then wget -N http://gforge.inria.fr/frs/download.php/5218/$$lib.6.tgz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib/src; \
ln -s Make.inc/Makefile.inc.i686_pc_linux2 Makefile.inc ; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log

############################
# SUNDIALS/PVODE
############################

# set compiler options (modify for your setup!)
# add CPU specific options, e.g. -march=pentium4 -msse2 (cf. man gcc)

sundialslib = $(SUNDIALS_DIR)/lib/libsundials_cvode.a
sundials: $(sundialslib)
$(sundialslib):
cd $(PD); \
echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"; \
echo "Please register on the SUNDIALS webpage:"; \
echo "http://www.llnl.gov/CASC/sundials/download/download.html"; \
echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"; \
lib=`basename $(SUNDIALS_DIR)`; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://www.magpar.net/static/magpar/libs/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $(PD)/$$lib; \
instdir=$(PD)/$$lib; \
./configure --prefix=$$instdir --with-mpi-root=$(MPI_DIR) CFLAGS="$(OPTFLAGS)" --with-precision=$(PRECISIO
echo "forcing make install with -i to ignore errors when overwriting include files"; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) -i install 2>&1 | tee makeinst.log

############################
# Python
############################

pythonver=2.6.4
python: FORCE
cd $(PD); \
libver=$(pythonver); \
lib=Python-$$libver; \
if [ ! -e $$lib.tgz ]; then wget -N http://www.python.org/ftp/python/$$libver/$$lib.tgz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tgz|tar xv ; fi ; \
cd $$lib; \
export CPP=cpp CC=gcc CXX=g++; \
./configure --prefix=$(PD)/python 2>&1 | tee configure.log; \
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log
```

```
############################
# NumPy
############################

numpy: FORCE
cd $(PD); \
libver=1.3.0; \
lib=numpy-$$libver; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://internap.dl.sourceforge.net/sourceforge/numpy/$$lib.tar.gz;
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
export CPP=cpp CC=gcc CXX=g++; unset CFLAGS; unset LDFLAGS; \
rm -f site.cfg ; \
echo "[atlas]"  >> site.cfg; \
echo "libraries = f77blas, cblas, atlas"  >> site.cfg; \
echo "library_dirs = $(PD)/atlas/lib"  >> site.cfg; \
echo "include_dirs = $(PD)/atlas/include"  >> site.cfg; \
$(PD)/python/bin/python setup.py install

############################
# matplotlib
############################

matplotlib: FORCE
cd $(PD); \
libver=0.99.1.2; \
lib=matplotlib-$$libver; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://internap.dl.sourceforge.net/sourceforge/matplotlib/$$lib.tar
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
export CPP=cpp CC=gcc CXX=g++; unset CFLAGS; unset LDFLAGS; \
$(PD)/python/bin/python setup.py build; \
$(PD)/python/bin/python setup.py install

############################
# PETSc
############################

#petsclib = $(PETSC_DIR)/lib/PETSc-config-magpar/libpetsc.a
petsc: FORCE
cd $(PD); \
lib=`basename $(PETSC_DIR)`; \
if [ ! -e $$lib.tar.gz ]; then wget ftp://ftp.mcs.anl.gov/pub/petsc/release-snapshots/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
cp $(MAGPAR_SRC)/PETSc-config-magpar.py $(PETSC_DIR)/config/; \
OPTFLAGS="$(OPTFLAGS)"; export OPTFLAGS; \
PRECISION="$(PRECISION)"; export PRECISION; \
./config/PETSc-config-magpar.py 2>&1 | tee config.log; \
$(MAKE) all 2>&1 | tee make.log

petsc_mingw: FORCE
cd $(PD); \
lib=`basename $(PETSC_DIR)`; \
if [ ! -e $$lib.tar.gz ]; then Xwget ftp://ftp.mcs.anl.gov/pub/petsc/release-snapshots/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $$lib; \
cp $(MAGPAR_SRC)/PETSc-config-magpar.py $(PETSC_DIR)/config/; \
OPTFLAGS="$(OPTFLAGS)"; export OPTFLAGS; \
PRECISION="$(PRECISION)"; export PRECISION; \
./config/PETSc-config-magpar.py 2>&1 | tee config.log; \
hfile=PETSc-config-magpar/include/petscconf.h; \
cp $$hfile $$hfile.bak; \
cat $$hfile.bak | \
sed "/PETSC_HAVE_GETPAGESIZE/,+2 d" | \
sed "/PETSC_HAVE_IEEEFP_H/,+2 d" | \
sed "/PETSC_HAVE_NETDB_H/,+2 d" | \
```

```
sed "/PETSC_HAVE_PWD_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_PROCFS_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_RESOURCE_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_TIMES_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_UTSNAME_H/,+2 d" \
> $$hfile; \
$(MAKE) all 2>&1 | tee make.log

# Installation instructions for PETSc versions older than 2.3.0
# have been moved to the FAQ page.

############################
# TAO
############################

# magpar requires
# (PETSc version 3.0.0 and TAO 1.10) or
# (PETSc version 2.3.3 and TAO 1.9) or
# (PETSc version 2.3.2 and TAO 1.8.2) or
# (PETSc version 2.3.0 and TAO 1.8) or
# (PETSc version 2.2.1 and TAO 1.7) or
# (PETSc version 2.2.0 and TAO 1.6) or
# (PETSc version 2.1.6 and TAO 1.5)

taolib = $(TAO_DIR)/lib/PETSc-config-magpar/libtao.a
tao: $(taolib)
$(taolib):
cd $(PD); \
echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"; \
echo "Please register on the TAO webpage:"; \
echo "http://www.mcs.anl.gov/research/projects/tao/download/"; \
echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"; \
lib=`basename $(TAO_DIR)`; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://www.mcs.anl.gov/research/projects/tao/download/$$lib.tar.gz;
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
cd $(TAO_DIR); \
echo "Enforcing '-j 1' because of problems"; \
echo "with simultaneous/parallel make jobs"; \
$(MAKE) -j 1 2>&1 | tee make.log

############################
# zlib
############################

zliblib = $(ZLIB_DIR)/libz.a
zlib: $(zliblib)
$(zliblib):
cd $(PD); \
lib=`basename $(ZLIB_DIR)`; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://www.zlib.net/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
ln -s $$lib zlib; \
cd $$lib; \
$(MAKE) CC=$(CC) CFLAGS="-O -fPIC" 2>&1 | tee make.log; \
$(MAKE) test 2>&1 | tee maketest.log

############################
# libpng
############################

pnglib = $(PNG_DIR)/lib/libpng12.a
libpng: $(pnglib)
$(pnglib): $(zliblib)
cd $(PD); \
lib=`basename $(PNG_DIR)`; \
if [ ! -e $$lib.tar.gz ]; then wget -N http://downloads.sourceforge.net/libpng/$$lib.tar.gz; fi; \
if [ ! -d $$lib ] ; then gunzip -c $$lib.tar.gz|tar xv ; fi ; \
```

```
cd $$lib; \
instdir=$(PD)/$$lib; \
true ./configure --prefix=$$instdir --enable-shared=no 2>&1 | tee configure.log; \
true CFLAGS="-I$(ZLIB_DIR)"; export CFLAGS; \
true LDFLAGS="-L$(ZLIB_DIR)"; export LDFLAGS; \
./configure CFLAGS="-I$(ZLIB_DIR)" LDFLAGS="-L$(ZLIB_DIR)" --prefix=$$instdir --enable-shared=no 2>&1 | te
$(MAKE) 2>&1 | tee make.log; \
$(MAKE) install 2>&1 | tee makeinst.log; \
$(MAKE) check 2>&1 | tee makecheck.log

############################
# Gmsh
############################

gmsh: FORCE
cd $(PD); \
ver=2.4.2; \
arc=gmsh-$$ver-Linux.tgz; \
if [ ! -e $$arc ]; then wget -N http://geuz.org/gmsh/bin/Linux/$$arc; fi; \
gunzip -c $$arc|tar xv

############################
# Netgen
############################

netgen: FORCE
cd $(PD); \
ver=4.9.11; \
arc=netgen-$$ver.tar.gz; \
if [ ! -e $$arc ]; then wget -N http://voxel.dl.sourceforge.net/sourceforge/netgen-mesher/$$arc; fi; \
gunzip -c $$arc|tar xv

# dummy target to enforce rebuild of other targets
FORCE:

.PHONY: all atlas atlas_compile lapack mpi mpich2 mvapich2 mpich2 openmpi parmetis mgridgen sundials petsc
```

Chapter 13

FAQ

Sections:

13.1 Successful installations

Please send me an email, if you have successfully installed magpar on a system different from those listed below: magpar(at)magpar.net

CPU	OS	magpar version	comment
various AMD and Intel dual and quad-core processors	CentOS 5 (64 bit)	0.8	using automated installation with Makefile.libs, GCC 4.2.1, and self compiled ATLAS 3.7.36 on 64-bit Linux
AMD Opteron Processor 250 (2 x 2.4 GHz)	Red Hat Enterprise Linux 3 (64 bit)	0.5-0.8	64-bit Linux
Sun UltraSPARC IV+	Sun Solaris 10 (64 bit)	0.8	InstallationOnSolarisSPARC
AMD Athlon64 3200	Linux (64 bit)	0.4	64-bit Linux
Intel Xeon, AMD Athlon MP	RedHat 7.3	0.2-0.8	using RedHat's GNU compilers 2.96, atlas3.6.0_Linux_-ATHLON and Intel compilers
Pentium M	Debian 4.0 (etch), and Debian testing (lenny/sid)	0.1-0.8	using atlas3.6.0_-Linux_P4SSE2, GCC 4.1.3, GCC 4.2.1
Pentium 4	RedHat 9	0.2-0.5	using RedHat's RPM packages for LAPACK, libpng, zlib; using MPICH, because mpi++.h is missing in RedHat's LAM/MPI
Apple PowerMac G5	Mac OS X 10.4.2 w/ Xcode 2.1	0.5	powerpc-apple-darwin8-gcc-4.0.0 (GCC) 4.0.0 (Apple Computer, Inc. build 5026) on Apple Macintosh running Mac OS X
Apple iBook G4 1.2 GHz	Mac OS X 10.3	0.5	Apple Macintosh running Mac OS X
Pentium III	Fedora Core 2	0.3-0.5	LAM/MPI 7.1.1
Alpha EV68	Tru64 5.1	0.1-0.5	DEC/Compaq/hp Alpha machines running OSF/1, Tru64 : using MPICH, dxml; Compaq AlphaServer SC45 V2.6 consisting of 11 ES45 nodes with 4 processors (1 GHz, 8 MB cache/CPU) each and 16 GB RAM/node, connected with Quadrics Supercomputer World high speed interconnect
Alpha EV6 (21264)	Tru64 5.0A	0.1-0.3	using MPICH, using dxml
Alpha EV56	RedHat 6.2	0.1-0.3	g++ is broken, compiled PETSc only with BOPT=O (no C++ support, no TAO, no energy minimization), self

13.2 64-bit Linux

On recent 64-bit Linux distributions, the Installation instructions apply and the installation works out of the box. Still, the following suggestions might be helpful.

Problem

Compilation of magpar and all required libraries on AMD 64-bit processors (Athlon64, Opteron) running 64-bit Linux.

13.2.1 Solution 1

AMD Opteron Processor 250 (2 x 2.4 GHz) running Red Hat Enterprise Linux WS release 3 (Taroon Update 4)

LAPACK: The (incomplete) implementation distributed with ATLAS works: http://prdownloads.sourceforge.net/math-atlas/atlas3.6.0_Linux_-HAMMER64SSE2.tar.gz?download

Otherwise use RedHat's LAPACK rpm "lapack-3.0-20" - also available from RPMfind: ftp://rpmfind.net/linux/redhat/enterprise/3/en/os/i386/SRPMS/lapack-3.0-20.src.rpm

The generic LAPACK from netlib does not work!

Everything else works as described in the Installation instructions.

13.2.2 Solution 2

submitted by Richard Boardman (thanks!):

Good news! I've managed to get magpar running on a pure 64-bit Linux system on AMD64. It was a bit of an adventure.

ATLAS: This needed to be built manually, as it looks like they've not used -fPIC (position independent code) everywhere. I needed to get the source and override (despite the warnings) the compiler flags to send -fPIC and -m64 to both the C and the Fortran components (IIRC)

LAPACK: Setting the -fPIC flags and the -m64 thing does the trick, apart from at the link stage. Two files needed to be compiled manually (and then the linking done manually):

```
g77 -c -m64 -fPIC dlamch.f
g77 -c -m64 -fPIC slamch.f
```

as LAPACK tries to compile these and these alone (weird) without the PIC.

MPI(CH): absolutely must have the -fPIC and the -m64 stuff in there. I think it's pretty much OK to go with:

```
CFLAGS="-fPIC -m64" FFLAGS="-fPIC -m64" RSHCOMMAND="ssh" \
./configure && make
```

though watch out - if PETSc fails, then check carefully where the R_X86_64_32 relocation error occurred and (a) try and make the natural build process cover this, (b) manually build it if this fails. I guess this is true for all of them.

PETSc: The above stages are necessary for PETSc 2.2.(0) - assuming everything above went to plan, then PETSc will build, but make sure the following lines in bmake/linux/variables are set:

```
C_CC            = gcc -fPIC -m64
C_FC            = g77 -Wno-globals -fPIC -m64
O_COPTFLAGS     = -O -Wall -Wshadow  -fomit-frame-pointer -fPIC -m64
O_FOPTFLAGS     = -O -fPIC -m64
CXX_CC          = g++ -fPIC -m64
CXX_FC          = g77 -Wno-globals -fPIC -m64
GCXX_COPTFLAGS  = -g -m64 -fPIC
OCXX_COPTFLAGS  = -O -m64 -fPIC
GCOMP_COPTFLAGS = -g -m64 -fPIC
OCOMP_COPTFLAGS = -O -m64 -fPIC
```

Here is a suggestion for (quite pedantic ;-) compile flags for PETSc:

```
export CFLAGS="-O3 -fPIC -march=k8 -msse2 -m64 -mfpmath=sse -m3dnow
-fexpensive-optimizations -fforce-addr -fforce-mem -finline-functions
-funroll-loops -Wall -Winline -W -Wwrite-strings -Wno-unused"
```

zlib, libpng and magpar: These should have -m64 -fPIC in their build process for consistency.

If all this is done, hopefully a nice static pure 64-bit binary should be built, Opteron-friendly :)

13.3 Intel compilers

(tested with Intel compiler (icc, ifort) version 9.0)

In order to compile magpar (and all the libraries) with the Intel compilers one can just add the following snippet to the host specific Makefile.in.$HOSTNAME or Makefile.in (update paths to your installation directories!)

```
CC=/usr/local/bin/icc
CPP=/usr/local/bin/icc -E
CXX=/usr/local/bin/icpc
FC=/usr/local/bin/ifort
TIMER=EXT_ETIME

# Generate code for Intel Pentium 4 processors and enables new
# optimizations in addition to Intel processor-specific optimizations.
OPTFLAGS += -O3 -xN
# http://icl.cs.utk.edu/lapack-forum/viewtopic.php?t=295
NOOPT = -O3 -fltconsistency
PETSC_XTRALIBS=/usr/local/lib/libifcore.a
```

These settings are required especially for the compilation of LAPACK (set the variables in the make.inc file) as discussed here: http://icl.cs.utk.edu/lapack-forum/viewtopic.php?t=295

It is possible to use the precompiled ATLAS libraries, even though the Intel Math Kernel Library, which implements all BLAS and LAPACK routines, might be a good option, too.

The configuration of PETSc might fail when it tests the linking of Fortran object files with mpicc. To solve this problem it is necessary to add "-lirc" to "MPI_OTHERLIBS" in mpif90.

If you use Makefile.libs to install the libraries, it will use the settings in Makefile.in.$HOSTNAME or Makefile.in.defaults. If you compile by hand, define the variables above on the command line (and export them if necessary).

13.4 Apple Macintosh running Mac OS X

Problem

Compilation of magpar and all required libraries on Apple Macintosh running Mac OS X.

13.4.1 Solution 1

submitted by Richard Boardman (thanks!)

tested on Apple iBook G4 1.2 GHz with Mac OS X 10.3

I got everything up and running on Darwin/G4 [OSX to everyone else ;)]; here are a few observations.

1] get Fink∗ from SourceForge.net: Fink is a Debian-like package manager for Mac OS X, and contains amongst other things ATLAS and LAPACK

2] get ATLAS from Fink

3] get LAPACK from Fink

4] download MPICH and configure as usual I believe LAM/MPI and/or the packages from Fink might work, too.

5] ParMetis-3.1

- one should adjust the include and lib paths in ParMetis' Makefile.in to point to /sw/lib and /sw/include to pick up ATLAS and LAPACK.
- note that on my Panther setup I needed to place /usr/include/malloc in my include path for ParMetis as malloc.h wasn't being picked up by default
- also note that the usual /usr/lib and /usr/include should be left in the Makefile.in

6] SUNDIALS - download and configure as usual

7] PETSc 2.3.0

Copy PETSc-config-magpar.py to darwin-gnu-magpar.py

```
cp PETSc-config-magpar.py darwin-gnu-magpar.py
```

update it (Fink installs the libraries, e.g. ATLAS, LAPACK, in /sw) and add the following line to the "configure_options":

```
'--with-cxx=g++', # mpiCC does not work (some case sensitivity issue)
```

Then, configure and run as usual.

8] TAO, zlib, and libpng as usual

13.4.2 Solution 2

submitted by Greg Parker (thanks!)

tested on Apple PowerMac G5 2x2.0 GHz with Mac OS X 10.4.1 w/ Xcode 2.1 and

Apple PowerMac G5 2x2.5 GHz with Mac OS X 10.4.1 w/ Xcode 2.0

Turns out things are pretty easy on Tiger. No need for Fink or a Fortran compiler. Apple supplies their own BLAS/LAPACK libraries and headers, so no need for Atlas or Lapack. If they are installed, the configure system of PETSc 2.3.0 finds them by itself.

LAM/MPI: Download lam-7.1.1.dmg.gz from http://www.lam-mpi.org/7.1/download.php and use the installer, which puts it in /usr/local/...

Then type:

```
cd $PD
mkdir mpi
cd mpi
ln -s /usr/local/bin
ln -s /usr/local/include
ln -s /usr/local/lib
PATH=$PD/mpi/bin:$PATH
export PATH
lamboot -v
```

ParMetis 3.1: Edit $MAGPAR_HOME/ParMetis-3.1/Makefile.in (do not modify CC or LD variables as described here: ParMetis)

```
# INCDIR = -I/usr/local/include -I/usr/include -I/usr/include/malloc
# LIBDIR = -L/usr/local/lib -L/usr/lib
```

and compile as usual.

SUNDIALS 2.1.0: Configure with the following command

```
./configure --disable-f77 --with-mpi-incdir=$PD/mpi/include --with-mpi-libdir=$PD/mpi/lib
```

and compile as usual.

PETSc 2.3.0

```
./config/configure.py --with-fortran=0 --with-mpi-include=$PD/mpi/include \
  --with-clanguage=cxx --with-debugging=0 \
  --with-mpi-lib=[$PD/mpi/lib/libmpi.a,$PD/mpi/lib/liblammpi++.a,$PD/mpi/lib/liblam.a] \
  --with-cxx=mpic++ -CXXOPTFLAGS="-O3 -Wno-long-double" --with-mpirun=mpirun
PETSC_ARCH=<arch> # where <arch> is whatever config said, e.g. darwin8.2.0
export PETSC_ARCH
PETSC_DIR=$PWD
export PETSC_DIR
make all
make test # be sure to have X11 running first, if it fails, may have to type 'lamboot -v' again
```

libpng-1.2.5

```
cp scripts/makefile.macosx ./makefile
make # you will get errors compiling pngtest, but you can ignore them
```

Compiling magpar itself

In Makefile.in change the PETSc architecture to whatever PETSc's config said above

```
PETSC_ARCH = <arch>
CFLAGS += -O2 -faltivec
```

13.5 Optimized BLAS libraries

It is highly recommended to use any machine specific (vendor provided and highly optimized) libraries. On most high performance machines there are optimized BLAS and LAPACK libraries available, e.g.:

- IBM AIX: ESSL
- DEC/Compaq Alpha: DXML, CXML
- SUN: SUNPERF

In addition, the following implementations are available:

- ATLAS: Automatically Tuned Linear Algebra Software - recommended
- Generic BLAS and LAPACK implementations from netlib - portable but slow
- BLAS by Kazushige Goto: Visiting Scientist, FLAME project, UT-Austin - untested
- Intel Math Kernel Library: Optimized library for Intel Intel processors - untested
- AMD Math Core Library: Optimized library for AMD processors - untested

13.6 Precompiled packages on Ubuntu/Debian

Using Debian-based distributions (including Ubuntu) one can save a lot of time by using some of the pre-built packages.

The following instructions pertain specifically to Debian Lenny and were used successfully in June 2008 and saved in Makefile.in.host_debian, which can serve as a template for your Makefile.in.$HOSTNAME.

Makefile.in.host_debian

PETSc and Sundials still have to be compiled from source, since the Debian packages were built for uniprocessor systems only and do not work with MPICH. TAO also must be built from source. For this purpose one can still use the Automated Installation for the individual libraries like this

```
cd $MAGPAR_HOME/src
make -f Makefile.libs sundials petsc tao
```

or install them using the Manual Installation method.

Thanks to Daniel Lenski for his installation report.

13.7 Compiling LAPACK on RedHat9 exits with an error

Problem

The GNU Fortran compiler shipping with RedHat 9 (GNU Fortran (GCC 3.2.2 20030222 (Red Hat Linux 3.2.2-5))) has problems with complex variables.

