magpar – an open source finite element micromagnetics package

Werner Scholz
Vienna University of Technology
http://magnet.atp.tuwien.ac.at/scholz/magpar/

Outline

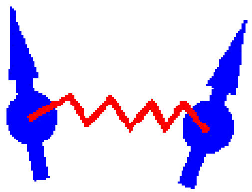
- Motivation
- Finite element micromagnetics
- Parallel linear algebra
- Program flow diagram
- Magnetization dynamics based on LLG
- Static energy minimzation
- Performance evaluation
- Implemented features
- System/software requirements, licenses

Motivation

- FE micromagnetics from scratch
- High performance parallel processing
- Static energy minimization + dynamic time integration (LLG)
- Free + open source software packages
- Platform independence

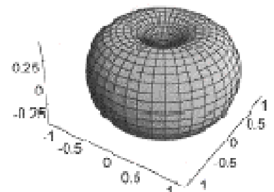
Finite Element Micromagnetics

exchange



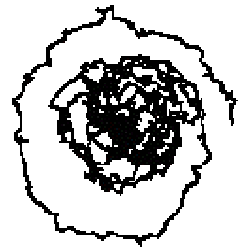
⇨ parallel spins

anisotropy



⇨ easy directions

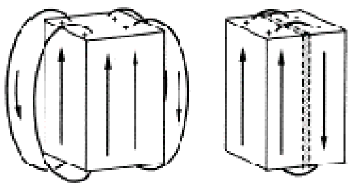
thermal activation



⇨ fluctuations

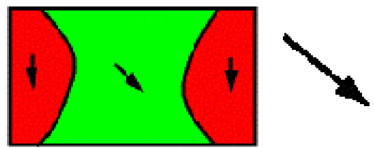
$$\frac{\partial \boldsymbol{J}}{\partial t} = -|\gamma| \boldsymbol{J} \times \boldsymbol{H}_{\mathrm{eff}} + \frac{\alpha}{J_s} \boldsymbol{J} \times \frac{\partial \boldsymbol{J}}{\partial t}$$

magnetostatics



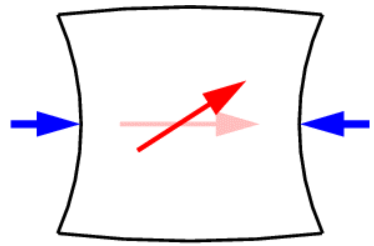
⇨ domains

external field



⇨ rotation

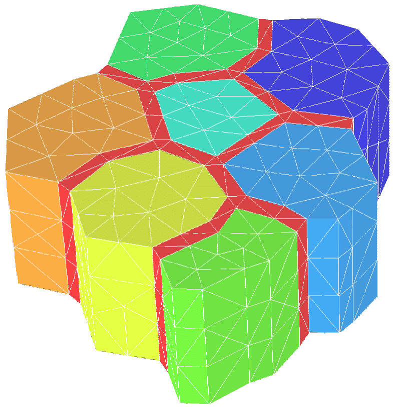
strain effects



⇨ "modified anisotropy"

magnet.atp.tuwien.ac.at

Finite Element Approach