```
...
SEP: Testing Symmetric Eigenvalue Problem routines
./xeigtstc < sep.in > csep.out 2>&1
make: *** [csep.out] Error 139
```

At this stage, the LAPACK library has been compiled successfully, but the complex test programs fail.

Solution

Since magpar does not use any complex variables, one can safely ignore this error message and continue with the installation.

13.8 PETSc does not compile on RedHat 6.2 with BOPT=g_c++ or BOPT=O_c++

Problem

The GNU C++ compiler shipping with RedHat 6.2 (gcc version egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)) is broken and fails with an internal error:

```
libfast in: /home/scholz/work/magpar/libs/petsc-2.1.6/src/vec/esi
eindexspace.c: In method `::esi::petsc::::esi::petsc::IndexSpace<int>::
IndexSpace(class ::esi::::esi::IndexSpace<int> &)':
eindexspace.c:21: Internal compiler error.
...
```

Solution

GCC 2.95.3 (shipping with SuSE 8.0), GCC 2.95.4 (shipping with Debian 3.0 woody) and GCC 3.2.2 have proved to work. Get some binary package for your platform or compile GCC from source.

You can get the latest GCC release from here:

http://gcc.gnu.org/

http://gcc.gnu.org/install/binaries.html

http://gcc.gnu.org/mirrors.html

13.9 DEC/Compaq/hp Alpha machines running OSF/1, Tru64

13.9.1 Problems linking various libraries with "ar"

Problem

The following error occured on a Compaq machine running Tru64 UNIX V5.1A (Rev. 1885):

```
/: write failed, file system is full
ar:  error writing archive member contents: [...]
*** Exit 1
Stop.
```

after compiling the source files of MPICH (and ParMetis-3.1, too) when "make" tried to create the libraries using "ar".

Solution

Set the environment variable "TMPDIR" to a directory/partition with sufficent free space, e.g. your home directory:

```
TMPDIR=$HOME
export TMPDIR
```

and recompile.

13.9.2 Problems with snprintf

Tru64 version 4.x does not provide an "snprintf" function. Thus, the compilation of PETSc fails in $PETSC_DIR/src/sys/src/viewer/impls/mathematica/. A simple replacement of snprintf by sprintf and omitting the size argument solves the problem.

in $PETSC_DIR/src/sys/src/viewer/impls/mathematica/mathematica.c
replace:

```
snprintf(linkname, 255, "%6d", ports[rank]);
```

with:

```
sprintf(linkname, "%6d", ports[rank]);
```

and all other occurences of snprintf in this file accordingly.

13.9.3 Compiling ParMetis

If you are using the ELAN libraries (for Quadrics high speed interconnect) instead of MPI(CH) over Ethernet:

```
# modify $MAGPAR_HOME/src/Makefile.in to use
# the normal C compiler instead of mpicc
CC = cc
XTRALIBS = -lmpi -lelan
LD = cc
# test example
prun -p scp1 -n 4 -s -o out -e err ./ptest rotor.graph
```

Otherwise use a "normal" MPI library (MPICH, LAM/MPI) and compile as usual.

13.9.4 Compiling SUNDIALS version 2.1

If you are using the ELAN libraries (for Quadrics high speed interconnect) instead of MPI(CH) over Ethernet:

```
# use the normal C and Fortran compilers instead of mpicc and mpif77
./configure --with-mpicc=cc --with-mpif77=f77
```

13.9.5 Compiling SUNDIALS version 1.0

```
cd $PD/sundials

# replace compiler "gcc" by "cc"
# remove compiler option "-ffloat-store"
files=`find . -name "*"`
for i in `grep -l "CFLAGS =" $files`; do
  sed "s/gcc/cc/g" $i | sed "s/-Wall -ffloat-store//g" > $i.tmp
  mv $i.tmp $i
done

# optional: compile with optimizations
# (no improvement/speedup found)
for i in `grep -r -l "CFLAGS =" $files`; do
  sed "s/CFLAGS = /CFLAGS = -O3 /g" $i > $i.tmp
  mv $i.tmp $i
done

unset files
```

13.9.6 Compiling PETSc 2.3.0

Install Python (if necessary) as described in the Installing Python.
Modify $MAGPAR_HOME/src/PETSc-config-magpar.py:

```
...
# use Compaq Extended Math Library (CPML/CXML)
'--with-blas-lib=libcxml',
# use ELAN libraries (if you have Quadrics high speed interconnect)
'--with-mpi-lib=libmpi.a,libelan.a',
...
```

Otherwise use a "normal" MPI library (MPICH, LAM/MPI) and compile as usual.

13.9.7 Compiling PETSc 2.2.1 and earlier

```
...
PETSC_DIR=$PD/petsc-2.2.1
export PETSC_DIR
# choose suitable predefined configuration (directory name) for your platform
PETSC_ARCH=alpha
export PETSC_ARCH
cd ..
ln -s predefined/$PETSC_ARCH
cd $PETSC_ARCH
cp packages packages.bak
# edit the file "packages":
# Location of BLAS and LAPACK
# BLASLAPACK_LIB = -ldxml
# use ELAN libraries (if you have Quadrics high speed interconnect)
# MPI_LIB        = -L${MPI_HOME}/lib -lmpi -lelan
# otherwise use MPICH (as usual)
...
cp petscconf.h petscconf.h.bak
# edit petscconf.h if you want to have static binaries (recommended):
# replace "#define PETSC_USE_DYNAMIC_LIBRARIES 1"
# by "#undef PETSC_USE_DYNAMIC_LIBRARIES"
# add "#define PETSC_HAVE_NETDB_H"
# (if you really have a "netdb.h" somewhere in /usr/include)
#
...
```

13.9.8 Compiling libpng

```
...
cp scripts/makefile.dec Makefile
# edit the Makefile if necessary:
# update the paths to zlib:
# ZLIBLIB=../zlib-1.1.4 -lz -lm
# ZLIBINC=../zlib-1.1.4
make
make test
```

13.9.9 Compiling magpar

Update Makefile.in if you are using PETSc 2.2.1 or earlier:

```
...
# set the proper PETSC_ARCH (which was used for compiling PETSc):
```

```
# PETSC_ARCH  = alpha
# link statically:
# CLINKER_STATIC = -non_shared
# optional: really tough checking with Tru64's cc:
# CFLAGS     += -portable -check -verbose
...
```

13.10 magpar crashes with a segmentation violation

Problem

This segmentation fault might happen during mesh refinement, mesh partitioning (or really just the first call to a METIS function). In parteleser::c, for example, METIS_MeshToDual is called to convert the mesh into its dual graph and obtain the adjacency structure of the mesh. If you happen to have a very weird mesh in which many elements share a single node, some static arrays in Metis are too small.

Solution

You can either generate a better mesh (if this problem occurs many elements are terribly degenerate with high aspect ratio and low quality factor anyway) or patch METIS:

Edit $PD/ParMetis-3.1.1/METISLib/mesh.c:79 and increase the array size:

```
idxtype *mark, ind[500], wgt[500];
```

Then you have to recompile the ParMetis package and recompile magpar:

```
cd $PD/ParMetis-3.1.1/
make
cd $MAGPAR_HOME/src/
# remove the magpar binary, so a new one is linked
rm magpar.exe
make
```

13.11 Single precision arithmetics

NB: magpar does not appear to work in single precision mode at all right now! There seem to be big problems with the KSP and TAO solvers (convergence issues!?).

To compile magpar with single precision floating point arithmetics:

set floating point precision in Makefile.in.$HOSTNAME

```
PRECISION=single
```

patch TAO to make it compile with PETSc in single precision (patch not available yet)

recompile PETSc, Sundials, TAO

```
make -f Makefile.libs petsc tao sundials
```

recompile magpar

```
make
```

13.12 Single processor version without MPI

Compile magpar according to the Installation instructions with the following modifications:

13.12.1 ParMetis

In $PD/ParMetis-3.1.1/METISLib/metis.h remove line 25:

```
#include "../parmetis.h"  /* Get the idxtype definition */
```

Compile serial version only:

```
cd METISLib
make CC=gcc LD=gcc
```

Do not use Metis-4.0.1 (November 1998), because it contains a couple of bugs, which have been ironed out in the Metis version included in ParMetis!

13.12.2 SUNDIALS

```
./configure --prefix=$PWD --disable-mpi
make
make -i install
```

13.12.3 PETSc

modify $MAGPAR_HOME/src/PETSc-config-magpar.py:

comment out all MPI related lines except for:

```
'--with-mpi=0',
```

Then configure and compile as usual.

13.13 Compiling magpar in Cygwin for Windows

Here is a recipe for compiling magpar in the Cygwin environment on a Windows machine:

- get Cygwin: http://cygwin.com/
- download the setup program http://cygwin.com/setup.exe and run it
- not all packages are required - here is an overview of recommended packages:

```
Category     Package
-----------------------------------------------
Admin        none
Archive      none
Audio        none
Base         all
Database     none
Devel        binutils, gcc, gcc-core, gcc-g++, gcc-g77, gcc-mingw-*, make
Doc          none (recommended: cygwin-doc, man)
Editors      none
Games        none
Gnome        none
Graphics     libpng12*
Interpreters gawk, python, perl
Libs         zlib
Mail         none
Math         lapack
```

```
Mingw         none
Net           none
Publishing    none
Shells        bash, (recommended: mc)
Text          less
Utils         bzip2, cygutils, diffutils (required by PETSc)
Web           wget
X11           none
```

In Makefile.in.defaults or your own Makefile.in.$HOSTNAME set

```
ATLAS_DIR=/usr/lib
```

to use Cygwin's LAPACK.

Follow the general Installation instructions and those for Single processor version without MPI. If magpar is compiled in the Cygwin environment, magpar will generate complete inp files (cf. project.INP.inp).

In addition, users have reported successful parallel magpar installations/runs with MPICH in the Cygwin environment:

- Download and install the Microsoft Visual C++ 2005 SP1 Redistributable Package (required by MPICH Windows binary packages).
- Download and install a suitable MPICH Windows binary package (∗.msi) from the MPICH2 Downloads page.
- Update the MPI_DIR variable in your magpar Makefile.in.$HOST or Makefile.in.defaults to the MPICH2 installation directory, e.g. "C:\Program Files\MPICH2\".
- Follow the general instructions for compilation in Cygwin above.
- Run a magpar example with 2 magpar processes on the local machine (e.g. a dual core processor) with a command like this:

  ```
  mpiexec.exe -n 2 -localonly ./magpar.exe
  ```

- Running parallel magpar processes on remote machines should only be a matter of a proper MPICH2 network installation. This requires the installation of MPDs (MPICH daemons) as a service on the remote machines.

magpar executables for Windows and the source code of all required libraries are available on the magpar homepage.

13.14 Compiling magpar using the MinGW compilers in Cygwin

This section describes the procedure for compiling magpar using the MinGW compilers in the Cygwin environment on a Windows machine. The advantage over compilation with the native Cygwin compilers is, that the MinGW compiled executables do not require the cygwin.dll any more and the other libraries (see Running magpar on Windows) are linked statically into the executable. Compilation in the native MinGW/MSYS environment does not work (easily), because of PETSc using Python for its configuration.

Currently, the Windows version of magpar has only been compiled and tested in serial model (without MPI support). Thus, the installation procedure described in Single processor version without MPI should be used with the following modifications:

Install Cygwin as described in Compiling magpar in Cygwin for Windows with the addition of packages of the MinGW compilers (included in category "Devel"). The package names are gcc-mingw, gcc-mingw-core, gcc-mingw-g++, gcc-mingw, gcc-mingw-g77.

Compile the required libraries with the following changes (the most important being the addition of the "-mno-cygwin" option for the compilers):

BLAS/LAPACK:

```
cp make.inc.example make.inc
# generate BLAS and LAPACK libraries (no optimized BLAS for now)
make FORTRAN=g77 LOADER=g77 TIMER=EXT_ETIME OPTS="-funroll-all-loops -O3 -mno-cygwin" blaslib lapacklib
cp blas_LINUX.a libblas.a
cp lapack_LINUX.a liblapack.a
```

ParMetis:

```
make CC=gcc LD=gcc CFLAGS="-O3 -I. -mno-cygwin"
```

Sundials:

```
./configure --prefix=$PWD --disable-mpi --with-cflags=-mno-cygwin --with-ldflags=-mno-cygwin
```

PETSc:

```
cd $PETSC_DIR
export PRECISION=double
export OPTFLAGS="-mno-cygwin"
export ATLAS_DIR=/usr/lib
export PETSC_XTRALIBS=""
./config/PETSc-config-magpar.py
```

After running ./config/PETSc-config-magpar.py remove the following flags from $PETSC_DIR/PETSc-config-magpar/include/petscconf.h (or $PETSC_DIR/bmake/PETSc-config-magpar/petscconf.h for PETSc 2.3.x):

```
PETSC_HAVE_GETPAGESIZE
PETSC_HAVE_IEEEFP_H
PETSC_HAVE_NETDB_H
PETSC_HAVE_PWD_H
PETSC_HAVE_SYS_PROCFS_H
PETSC_HAVE_SYS_RESOURCE_H
PETSC_HAVE_SYS_TIMES_H
PETSC_HAVE_SYS_UTSNAME_H
```

simple sed script:

```
hfile=bmake/PETSc-config-magpar/petscconf.h; \
cp $hfile $hfile.bak; \
cat $hfile.bak | \
sed "/PETSC_HAVE_GETPAGESIZE/,+2 d" | \
sed "/PETSC_HAVE_NETDB_H/,+2 d" | \
sed "/PETSC_HAVE_PWD_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_PROCFS_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_RESOURCE_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_TIMES_H/,+2 d" | \
sed "/PETSC_HAVE_SYS_UTSNAME_H/,+2 d" \
> $hfile
```

zlib:

```
make CFLAGS="-O -mno-cygwin"
```

libpng:

```
# make sure $PD is set correctly
lib=libpng-1.2.33  # adjust to your version of libpng
zlib=zlib-1.2.3    # adjust to your version of zlib
./configure --prefix=$PD/$lib --enable-shared=no CFLAGS="-I$PD/$zlib -mno-cygwin" LDFLAGS="-L$PD/$zlib -
make && make install && make check
```

magpar:

```
# a simple "make" will compile and link the magpar executable
make
```

After compiling magpar (or really any program) one can check which DLLs the program depends on using

```
objdump -p magpar.exe
```

see also: http://www.delorie.com/howto/cygwin/mno-cygwin-howto.html

13.15 Running magpar on Windows

To run magpar on Windows either check Compiling magpar in Cygwin for Windows or download the archives containing the precompiled binaries from the magpar download page.

One zip archive contains the (executable) program itself and the required libraries:

- magpar.exe: magpar executable
- cygwin1.dll: Cygwin library
- cygblas.dll: BLAS library
- cyglapack.dll: LAPACK library
- cygpng12.dll: PNG library
- cygz.dll: zlib library

To run a magpar simulation

- extract the files from the archive
- copy the magpar executable into a simulation directory (e.g. one of the examples provided separately)
- copy the DLLs to C:\Windows\ (once) or also into each simulation directory
- double click on magpar.exe in the simulation directory
- or run the executable in a "Command Prompt" window (select "Start/Programs/Accessories/Command Prompt" or do "Run.../cmd").
- ideally run the executable in a "Command Prompt" like this "./magpar.exe > stdout.txt" so the informational output of magpar is saved in the file stdout.txt. Then inspect the file stdout.txt as the simulation runs (e.g. with WordPad - not Notepad due to Unix newlines)

The precompiled magpar executable provided

- is only a serial version (it does not support parallelization on SMP machines or clusters)
- generates complete project.INP.inp files
- calculates exchange field and energy separately from anisotropy
- includes K_2 in the uniaxial and cubic anisotropy field and energy

13.16 Graphical User Interface for Windows

The Windows executables of magpar are simple programs which just run in a terminal window without any nice user interface. However, thanks to the work of Tomasz Blachowicz and Bartlomiej Baron there is now also a nice graphical user interface for magpar available: MagParExt

Here is a quick introduction how to use it:

1. Download MagParExt from its homepage and install it.
2. Copy magpar.exe and the DLLs into a project (e.g. example) directory
3. Launch MagParExt
4. assign a new name to your project (independent of the magpar project/simName)
5. Point MagParExt to the project directory using the "Project files/Browse..." button
6. Check/modify the simulation parameters using the "Simulation/Configure..." button
7. Check/modify the material parameters using the "Materials/Configure..." button
8. Configure the application path to magpar.exe using "Options/General"
9. If a copy of the Cygwin DLLs is installed in "C:\Windows" there is no need to copy them into each project diretory.
10. Run the simulation using the "Simulation/Run" button
11. When the simulation finishes import the output data using the "Output data/Import from file..." button (make sure you open the $*$.log file in the correct directory)
12. Click on "Output data/Preview" to have a look at the log file
13. Visualize the results using "Graphs"
14. Add "Add new graph..." and assign a name to the graph
15. Set up "Data Series" by selecting data for x- and y-axis and click "Add"
16. Click "Ok" to view the plot
17. Go to "Animations" and press play to view them.

References:

See the paper by Tomasz Blachowicz and Bartlomiej Baron in the list of Publications and online at [arXiv].

13.17 Installing Python

If you do not have Python installed on your system or your Python version is older than 2.2. then download the latest source package of Python, configure, compile, and install it with

```
./configure --prefix=$PD/python
make
make install
```

and update $MAGPAR_HOME/src/PETSc-config-magpar.py with the full path to the python binary, e.g.

```
#!/home/scholz/work/magpar/libs/python/bin/python
```

(the environment variable "$MAGPAR_HOME" does not work here!)

13.18 Additional solvers and libraries for PETSc

These libraries are optional and they make additional preconditioners/linear solvers available to PETSc. They are not required for magpar.

Please consult the PETSc manual on how to make use of these additional solvers.

13.18.1 BlockSolve95

```
cd $PD
wget ftp://info.mcs.anl.gov/pub/BlockSolve95/BlockSolve95.tar.Z
tar xzvf BlockSolve95.tar.Z
cd BlockSolve95/bmake/linux
# edit linux.site:
#
# Location of BLAS and LAPACK.  These libraries are available via Netlib,
# or see $(BS_DIR)/readme for information on retrieving a subset.
#
# FC_LIB         =  -lg2c -lm
#
# BLAS_LIB     = -L${MAGPAR_HOME}/libs/atlas/lib -llapack -lf77blas -latlas $(FC_LIB)
#
# Location of MPI (Message Passing Interface) software
#
# MPI_HOME     = ${MAGPAR_HOME}/libs/mpi
# for MPICH:
# MPI_LIB      = -L${MPI_HOME}/lib -lmpich -lpmpich # -lfmpich
# for LAM/MPI:
# MPI_LIB      = -L${MPI_HOME}/lib -lmpi -llamf77mpi -llam
#
# MPI_INCLUDE  = -Dmpi -I${MPI_HOME}/include
# MPIRUN       = ${MPI_HOME}/bin/mpirun
#
# Location of BlockSolve95
#
# BS_INCLUDE     = -I${MAGPAR_HOME}/libs/BlockSolve95/include
# BS_LIB         = -L${MAGPAR_HOME}/libs/BlockSolve95/lib/libO/${PETSC_ARCH} -lBS95
cd $PD/BlockSolve95
PETSC_ARCH=linux
export PETSC_ARCH
make ACTION=lib BOPT=O tree # use BOPT=O even if you use O++ for PETSc below!
```

13.18.2 hypre

This library is optional and makes additional preconditioners/linear solvers available to PETSc. It is not required for magpar.

Please consult the PETSc manual on how to make use of these additional solvers.

```
cd $PD
wget http://www.llnl.gov/CASC/hypre/download/hypre-1.8.2b.tar.gz
tar xzvf hypre-1.8.2b.tar.gz
cd hypre-1.8.2b/src
# for MPICH
./configure --with-mpi-include=${MAGPAR_HOME}/libs/mpich/include \
            --with-mpi-libs="mpich pmpich fmpich" \
            --with-mpi-lib-dirs=${MAGPAR_HOME}/libs/mpich/lib \
            --with-blas="-I${MAGPAR_HOME}/libs/atlas/include \
            -L${MAGPAR_HOME}/libs/atlas/lib -llapack -lf77blas -latlas -lg2c"
# for LAM/MPI
./configure --with-mpi-include=${MAGPAR_HOME}/libs/lam/include \
            --with-mpi-libs="mpi lamf77mpi lam" \
```

```
            --with-mpi-lib-dirs=${MAGPAR_HOME}/libs/lam/lib \
            --with-blas="-I${MAGPAR_HOME}/libs/atlas/include \
            -L${MAGPAR_HOME}/libs/atlas/lib -llapack -lf77blas -latlas -lg2c"
# fix utilities/fortran.h (remove one underscore at end of line)
# line 31: define hypre_NAME_C_CALLING_FORT(name,NAME) name##_
# line 32: define hypre_NAME_FORT_CALLING_C(name,NAME) name##_
make
```

13.18.3 SuperLU

This library is optional and makes additional preconditioners/linear solvers available to PETSc. It is not required for magpar.

Please consult the PETSc manual on how to make use of these additional solvers.

```
cd $PD
wget http://crd.lbl.gov/~xiaoye/SuperLU/superlu_dist_2.0.tar.gz
tar xzvf superlu_dist_2.0.tar.gz
cd SuperLU_DIST_2.0
# edit make.inc:
# PLAT            = _linux
# DSuperLUroot    = ${MAGPAR_HOME}/libs/SuperLU_DIST_2.0
# BLASLIB         = -L${MAGPAR_HOME}/libs/atlas/lib -llapack -lf77blas -latlas -lg2c
# for MPICH
# MPILIB          = -L${MAGPAR_HOME}/libs/mpi/lib -lmpich -lpmpich
# for LAM/MPI
# MPILIB          = -L${MAGPAR_HOME}/libs/mpi/lib -lmpi -llamf77mpi -llam
# CC              = ${MAGPAR_HOME}/libs/mpi/bin/mpicc
# CFLAGS          = -O3
# FORTRAN         = ${MAGPAR_HOME}/libs/mpi/bin/mpif77
# FFLAGS          = -O3
# LOADER          = ${MAGPAR_HOME}/libs/mpi/bin/mpicc
# LOADOPTS        = #
# CDEFS           = -DAdd_
```

13.19 Links to other FAQs, troubleshooting guides

An extensive list of known problems and difficulties is listed on the PETSc website:

http://www.mcs.anl.gov/petsc/petsc-as/documentation/troubleshooting.html

http://www.mcs.anl.gov/petsc/petsc-as/documentation/faq.html

13.20 What is "magpar" worth (at least ;-) ?

```
Total Physical Source Lines of Code (SLOC)                = 11,632
Development Effort Estimate, Person-Years (Person-Months) = 2.63 (31.56)
 (Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))
Schedule Estimate, Years (Months)                         = 0.77 (9.28)
 (Basic COCOMO model, Months = 2.5 * (person-months**0.38))
Estimated Average Number of Developers (Effort/Schedule)  = 3.40
Total Estimated Cost to Develop                           = $ 355,287
 (average salary = $56,286/year, overhead = 2.40).
```

generated using David A. Wheeler's SLOCCount version 2.26

13.21 Other micromagnetics software

Here are a few links to other free and commercial micromagnetics packages:

- OOMMF (Object Oriented MicroMagnetic Framework)
- nmag - a flexbile micromagnetic simulation package
- JaMM is Java Micromagnetics
- RKMAG
- LLG Micromagnetics Simulator
- MicroMagus
- MagOasis
- FEMME - multiscale finite element micromagnetic package
- PC Micromagnetics Simulator
- AlaMag

13.22 Installation of old library versions

13.22.1 MPICH1

```
cd $PD
wget ftp://ftp.mcs.anl.gov/pub/mpi/mpich.tar.gz
tar xzvf mpich.tar.gz
cd mpich-1.2.6
# patch -p0 < ../patchfile
#  if you experience problems applying the patch
#  go back to the original mpich.tar.gz
./configure
make
make testing
# Problems with RedHat 6.2 bash: testing fails
# solution: replace bash version 1 with version 2:
# cd /bin; mv bash bash.bak; ln -s bash2 bash
#
# install the binaries in $PD/mpich
./bin/mpiinstall -prefix=$PD/mpich
# set symbolic link to MPICH installation directory
cd $PD
ln -s mpich mpi
```

13.22.2 LAM/MPI

```
cd $PD
wget http://www.lam-mpi.org/download/files/lam-7.1.2.tar.gz
tar xzvf lam-7.1.2.tar.gz
cd lam-7.1.2
./configure --prefix=$PD/lam
make
make install
# set symbolic link to LAM installation directory
cd $PD
ln -s lam mpi
# start LAM universe on local machine:
lamboot -v
```

13.23 How does boundary matrix size scale with mesh length?

Since the boundary matrix contains one row and one column for every surface node, it is clear that boundary matrix size $\sim$ (number of surface nodes)^2.

The surface of the mesh consists of triangles, edges, and points: Their numbers are:

$$\begin{aligned} t &= \text{number of triangles} \\ e &= \text{number of edges} = \frac{3}{2}t \qquad \text{(3 edges/triangle, each shared by exactly two triangles)} \\ n &= \text{number of nodes} \end{aligned}$$

If the mesh is a continuous, perfect simple hexagonal mesh, then each node is shared by six triangles. There are three nodes per triangle and thus:

$$\begin{aligned} n &= (\text{number of triangles}) \times (\text{number of nodes/triangle})/(\text{number of triangles sharing each node}) \\ &= t \times 3/6 \\ &= \frac{t}{2} \end{aligned}$$

Consider the process of subdividing every triangle into four smaller triangles. This is what Magpar does when the `-refine 1` option is given, and is effectively the same as reducing the mesh length by a factor of 2:

How are e_f, t_f, n_f related to e_0, t_0, n_0?

1. every triangle becomes four new triangles, thus $tf = 4t_0$
2. a new node is placed along every edge, thus $n_f = n_0 + e_0$

Since $e = \frac{3}{2}t$, we get $n_f = n_0 + \frac{3}{2}t_0$.

In the case of a simple hexagonal mesh: $t = 2n$, thus $n_f = n_0 + 3n_0$, or $n_f = 4n_0$. In this case, $n \sim$ (mesh length)$^{-2}$.

This is **neither an upper bound nor a lower bound**, since it is possible for there to be more than 6 triangles sharing each surface on average, or less than 6 on average. But most of the non-hexagonal cases I can think of are pretty pathological, and a quick glance at a mesh produced by Gmsh shows that surface nodes overwhelmingly are shared by 6 triangles.

From this we conclude that $n \sim$ (mesh length)$^{-2}$ to an excellent approximation. Since (boundary matrix size) $\sim$ (number of surface nodes)^2, we conclude that: (boundary matrix size) $\sim$ (mesh length)$^{-4}$

13.24 Interpolating in a tetrahedral mesh with barycentric coordinates

Barycentric coordinates are ideal for interpolation within a tetrahedral finite-element mesh (such as used by Magpar). Barycentric interpolation expresses the interpolated function as a weighted sum of the function's values at the four vertices of the tetrahedral cell in which the interpolation point lies. Basically, it assumes linearity of the function within the cell:

$$f(\vec{x}) = f\left(\sum_{k=1}^{4} \lambda_k \vec{v}_k\right) = \sum_{k=1}^{4} \lambda_k f(\vec{v}_k) \quad \forall\ \vec{x} \in \text{ the tetrahedral volume with vertices } \vec{v}_1 ... \vec{v}_4$$

From this basic description, we infer:

- The sum of the n barycentric coordinates, denoted $\lambda_{1...n}$, is exactly 1.
- For any point on the surface/perimeter, $\lambda_i = 0$ for at least one i.
- For any point within the volume/area, $0 < \lambda_i < 1\ \forall\ i$.

Given the linearity of this coordinate system, there must be a matrix A that will transform from barycentric coordinates to Cartesian coordinates, that is $A \cdot \vec{\lambda} = \vec{x}$. For the 3D case, $\vec{x}$ is 3D and $\vec{\lambda}$ is 4D, thus A is 3×4.

We know how A maps the tetrahedron's vertices to barycentric coordinates:

$$A\begin{pmatrix}1\\0\\0\\0\end{pmatrix} = \vec{v}_1, \quad A\begin{pmatrix}0\\1\\0\\0\end{pmatrix} = \vec{v}_2, \quad A\begin{pmatrix}0\\0\\1\\0\end{pmatrix} = \vec{v}_3, \quad A\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \vec{v}_4, \quad \text{thus} \quad AI = A = \begin{pmatrix} v_{1x} & v_{2x} & v_{3x} & v_{4x} \\ v_{1y} & v_{2y} & v_{3y} & v_{4y} \\ v_{1z} & v_{2z} & v_{3z} & v_{4z} \end{pmatrix}$$

So far so good, we can map barycentric coordinates to tetrahedral vertices. However, the reverse isn't there yet. We need to recall the additional constraint on the sum of the barycentric coordinates:

$$\sum_{k=1}^{4} \lambda_k = 1$$

We can combine this with the matrix A to get:

$$T\vec{\lambda} = \begin{pmatrix}x\\y\\z\\1\end{pmatrix} \quad \text{where} \quad T = \begin{pmatrix} v_{1x} & v_{2x} & v_{3x} & v_{4x} \\ v_{1y} & v_{2y} & v_{3y} & v_{4y} \\ v_{1z} & v_{2z} & v_{3z} & v_{4z} \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad \text{and thus} \quad \vec{\lambda} = T^{-1}\begin{pmatrix}x\\y\\z\\1\end{pmatrix}$$

Now the inverse matrix, T^{-1}, maps from Cartesian coordinates to barycentric coordinates. Here's the complete algorithm to do barycentric interpolation of a point $\vec{x}$ within a tetrahedral mesh:

1. Compute the barycentric coordinate transformation matrices T for all the (nearby) cells in the mesh, and invert them to get T^{-1}.
2. For each cell and interpolation point, compute $\vec{\lambda} = T^{-1}\vec{x}$.
3. For each x, there will only be at most one i such that $0 < \lambda_k < 1$ for $k = 1...4$. If point $\vec{x}$ lies on the surface of a tetrahedron, there may be several choices with $0 \leq \lambda_k \leq 1$. Pick any one.
4. Now we know (a) which cell $\vec{x}$ is in, (b) its barycentric coordinates $\vec{\lambda}$ relative to that cell, and (c) the vertices $\vec{v}_1$, $\vec{v}_2$, $\vec{v}_3$, and $\vec{v}_4$ of that cell. Now it's easy: the interpolated function at point $\vec{x}$ is simply the $\vec{\lambda}$-weighted average of the function's value at the vertices,

$$f_{\text{interp}}(\vec{x}) = \sum_{k=1}^{4} \lambda_k f(\vec{v}_k)$$

Chapter 14

Preprocessing

Sections:

Several files are required for the configuration and initialization of a simulation (cf. Input Files for Simulations). However, they are very easy to adapt from existing Examples and finite element meshes of several simple geometries are included in the package of Examples available on the download page.

If a new geometry is required, some more advanced tools are necessary. magpar can import UCD files and MSC.Patran neutral files. The UCD format is very simple, so it should be very easy to write export filters/converters for other mesh generator, too.

The following CAD software and mesh generators have been used to generate geometries and finite element meshes for magpar.

14.1 Gmsh

From the Gmsh homepage:

```
Gmsh: a three-dimensional finite element mesh generator with built-in pre-
and post-processing facilities

Gmsh is an automatic 3D finite element grid generator with a built-in
CAD engine and post-processor. Its design goal is to provide a
simple meshing tool for academic problems with parametric input and
advanced visualization capabilities.

Gmsh is built around four modules: geometry, mesh, solver and post-processing.
The specification of any input to these modules is done either interactively
using the graphical user interface or in ASCII text files using Gmsh's own
scripting language.
```

After defining the geometry the 3D tetrahedral mesh has to be generated. Gmsh uses TetGen or NETGEN to generate the tetrahedral finite element mesh.

The mesh is then exported using Mesh/Save or File/Save Mesh as a $*$.msh file. In order to assign different property ids to different parts of the volume it is necessary to define corresponding "Physical groups" (Add/Volume). In addition, it might be necessary to tweak the ids of these physical volumes by hand in the $*$.geo file to ensure that these physical volumes are numbered consecutively starting from 1.

Sometimes, Gmsh completes the mesh generation but it cannot save the mesh. If this happens, it can help to increase the Linux/Unix resources - in particular the maximum number of open files - in /etc/security/limits.conf:

```
*               hard    nofile          10000
```

see: http://www.geuz.org/pipermail/gmsh/2008/003318.html

The Python program gmsh: gmshtoucd.py (provided in the $MAGPAR_HOME/src/tools/ directory), can then be used to convert Gmsh mesh files (version 2.0) into UCD/inp files (cf. project.inp, project.out: finite element mesh).

The examples sphere_larmor: Larmor precession and mumag3b: mumag standard problem #3 with 2 cubes use finite element meshes generated with Gmsh.

14.1.1 Optimizing meshes

Sometimes Gmsh will generate a suboptimal mesh! This will often cause magpar to run very slowly (because the mesh structure results in larger numerical errors in the magnetostatic field, which causes the time integrator to take very small time steps) or to fail when the linear solvers (for the calculation of the magnetostatic field, e.g. Au1∗u1=divM in Hdemag()) do not converge. You can check for a bad mesh by looking at the .felog generated by magpar at startup:

```
simname: strip
[...]
elevol:
  id_min: 7241
  id_max: 5335
  Vmax:   22.4616
  Vmin:   0.0000158533
  ratio:  1416840
  Vavg:   6.43932
  Vtot:   62500
[...]
edge_len_min: 0.003124
edge_len_max: 7.59121
edge_len_avg: 4.32508
[...]
```

Notice in this example that the ratio of the volume of the largest cell in the mesh to that of the smallest cell is **1416840**. Furthermore, the shortest edge (`edge_len_min`) is 2500 times shorter than the longest edge (`edge_len_max`). This will probably cause problems for magpar.

If Gmsh produces a poor mesh, there are a few things to try (options to insert at the beginning of the .geo file):

- try changing the meshing algorithm used by setting the variable `Mesh.Algorithm3D` in your .geo file:

  ```
  Mesh.Algorithm3D=1; // for Tetgen+Delaunay (default)
  Mesh.Algorithm3D=4; // for Netgen
  ```

- if this doesn't work, try optimizing the mesh by setting the variable `Mesh.Optimize` or `Mesh.OptimizeNetgen`:

  ```
  Mesh.Optimize=1;
  Mesh.OptimizeNetgen=1;
  ```

14.1.2 Command-line meshing

Gmsh can be invoked from the command line to generate a 3D mesh from a .geo file, which allows the process to be scripted. This is done by invoking gmsh with the `-3` option, as in:

```
$ gmsh -3 filename.geo
```

The `-optimize` and `-optimize_netgen` options may be added to produce an optimized mesh (equivalent to setting the `Mesh.Optimize` or `Mesh.OptimizeNetgen` variable in the .geo file).

14.2 NETGEN

From the NETGEN homepage (Linz site):

```
NETGEN is an automatic 3d tetrahedral mesh generator. It accepts input from
constructive solid geometry (CSG) or boundary representation (BRep) from
STL file format. The connection to a geometry kernel allows the handling of
IGES and STEP files. NETGEN contains modules for mesh optimization and
hierarchical mesh refinement. Netgen is open source based on the LGPL license.
It is available for Unix/Linux and Windows.
```

Richard Boardman and Hans Fangohr have implemented the tool ngtoucd.py, which converts neutrally-exported NETGEN meshes into AVS/UCD files (cf. project.inp, project.out: finite element mesh).

NETGEN can be found here:

- http://www.mathcces.rwth-aachen.de/netgen/
- http://www.hpfem.jku.at/netgen/
- http://netgen-mesher.wiki.sourceforge.net/
- http://sourceforge.net/projects/netgen-mesher

14.3 Salome

Salome provides a generic platform for Pre- and Post-Processing for numerical simulations. It is based on an open and flexible architecture made of reusable components.

Salome has powerful tools to generate 3D geometries (even with combinations of bottom-up [points > lines > faces > volumes] and top-down approaches [3D shapes > volumes > faces > lines > points] and tools to generate unions, intersections, cuts, etc.), mesh them, and export them in various formats.

Currently, there are no converters for the FE meshes generated by Salome yet, but Gmsh can import Salome's output.

14.4 GiD

GiD is the "personal pre- and postprocessor" from the "International Center for Numerical Methods in Engineering" (CIMNE). An evaluation version, which is restricted to 3000 finite elements, is available on the web for free. The installation of GiD is very easy on Windows as well as Linux platforms.

After the installation of GiD the new problem type "inp.gid" should be installed. This is done by copying the directory "$MAGPAR_HOME/src/tools/gid/inp.gid" to the subdirectory "problemtypes" of your GiD installation. If GiD recognized the new problem type, there should be an entry "inp" in the "Data/Problem type" menu.

In order to create a the finite element mesh of a simple geometry and export it in inp format, the following steps are required:

1. start GiD
2. set problem type:
   "Data/Problem type/inp"

3. create a sphere:
   "Geometry/Create/Object/Sphere"
4. enter the center of the sphere, e.g.
   "0,0,0"
5. enter the desired radius, e.g.
   "1"
6. create more geometric objects, e.g. a second sphere:
7. create a sphere:
   "Geometry/Create/Object/Sphere"
8. enter the center of the sphere, e.g.
   "3,0,0"
9. enter the desired radius, e.g.
   "1"
10. assign "Conditions" (=property ids=grain numbers) - the corresponding material parameters are later defined in project.krn: material properties in the appropriate line (line number=property id):
    "Data/Conditions"
11. set grain number
    "grain number 1"
12. "Assign"
13. select all objects, which should have property id "1", e.g. the left sphere
14. press the "Finish" button or the Esc key when finished
15. set grain number
    "grain number 2"
16. "Assign"
17. select all objects, which should have property id "2", e.g. the right sphere
18. press the "Finish" button or the Esc key when finished, "Close" the dialog window
19. If you have created independent points, lines, or surfaces delete them all
    (this is not necessary for the two spheres in this simple example):
    "Geometry/Delete/Point"
    "Geometry/Delete/Line"
    "Geometry/Delete/Surface"
    only volumes should be left over!
20. generate a tetrahedral finite element mesh:
    "Meshing/Generate"
21. Enter the size of elements, e.g.
    "0.3"

22. save the project:
    "Files/Save" "sphere"
23. generate the inp file:
    Calculate/Calculate
24. the inp file will be stored in the project directory "sphere.gid" with the suffix "dat", e.g. "sphere.dat"
25. rename the inp file:
    mv sphere.dat sphere.inp

Now you can check the validity of the inp file by rendering it with ParaView (cf. section Postprocessing).

14.5 MSC.Patran

`MSC.Patran` is another CAD program, which includes a mesher to generate finite element meshes. It can export the finite element meshes as `"neutral files"`, which can also be imported by magpar.

Chapter 15

Input Files for Simulations

Sections:

15.1 allopt.txt: simulation parameters

All simulation parameters can be set in the configuration file allopt.txt . The default configuration file allopt.txt in the $MAGPAR_HOME/src/doc/ subdirectory is thoroughly documented. Any option defined in this file can be overridden by an environment variable or command line option (cf. PETSc manual chapter 14 - Other PETSc Features). This useful feature is used in example mumag3: mumag standard problem #3 .

Additional PETSc internal logging/info/diagnostic options, which may slow down the simulations (!), are given in allopt_log.txt .

Deprecated and removed options can be found in Sec. allopt_ret.txt .

15.2 project.krn: material properties

For each grain (or part of the model with distinct property id) this file contains a line defining its material properties.

The grain with property id 1 is assigned the properties in line 1, the grain with property id 2 is assigned the properties in line 2, etc.

psi defines the third Euler angle for cubic anisotropy

example.krn:

```
  0.0   0.0   5.00E+06 0.0     1.32 1.40E-11 0.1   uni   # property 1
  0.0   0.0   1.50E+07 0.0     1.32 1.40E-11 0.1   uni   # property 2
#
# theta phi   K1       K2      Js   A        alpha psi   # parameter
# (rad) (rad) (J/m^3)  (J/m^3) (T)  (J/m)    (1)   (rad) # units

# theta and phi: direction of the uniaxial magnetocrystalline anisotropy axis in spherical coordinates (ra
#     theta measured from the z-axis, phi measured from the x-axis in the x-y-plane
# K1: first magnetocrystalline anisotropy constant (J/m3)
#     set <0 for in-plane anisotropy
# K2: second magnetocrystalline anisotropy constant (J/m3)
# Js: saturation polarization (Tesla)
#     set to -1 to have volumes with the given property id removed
# A:  exchange constant (J/m)
# alpha: Gilbert damping constant (dimensionless); set to 999 to "lock" magnetization
# psi: third Euler angle for cubic anisotropy (see http://mathworld.wolfram.com/EulerAngles.html)
#      or use "uni" for uniaxial anisotropy

# additional material parameters
# just examples, no warranty for accuracy!
#
# theta phi   K1       K2      Js   A        alpha psi   # parameter
```

```
# (rad)  (rad)  (J/m^3)   (J/m^3) (T)   (J/m)     (1)    (rad) # units
#
  0.0    0.0    4.60E+04 1.5e4   2.15 2.50E-11 0.1    0.0   # Kneller, Ferromagnetismus. Berlin, Springer, 19
                                                             # E. F. Kneller, R. Hawig, IEEE Trans. Magn. 27 (
                                                             # L. W. McKeehan, Phys. Rev. 51, 136-139 (1937)
  0.0    0.0    4.00E+04 0.0     0.40 1.32E-11 0.1    uni   # gamma-Fe2O3 (Maghemite): Johansson, J.M.M.M. 17
  0.0    0.0    4.50E+05 0.0     1.76 1.30E-11 0.1    uni   # Co: Yang, J. Appl. Phys. 87 (2000) 6884
  0.0    0.0    4.60E+06 0.0     1.60 9.86E-12 0.1    uni   # Nd2Fe14B: Klemmer, Script. Met. 33 no. 10/11 (1
                                                             # Weller, IEEE Trans. Magn. 36 (2000) 10-15
  0.0    0.0    0.0      0.0     1.00 1.05E-11 0.1    uni   # Permalloy: N. Smith, D. Markham, and D. LaToure
  0.0    0.0    5.00E+06 0.0     1.32 1.40E-11 0.1    uni   # Sm2Co17: Durst, Phys. Stat. Sol. (a) 108 (1988)
  0.0    0.0    9.00E+06 0.0     0.80 1.40E-11 0.1    uni   # SmCo5: Durst, Phys. Stat. Sol. (a) 108 (1988) 4
  0.0    0.0    1.50E+07 0.0     1.32 1.40E-11 0.1    uni   # SmCo5: Tang, IEEE Trans. Magn. 37 (2001) 2515
  0.0    0.0    1.80E+06 0.0     1.38 1.03E-11 0.1    uni   # FePd (L1_0): Klemmer, Script. Met. 33 no. 10/11
                                                       uni   # Weller, IEEE Trans. Magn. 36 (2000) 10-15
  0.0    0.0    6.60E+06 0.0     1.43 1.02E-11 0.1    uni   # FePt (L1_0): Klemmer, Script. Met. 33 no. 10/11
                                                       uni   # Weller, IEEE Trans. Magn. 36 (2000) 10-15
  1.5707963 0 3.9788736e4 0.0   1.00 1.00E-11 0.1    uni   # mumag standard problem #3 http://www.ctcms.nist
```

15.3 project.kst: magnetoelastic properties

For each grain (or part of the model with distinct property id) this file contains a line defining its magnetoelastic properties.

example.kst

```
  7        17.5e-6   103.7e-6  -1e9    0.0     0.0      # Fe65Co35: Bai, J. Appl. Phys. 95 (2004) 6864-686
#
# texture  lambda100 lambda111 sigmaX  sigmaY  sigmaZ   # parameter
# (-)      (erg/cm^3)(erg/cm^3)(Pa)    (Pa)    (Pa)     # units
#
# texture
# =======
# id   theta            phi     psi               comment
# ------------------------------------------------------------------------
#  0:                                              disable stress effects
#  1: 0                0       0
#  2: 0                random  random            100 texture, the film normal z is [001] direction
#  3: pi/2             0       3pi/4
#  4: pi/2             random  3pi/4             110 texture, the film normal z is [110] direction
#  5: pi/2             pi      asin(sqrt(2/3)))
#  6: asin(sqrt(2/3))) random  3pi/4             111 texture, the film normal z is [111] direction
#  7: random           random  random            random texture
#
# lambda100: magnetoelastic constant in 100 direction
# lambda111: magnetoelastic constant in 111 direction
# sigmaX:    stress in x direction (unit Pa)
# sigmaY:    stress in y direction (unit Pa)
# sigmaZ:    stress in z direction (unit Pa)
```

15.4 project.inp, project.out: finite element mesh

The finite element mesh can be imported in two formats:

- UCD file, the default file format of AVS. Various 3D CAD and meshing tools can be used for Preprocessing and mesh generation.

  see also: http://people.scs.fsu.edu/~burkardt/html/ucd_format.html

- "neutral file", which can be generated with MSC.Patran, for example.

A short tutorial can be found in section Preprocessing .

15.5 project.0001.inp: initial magnetization distribution

(optional)

magpar stores the magnetization distribution, magnetostatic potential, and local fields in UCD files. These UCD files, which contain the finite element mesh, can be used to read the finite element mesh as well as the magnetization distribution. This allows you to modify some parameters (e.g. the external field) and restart simulations from any magnetization distribution. The Cartesian components of the magnetization (M_x ,M_y ,M_z) are read from the second, third and fourth column of the vertex data section of the inp file. The first column contains the vertex id.

A short tutorial on postprocessing and visualization of magpar's output can be found in section Postprocessing.

Chapter 16

allopt.txt

Sections:

- allopt.txt : default configuration file
- allopt_log.txt : additional options
- allopt_ret.txt : deprecated and removed options

16.1 allopt.txt

Default configuration file for simulation parameters:
doc/allopt.txt:

```
#########################################################################
#
# magpar configuration file: allopt.txt
#
# default settings
#
#########################################################################

##### project name (required)
# (init/serinit.c)
#
-simName test

#########################################################################
# finite element mesh
#########################################################################

##### mesh type
# (init/serinit.c)
#
# 0: Patran: <simName>.out
# 1: AVS:    <simName>.inp
-meshtype 1

##### regular mesh refinement
# (init/regrefine.c)
#
# number of regular refinement steps
# (every step generates 8x as many elements and about 8x as many nodes!!!)
-refine 0

##### mesh manipulation
# (init/distortmesh.c)
#
# scale mesh in x,y,z direction by scaling factors given
-mesh_scale 1,1,1
#
# shift mesh in x,y,z direction
-shift 0,0,0
#
# distort mesh (shift vertices)
# 0: no distortion
# 1: distort only interior mesh
# 2: distort only boundary mesh
# 3: distort whole mesh
-meshdist 0
#
# max. distortion = distpar*(min. edge length of all tetrahedra)
-distpar 0
```

```
##### renumber vertices
# (init/reorder.c)
#
# matrix bandwidth reduction
# 0: disabled
# 1: enabled (Reverse Cuthill-McKee as implemented in PETSc, others available)
-optimizebw 1
#
# (init/parteleser.c)
# mesh partitioning using METIS
# 0:   disabled
# 1:   enabled, partition for actually used number of processors
# n>1: partition for n processors,
#      automatically mapped to actually used number of processors
-metispartition 0

#########################################################################
# materials, magnetization
#########################################################################

#### size scaling of finite element mesh (unit: m)
# (io/readkrn.c)
#
-size 1e-9

##### initial magnetization
# (init/maginit.c)
#
#  - specify a single value to initialize all materials in the same way or
#  - specify a list of values (comma separated, no spaces) to define
#    different init_mag (and init_magpar) parameters for each volume
#    with distinct property id (number of values has to exactly
#    the number of property ids)
#  - negative values: select abs(init_mag), but reverse the magnetization
#  0: magnetization from inp (set file number by -inp below)
#  1: Mx=1
#  2: My=1
#  3: Mz=1
#  4: Mx=My=Mz=sqrt(1/3)=0.57735027
#  5: artificial flower state, center: x=y=z=init_magparm
#  6: set magnetization in x-z plane to theta=init_magpar (in rad from z-axis)
#  7: vortex state: core radius = init_magparm, center at (x=0,y=0)
#  8: random magnetization
#  9: Bloch wall: center at x = init_magparm, width=x/10
# 10: M // anisotropy axes
# 11: set magnetization in x-y plane to theta=init_magpar (in **degrees** from x-axis)
# 12: Head-to-head transverse wall: center at x = init_magparm, width=x/10
# 13: Head-to-head vortex wall: core radius = init_magparm, center at (x=0,y=0)
# 14: set magnetization using a Python function called "initmag"
#     The function must be defined in the magpar Python module and
#     accept 3 arguments for space coordinates (in units of FE mesh)
#     and return 3 values (Cartesian components of the normalized magnetization)
#     like this:
#     ------------------------------------
#     def initmag(x,y,z):
#             return 0.0,1.0,0.0
#     ------------------------------------
-init_mag 3
-init_magparm 0
#
# (init/serinit.c)
# number of first inp file (to be read or written)
# full file name: <simName>.<inp>.inp
-inp 0001

##### modify material properties
#  0: do not modify any
```

```
# >0: pid of volume to be modified
#
# (init/modifyprop_ser.c)
# modify init/modifyprop_ser.c according to your needs
-nslicepropser 0
#
# (init/modifyprop_par.c)
# modify init/modifyprop_par.c according to your needs
-nsliceproppar 0

#########################################################################
# solution method
#########################################################################

##### run mode / minimization method
# (main.c, init/parinit.c)
# check for more options for each solver below
#
# 0:  PVODE (LLG time integration)
# 1:  TAO (energy minimization)
# 2:  relax with high damping and no precession during t<0,
#     then switch to normal damping (as defined in *.krn)
# 3:  relax with energy minimization, then switch to LLG time integration
# 99: exit immediately after calculating fields, energies, etc.
#     of initial magnetization
-mode 99

#########################################################################
# magnetic fields
#########################################################################

# /* mu0 = 4*M_PI*1e-7 = 12.566371e-7 Tm/A (=Vs/Am)*/
# /* gamma = mu0*g*|e|/(2*me) [m/As]  (cf. Diplomarbeit Scholz S. 14) */
# 795.77472 kA/m = 1 T

##### switch demagnetizing=magnetostatic field on/off
# (field/hdemag.c)
#
# 0: off
# 1: on
-demag 1
#
# select solver for solution of linear system (Poisson, Laplace equation)
# cf. PETSc manual chapter 4.3 Krylov Methods p. 63
# choices:
#    richardson chebychev cg gmres tcqmr bcgs cgs tfqmr cr lsqr
#    preonly qcg bicg fgmres minres symmlq lgmres
#
#   gmres:    good for small systems
#   cg:       shows better convergence rate for large systems
#   preonly:  for use with direct solvers (e.g. "-pc_type lu", see below)
#
# all standard ksp options (see below, just add prefix "-hdemag_u{1,2}")
-hdemag_u1_ksp_type cg
-hdemag_u1_ksp_rtol 1e-05
-hdemag_u1_ksp_atol 1e-50
-hdemag_u2_ksp_type cg
-hdemag_u2_ksp_rtol 1e-08
-hdemag_u2_ksp_atol 1e-50

##### hext_ho: homogeneous external field
# (field/hext_ho.c)
#
# initial field amplitude in kA/M (795.77472 kA/m = 1 T)
-hextini 0
# direction of external field (rad)
# theta measured from the z-axis
```

```
# phi measured from the x-axis in the x-y-plane
#        0.017453293 rad =  1 deg
#        0.034906585 rad =  2 deg
#        0.052359878 rad =  3 deg
#        0.087266463 rad =  5 deg
#        0.17453293  rad = 10 deg
# Pi/4 = 0.78539816  rad = 45 deg
# Pi/2 = 1.5707963   rad = 90 deg
# Pi   = 3.1415927   rad = 180 deg
-htheta 0
-hphi 0
# change external field in these steps (kA/m)
-hstep 0
# change external field at this speed (kA/(m*ns))
# (set "-tol 0" to avoid magpar exiting when the torque becomes very small)
-hsweep 0
# stop simulation if Hext < Hfinal (kA/m)
-hfinal 0
# scale external field (value given by -hextini) according to input file
# file format:
# <number of data pairs>
# <time> <scaling factor>
# ...
-hext_ho_htfile <filename>
# change field in steps when system reaches equilibrium
# (similar to -hext_ho_htfile; also fileformat)
# file format:
# <number of data pairs>
# <step number> <scaling factor>
# ...
-hext_ho_hstepfile <filename>

##### activate magnetoelastic (stress) parameters/effects
# (field/helastic.c)
#
# read material parameters from file given
-helastic_propfile <filename>

##### activate custom external field
# (field/hext_cu.c)
#
# 0: off
# 1: on
-hext_cu 0

##### activate custom external field defined by a Python function
# (field/hext_py.c)
#
# The function must be defined in the magpar Python module and
# accept 3 arguments for space coordinates (in units of FE mesh)
# and return 3 values (Cartesian components of field in Tesla)
# like this:
# ----------------------------------
# def hext(x,y,z):
#         return 0.0,0.2,0.1
# ----------------------------------
-hext_py <Python_function>

#########################################################################
# output
#########################################################################

##### output of data files (inp, png, dat files)
# (llg/mytssteppvode.c)
#
# save only every n'th line of output in log files
# yet, equilibrium states will always be saved
```

```
-ts_logsteps 1
#
# take this many timesteps at once
# (>1: makes log files correspondingly smaller)
-ts_nsteps 1
#
# (llg/checkiterationllg.c)
# (emini/checkiterationemini.c)
#
# write inp file in equilibrium (torque=max(|dM/dt|)<condinp_equil)
# 0: off
# 1: on
-condinp_equil 1
# set the following options to very high values (e.g. 1e99)
# if output should be disabled:
# write inp file if |M_current - M_lastinp| > condinp_j  (units of Ms // Hext)
-condinp_j 0.1
# write inp file in equilibrium if |M_current - M_lastinp| > condinp_j  (units of Ms // Hext)
-condinp_equil_j 0.0
# write inp file if (t_current - t_lastinp) > condinp_t (nanoseconds)
-condinp_t 1e300
# write inp file at times specified in the input file
# file format:
# <number of time values>
# <time> (nanoseconds)
# ...
-condinp_file_t_ns <filename>

##### definition of slice plane for PNG output
# (png/writedatapng.c)
#
# definition: nx*x+ny*y+nz*z = nx*vx+ny*vy+nz*vz
# slice_n (nx,ny,nz): normal vector on slice plane
# slice_p (vx,vy,vz): any point in the slice plane
# Coordinates in (dimensionless) units of the finite element mesh.
# The values must be separated with commas with no intervening spaces.
-slice_n 0,0,1
-slice_p 1e99,1e99,1e99
-slice2_n 0,0,1
-slice2_p 1e99,1e99,1e99
# select area to be drawn in PNG files based on property id
# (line number in krn file)
# 0:  plot magnetization of grains with any property id
# >0: only grain with given property id
# <0: all grains except the one with the given property id
-slice_g 0
-slice2_g 0
# image resolution:
# number of pixels on longer edge
# -> max. image size: res x res pixels
-res 50

##### definition of data sampling line
# (io/writedatadat.c)
#
# vector line_v (vx,vy,vz), point line_p (px,py,pz)
# Coordinates in (dimensionless) units of the finite element mesh.
# The values must be separated with commas with no intervening spaces.
-line_v 1,0,0
-line_p 1e99,1e99,1e99

##### save energy, average magnetization for each volume in simname.log_<pid>
# (io/writelog_pid.c)
#
# 0: off
# 1: on
-logpid 0
```

```
########################################################################
# tolerances
########################################################################

##### tolerances
# (llg/checkiterationllg.c)
#
# if (torque=max(|dM/dt|) < tol): equilibrium
-tol 1e-5
#
# if (max(|M|) > renormtol): renormalize all M on all nodes and restart PVODE
-renormtol 1e-2

##### dimensionless Landau-Lifshitz-Gilbert damping constant
# define for every grain together with material parameters in *.krn

##### exit condition
# (main.c)
#
# stop simulation if |J//Hext| < jfinal (1)
-jfinal -0.95

##### exit condition
# (main.c)
#
# stop simulation if (Mx<mfinal[0] || My<mfinal[1] || Mz<mfinal[2])
-mfinal -1e99,-1e99,-1e99

########################################################################
#
# options for solvers
#
########################################################################

########################################################################
# PETSc (Krylov subspace) solvers

# The following options are not used!
# They are only the generic options for PETSc's KSP solvers.
# However, they are available with appropriate prefixes,
# e.g. -hdemag_u1_,-hdemag_u2_,-psolve_
# as described elsewhere in this file.

# Krylov subspace solver for linear systems
# important for accuracy and speed of calculation of the demagnetizing field
#

# monitor convergence (print residual at every iteration)
#-ksp_monitor

# Maximum number of iterations (KSPSetTolerances)
# <10000>
#-ksp_max_it 10000
# Relative decrease in residual norm (KSPSetTolerances)
# <1e-05>
#-ksp_rtol 1e-05
# Absolute value of residual norm (KSPSetTolerances)
# <1e-50>
#-ksp_atol 1e-50
# Residual norm increase cause divergence (KSPSetTolerances)
#  <10000>
#-ksp_divtol 10000

#Preconditioner (PC) Options for linear systems ------------------------------
#  -pc_type Preconditioner:(one of) none jacobi pbjacobi bjacobi sor lu shell mg
#       eisenstat ilu icc cholesky asm sles composite redundant nn mat milu jacobic multilevel schur (see m
```

```
#
# default on single proc: icc (good, even better for small systems: lu - i.e. direct solver)
# default on multi-proc:  bjacobi
# direct solver:          lu (use together with "-ksp_type preonly")
#   (PETSc LU solver only available for single processor,
#   use SuperLU_DIST as a direct parallel solver)

#-pc_type icc
# apply Manteuffel shift to diagonal to force positive definite preconditioner
# (required by ICC, automatic default for PETSc >=2.2.1)
#-pc_icc_shift

#-pc_type lu
#-pc_type bjacobi

# default on multi-proc: icc
#-sub_pc_type icc
# apply Manteuffel shift to diagonal to force positive definite preconditioner
# (required by ICC, automatic default for PETSc >=2.2.1)
#-sub_pc_icc_shift

# cf. PETSc manual chapter 13: Hints for Performance Tuning
# section 13.7, p. 141: Tips for Efficient Use of Linear Solvers

########################################################################
# TAO: Energy minimization

##### distortion parameter for energy minimization (TAO)
# (emini/eminisolve.c)
#
# the magnetization on all nodes is distorted whenever the TAO solver returns
# magdist==0: do not distort
# magdist>0:  distort randomly
# magdist<0:  distort in direction MxMxH
-magdist 0.0

##### TAO methods ------------------------------------------------------------
# tao_nls - Newton's method with line search for unconstrained minimization
# tao_ntr - Newton's method with trust region for unconstrained minimization
# tao_lmvm - Limited memory variable metric method for unconstrained minimization
# tao_cg_fr - Fletcher-Reeves Nonlinear conjugate gradient method for unconstrained minimization
# tao_cg_pr - Polak-Ribiere Nonlinear conjugate gradient method for unconstrained minimization
# tao_cg_prp - Polak-Ribiere-Plus Nonlinear conjugate gradient method for unconstrained minimization
# tao_tron - Newton Trust Region method for bound constrained minimization
# tao_gpcg - Newton Trust Region method for quadratic bound constrained minimization
# tao_blmvm - Limited memory variable metric method for bound constrained minimization
# tao_kt - Formulate a bound constrained problem as a complementarity problem
# tao_bqpip - Interior point method for quadratic bound constrained minimization
# tao_ssils - Infeasible semismooth method with a linesearch for complementarity problems
# tao_ssfls - Feasible semismooth method with a linesearch for complementarity problems

#TAO solver  ---------------------------------------------------
#  -tao_method Select TAO method:(one of) tao_lmvm tao_nls tao_cg tao_bqpip tao_blmvm tao_bnls tao_tron ta
#      tao_ntr tao_gpcg tao_ssils tao_ssfls tao_asils tao_asfls tao_isils tao_kt tao_rscs tao_icp tao_fd_t
#  Limited-memory variable-metric method for unconstrained optimization
#  More-Thuente line line search options for unconstrained minimization
#  -tao_ls_maxfev <30>: max function evals in line search ()
#  -tao_ls_ftol <0.001>: tol for sufficient decrease ()
#  -tao_ls_gtol <0.99>: tol for curvature condition ()
#  -tao_ls_rtol <1e-10>: relative tol for acceptable step ()
#  -tao_ls_stepmin <1e-20>: lower bound for step ()
#  -tao_ls_stepmax <1e+20>: upper bound for step ()
#  -tao_view: view TAO_SOLVER info after each minimization has completed (TaoView)

#  -tao_fatol <1e-4>: Stop if solution within (TaoSetTolerances)
#  -tao_frtol <1e-4>: Stop if relative solution within (TaoSetTolerances)
#  -tao_catol <0>: Stop if constraints violations within (TaoSetTolerances)
```

```
#  -tao_crtol <0>: Stop if relative contraint violations within (TaoSetTolerances)
#  -tao_gatol <0>: Stop if norm of gradient less than (TaoSetGradientTolerances)
#  -tao_grtol <0>: Stop if norm of gradient divided by the function value is less than (TaoSetGradientTole
#  -tao_gttol <0>: Stop if the norm of the gradient is less than the norm of the initial gradient times (T
#  -tao_max_its <2000>: Stop if iteration number exceeds (TaoSetMaximumIterates)
#  -tao_max_funcs <4000>: Stop if number of function evaluations exceeds (TaoSetMaximumFunctionEvaluations
#  -tao_fmin <-1e+30>: Stop if function less than (TaoSetFunctionLowerBound)
#  -tao_steptol <0>: Stop if step size or trust region radius less than (TaoSetTrustRegionRadius)
#  -tao_trust0 <1e-06>: Initial trust region radius (TaoSetTrustRegionRadius)
#  -tao_unitstep: Always use unit step length (TaoCreateUnitLineSearch)
#  -tao_lmmax <5>: Maximum number of vector pairs to use in limited memory variable metric matrix (TaoSetL
#  -tao_view_hessian: view Hessian after each evaluation (None)
#  -tao_view_gradient: view gradient after each evaluation (None)
#  -tao_view_jacobian: view jacobian after each evaluation (None)
#  -tao_view_constraints: view constraint function after each evaluation (None)
#  -tao_cancelmonitors: cancel all monitors hardwired in code (TaoClearMonitor)
#  -tao_monitor: Use the default convergence monitor (TaoSetMonitor)
#  -tao_smonitor: Use short monitor (None)
#  -tao_vecmonitor: Plot solution vector at each iteration (TaoVecViewMonitor)
#  -tao_vecmonitor_update: plots step direction at each iteration (TaoVecViewMonitorUpdate)
#  -tao_xmonitor: Use graphics convergence (TaoPetscXMonitor)

# most useful options and default settings for magpar
-tao_method tao_lmvm
-tao_ls_ftol 1e-20
-tao_ls_rtol 1e-01
-tao_ls_gtol 0.99
-tao_fatol 1e-4
-tao_frtol 1e-4
-tao_max_its 500
-tao_monitor

#  -tao_lmm_vectors: number of vectors stored for the Hessian approximation
-tao_lmm_vectors 100

# more information about these options can be found in the TAO manual

#########################################################################
# PVODE: LLG time integration

###### PETSc standard options for timesteppers
# (llg/mytscreatepvode.c)
# (main.c)
#
-ts_init_time <0>: Initial time (TSSetInitialTime) (unit: ns)
-ts_max_time <5>: Time to run to (TSSetDuration) (unit: ns)
-ts_dt <0.020944>: Initial time step (TSSetInitialTimeStep)
#
-ts_pvode_type <bdf> (one of) bdf adams
-ts_pvode_atol <1e-06>: Absolute tolerance for convergence (TSPVodeSetTolerance)
-ts_pvode_rtol <1e-06>: Relative tolerance for convergence (TSPVodeSetTolerance)
-ts_pvode_linear_tolerance <0.05>: Convergence tolerance for linear solve (TSPVodeSetLinearTolerance)

##### additional PVode parameters
# (llg/mytscreatepvode.c)
#
# maximum Krylov dimension
-maxl 300
# minimum absolute value of step size allowed (ns)
-mintimestep 0
# maximum absolute value of step size allowed (ns)
-maxtimestep 1e99
# maximum lmm order to be used by the solver
# Default (=max.) = 12 for ADAMS, 5 for BDF
-maxorder 2
# set preconditioning type
# 0: none
```

```
# 1: jacobi
# 2: band-block-diagonal (not implemented, yet)
-precon 1

# choose linear solver for psolve
# (llg/precond.c)
# all standard ksp options (see above, just add prefix "psolve_")
#
# gmres and bcgs are usually a good choice
# for complete list see PETSc Krylov solvers above
-psolve_ksp_type gmres
#
# set tolerances for psolve
-psolve_ksp_atol 1e-7
-psolve_ksp_rtol 0.01
-psolve_ksp_divtol 100

# more information about these options can be found in the PVODE manual
```

16.2 allopt_log.txt

Additional configuration file with PETSc internal logging/info/diagnostic options:

(Note, that some of these options (especially "-tr∗") may slow down the simulations considerably!)

doc/allopt_log.txt:

```
########################################################################
#
# magpar configuration file: allopt_log.txt
#
# PETSc internal logging/info/diagnostic options
#
# for PETSc version 2.3.2
#
########################################################################

-help         # unlisted but active

##### memory checking
#
-malloc_dump
-malloc
-malloc_log   # unlisted but active
-malloc_info  # unlisted but active
# -mallocinfo ignored
-malloc_debug

-fp_trap

##### print more info during initialization and solution
#
-info
-log_trace

##### enable profiling of important magpar functions
# even if -info is not defined/disabled
# implemented in macro MagparFunctionProfBegin in griddata.h
#
-profile

##### print info at end of run
#
-memory_info
```

```
-options_table
-options_left
-get_total_flops
-log_summary

##### shortcuts for analyzing malloc information
#
# cat %f | grep bytes | grep -v "src/" > %f.b
# cat %f | grep bytes | colrm 1 4 | sort -n > %f.c
# cat %f | grep bytes | grep "src/" > %f.d
# grep mallocs %f | grep -v " 0 mallocs" | grep -v "is 0" | less


##### Options for all PETSc programs:
# -on_error_abort: cause an abort when an error is detected. Useful
#       only when run in the debugger
# -on_error_attach_debugger [gdb,dbx,xxgdb,ups,noxterm]
#       start the debugger in new xterm
#       unless noxterm is given
# -start_in_debugger [gdb,dbx,xxgdb,ups,noxterm]
#       start all processes in the debugger
# -on_error_emacs <machinename>
#    emacs jumps to error file
# -debugger_nodes [n1,n2,..] Nodes to start in debugger
# -debugger_pause [m] : delay (in seconds) to attach debugger
# -stop_for_debugger : prints message on how to attach debugger manually
#                      waits the delay for you to attach
# -display display: Location where graphics and debuggers are displayed
# -no_signal_handler: do not trap error signals
# -mpi_return_on_error: MPI returns error code, rather than abort on internal error
# -fp_trap: stop on floating point exceptions
#           note on IBM RS6000 this slows run greatly
# -malloc_dump <optional filename>: dump list of unfreed memory at conclusion
# -malloc: use our error checking malloc
# -malloc no: don't use error checking malloc
# -mallocinfo: prints total memory usage
# -malloc_debug: enables extended checking for memory corruption
# -options_table: dump list of options inputted
# -options_left: dump list of unused options
# -options_left no: don't dump list of unused options
# -tmp tmpdir: alternative /tmp directory
# -shared_tmp: tmp directory is shared by all processors
# -not_shared_tmp: each processor has separate tmp directory
# -memory_info: print memory usage at end of run
# -get_total_flops: total flops over all processors
# -log[_all _summary]: logging objects and events
# -log_trace [filename]: prints trace of all PETSc calls
# -info <optional filename>: print informative messages about the calculations
# -v: prints PETSc version number and release date
# -options_file <file>: reads options from file
# -petsc_sleep n: sleeps n seconds before running program
#
#------Additional PETSc component options--------
# -log_summary_exclude: <vec,mat,pc.ksp,snes>
# -info_exclude: <null,vec,mat,pc,ksp,snes,ts>
#
#Options database options ------------------------------------------------
#  -options_monitor <stdout>: Monitor options database (PetscOptionsSetMonitor)
#  -options_cancelmonitors: Cancel all options database monitors (PetscOptionsClearMonitor)
#
#Vector Options ----------------------------------------------------
#  -vec_view: Print vector to stdout (VecView)
#  -vec_view_matlab: Print vector to stdout in a format Matlab can read (VecView)
#  -vec_view_socket: Send vector to socket (can be read from matlab) (VecView)
#  -vec_view_binary: Save vector to file in binary format (VecView)
#
#Options for SEQAIJ matrix ----------------------------------------------
```

```
#  -mat_no_unroll: <TRUE> Do not optimize for inodes (slower) ((null))
#  -mat_no_inode: <TRUE> Do not optimize for inodes (slower) ((null))
#  -mat_inode_limit <5>: Do not use inodes larger then this value ((null))
#Matrix Options ---------------------------------------------------
#  -mat_view_info: Information on matrix size (MatView)
#  -mat_view_info_detailed: Nonzeros in the matrix (MatView)
#  -mat_view: Print matrix to stdout (MatView)
#  -mat_view_matlab: Print matrix to stdout in a format Matlab can read (MatView)
#  -mat_view_socket: Send matrix to socket (can be read from matlab) (MatView)
#  -mat_view_binary: Save matrix to file in binary format (MatView)
#  -mat_view_draw: Draw the matrix nonzero structure (MatView)
#
#Graphics (PetscDraw) Options -------------------------------------------------
#  -draw_type Type of graphical output:(one of) x null ps (PetscDrawSetType)
#  -nox: Run without graphics (None)
#
#Preconditioner (PC) Options --------------------------------------------------
#  -pc_type Preconditioner:(one of) none jacobi pbjacobi bjacobi sor lu shell mg
#      eisenstat ilu icc cholesky asm ksp composite redundant nn mat fieldsplit tfs (PCSetType)
#  ICC Options
#  -pc_factor_levels <0>: levels of fill (PCFactorSetLevels)
#  -pc_factor_fill <1>: Expected fill in factorization (PCFactorSetFill)
#  -pc_factor_mat_ordering_type Reorder to reduce nonzeros in ICC:(one of) natural nd 1wd rcm qmd rowlengt
#  -pc_factor_shift_nonzero: Shift added to diagonal (PCFactorSetShiftNonzero)
#  -pc_factor_shift_nonzero <0>: Shift added to diagonal (PCFactorSetShiftNonzero)
#  -pc_factor_shift_positive_definite: Manteuffel shift applied to diagonal (PCFactorSetShift)
#  -pc_factor_zeropivot <1e-12>: Pivot is considered zero if less than (PCFactorSetZeroPivot)
#Krylov Method (KSP) Options --------------------------------------------------
#  -ksp_type Krylov method:(one of) cg cgne stcg richardson chebychev gmres tcqmr bcgs
#      bcgsl cgs tfqmr cr lsqr preonly qcg bicg fgmres minres symmlq lgmres lcd (KSPSetType)
#  -ksp_max_it <10000>: Maximum number of iterations (KSPSetTolerances)
#  -ksp_rtol <1e-05>: Relative decrease in residual norm (KSPSetTolerances)
#  -ksp_atol <1e-50>: Absolute value of residual norm (KSPSetTolerances)
#  -ksp_divtol <10000>: Residual norm increase cause divergence (KSPSetTolerances)
#  -ksp_converged_use_initial_residual_norm: Use initial residual residual norm for computing relative cor
#  -ksp_converged_use_min_initial_residual_norm: Use minimum of initial residual norm and b for computing
#  -ksp_knoll: <FALSE> Use preconditioner applied to b for initial guess (KSPSetInitialGuessKnoll)
#  -ksp_norm_type <preconditioned> (choose one of) none preconditioned unpreconditioned natural
#  -ksp_diagonal_scale: Diagonal scale matrix before building preconditioner (KSPSetDiagonalScale)
#  -ksp_diagonal_scale_fix: Fix diagonaled scaled matrix after solve (KSPSetDiagonalScaleFix)
#  -ksp_constant_null_space: Add constant null space to Krylov solver (KSPSetNullSpace)
#  -ksp_converged_reason: Print reason for converged or diverged (KSPSolve)
#  -ksp_cancelmonitors: Remove any hardwired monitor routines (KSPClearMonitor)
#  -ksp_monitor <stdout>: Monitor preconditioned residual norm (KSPSetMonitor)
#  -ksp_vecmonitor: Monitor solution graphically (KSPSetMonitor)
#  -ksp_truemonitor <stdout>: Monitor preconditioned residual norm (KSPSetMonitor)
#  -ksp_singmonitor <stdout>: Monitor singular values (KSPSetMonitor)
#  -ksp_smonitor <stdout>: Monitor preconditioned residual norm with fewer digits (KSPSetMonitor)
#  -ksp_xmonitor: Monitor graphically preconditioned residual norm (KSPSetMonitor)
#  -ksp_xtruemonitor: Monitor graphically true residual norm (KSPSetMonitor)
#  Pick at most one of -------------
#    -ksp_left_pc: Use left preconditioning (KSPSetPreconditionerSide)
#    -ksp_right_pc: Use right preconditioning (KSPSetPreconditionerSide)
#    -ksp_symmetric_pc: Use symmetric (factorized) preconditioning (KSPSetPreconditionerSide)
#  -ksp_compute_singularvalues: Compute singular values of preconditioned operator (KSPSetComputeSingularV
#  -ksp_compute_eigenvalues: Compute eigenvalues of preconditioned operator (KSPSetComputeSingularValues)
#  -ksp_plot_eigenvalues: Scatter plot extreme eigenvalues (KSPSetComputeSingularValues)
#  KSP GMRES Options
#  -ksp_gmres_restart <30>: Number of Krylov search directions (KSPGMRESSetRestart)
#  -ksp_gmres_haptol <1e-30>: Tolerance for exact convergence (happy ending) (KSPGMRESSetHapTol)
#  -ksp_gmres_preallocate: Preallocate Krylov vectors (KSPGMRESSetPreAllocateVectors)
#  Pick at most one of -------------
#    -ksp_gmres_classicalgramschmidt: Classical (unmodified) Gram-Schmidt (fast) (KSPGMRESSetOrthogonaliza
#    -ksp_gmres_modifiedgramschmidt: Modified Gram-Schmidt (slow,more stable) (KSPGMRESSetOrthogonalizatio
#  -ksp_gmres_cgs_refinement_type <REFINE_NEVER> (choose one of) REFINE_NEVER REFINE_IFNEEDED REFINE_ALWAY
#  -ksp_gmres_krylov_monitor: Plot the Krylov directions (KSPSetMonitor)
#  -ksp_view: View linear solver parameters (KSPView)
```

16.3 allopt_ret.txt

Deprecated and removed options:
doc/allopt_ret.txt:

```
########################################################################
#
# removed magpar options
#
########################################################################

# start in debugger (the debugger ddd must be in your path)
#-ddd

# replaced by -nslicepropser and -nsliceproppar
#-nsliceprop

# Different implementations of the external field can now be active
# simultaneously. By default all external fields are off. Just set the
# appropriate options (e.g. -hextini, -hext_cu, etc.) to activate them.
#-hextshape

# GeomView output has been removed
#-offdata

# -ksp_* replaced by -hdemag_u{1,2}_ksp_*
-ksp_type
#-ksp_atol
#-ksp_rtol

# set damping parameter in *.krn for individual materials
-alpha
```

Chapter 17

Output Files of Simulations

Sections:

- project.INP.X.png
- project.INP.d
- project.INP.datmsh
- project.INP.fedat
- project.INP.felog
- project.INP.femsh
- project.INP.gz
- project.INP.inp
- project.log
- project.log_pvode
- project.log_PID
- stdout
- project.9999.∗

In the following "INP" stands for a four-digit integer.

"project" stands for any project name defined as "-simName" in the configuration file allopt.txt: simulation parameters .

17.1 project.INP.X.png

The .png files are graphics files, which show a color coded snapshot of the magnetization. The slice plane through the model is defined in the configuration file allopt.txt: simulation parameters . The x-,y-, and z-component of the magnetization are stored in ∗.0.png, ∗.1.png, and ∗.2.png, respectively. It is color coded from red (M=-1) to blue (M=+1).

These graphics files can be viewed with any picture viewer or graphical web browser.

17.2 project.INP.d

The .d files contain the magnetization (or whatever is implemented in writedatadat.c) along the "sampling line". The "sampling line" is defined in the configuration file allopt.txt: simulation parameters .

17.3 project.INP.datmsh

This file contains the Cartesian coordinates of the measurement points along the sampling line.

```
point_id  dist  x  y  z
```

- **point_id:** simple id for numbering the measurement points
- **dist:** distance from the first measurement point (in units of the FE mesh)
- **x**, **y**, and **z:** Cartesian coordinates

17.4 project.INP.fedat

This files contains some mesh related data in UCD format. The "mkinp.sh" script (cf. UCD/inp Files) can be used to create a complete UCD file, which can be visualized as explained in section Postprocessing .

17.4.1 node data

- **id:** node id
- **vert_vol:** volume assigned to the node
- **proc:** processor to which the node is assigned

17.4.2 element data

- **id:** element id
- **proc:** processor to which the element is assigned
- **vol:** volume
- **prop:** property id
- **qual:** quality factor (cf. elevertvol.c)
- **a_x**, **a_y**, and **a_z:** Cartesian components of the anisotropy axis
- **K1:** first magnetocrystalline anisotropy constant (reduced units)
- **K2:** second magnetocrystalline anisotropy constant (reduced units)
- **Js:** saturation polarization (reduced units)
- **A:** exchange constant (reduced units)

17.5 project.INP.felog

This file contains a lot of useful information about the finite element mesh:

- **simname:** project name
- **n_vert:** number of vertices
- **n_ele:** number of elements
- **n_fac:** number of triangular faces (all, also interior faces)
- **n_vert_bnd:** number of vertices on the boundary
- **n_ele_bnd:** number of elements on the boundary
- **n_bnd_fac:** number of triangular faces on the boundary
- **n_prop:** number of property ids (grains)

- **bounding box:** the smallest rectangular box surrounding the model
- **elevol:** element volumes: minimum, maximum, ratio, element ids, average, total volume
- **vertvol:** vertex volumes
- **edge_len:** minimum, maximum, and average of all edges (connection two vertices) in the finite element mesh
- **elequal:** minimum, maximum, and average of element quality factors
- **volume by property id:**
- **average magnetization:** weighted with element volume

17.6 project.INP.femsh

This file contains the finite element mesh in `UCD format`. Since the mesh does not change during the simulation, it is stored only once in this file and used by the "mkinp.sh" (cf. UCD/inp Files) script to create complete UCD files.

17.7 project.INP.gz

These files contain various simulation data (magnetization distribution, magnetostatic, anisotropy, exchange, external field) in `UCD format`. In order to save disk space (which becomes important for large models and long simulations with many output files) the mesh data is not included (it is stored only once in project.INP.femsh) and the remaining data are compressed using zlib (gzip format) (if this option has been compiled in). The "mkinp.sh" script (cf. UCD/inp Files) can be used to create a complete UCD file, which can be visualized as explained in section Postprocessing .

The following data are stored in theses files:

- **M_x :** x component of magnetization vector (dimensionless: magnetization vector normalized to 1)
- **M_y :** y component of magnetization vector (dimensionless: magnetization vector normalized to 1)
- **M_z :** z component of magnetization vector (dimensionless: magnetization vector normalized to 1)

analogously the Cartesian components of

- **Hdemag:**magnetostatic field (T)
- **Hani_x :**magnetocrystalline anisotropy field (T)
- **Hexch_x :**exchange field (T)

  or
- **Hexchani_x:**exchange+magnetocrystalline anisotropy field (T)
- **Hext_x :**external field (T)

The structure and contents can change depending on compilation and simulation options. E.g., if the magnetostatic field is switched of in the allopt.txt file then the divergence and scalar magnetic potential are omitted. If the magpar executable has been compiled with the EXCH option, then the exchange and magnetocrystalline anisotropy field are saved separately.

17.8 project.INP.inp

Complete inp files (including the mesh information) are created by magpar if zlib compression is not supported (magpar is not linked with the zlib library) or magpar is compiled in the Cygwin environment under Windows. These complete inp files can be immediately visualized as explained in section Postprocessing .

Otherwise these complete inp files can be generated from project.INP.femsh and project.INP.gz as described in UCD/inp Files.

17.9 project.log

This log file contains the most important simulation data in a columnar format. magpar always appends new data to the end of the file if it does already exit. This useful feature is used in example mumag3: mumag standard problem #3 and sphere_sw: Stoner-Wohlfarth behavior .

columns:

- **eq:** >=1 if the system has reached equilibrium, i.e. mx|dM/dt| is smaller than the threshold -tol (defined in allopt.txt: simulation parameters) (see below); if the system is in equilibrium for successive iterations this counter is increased; it is reset to zero as soon as the system is not in equilibrium
- **inp:** number of the corresponding UCD (.inp), .png, .d output files
- **time:** simulated time
- **Hext:** external field (sum of contributions from all external field sources)
- **Etot:** total energy
- **J//Hext:** reduced average polarization parallel to the external field
- **Mx**, **My**, and **Mz:** Cartesian components of the average magnetization (not weighted with Js!)
- **Edem:** magnetostatic (=demag=stray field) energy
- **Exchani:** exchange+anisotropy energy
- **Eext:** Zeeman energy
- **devNorm:** maximum deviation of the norm of the reduced magnetization |M| on any node from 1.0
- **mx|dM/dt|:** maximum of |dM/dt|, the difference quotient of M with respect to the last time step dt ("torque") on any node
- **timestep:** time step
- **tCPU:** elapsed time ("wall clock time")
- **CPUtstep:** CPU time ("wall clock time") required for the last time step

17.10 project.log_pvode

SUNDIALS/PVODE data:

- **time:** simulated time

- **nstep:** number of time steps
- **qu:** integration order
- **nfe:** number of calls to the user's f function (i.e. RHSfunction, calc_dMdt)
- **nni:** number of (non-linear) Newton iterations
- **ncfn:** number of nonlinear convergence failures
- **nli:** number of linear iterations of CVSPGMR solver
- **npe:** number of preconditioner evaluations
- **nps:** number of calls to PSolve
- **ncfl:** number of linear convergence failures

17.11 project.log_PID

These log files list the total energy, average magnetization and field, magnetostatic, anisotropy, exchange and Zeeman energy separately for each volume with the property id PID.

17.12 stdout

A lot of information is sent to stdout during the simulation:

- magpar version, library versions, included features
- settings of the most important parameters
- scaling factors for field, time, and energy
- material parameters, anisotropy field, exchange lengths, Larmor frequencies
- size of the boundary matrix
- sampling line and slice plane data
- solver output
- exit conditions
- etc.

See allopt.txt: simulation parameters and the PETSc manual for more.

17.13 project.9999.*

Every 2000 seconds (33 minutes) a full set of output files is written. This allows to check the progress of the simulation even if the cond_inp parameters in allopt.txt: simulation parameters have been set to disable any output. Even more important, it is a checkpointing type behavior since one can restart the simulation from the UCD file (project.INP.gz) if the program or one of the executing machines have crashed.

Chapter 18

Examples

Sections:

The examples described in this section can be downloaded from the magpar download page. In order to run them, an executable of magpar is required, which can be created as described in the Installation guide. The executable should be copied into the subdirectory, which contains the example to be run. Finally, it might be necessary to update the (very simple) "run" scripts and modify the "prg" variable, which contains the name of the magpar executable.

18.1 Download

Download and extract the examples:

```
# install examples package parallel to magpar source package
# (this is required for the local html documentation
# to display the figures properly)
cd $MAGPAR_HOME/../
wget http://www.magpar.net/static/magpar/download/magpar-0_9_ex.tar.gz
tar xzvf magpar-0_9_ex.tar.gz
```

18.2 Sphere

These examples consist of the following files:

- **allopt.txt: simulation parameters**
- **run**
  shell script to run magpar (not required)
- **sphere.krn**
  material parameters: zero anisotropy and $J_S = 1$ T ($M_S = J_S / mu_0 = 795774.72$ A/m) have been chosen.

- **sphere.out**

  symbolic link to one of the finite element meshes:

- **sphere_coarse.out**

  coarse finite element mesh (849 nodes, 3945 tetrahedral elements) generated by Patran, exported in "neutral file" format

- **sphere_fine.out**

  coarse finite element mesh (2016 nodes, 10142 tetrahedral elements) generated by Patran, exported in "neutral file" format

Using this spherical model we can check the simulation results against several analytical calculations:

- For a homogeneous magnetization distribution, the demagnetizing field should be homogeneous within the sphere.
- The strength of the demagnetizing field (=strayfield) should be H_{demag} =-M_S /3 .
- The magnetostatic energy should be mu_0 M_S ^{2} /6.
- If we tilt the magnetization against the external field and do the dynamic time integration of the Landau-Lifshitz-Gilbert equation using PVODE, we can check the precession frequency (Larmor frequency).
- We can check the Stoner-Wohlfarth behavior by calculating demagnetization curves using the static energy minimization (which is much faster than the dynamic time integration - even with large damping).

18.2.1 sphere_demag: Demagnetizing field

```
######################################################################
# magpar configuration file: allopt.txt
######################################################################

-simName sphere

-meshtype 0         # read finite element mesh from Patran neutral file
```

After running the simulation in "sphere_demag" the demag field can be found in "sphere.0001.gz". This file contains the "second half" of the inp file (the data section - not the finite element mesh, cf. UCD/inp Files). In the first column there is the id of the node, for which the data are given. In the second, third, and fourth column, the Cartesian components of the magnetization are given. In the fourth column the divergence of M is given (including surface charge contributions). The fifth and sixth column give the contributions u_1 and u_2 to the total magnetostatic potential u, which can be found in the seventh column. In the following three columns the Cartesian components of the demagnetizing and in the next three the (combined) exchange+anisotropy field is given. The z component of the demagnetizing field is approximately $H_{demag,z}$ =-1/3 M_z , as expected for a homogeneously magnetized sphere. The x and y component should be zero, but they are not exactly zero due to numerical errors and the coarse model of the sphere. (The sphere is approximated by a polyhedron.)

The strayfield energy should be

$$E_{\mathrm{demag}} = -\frac{\mu_0}{2}\mathbf{M}\cdot\mathbf{H}_{\mathrm{demag}} = -\frac{\mu_0}{2}\mathbf{M}\cdot(-\frac{1}{3}\mathbf{M}) = \frac{\mu_0}{6}M_s^2 = 132629.12\ \mathrm{J/m^3}$$

The simulation gives 1.311535e+05 J/m^3 (see sphere.log) for the coarse model of the sphere.

We can improve the calculation by using a finer finite element mesh. This can be easily achieved by globally refining the finite element mesh. The refinement scheme implemented in magpar splits each tetrahedron into 8 smaller tetrahedra. Thus, after every refinement step we have 8 times as many elements and (approximately) 8 times as many nodes.

You can either modify the "-refine" option in allopt.txt: simulation parameters or override the setting in this file by a command line argument:

```
magpar.exe -refine 1
```

refine	nodes	elements	E_{demag} (J/m^3)
0	849	3945	1.311535e+05
1	5988	31560	1.321752e+05

If you want to try a second refinement you will need about 300 MB of memory, because the number of nodes in the finite element mesh increases to 44919 with 5538 nodes on the boundary, which gives a boundary matrix of about 233 MB.

For the fine finite element mesh (sphere_fine.out) we find the following results:

```
rm sphere.out
ln -s ../sphere_fine.out sphere.out
magpar.exe
magpar.exe -refine 1
```

refine	nodes	elements	E_{demag} (J/m^3)
0	2016	10142	1.318847e+05
1	14796	81136	1.324505e+05

If you want to try a second refinement you will need about 1 GB of memory, because the number of nodes in the finite element mesh increases to 113219 with 9970 nodes on the boundary, which gives a boundary matrix of about 758 MB.

18.2.2 sphere_larmor: Larmor precession

For the simulation of the Larmor precession the following parameters have been updated:

```
#########################################################################
# magpar configuration file: allopt.txt
#########################################################################

-simName sphere

-meshtype 1          # read finite element mesh from AVS inp file

-size 10e-9          # set length scaling factor to 10 nm

-init_mag 4          # start with a tilted initial magnetization

-mode 0              # select LLG time integration (PVode solver)

-demag 0             # switch off demagnetizing field

-hextini 1000        # apply a homogeneous external field of 1000 kA/m

-ts_max_time 0.03    # stop the simulation after 0.03 ns
```

The damping constant alpha is set to 0.0 in sphere.krn.

The simulation starts with $M_X/M_S = 5.773503e\text{-}01$. As the magnetization precesses around the external field, M_X decreases, increases, and decreases again until it reaches its initial value again. This corresponds to one complete precession around the external field and it is completed after about 0.028435 ns.

The analytical calculation gives:

$$f_{\text{Larmor}} = \frac{\gamma}{2\pi}|H_{\text{ext}}| = 35.176\ \text{GHz} \quad , \quad \gamma = \mu_0 * g * |e|/(2 * m_e) = 2.210173\text{e}5\ \text{m/As}$$

$$\tau_{\text{Larmor}} = 1/f_{\text{Larmor}} = 0.028428477\ \text{ns}$$

Note, that the warning messages

```
...
Warning: Etot increased at t=0.0133301 ns by 1.10144e-06 (-1.90776e-12*Etot) from -577350 to -577350 J/m^3
Warning: Etot increased at t=0.0137142 ns by 2.33329e-07 (-4.04138e-13*Etot) from -577350 to -577350 J/m^3
...
```

indicate, that the total energy is not perfectly conserved. At every timestep there is a relative error of about 1e-12. Moreover, the magnetization does not stay perfectly homogeneous, which becomes apparent in the (slightly) increasing exchange energy.

The finite element mesh of this example has been generated using Gmsh and converted using gmsh: gmsh-toucd.py (see also Preprocessing).

18.2.3 sphere_sw: Stoner-Wohlfarth behavior

Finally we can assume some magnetocrystalline anisotropy and calculate the switching field, if the external field is applied at different angles.

For a magnetocrystalline anisotropy constant of $K_1 = 1e5\ J/m^3$ and $J_S = 1\ T$ we get an anisotropy field of $H_{ani} = 2K_1/J_S = 200\ kA/m$.

For this simulation the following options have been modified:

```
########################################################################
# magpar configuration file: allopt.txt
########################################################################

-simName sphere

-meshtype 0          # read finite element mesh from Patran neutral file

-size 10e-9          # set length scaling factor to 10 nm

-mode 1              # select energy minimization using TAO

-demag 0             # switch off demagnetizing field

-hextini -90.0       # apply a homogeneous external field of -90 kA/m
-htheta 0.1          # apply homogeneous external field at 0.1 rad from z-axis
-hstep -1.0          # change field amplitude in steps of -1 kA/m
-hfinal -3000        # final field value -3000 kA/m

-condinp_equil 0     # do not save output files when equilibrium is reached

-jfinal -0.1         # stop simulation if J//Hext < -0.1

-tao_fatol 0.0       # set absolute tolerance to 0 (disabled)
-tao_frtol 1e-12     # set strict relative tolerance
```

The results are summarized in the following graph:

Stoner-Wohlfarth behavior of a spherical single domain particle

18.2.4 sphere_cubic: Single domain particle with cubic anisotropy

In this example we can compare the anisotropy energy of a given homogeneous magnetization distribution with the analytical result. The cubic anisotropy energy is given by [1]

$$E_{ani} = K_1 \left(m_x^2 m_y^2 + m_x^2 m_z^2 + m_y^2 m_z^2\right) + K_2 m_x^2 m_y^2 m_z^2 =$$

$$= K_1 \left(\sin^2\theta\cos^2\phi\sin^2\theta\sin^2\phi + \sin^2\theta\cos^2\phi\cos^2\theta + \sin^2\theta\sin^2\phi\cos^2\theta\right) + K_2 \sin^2\theta\cos^2\phi\sin^2\theta\sin^2\phi\cos^2\theta$$

with the magnetocrystalline anisotropy constants K_1 and K_2 .

We just change the parameter for the initial magnetization

```
########################################################################
# magpar configuration file: allopt.txt
########################################################################

-simName sphere

-meshtype 0          # read finite element mesh from Patran neutral file

-init_mag 6          # set magnetization in x-z plane to...
-init_magparm 0.785 # theta = 0.785 rad = 45 deg (from z-axis)

-demag 0             # switch off demagnetizing field
```

to set the magnetization in the x-z plane to a desired angle theta (defined by `-init_magparm` measured from the z-axis). It is assumed that the axes of the cubic lattice coincide with Cartesian coordinate system (theta=phi=psi=0). In general the cubic axes are defined by the `Euler angles` in project.krn: material properties.[2,3]

The `run` script completes a series of 20 simulations which vary the angle theta between 0 and 180 deg. The following figure shows the results for K_1 =4.6e5 J/m^3 , K_2 =1.5e4 J/m^3 , which are in perfect agreement with the analytical calculation.

Cubic anisotropy energy as a function of polar angle theta

[1] L. W. McKeehan, "Ferromagnetic Anisotropy in Nickel-Cobalt-Iron Crystals at Various Temperatures", `Phys. Rev. 51 (1937) 136-139`. [2] Eric W. Weisstein. `Euler Angles`. From `MathWorld` - A `Wolfram` Web Resource.

[3] `Euler angles` in `Wikipedia`.

18.3 iface: Domain wall pinning

For yet another problem, we have an analytical result to compare with: The pinning field of a Bloch wall at a perfect planar interface (e.g. a grain boundary). The analytical solution can be found in Ref. [4].

They find:

$$H_{\mathrm{pin}} = \frac{2K_1^{\mathrm{II}}}{J_{\mathrm{s}}^{\mathrm{II}}} \frac{1-\varepsilon_A\varepsilon_K}{(1+\sqrt{\varepsilon_A\varepsilon_J})^2} \quad ,$$

where

$$\varepsilon_J = \frac{J_{\mathrm{s}}^{\mathrm{I}}}{J_{\mathrm{s}}^{\mathrm{II}}} \quad , \quad \varepsilon_A = \frac{A^{\mathrm{I}}}{A^{\mathrm{II}}} \quad , \quad \varepsilon_K = \frac{K_1^{\mathrm{I}}}{K_1^{\mathrm{II}}}$$

and (I) denotes the material parameters of the softer material and (II) those of the harder material.

The simulation is initialized with a domain wall and an external field moves the domain wall towards the interface, where it gets pinned. As the external field increases the Bloch wall is more and more forced into the "harder material" until it depins and propagates further through the "harder material".

This example consists of

- **allopt.txt: simulation parameters**
- **iface.gid**
  the GiD model
- **iface.inp**
  the INP file, which has been generated by GiD
- **iface.krn**
  material parameters: Sm_2 Co_{17} and $SmCo_5$
- **run**
  shell script to run magpar (not required)

For the material parameters given in iface.krn, the analytical formulas above give a pinning field of H_{pin} =1933 kA/m.

```
#########################################################################
# magpar configuration file: allopt.txt
#########################################################################

-simName iface

-refine 2           # refine mesh 2 times

-size 13e-9         # set length scaling factor to 13 nm

-init_mag -9        # initialize magnetization with Bloch wall and reverse
-init_magparm 0.7

-mode 1             # select energy minimization using TAO

-demag 0            # switch off demagnetizing field

-hextini -1700      # apply a homogeneous external field of 12 kA/m
-hstep   -5         # change field amplitude in steps of -5 kA/m
-hfinal  -5000      # final field value -5000 kA/m

-condinp_equil 0    # do not save output files when equilibrium is reached
-condinp_t  1e99    # do not save output files at regular time intervals

-slice_n 0,0,1      # slice plane for png files: x-y-plane
-slice_p 0,0,0.5

-line_v 1,0,0       # set measurement line parallel to x-axis
-line_p 0,0.5,0.5

-logpid 1           # save average magnetization in output files (*.log_XXX)

-tao_fatol 1e-10    # set absolute tolerance for energy minimizer TAO
-tao_frtol 1e-10    # set relative tolerance for energy minimizer TAO
```

The "run" script automatically varies the mesh density using the "-refine" option. It runs three simulations starting with a very coarse mesh, which is refined once and twice in the following runs. One can observe, that the simulation result gets closer to the analytical result as the mesh density increases.

[4] H. Kronmüller, D. Goll, "Micromagnetic theory of the pinning of domain walls at phase boundaries", Physica B: Condensed Matter, 319 (2002) 122-126.

18.4 mumag3: mumag standard problem #3

mumag standard problem #3 asks to calculate the single domain limit of a cubic magnetic particle. For a certain size L the total energy of the so-called flower state on the one hand, and the vortex or curling state on the other hand is equal. The material parameters have been chosen according to the definition on the mumag website. Just the anisotropy axis has been rotated by 90 deg into the x-axis to make the initialization of the vortex state easier.

With M_S =1 T and A=1e-11 J/m we find K_m =1/2$*$$mu_0$ $*M_S$ ^{2} =3.9788736e5 and set K_1 =0.1$*K_m$ =3.9788736e4 J/m^3 . l_{ex} =$(A/K_m)^{1/2}$ =5.013256 nm.

```
######################################################################
# magpar configuration file: allopt.txt
######################################################################

-simName mumag3

-meshtype 0          # read finite element mesh from Patran neutral file

-refine 4            # refine mesh 4 times

-size 30e-9          # set length scaling factor to 30 nm

-init_mag 1          # initialize magnetization parallel to x-axis

-mode 1              # select energy minimization using TAO

-slice_n 0,0,1       # slice plane for png files: x-y-plane
-slice_p 0,0,0

-line_v 0,0,1        # set measurement line parallel to z-axis
-line_p 0,0,0

-tao_fatol 1e-8      # set absolute tolerance for energy minimizer TAO
-tao_frtol 0.0       # set relative tolerance for energy minimizer TAO
```

In this example we use the global refinement feature of magpar. The neutral file "mumag3.out" describes a very simple finite element discretization of a cube with 8 nodes and 5 tetrahedral finite elements. The option "-refine 4" makes magpar refine this mesh 4 times, which gives $5*8^4$ =5$*$4096=20480 elements and 4233 nodes.

The shell script "run" should be used to run this example. It takes advantage of some useful features of magpar (rather PETSc) to run several simulations with different parameters in one go:

It contains two loops. One which selects the initial magnetization:

- `-init_mag 1`
  homogeneous magnetization in x-direction
- `-init_mag 7`
  approximate flower state

And another loop which varies the size of the cube from 30 nm to 50 nm.

magpar is then called with the command line options "-init_mag" and "-size", which override the settings in the allopt.txt: simulation parameters configuration file. All simulations append their results to the existing

"mumag3.log" file. In order to store the setting of the initial magnetization and size a simple echo command adds some comment lines to the log file just before a new simulation is started.

Finally, one just has to extract the settings of the size and total energy in equilibrium to create a plot, which shows the total energy as a function of size for flower and vortex state. With the parameters in this example we find the following result:

Total energy as a function of size for mumag standard problem #3

18.5 mumag3b: mumag standard problem #3 with 2 cubes

This example is very similar to the previous example mumag3: mumag standard problem #3. However, in this example the geometry consists of two disjoint cubes, where the material parameters of one are set to air/vacuum, while the other one is magnetic.

```
  0.0       0 0.0          0.0  0.0  0.0      0.1   uni    # cube1: air
  1.5707963 0 3.9788736e4 0.0  1.00 1.00E-11 0.1   uni    # cube2: mumag standard problem #3 http://www.ct
#
# theta phi   K1       K2      Js   A         alpha psi    # parameter
# (rad) (rad) (J/m^3)  (J/m^3) (T)  (J/m)     (1)   (rad)  # units
```

The geometry and finite element mesh of this example have been generated with Gmsh and converted with gmsh: gmshtoucd.py (see Preprocessing).

This example also demonstrates the use of the option "-logpid 1" in allopt.txt and how non magnetic volumes can be included in simulations with magpar (e.g. to calculate the magnetostatic field surrounding a magnet).

18.6 nanodot: Nanodot

In case a simulation was interrupted or it should be continued at a certain point, it can be restarted from any UCD file (project.INP.gz).

For example, we restart the simulation of the magnetization reversal of a magnetic nanodot.

We need the following files:

- **allopt.txt: simulation parameters**
- **run**

  shell script to run magpar (not required)
- **nanodot.krn**

  material parameters: permalloy
- **nanodot.0050.inp**

  UCD file with the magnetization distribution of some magnetization state in a positive field
- **nanodot.inp**

  UCD file with the finite element mesh

The UCD file with the magnetization distribution has to be created from a previous simulation run. First one has to search the project.log file of the old simulation for the correct number of the UCD file (e.g.,

search the fourth column where Hext==12 and the first column where eq==1 to find the equilibrium state in a field of 12 kA/m). If an inp file has been written, its number can be found in the second column. Otherwise one has to take some other inp number nearby.

Then, a full UCD file is created using the mkinp.sh shell script (cf. UCD/inp Files).

```
mkinp.sh nanodot.0001.femsh nandot.0050.gz
```

As a result, the file nanodot.0050.inp is created.

Since this UCD file also contains the full geometry of the finite element mesh, it can also be used for the nanodot.inp file. A simple copy

```
cp nanodot.0050.inp nanodot.inp
```

or a symbolic link (which saves some disk space)

```
ln -s nanodot.0050.inp nanodot.inp
```

should work.

However, it should be noted, that the original UCD files (generated with GiD) or neutral files (generated with Patran) cannot be used for the nanodot.inp file, if the simulation has been run in parallel. The reason is, that the nodes of the finite element mesh are renumbered after mesh partitioning. Thus, the numbering of the nodes in the output files is different from the numbering in the original UCD or neutral file! Therefore, any time a simulation is restarted (continued), the mesh has to be read from the same UCD file, from which the magnetization distribution is read.

Finally, the simulation parameters in allopt.txt: simulation parameters have to be updated:

```
#########################################################################
# magpar configuration file: allopt.txt
#########################################################################

-simName nanodot

-size 100e-9        # set length scaling factor to 100 nm

-init_mag 0         # read magnetization from inp file with...

-inp 0059           # ...inp number 0059

-mode 0             # select LLG time integration (PVode solver)

-hextini 32         # apply a homogeneous external field
-htheta  1.5707963  # apply homogeneous external field parallel to x-axis
-hstep   -5         # change field amplitude in steps of -5 kA/m
-hfinal  -200       # final field value -200 kA/m

-condinp_j  0.05    # save output files when J//Hext changes by 0.05
-condinp_t  1e99    # do not save output files at regular time intervals

-slice_n 0,0,1      # slice plane for png files: x-y-plane
-slice_p 0,0,0.1

-jfinal -0.2        # stop simulation if |J//Hext| < jfinal

-ts_max_time 1e99   # effectively disable maximum simulation time
```

Then, the simulation can be restarted.

The simulation parameters have been set to calculate the demagnetization curve of a magnetic nanodot. At the nucleation field (still a positive field!) a magnetic vortex structure nucleates on the boundary of the

nanodot and quickly moves towards the center (it really precesses depending on the damping parameter!). As the external field is reduced, the vortex moves towards the boundary again until it is annihilated.

Section Postprocessing presents some tools to visualize the results of the simulation, which are quite pretty and colorful.

More details about the properties of magnetic nanodots can be found in the following paper:

W. Scholz, K. Y. Guslienko, V. Novosad, D. Suess, T. Schrefl, R. W. Chantrell, J. Fidler,

"Transition from single-domain to vortex state in soft magnetic cylindrical nanodots",

J. Magn. Magn. Mater. 266 (2003) 155-163.

A preprint is available here.

18.7 nanodot_demag: Nanodot demag energy

This simple example just calculates the magnetostatic field and energy of a homogenneously magnetized cylinder (nanodot).

The configuration file is very short since we can use the default settings of all other options.

```
#########################################################################
# magpar configuration file: allopt.txt
#########################################################################

-simName nanodot

-size 100e-9         # set length scaling factor to 100 nm

# Du-Xing Chen, James A. Brug, and Ronald B. Goldfarb,
# "Demagnetizing Factors for Cylinders,"
# IEEE Trans. Magn. 21 (1991) 3601-3619.
# http://dx.doi.org/10.1109/20.102932
#
# p. 3604, Tab. 1
#
# gamma   N_m
# --------------
# 0.10    0.7967
#
# gamma ... dot aspect ratio (thickness/diameter)
# N_m   ... demagnetizing factor
#
# E_dem=0.7967/2*M_s^2*mu_0=0.39835*mu_0=316997 J/m^3
#
# more values from Table 1 in the paper cited above:
#
# gamma   N_m         gamma   N_m         gamma   N_m
# --------------      --------------      --------------
# 0.00001 0.9999      0.26    0.6262      1.4     0.2429
# 0.0001  0.9994      0.28    0.6101      1.6     0.2186
# 0.001   0.9950      0.30    0.5947      1.8     0.1986
# 0.01    0.9650      0.32    0.5801      2.0     0.1819
# 0.02    0.9389      0.34    0.5662      2.5     0.1501
# 0.03    0.9161      0.36    0.5530      3.0     0.1278
# 0.04    0.8954      0.38    0.5403      3.5     0.1112
# 0.05    0.8764      0.40    0.5281      4       0.09835
# 0.06    0.8586      0.45    0.4999      5       0.07991
# 0.07    0.8419      0.50    0.4745      6       0.06728
# 0.08    0.8261      0.55    0.4514      7       0.05809
# 0.09    0.8110      0.60    0.4303      8       0.05110
# 0.10    0.7967      0.65    0.4110      9       0.04562
```

```
# 0.12    0.7698      0.70    0.3933      10      0.04119
# 0.14    0.7450      0.75    0.3770      20      0.02091
# 0.16    0.7219      0.80    0.3619      50      0.008438
# 0.18    0.7004      0.90    0.3349      100     0.004232
# 0.20    0.6802      1.0     0.3116      200     0.002119
# 0.22    0.6611      1.1     0.2911      500     0.0008483
# 0.24    0.6432      1.2     0.2731      1000    0.0004243
```

The result of the simulation can be compared with the analytical result given in the allopt.txt file above. With the given FE mesh we obtain a pretty good result (3.158828e+05 J/m^3: error of -0.6%), with one refinement the result is even closer (3.166590e+05 J/m^3: error of -0.1%).

18.8 stress: Magnetoelastic effects on domain structure

Ahmet Kaya from the research group of Jim Bain and Jimmy Zhu at the Data Storage Systems Center (DSSC) at Carnegie Mellon University implemented magnetoelastic/magnetostriction effects in magpar based on the following papers:

Daniel Z. Bai, Jian-Gang Zhu, Winnie Yu, and James A. Bain

Micromagnetic simulation of effect of stress-induced anisotropy in soft magnetic thin films

Journal of Applied Physics, Volume 95, Number 11, June 2004, pp. 6864-6866

[J. Appl. Phys. 1] [J. Appl. Phys. 2]

Daniel Bai,

Micromagnetic Modeling of Write Heads for High-Density and High-Data-Rate Perpendicular Recording

dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, Aug. 2004.

This example uses material parameters typical of Fe_{65} Co_{35} (M_S =2.4 T) and random texture (as defined in the project.kst: magnetoelastic properties file stress.kst).

```
  7        17.5e-6   103.7e-6  -1e9    0.0     0.0      # Fe65Co35: Bai, J. Appl. Phys. 95 (2004) 6864-686
#
# texture  lamda100  lamda111  sigmaX  sigmaY  sigmaZ   # parameter
# (-)      (erg/cm^3)(erg/cm^3)(Pa)    (Pa)    (Pa)     # units
```

The example reproduces the results shown in Fig. 2 in the paper. It is interesting to compare with the zero stress case, which gives the classic closure domain pattern. FIXME: requires fine mesh with -refine 1 !!!

```
#########################################################################
# magpar configuration file: allopt.txt
#########################################################################

-simName stress

-size 1e-6          # set length scaling factor to microns

-init_mag 0         # read magnetization from inp file with...

-inp 0002           # ...inp number 0002 (stress.0002.inp)

-mode 0             # select LLG time integration (PVode solver)

-htheta 1.5707963   # set htheta to measure J//Hext parallel to x-axis

#-helastic_propfile stress.kst  # read magnetoelastic material parameters from file
```

```
-slice_n 0,0,1      # slice plane for png files: x-y-plane
-slice_p 0,0,0.04

-ts_max_time 9.0    # stop the simulation after 9 ns
```

Equilibrium domain structure with magnetoelastic effects (1 GPa compressive stress in x-direction with random texture):

Mx

My

Mz x-direction: parallel to the long axis

y-direction: parallel to the short axis

z-direction: perpendicular to the thin film

Without magnetoelastic effects:

Mx

My

Mz

Another interesting reference:

Pei Zou, Winnie Yu, and James A. Bain,

Influence of Stress and Texture on Soft Magnetic Properties of Thin Films

IEEE Trans. Magn. 38 (2002) 3501-3520.

[IEEE Trans. Magn. 1] [IEEE Trans. Magn. 2]

18.9 thinfilm: Thin magnetic film

The calculation of the demagnetizing factor of a thin magnetic film is shown in this example. A thin film of infinite size (in the plane) has a demagnetizing factor of D_z =1. However, for a small platelet of finite size (1000 x1000 x50) the demagnetizing factor is smaller ($D_z < 1$) and we can calculate its value analytically using Rok Dittrich's demagcalc. For the given aspect ratio, a material with a saturation magnetization of 1 T and a homogeneous magnetization distribution perpendicular to the plane we find D_z =0.881 and a magnetostatic energy density of E=350.54 kJ/m^3 .

```
#########################################################################
# magpar configuration file: allopt.txt
#########################################################################

-simName thinfilm

-size 1e-6          # set length scaling factor to microns

-slice_n 0,0,1      # slice plane for png files: x-y-plane
-slice_p 0,0,0.025
```

The numerical simulation with magpar gives for different refinement levels (using the "-refine" option):

refine	nodes	elements	boundary matrix	E_{demag} (J/m^3)
0	1014	2974	$<$ 1MB	3.444659e+05
1	5962	23792	89 MB	3.486828e+05
2	39559	190336	1801 MB	3.499780e+05

The simulation of models with a very flat geometry requires lots of memory because the size of the boundary matrix, which is required for the calculation of the demagnetizing field, scales quadratically with the number of nodes on the surface.

18.10 Running a simulation in parallel

In order to run a simulation in parallel on a multiprocessor machine, it is sufficient to set the variable "np" in the "run" scripts to any desired number of processes. This can also be tested on single processor machines, where the processes have to share a single CPU. Of course, this is not very efficient but very useful during development and debugging!

On workstation clusters the proper invocation of parallel programs depends on the local configuration. Usually, there is a job queueing system installed, which takes care of the scheduling and distribution. Most queueing systems support the execution of parallel jobs. Please, ask the system administrator of your machine how to run MPI jobs.

MPI usually requires a machine-file, which contains a simple list of possible machines to run on (e.g. machines.txt). These machines must be accessible by rlogin or ssh without a password (cf. section MPI).

To run magpar.exe on 10 processors distributed over the machines listed in machines.txt:

MPICH1 syntax:

```
mpirun -machinefile machines.txt -np 10 magpar.exe
```

MPICH2 syntax:

```
# create file with secret passphrase (change "mysecret")
echo "MPD_SECRETWORD=mysecret" > ~/.mpd.conf
chmod 600 ~/.mpd.conf
# create list of available machines
cp machines.txt mpd.hosts
# start daemons on 3 machines on (some of) the hosts in mpd.hosts
mpdboot -n 3
# check that all daemons are up and running
mpdtrace; mpdringtest 100
mpiexec -l -n 3 magpar.exe
```

LAM/MPI syntax:

```
$LAMBIN/lamboot machines.txt
$LAMBIN/mpirun -c 10 magpar.exe
$LAMBIN/lamhalt
```

Chapter 19

Postprocessing

Sections:

In order to visualize the simulation results, which are stored in the Output Files of Simulations , the data have to be converted into a suitable format for the Visualization Tools mentioned below.

19.1 PNG files

The magnetization snapshots stored in PNG files project.INP.X.png can be viewed with any picture viewer or web browser, which knows the PNG graphics format.

An animated gif can be created with the shell script "$MAGPAR_HOME/src/tools/shutil/mkanigif.sh". It requires the "convert" tool from the `ImageMagick` toolbox and `WhirlGIF`.

Of course, you could also use the "Animation Shop" included in "Paint Shop Pro" which is now distributed by `Corel` or one of the many other animation tools.

19.2 Sampling Line

The sampling line data consist of the definition of the measurement points project.INP.datmsh and the corresponding data files project.INP.d . In order to get complete data files, the shell script "mkdat.sh" in $MAGPAR_HOME/src/tools/shutil/ can be used. It simply uses the standard Unix/Linux tool "paste" to combine the .datmsh and .d files into .dat files. These data can then be plotted with any 2D plotting tool. Grace has shown to be a simple yet useful free and open source tool.

The following command will convert all data files in one go:

```
mkdat.sh project.0001.datmsh *.d
```

19.3 Log File

The log file project.log contains all data in a simple columnar format. These data can be analyzed with any 2D plotting tool. Grace is a free open-source 2D plotting tool for scientific data, which is easy to use, features a convenient GUI, yet is fully scriptable.

19.4 UCD/inp Files

The UCD format (e.g. project.INP.fedat , project.INP.femsh , project.INP.gz) usually require lots of disk space. Therefore, these files have been split and compressed to save disk space. The data of the finite element mesh are stored once in project.INP.femsh at the beginning of the simulation. The simulation data (magnetization, magnetostatic potential, local fields) are stored in project.INP.gz and compressed using zlib (gzip format) (if this option has been compiled in). In order to get complete UCD files, the shell script "mkinp.sh" in $MAGPAR_HOME/src/tools/shutil/ can be used. It unzips the data files and combines them with the mesh data into complete UCD files. These can then be visualized using ParaView , MicroAVS or AVS , for example.

The following command will convert all data files in one go:

```
mkinp.sh project.0001.femsh *.gz
```

19.5 Visualization Tools

19.5.1 Grace

Grace is a WYSIWYG 2D plotting tool, which runs on practically any version of Unix-like OS, as well as VMS, OS/2, and Win9x/NT/2000/XP.

19.5.2 gnuplot

gnuplot is a command-line driven interactive data and function plotting utility for Linux, UNIX, MS Windows, and many other platforms. It supports 2D and 3D plots using lines, points, boxes, contours, vector fields, surfaces, etc. It is highly recommended to use version 4.2 or later which allows the specification of explicit RGB colors.

Various gnuplot tutorials can be found on the web.

19.5.3 ParaView

ParaView by Kitware Inc. is an open-source, multi-platform 3D visualization application, which can directly read UCD files and visualize them in many different ways. It runs on Linux, Windows, Mac OS X, and other Unix-like platforms.

19.5.4 Mayavi

Mayavi by Kitware Inc. is a general purpose, cross-platform tool for 3-D scientific data visualization. It features visualization of scalar, vector and tensor data in 2 and 3 dimensions, file import of UCD/inp Files and vtu files, and easy scriptability using Python.

19.5.5 OpenDX

OpenDX is an open source software project based on IBM's Visualization Data Explorer - a powerful 3D visualization package.

Some useful tools are provided in src/tools/opendx/:

inp2dx.pl is a Perl script, which converts AVS/inp files to OpenDX format. In addition, the visual program "vp_mumag.net" is provided for convenient visualization of scalar and vector data (e.g. magnetization and magnetic fields) as 2D and 3D contour plots, vector fields (hogs), rubber sheets, 2D plots, etc. This visual program has been developed with OpenDX version 4.3.2 - later OpenDX versions seem to have some issues with it.

19.5.6 MicroAVS

MicroAVS by Kubota Graphics Technologies is a simplified version of AVS (see section AVS below) for Windows platforms. It can be used to visualize the UCD files generated by magpar.

19.5.7 AVS

AVS/Express by Advanced Visual Systems is comprehensive and versatile data visualization tools, which can import files in UCD format.

19.5.8 GiD

GiD can also be used as a postprocessing tool for visualization. There is a C/C++/Fortran library to create postprocess files for GiD from the authors of GiD: GiDpost. However, suitable output routines still need to be implemented.

Chapter 20

Tools

Sections:

The tools discussed below can be found in the $MAGPAR_HOME/src/tools/ directory in their respective subdirectories.

20.1 gmsh: gmshtoucd.py

gmshtoucd.py is a small Python utility (implementation based on ngtoucd.py) to convert Gmsh mesh files into AVS inp files for use with magpar.

20.2 gmsh: msh2inp.awk

msh2inp.awk is a utility in awk to convert Gmsh mesh files into AVS inp files for use with magpar - implemented by Stefan Tibus.

20.3 ngtoucd.py

ngtoucd.py is a small Python utility to convert neutrally-exported NETGEN meshes into AVS unstructured cell data meshes for use with magpar.

Authors:

Richard Boardman and Hans Fangohr

documentation:

```
ngtoucd v0.2
------------

ngtoucd is a small Python utility to convert neutrally-exported NETGEN
meshes into AVS unstructured cell data meshes for use with magpar;
although in theory the meshes should be accessible to anything
requiring AVS/UCD input files, this has not been tested.

ngtoucd requires Python to run. It has been tested with Python 2.3, but
Python 2.1 and 2.2 should also be fine.

NETGEN can be found here:
http://www.mathcces.rwth-aachen.de/netgen/
http://www.hpfem.jku.at/netgen/

Usage
```

```
-----

To use, first ensure that the file ngtoucd has been marked executable:

chmod +x ngtoucd

will set the executable flag on the file. To convert a mesh called
foo.msh:

./ngtoucd foo.msh

ngtoucd will then convert the mesh and leave a file called foo.inp -
this is the AVS/UCD mesh. By copying ngtoucd into the path, perhaps one
of:

~/bin
/usr/bin
/usr/local/bin

(the top one is preferred for a personal setup, the last one for a
system-wide setup) then this should be callable without the preceding
./ as:

ngtoucd foo.msh

An example NETGEN mesh has been included (sphere.msh) for testing
purposes.


Changes since version 0.1
-------------------------

ngtoucd now no longer ignores the material parameter for UCD meshes. This
can be interpreted loosely as top-level objects (TLOs) in netgen. The ID
is passed straight through to the UCD mesh (after parsing as an integer).


Contact details
---------------

Please send any bug reports and other comments to
richboardman@gmail.com or hans.fangohr@physics.org
```

NETGEN can be found here:

- http://www.mathcces.rwth-aachen.de/netgen/
- http://www.hpfem.jku.at/netgen/

20.4 vtk tools

The two Perl scripts `inp2vtu.pl` and `inp2vtu_gz.pl` convert project.INP.inp files of the form `<project>.????.inp` into XML formatted files for use with ParaView.

Syntax:

```
# create VTK XML ASCII files:
./inp2vtu.pl
Usage: inp2vtu.pl <project>

# create VTK XML binary files (about 1/5th the size of the ASCII files):
./inp2vtu_gz.pl
Usage: inp2vtu_gz.pl <project>
```

The scripts generate the following output files:

```
<project>_????.vtu    # ???? stands for the 4 digit INP serial number
<project>.pvd         # ParaView control file (defining timestep and
                      # corresponding vtu file
```

Then you can open the file `<project>.pvd` in ParaView and use animation controls to browse through the ∗.vtu files and create avi movies with "File/Save Animation".

Note, that the scripts automatically ignore the checkpoint file project.9999.∗.

These scripts were contributed by Stefan Tibus (Univ. of Konstanz) from the research group of Manfred Albrecht at the `Univ. of Chemnitz` - thanks!

20.5 opendx: converter, visual program

This directory contains the converter inp2dx.pl, the shell script mkdx.sh and the OpenDX visual program "vp_mumag". See section OpenDX .

20.6 shutil: shell scripts

This directory contains various shell scripts, mainly for Postprocessing of magpar output files.

20.7 gid: GiD inp problem type

This tool defines the problem type "inp", which enables GiD to export finite element meshes in AVS/inp format as described in the section on Preprocessing.

Chapter 21

Publications

If you use magpar and publish results, which have been obtained/derived from its simulations, please cite the following paper:

Reference

- W. Scholz, J. Fidler, T. Schrefl, D. Suess, R. Dittrich, H. Forster, V. Tsiantos,
  "Scalable Parallel Micromagnetic Solvers for Magnetic Nanostructures",
  Comp. Mat. Sci. 28 (2003) 366-383.
  Preprint: [Paper], [ScienceDirect]

Additional Papers

Papers about magpar and using magpar. I would be happy, if anyone who is using magpar and published results, which have been obtained/derived from its simulations, could send me references and a preprint of the papers (PDF file by email to magpar(at)magpar.net).

1. Y. Kanai, K. Hirasawa, Y. Jinbo, K. Yoshida, S.J. Greaves and H. Muraoka,
   "Write head modeling for shingled recording."
   Intermag 2009 - paper DC-08
   [Conf. program]
2. V. Russier,
   "Magnetization in uniaxial spherical nanoparticles: Consequence on the interparticle interaction",
   J. Magn. Magn. Mater. 2009 - in press.
   [ScienceDirect] [Preprint 1] [Preprint 2] [arXiv]
3. V. Russier,
   "Spherical magnetic nanoparticles: magnetic structure and interparticle interaction",
   J. Appl. Phys. 105 (2009) 073915.
   [J. Appl. Phys.] [Preprint] [arXiv]
4. Pawel Steblinski, Tomasz Blachowicz,
   "Wavelets spectra of magnetization dynamics in geometry driven magnetic thin layers",
   preprint 2009
   [arXiv]
5. Thomas Fischbacher, Hans Fangohr,
   "Continuum multi-physics modeling with scripting languages: the Nsim simulation compiler prototype for classical field theory.",
   preprint 2009 [arXiv] [preprint]
6. O. S. Trushin, N. Barabanova, V. P. Alekseev,
   "ISSLEDOVANIE MAGNITNYKh KhARAKTERISTIK PERMALLOEVYKh NANOSTRUKTUR METODOM MIKROMAGNITNOGO MODELIROVANIYa."
   Russian Mikroelektronika vol. 38, no. 3 (2009) 198-203.
   [Abstract] in [Russian Microelectronics] [Springer]
7. Chen Ren-Jie, Zhang Hong-Wei, Shen Bao-Gen, Yan A-Ru and Chen Li-Dong,
   "Simulation of magnetization behaviours of Sm(Co,Fe,Cu,Zr)z magnet with low Cu content"
   Chinese Phys. B 18 (2009) 2582-2588
   [IOP]

8. Michael J. Donahue
"Parallelizing a micromagnetic program for use on multi-processor shared memory computers"
preprint 2009
[PDF]

9. R. Yanes, F.Garcia-Sanchez, R.Rozada, O.Chubykalo-Fesenko, P.Martin Pimentel, B.Leven, B.Hillebrands,
"Modelling of microwave-assisted switching in magnetic ellipsoids",
MSNOW 2008 - Magnetic Single Nano-Object Workshop and School
[arXiv] [Slides] [Slides_b]

10. Tomasz Blachowicz, Bartlomiej Baron,
"A graphical extension for the Windows version of the Parallel Finite Element Micromagnetics Package (MagParExt)",
[MagParExt website] [arXiv]

11. M Jaafar, R Yanes, A Asenjo, O Chubykalo-Fesenko, M Vazquez, E M Gonzalez and J L Vicent,
"Field induced vortex dynamics in magnetic Ni nanotriangles"
Nanotechnology 19 (2008) 285717 (8pp)
[IOP Nanotechnology]

12. Lin Yuan, Jianjun Jiang, Gang Du, Zhongyou Wang and Huahui He,
"Micromagnetic Simulation of Magnetic Particles with Surface Anisotropy"
Journal of the Magnetics Society of Japan, Vol. 32 (2008) , No. 2_1 pp. 50-53
[J-Stage] [J. Magn. Soc. Jpn.]

13. D. Makarov, S. Tibus, C. T. Rettner, T. Thomson, B. D. Terris, T. Schrefl, and M. Albrecht
"Magnetic strip patterns induced by focused ion beam irradiation",
J. Appl. Phys. 103, 063915 (2008)
[J. Appl. Phys.]

14. L Yuan, J Jiang, Z Wang, B Tian, H He
"Micromagnetic Simulation of a Magnetic Film With Surface Roughness"
IEEE Trans. Magn. 44 (2008) 2130-2133.
[IEEE]

15. S. Tibus, D. Makarov, C. T. Rettner, T. Thomson, B. D. Terris, T. Schrefl, M. Albrecht,
"Patterning of magnetic films by focused ion beam irradiation",
Intermag 2008 paper no. CM-05, submitted to IEEE Trans. Magn.

16. S. Tibus, D. Makarov, C.T. Rettner, T. Thomson, B.D. Terris, and M. Albrecht,
"Magnetic Patterning by Focused Ion Beam Irradiation",
IMST 2007 - Innovative Mass Storage Technologies 2007
[Abstract]

17. R. Yanes, M. Jaafar, O. Chubykalo-Fesenko, A. Asenjo, M. Vazquez, J.L. Vicent
"Micromagnetic Modelling and MFM Images of Magnetic Structures in Ni Dots",
Poster P102 presented at GEFES 2008 (V Reunion del Grupo Especializado de Fisica del Estado Solido)
[Abstracts]

18. Yuan, Lin; Jiang, Jianjun; Di, Yongjiang; Bie, Shaowei; He, Huahui

"Micromagnetic simulation of the magnetic dispersion angle and effective damping factor for the single-phase soft magnetic films",

J. Magn. Magn. Mater. Volume 320, Issue 7, p. 1393-1397 (2008).

[ScienceDirect]

19. Deng Lian-wen, Huang Xiao-zhong, Zhou Ke-sheng, Yang Bing-chu,

"Micromagnetic simulation and experimental study on microwave permeability of nano-magnetic films",

Trans. Nonferrous Met. Soc. China 17 (2007) s756-s759.

[PDF]

20. Ching-Ming Lee, L. X. Ye, J. M. Lee, and Te-Ho Wu,

"Micromagnetic study of magnetization reversal in patterned DyFeCo thin films",

phys. stat. sol. (a) 204, No. 12, 4049-4052 (2007)

[phys. stat. sol.]

21. Simon Greaves,

"Micromagnetic Simulations of Magnetic Recording Media" in "High Performance Computing on Vector Systems 2007",

Springer Berlin Heidelberg

[Book @ Springer]

22. Nadjib Benatmane, Werner Scholz, and T.W. Clinton,

"Magnetic Configurations and Phase Diagrams of sub-100 nm NiFe Nanorings",

IEEE Trans. Magn. 43 (2007) 2884-2886.

[IEEE Trans. Magn.]

10th Joint MMM/Intermag Conference (paper no. GE-07), Jan. 11, 2007, Baltimore, MD.

23. T. Fischbacher, M. Franchin, G. Bordignon, H. Fangohr,

"A Systematic Approach to Multiphysics Extensions of Finite-Element-Based Micromagnetic Simulations: Nmag."

IEEE Trans. Magn. Volume: Volume: 43, Issue: 6, pp. 2896-2898

[IEEE Trans. Magn.] [Preprint]

24. H. H. Long, E. T. Ong, Z. J. Liu, and E. P. Li,

"Fast Fourier Transform on Multipoles for Rapid Calculation of Magnetostatic Fields"

IEEE Trans. Magn. Volume, vol. 42 no. 2 (2006) 295-300.

[IEEE Trans. Magn.]

25. R. Dittrich, J. Fidler, D. Suess, W. Scholz, and H. Forster, T. Schrefl,

"Computational Aspects of Micromagnetics",

in "The Science of Hysteresis," G. Bertotti, I. Mayergoyz (eds), Academic Press, London, vol. 2, pp. 383-433, 2006, ISBN: 0-12-480874-3.

[Book@Academic Press] [Book@Amazon]

26. J. Fidler, T. Schrefl and W. Scholz,

"Computational Micromagnetics",

in Handbook of Theoretical and Computational Nanotechnology, M. Rieth, W. Schommers (eds), American Scientific Publishers, 2006, ISBN: 1-58883-042-X.

[Book@American Scientific Publishers] [Book@Amazon]

27. Thomas Schrefl, Gino Hrkac, Simon Bance, Dieter Suess, Otmar Ertl, Josef Fidler,
"Numerical Methods in Micromagnetics (Finite Element Method)",
in "Handbook of Magnetism and Advanced Magnetic Materials", Helmut Kronmueller, Stuart Parkin (eds.), ISBN: 978-0-470-02217-7
[Wiley InterScience]

28. D. Suess, J. Fidler, T. Schrefl,
"Micromagnetic Simulation of magnetic materials",
in "Handbook of Magnetic Materials", by K. H. J. Buschow (editor); Elsevier Science, ISBN-10: 0444518509
[Book@Amazon]

29. J. Fernandez-de-Castro, Xiao Shen, Jianhua Xue, Yuming Zhou, and S. Sharma,
"Writer Flux Closure and Transition Degradation Mechanism in Perpendicular Recording."
IEEE Trans. Magn. Volume: 43, Issue: 6, 2175-2177.
[IEEE Trans. Magn.]
MMM/Intermag 2007 Conference, Baltimore, MD, paper no. AB-09.

30. M.J. Donahue, R.D. McMichael
"Micromagnetics on Curved Geometries Using Rectangular Cells: Error Correction and Analysis"
IEEE Transactions on Magnetics 43 (2007) 2878-2880.
[IEEE] [PDF]

31. C.B. Muratov and V.V. Osipov
"Optimal grid-based methods for thin film micromagnetics simulations"
Journal of Computational Physics, Volume 216, Issue 2, 10 August 2006, Pages 637-653
[PDF]

32. HH Long, ZJ Liu, ET Ong, EP Li
"Micromagnetic modeling simulations and applications"
17th International Zurich Symposium on Electromagnetic Compatibility, 2006. pp. 398-401
[IEEE] [IEEE] [arXiv]

33. M Albrecht, G Hu, IL Guhr, TC Ulbrich, et al.
"Magnetic multilayers on nanospheres"
Nature Materials 4, 203 - 206 (2005)
[Nat. Mat.] [PDF]

34. L Clime, B Le Drogoff, S Zhao, Z Zhang
"Magnetic nanocarriers: from material design to magnetic manipulation"
International Journal of Nanotechnology, 2008 - Inderscience
[Online]

35. G. Rodriguez-Rodriguez, M. Velez, O. Chubykalo-Fesenko, and J. M. Alameda,
"Micromagnetic simulation of MFM tip hysteresis and stray field",
Poster presented at the "Trends in Nanotechnology" conference September 1-5, 2008, Oviedo, Spain.
[TNT 2006]

36. W. Scholz and S. Batra,

"Effect of write current waveform on magnetization and head field dynamics of perpendicular recording heads",

IEEE Trans. Magn. 42 (2006) 2264 - 2266.

[PDF]

©2005 IEEE see copyright notice

Intermag 2006 Conference, San Diego, CA, paper no. EF-03.

37. S. Batra, W. Scholz, T. Roscamp,

"Effect of thermal fluctuation field on the noise performance of a perpendicular recording system",

J. Appl. Phys. 99, 08E706 (2006)

©(2004) American Institute of Physics see copyright notice

This article may also be found at the publisher's website.

[PDF]

presented at: MMM 2005 (paper no. CC-10), Nov. 1, 2005, San Jose, CA.

38. F. Johnson, S. Amancherla, G.J. Parker, L.E. Iorio, and P.R. Subramanian,

"Micromagnetic simulations of the dependence of domain wall width with grain size in systems with cubic and uniaxial anisotropy.",

Intermag 2006 Conference, San Diego, CA, paper no. AD-10.

39. A. Kaya, M. Benakli, M.L. Mallary and J.A. Bain,

"A Micromagnetic Study of the Effect of Spatial Variations in Damping in Perpendicular Recording Heads",

IEEE Trans. Magn. 42 (2006) 2428 - 2430.

[IEEE Trans. Magn.]

Intermag 2006 Conference, San Diego, CA, paper no. BC-05.

40. A. Kaya and J. A. Bain,

"Using Magpar to Model Write Heads",

[PDF]

Carnegie Mellon University, DSSC 2006 Spring Technical Review

41. C. C. Chang, Y. C. Chang, C. M. Lee, C. C. Chen, and J. Wu,

"Control of Domain Wall Motion in Permalloy Ring",

IEEE Trans. Magn. 42 (2006) 2960 - 2962.

[IEEE Trans. Magn.]

Intermag 2006 Conference, San Diego, CA, paper no. HQ-08.

42. Richard P. Boardman, Hans Fangohr, Matthew J. Fairman, Jürgen Zimmermann, Simon J. Cox, Alexander A. Zhukov and Peter A.J. de Groot

"Micromagnetic modelling of ferromagnetic cones",

J. Magn. Magn. Mater. Volume 312, Issue 1, May 2007, Pages 234-238

[ScienceDirect] [e-prints Soton]

43. Richard Boardman,

"Computer simulation studies of magnetic nanostructures",

Ph.D. thesis, Computational Engineering and Design Group, School of Engineering Sciences, University of Southampton, United Kingdom (2005).

[PDF]

44. Richard Boardman, Juergen Zimmermann, Hans Fangohr, Alexander Zhukov and Peter de Groot,
"Micromagnetic simulation studies of ferromagnetic part-spheres",
J. Appl. Phys. 97 (2005) 10E305.
[J. Appl. Phys.]
Presented at MMM 2004, Jacksonville, FL, USA.

45. I-Hsin Chung, Robert E. Walkup, Hui-Fang Wen, Hao Yu,
"MPI Performance Analysis Tools on Blue Gene/L,"
sc, p. 16, ACM/IEEE SC 2006 Conference (SC'06), 2006
[Paper] [IEEE Computer Society]

46. I.-H. Chung, J.K. Hollingsworth,
"A Case Study Using Automatic Performance Tuning for Large-Scale Scientific Programs",
High Performance Distributed Computing, 2006 15th IEEE International Symposium on Volume, Issue, 0-0 0 Page(s):45 - 56.
[IEEE]

47. Krzysztof M. Lebecki, Jacek Kossut,
"Periodic boundary conditions in micromagnetic modeling",
Poster at E-MRS Fall Meeting 2006, Symposium E
[science24.com]

48. Krzysztof M. Lebecki,
"Mikromagnetyczne modelowanie rozkladu namagnesowania w kwazijednowymiarowych...",
PhD thesis
[Thesis]

49. D. Bueno-Baques, L. Maldonado,J. L. Hidalgo-Gonzalez, V. Corral-Flores and J. A. Matutes-Aquino,
"Magnetic Properties of Screen-Printed (Y0.5Sm0.5)Co5 Magnet Arrays",
Poster presented at the "Trends in Nanotechnology" conference 04-08 September2006, Grenoble-France
[TNT 2006]

50. C. K. Lim, Y. S. Kim, N. Y. Park, and J. Lee,
"Multibit MRAM using a pair of memory cells",
IEEE Trans. Magn. 41 (2005) 2670-2672.
[IEEE Trans. Magn.]
Presented at Intermag 2005 Conference, Nagoya, Japan, paper no. HP-04.

51. W. Scholz, S. Batra,
"Micromagnetic Simulation of Head Field and Write Bubble Dynamics in Perpendicular Recording",
IEEE Trans. Magn. 41 (2005) 2839-2844.
[PDF]
©2005 IEEE see copyright notice
Invited talk at Intermag 2005 Conference, Nagoya, Japan, paper no. BB-05.

52. W. Scholz, S. Batra,
"Micromagnetic Modeling of Head Field Rise Time for High Data-Rate Recording",
IEEE Trans. Magn. 41 (2005) 702-706.
[PDF]
©2005 IEEE see copyright notice
Invited talk at 3rd North American Perpendicular Magnetic Recording Conference (NAPMRC '04), August 10, 2004, Boulder, CO, USA.

53. M. d'Aquino, W. Scholz, T. Schrefl, C. Serpico, J. Fidler,
"Micromagnetic Analysis of Fast Precessional Switching",
J. Magn. Magn. Mater. vol. 290-291 (2005) 510-513.
Preprint: [Paper], [ScienceDirect]

54. K. Yu. Guslienko, W. Scholz, R. W. Chantrell, and V. Novosad,
"Vortex state oscillations in soft magnetic cylindrical dots",
Phys. Rev. B 71 (2005) 144407.
Preprint: [Abstract], [PDF]
Featured in the Virtual Journal of Nanoscale Science & Technology: Volume 5, Issue 21

55. Massimiliano d'Aquino,
"Nonlinear Magnetization Dynamics in Thin-Films and Nanoparticles",
PhD thesis, Universita di Napoli Federico II.
[Abstract], [PDF]

56. M. d'Aquino, W. Scholz, T. Schrefl, C. Serpico, J. Fidler,
"Numerical and analytical study of fast precessional switching",
J. Appl. Phys. 95 (2004) 7055-7057.
©(2004) American Institute of Physics see copyright notice
This article may also be found at the publisher's website.
[PDF]

57. M. d'Aquino, W. Scholz, T. Schrefl, C. Serpico, G. Miano,
"Analysis of fast precessional switching in magnetic thin films",
Proceedings of PIERS (Progress In Electromagnetic Research Symposium) 2004 pp. 257-260.
Preprint: [Paper]

58. M Dimian
"Nonlinear spin dynamics and ultra-fast precessional switching"
PhD thesis, University of Maryland, College Park, 2005
[PDF]

59. ED Boerner, KZ Gao, RW Chantrell
"Effect of damping on the time variation of fields produced by a small pole tip with a soft underlayer"
IEEE Trans. Magn., Vol. 40, No. 4 (2004) 2371-2373.
[IEEE] [PDF]

60. W. Scholz, J. Fidler, T. Schrefl, D. Suess, H. Forster, R. Dittrich, V. Tsiantos,
"Numerical Micromagnetic Simulation of Fe-Pt Nanoparticles with Multiple Easy Axes",
J. Magn. Magn. Mater. 272-276 (2004) 1524-1525.
Preprint: [Paper], [Poster], [ScienceDirect]

61. W. Scholz, D. Suess, T. Schrefl, and J. Fidler,
"Micromagnetic simulation of magnetization reversal in small particles with surface anisotropy",
J. Appl. Phys. 95 (2004) 6807-6809
©(2004) American Institute of Physics see copyright notice
This article may also be found at the publisher's website.
[PDF]

62. W. Scholz, D. Suess, R. Dittrich, T. Schrefl, V. Tsiantos, H. Forster, J. Fidler,
"Implementation of a high performance parallel finite element micromagnetics package",
J. Magn. Magn. Mater. 272-276 (2004) 693-694

Preprint: [Paper], [Poster], [ScienceDirect]

63. J. Fidler, T. Schrefl, W. Scholz, D. Suess, R. Dittrich and M. Kirschner,
"Micromagnetic modelling and magnetization processes",
J. Magn. Magn. Mater. 272-276 (2004) 641-646.

Preprint: [Paper], [Talk], [ScienceDirect]

64. W. Scholz,
"Scalable Parallel Micromagnetic Solvers for Magnetic Nanostructures",
dissertation, Vienna University of Technology, Vienna, Austria, 2003.

[Abstract], [HTML], [PDF] (7 MB)

65. T. Schrefl, W. Scholz, R. Dittrich, H. Forster, D. Suess, M. Stehno, J. Fidler,
"Micromagnetic Simulations of Domainstructures in Softmagnetic Thin Films",
Project report 2003.

[PDF]

66. W. Scholz, T. Schrefl, J. Fidler, T. Matthias, D. Suess, V. Tsiantos,
"Micromagnetic simulation of the pinning and depinning process in permanent magnets",
IEEE Trans. Magn. 39 (2003) 2920-2922

[PDF]

©2003 IEEE see copyright notice

67. W. Scholz, K. Y. Guslienko, V. Novosad, D. Suess, T. Schrefl, R. W. Chantrell, J. Fidler,
"Transition from single-domain to vortex state in soft magnetic cylindrical nanodots",
J. Magn. Magn. Mater. 266 (2003) 155-163.

Preprint: [Paper], [Talk], [ScienceDirect]

Talks

1. W. Scholz and S. Batra

"Effect of write current waveform on magnetization and head field dynamics of perpendicular recording heads",
IEEE Trans. Magn., submitted.

Intermag 2006 Conference, San Diego, CA, paper no. EF-03.

2. W. Scholz, S. Batra,
"Micromagnetic Simulation of Head Field and Write Bubble Dynamics in Perpendicular Recording",
IEEE Trans. Magn. 41 (2005) 2839-2844.

invited talk at Intermag 2005 (paper no. BB-05), April 5, 2005, Nagoya, Japan.

3. W. Scholz, S. Batra,
"Micromagnetic Modeling of Head Field Rise Time for High Data-Rate Recording",
IEEE Trans. Magn. 41 (2005) 702-706.

invited talk at 3rd North American Perpendicular Magnetic Recording Conference (NAPMRC '04), August 10, 2004, Boulder, CO, USA.

4. W. Scholz, D. Suess, J. Fidler, and T. Schrefl
   "Micromagnetic Simulation of Magnetization Reversal in Small Particles with Surface Anisotropy",
   J. Appl. Phys. 95 (2004) 6807-6809.
   9th Joint MMM/Intermag Conference, January 5-9, 2004, Anaheim, CA, USA.
   [PDF] (900 KB)

5. W. Scholz
   "magpar - an open source finite element micromagnetics package",
   9th Joint MMM/Intermag Conference, January 5-9, 2004, Anaheim, CA, USA.
   [PDF] (1 MB)

6. W. Scholz, T. Schrefl, D. Suess, R. Dittrich, H. Forster, V. Tsiantos, J. Fidler,
   "Implementation of a scalable parallel finite element package for micromagnetic simulations",
   9th Joint MMM/Intermag Conference, January 5-9, 2004, Anaheim, CA, USA.
   [PDF] (1 MB)

7. Werner Scholz, Thomas Schrefl, Josef Fidler, Thorsten Matthias, Dieter Suess, and Vassilios Tsiantos
   "Micromagnetic simulation of the pinning and depinning process in permanent magnets",
   IEEE Trans. Magn. 39 (2003) 2920-2922
   Intermag 2003, March 30 - April 3, 2003. Boston, MA, USA.
   [PDF] (1.4 MB)

8. W. Scholz, K. Y. Guslienko, V. Novosad, D. Suess, T. Schrefl, R. W. Chantrell, J. Fidler,
   "Transition from single-domain to vortex state in soft magnetic cylindrical nanodots",
   J. Magn. Magn. Mater. 266 (2003) 155-163.
   ICFPM 2002, 4th International Conference on Fine Particle Magnetism, August 14-16, 2002, Pittsburgh, PA, USA.
   [PDF] (1.5 MB)

Posters

1. W. Scholz, D. Suess, R. Dittrich, T. Schrefl, V. Tsiantos, H. Forster, J. Fidler
   "Implementation of a high performance parallel finite element micromagnetics package"
   J. Magn. Magn. Mater. 272-276 (2004) 693-694.
   ICM 2003 conference, July 27 - August 1, 2003, Rome, Italy
   [Poster]

2. W. Scholz, J. Fidler, T. Schrefl, D. Suess, H. Forster, R. Dittrich, V. Tsiantos,
   "Numerical Micromagnetic Simulation of Fe-Pt Nanoparticles with Multiple Easy Axes",
   J. Magn. Magn. Mater. 272-276 (2004) 1524-1525.
   ICM 2003 conference, July 27 - August 1, 2003, Rome, Italy
   [Poster (1.5 MB)]

The research group of Josef Fidler and Thomas Schrefl presents many more publications and results in the field of magnetic materials and micromagnetics on its website:

http://magnet.atp.tuwien.ac.at/

Sites, which mention and/or link to magpar

1. IEEE Magnetics Society Newsletter January 2004, Conference report
2. IEEE Magnetics Society Newsletter January 2004
3. Joint MMM-Intermag conference 2004, muMAG session report
4. NIST: Micromagnetic Modeling Activity Group
5. OOMMF: Other Online Resources
6. A Case Study Using Automatic Performance Tuning for Large-Scale Scientific Programs
7. MPI tools and performance studies---MPI performance analysis tools on Blue Gene/L, link2
8. Progress report on the U.S. Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) Program
9. TACC Tutorial
10. GUCAS Tutorial
11. PDE Simulation with Python
12. PETSc and its Ongoing Research and Development
13. External Impact of TOPS Scalable Solver Software
14. What's in NPACI That Rocks and How Do I Use It?
15. PETSc Bibliography
16. TAO Bibliography
17. Getting Started with PETSc
18. Technologies and Tools for High-Performance Distributed Computing
19. Ultrascale Tsunami of Data
20. Using the NCSU HPC machines
21. PETSc Overview
22. PETSc Tutorial
23. PETSc Tutorial
24. PDE libraries in Python
25. PDE Simulation with Python
26. Boyana Norris: PETSc
27. PETSc Tutorial
28. SUNDIALS: SUite of Nonlinear and DIfferential / ALgebraic Equation Solvers
29. The SUNDIALS integration and sensitivity analysis solvers

30. Adjoint-based methods for analysis of dynamical systems
31. SUNDIALS
32. Projects using SUNDIALS
33. Computational Nanomagnetism (Hans Fangohr)
34. nmag manual
35. Richard Boardman
36. Richard Boardman's Ph.D. thesis
37. DSSC 2006 Spring Technical Review
38. Using Magpar to Model Write Heads
39. Samir Y. Garzon
40. Massimiliano d'Aquino
41. Comparison between Damping and Precessional switching in magnetic thin-films
42. Nonlinear Magnetization Dynamics in Thin-films and Nanoparticles
43. Ruslan PUSCASU
44. Laurentiu STOLERIU
45. CARPATH Activities
46. Towards Automatic Performance Tuning
47. Tomasz Blachowicz
48. MagParExt
49. Simulations of magnetization dynamics
50. Micromagnetic Simulations using the Exeter Supercomputer
51. Artificial cilia for microfluidics
52. Laboratoire Leon Brillouin - Neutron scattering on magnetic nanostructures
53. Project proposal
54. Micromagnetism article in Wikipedia
55. Eric Meloche, "Perpendicular Recording at High Areal Densities"
56. ComputationalMagneticsLinks
57. MAGPAR - SPCC Dept. of Physics, University of Seoul
58. Univ. of Seoul
59. Wikipedia

Chapter 22

Programming, Debugging, Bug Reporting

Sections:

22.1 General programming guidelines

Formatting:

- There are no "tabs" (except in Makefiles where necessary).
- Indent each level by 2 spaces
- Opening parentheses, braces are never on a line of their own

Variables:

- Avoid global variables (especially in the GridData structure)
- Use local variables wherever possible

Simple templates for magpar functions:
Initialization function (called during serial or parallel initialization):

```
int MagparFunctionInit(GridData *gdata,Vec vec1,PetscReal *real1)
{
  /* First executable line of each PETSc function used for error handling. */
  /* macros defined in griddata.h, see also PetscFunctionBegin */
  /* print log information on stdout (source file, function name) */
  MagparFunctionLogBegin;

  [do stuff]

  /* print timing information */
  MagparFunctionLogReturn(0);
}
```

Solver function (called many times during solution):

```
int MagparFunction(GridData *gdata,Vec vec1,PetscReal *real1)
{
  /* by default do not print any information on stdout */
  /* if "-info" option is active: print log information on stdout */
  MagparFunctionInfoBegin;

  [do stuff]

  /* by default do not print any information on stdout */
  /* if "-info" option is active: print timing information on stdout */
  MagparFunctionInfoReturn(0);
}
```

22.2 Debugging

The following suggestions might be helpful for debugging:

- Additional PETSc options

  Before submitting any bug reports, please rerun your simulation including the additional configuration file with PETSc internal logging/info/diagnostic options contained in allopt_log.txt (especially the memory checking and "-info" options), which might give some more detailed information about the cause of the problem.

- Running magpar in a debugger for serial runs

  Just like any other program you can run magpar in a debugger (e.g. gdb) or use the PETSc option "-on_error_attach_debugger [gdb,dbx,xxgdb,ups,noxterm]" to attach to a debugger on error (see allopt_log.txt, PETSc Documentation, PETSc users manual). Use the PETSc option "-display" to define the location where graphics and debuggers are to be displayed.

- Running magpar in a debugger for parallel runs

  The "-on_error_attach_debugger" option even works for parallel runs, where separate xterms will pop up for each rank. For convenience it is useful to run parallel debugging runs on the local development machine. The number of parallel threads can exceed the number of processors/cores, of course. So you can debug a parallel run also on a single processor machine.

- printf

  Often it is more convenient and efficient to debug using the "printf" technique. In the context of PETSc based programs like magpar this includes functions, which print also more sophisticated parallel data structures, e.g. hdemag.c::Hdemag_Init :

```
PetscPrintf(PETSC_COMM_WORLD,"deb01: Au1\n");
ierr = MatView(Au1,PETSC_VIEWER_STDOUT_WORLD);CHKERRQ(ierr);
PetscPrintf(PETSC_COMM_WORLD,"deb02: u1\n");
ierr = VecView(u1,PETSC_VIEWER_STDOUT_WORLD);CHKERRQ(ierr);

/* get output from all processors */
PetscBarrier(PETSC_NULL);
PetscSynchronizedPrintf(PETSC_COMM_WORLD,
  "[%i]deb03:  %i %g!\n",rank,intvar,doublevar
);
PetscSynchronizedFlush(PETSC_COMM_WORLD);
```

In addition, it is often useful to ensure that all parallel processes are synchronized at a certain position in the program. This is easily accomplished by using the PETSc function "PetscBarrier", which blocks until this routine is executed by all processors owning a certain PETSc object, e.g. the global magnetization vector "gdata->M".

```
PetscPrintf(PETSC_COMM_WORLD,"deb03: barrier\n");
PetscBarrier(PETSC_NULL);
```

22.3 Guidelines for bug reports

Required information:

- magpar version, original distribution or patched/modified version
- a detailed description with copies of the error messages from compiler, program, etc.

- compiler name and version (e.g. GNU gcc, GNU g++, cc)
- versions of all used libraries: PETSc, MPI, BLAS, LAPACK, ATLAS, Sundials, etc.
- operating system: name, version, release, (Linux distribution)
- machine/hardware type

The more information you provide, the easier it is to track down the problem.

Bug reports may be sent to: magpar(at)magpar.net

Chapter 23

Directory Hierarchy

23.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Chapter 24

Data Structure Index

24.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 25

Directory Documentation

25.1 emini/ Directory Reference

Files

- file **checkiterationemini.c**
- file **emini.h**
- file **eminisolve.c**

25.2 field/ Directory Reference

Files

- file **bele.c**
- file **bmatrix.c**
- file **field.h**
- file **hcubic.c**
- file **hdemag.c**
- file **helastic.c**
- file **hexch_ani.c**
- file **hext_cu.c**
- file **hext_ho.c**
- file **hext_py.c**
- file **hexternal.c**
- file **hstep_file.c**
- file **htot.c**

25.3 tools/gmsh/ Directory Reference

Files

- file **gmshtoucd.py**
- file **msh2inp.awk**

25.4 init/ Directory Reference

Files

- file **destroyinit.c**
- file **distortmesh.c**
- file **elevertvol.c**
- file **facnb.c**
- file **filterelements.c**
- file **filternodes.c**
- file **init.h**
- file **initinfo.c**
- file **maginit.c**
- file **magset.c**
- file **modifyprop_par.c**
- file **modifyprop_ser.c**
- file **movedata.c**
- file **parinit.c**
- file **parteleser.c**
- file **regrefine.c**
- file **reorder.c**
- file **serinit.c**
- file **vertprop.c**

25.5 io/ Directory Reference

Files

- file **magpario.h**
- file **readinp.c**
- file **readkrn.c**
- file **readmesh.c**
- file **readpatran.c**
- file **writedata.c**
- file **writedataavs.c**
- file **writedatadat.c**
- file **writefemavs.c**
- file **writelog.c**
- file **writelog_pid.c**

25.6 llg/ Directory Reference

Files

- file **calc_dMdt.c**
- file **checkiterationllg.c**
- file **llg.h**
- file **myllgjacobian.c**
- file **mytscreatepvode.c**
- file **mytssteppvode.c**
- file **precond.c**
- file **rhsfunction.c**
- file **writelog_pvode.c**

25.7 tools/ngtoucd/ Directory Reference

Files

- file **ngtoucd.py**

25.8 png/ Directory Reference

Files

- file **writedatapng.c**
- file **writedatapng2.c**
- file **writepng.c**
- file **writepng.h**

25.9 pytools/ Directory Reference

Files

- file **magpar_scripts.py**

25.10 tools/ Directory Reference

Directories

- directory gmsh
- directory ngtoucd

25.11 util/ Directory Reference

Files

- file **area.c**
- file **ascat.c**
- file **axesrot.c**
- file **barycent.c**
- file **bbox2.c**
- file **calAfe2fe.c**
- file **calAfe2sq.c**
- file **calAsq2fe.c**
- file **calcbbox.c**
- file **cart2sphere.c**
- file **distint.c**
- file **distortvec.c**
- file **distpointline.c**
- file **ipol.c**
- file **matcreateseqadj.c**
- file **matviewstruct.c**
- file **mesh2dual.c**
- file **printmatinfo.c**
- file **progressbar.c**
- file **renormvec.c**
- file **solidangle.c**
- file **syncffprintf.c**
- file **tettri.c**
- file **util.h**
- file **vecsetvec.c**

Chapter 26

Data Structure Documentation

26.1 GridData Struct Reference

Data Fields

- char simname [256]
- int inp
- PetscReal time
- int mode
- PetscReal hscale
- PetscReal tscale
- PetscReal escale
- PetscReal lenscale
- PetscReal totvol
- int equil
- PetscReal vequil

global counters of finite element mesh

- int n_vert
- int n_ele

local counters (on each processor)

- int ln_vert
- int ln_ele

vertex data

- PetscReal * vertxyz
- int * vertl2g
- int * vertnewproc
- Vec vertvol
- int * vertprop

element data

- int * elevert

- int * eleprop
- int * elel2g
- int * elenewproc
- Vec elevol
- PetscReal elenmax
- int * elevertall
- int * **mesh2nodal_ia**
- int * **mesh2nodal_ja**
- int * **mesh2dual_ia**
- int * **mesh2dual_ja**

surface data

- int n_vert_bnd
- int n_bnd_fac
- int ln_bnd_fac
- int * vertbndg2bnd
- int * bndfacvert

material properties

- int n_prop
- PetscReal * propdat

Matrices and vectors

- Vec M
- Vec VHdem
- Vec VMs3
- PetscReal Edem
- Vec VHexchani
- PetscReal Eexchani
- Vec VHext
- PetscReal Eext
- Vec VHtot
- PetscReal Etot

26.1.1 Field Documentation

26.1.1.1 int* GridData::bndfacvert

table of vertices which belong to each triangular face

26.1.1.2 PetscReal GridData::Edem

magnetostatic energy

26.1.1.3 PetscReal GridData::Eexchani

exchange + uniaxial anisotropy energy

26.1.1.4 PetscReal GridData::Eext

Zeeman energy

26.1.1.5 int* GridData::elel2g

local to global element id mapping

26.1.1.6 int* GridData::elenewproc

assignment of elements to new processors

26.1.1.7 PetscReal GridData::elenmax

maximum edge length (vertex to vertex) in the mesh

26.1.1.8 int* GridData::eleprop

property id assigned to each element

26.1.1.9 int* GridData::elevert

each element's corner vertices (global ids)

26.1.1.10 int* GridData::elevertall

each element's corner vertices (bcast to all procs during init.)

26.1.1.11 Vec GridData::elevol

element volumes

26.1.1.12 int GridData::equil

equilibrium reached ? equil++ : equil=0

26.1.1.13 PetscReal GridData::escale

scaling factor for energy

26.1.1.14 PetscReal GridData::Etot

total energy

26.1.1.15 PetscReal GridData::hscale

scaling factor of all fields and magnetization to dimensionless units

26.1.1.16 int GridData::inp

number of AVS inp file (initially read or next number to be written)

26.1.1.17 PetscReal GridData::lenscale

scaling factor for length

26.1.1.18 int GridData::ln_bnd_fac

local number of faces on the boundary

26.1.1.19 int GridData::ln_ele

local number of elements

26.1.1.20 int GridData::ln_vert

local number of vertices

26.1.1.21 Vec GridData::M

magnetization

26.1.1.22 int GridData::mode

integration method (defined in allopt.txt)

26.1.1.23 int GridData::n_bnd_fac

number of boundary faces

26.1.1.24 int GridData::n_ele

number of elements

26.1.1.25 int GridData::n_prop

number of material properties (grains)

26.1.1.26 int GridData::n_vert

number of vertices

26.1.1.27 int GridData::n_vert_bnd

number of vertices on the boundary

26.1.1.28 PetscReal* GridData::propdat

NP material properties: 0: theta; 1: phi; 2: K1; 3: K2; 4: Js; 5: A; 6: e_cubic_1_x; 7: e_cubic_1_y; 8: e_cubic_1_z; 9: alpha; 10: e_cubic_2_x; 11: e_cubic_2_y; 12: e_cubic_2_z; 13: e_cubic_3_x; 14: e_cubic_3_y; 15: e_cubic_3_z; 16: anisotropy cubic ? 0-false-uniaxial : 1-cubic

26.1.1.29 char GridData::simname[256]

project name, basename for output files (*.inp, *.off, etc.)

26.1.1.30 PetscReal GridData::time

absolute time

26.1.1.31 PetscReal GridData::totvol

total volume of magnetic material (Ms>0)

26.1.1.32 PetscReal GridData::tscale

scaling factor for dimensionless time

26.1.1.33 PetscReal GridData::vequil

measure for equilibrium: max(dM/dt) TODO: RM

26.1.1.34 int* GridData::vertbndg2bnd

mapping global vertex id -> "boundary vertex counter"

26.1.1.35 int* GridData::vertl2g

mapping local vertex id -> global vertex id

26.1.1.36 int* GridData::vertnewproc

assignment of vertices to new processors

26.1.1.37 int* GridData::vertprop

property id assigned to each vertex

26.1.1.38 Vec GridData::vertvol

"volume" assigned to each vertex

26.1.1.39 PetscReal* GridData::vertxyz

(x,y,z) coordinates of all vertices

26.1.1.40 Vec GridData::VHdem

demagnetizing=magnetostatic field

26.1.1.41 Vec GridData::VHexchani

exchange + uniaxial anisotropy field

26.1.1.42 Vec GridData::VHext

external field

26.1.1.43 Vec GridData::VHtot

total field

26.1.1.44 Vec GridData::VMs3

vertex volume times Ms

The documentation for this struct was generated from the following file:

- griddata.h

26.2 magpar_scripts::Line Class Reference

Public Member Functions

- def **__init__**
- def **__str__**

Data Fields

- **start_point**
- **end_point**
- **vector**
- **length**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.3 magpar_scripts::Polygon Class Reference

Public Member Functions

- def **__init__**
- def **append_segment**
- def **point**
- def **vec**
- def **__str__**

Data Fields

- **segments**
- **length**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.4 magpar_scripts::Segment Class Reference

Public Member Functions

- def **__init__**
- def **__str__**

Data Fields

- **start**
- **line**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.5 magpar_scripts::Test_external_field_wire Class Reference

Public Member Functions

- def **setUp**
- def **testfield**

Data Fields

- **polygon**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.6 magpar_scripts::TestBiotSavart Class Reference

Public Member Functions

- def **setUp**
- def **testfield**

Data Fields

- **polygon**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.7 magpar_scripts::TestLineClass Class Reference

Public Member Functions

- def **setUp**
- def **testvector**
- def **testlength**

Data Fields

- **a**
- **b**
- **line**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.8 magpar_scripts::TestPolygonClass Class Reference

Public Member Functions

- def **setUp**
- def **testpoints**

Data Fields

- **polygon**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.9 magpar_scripts::TestVectorClass Class Reference

Public Member Functions

- def **setUp**
- def **testNeg**
- def **testAbs**
- def **testPow**
- def **testSimpleArithmetics**
- def **testOuter**
- def **testNormalize**

Data Fields

- **x**
- **y**
- **z**
- **a**
- **b**
- **c**

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

26.10 magpar_scripts::Vector Class Reference

Public Member Functions

- def **__init__**
- def **__getitem__**
- def **__setitem__**
- def **copy**
- def **__neg__**
- def **__abs__**
- def **__pow__**
- def **__add__**
- def **__iadd__**
- def **__sub__**
- def **__isub__**
- def **__mul__**
- def **__rmul__**
- def **__imul__**
- def **__div__**
- def **__idiv__**
- def **outer**
- def **normalize**
- def **__eq__**
- def **__ne__**
- def **__repr__**
- def **__str__**

Data Fields

- **x**
- **y**
- **z**

26.10.1 Detailed Description

The documentation for this class was generated from the following file:

- pytools/magpar_scripts.py

Index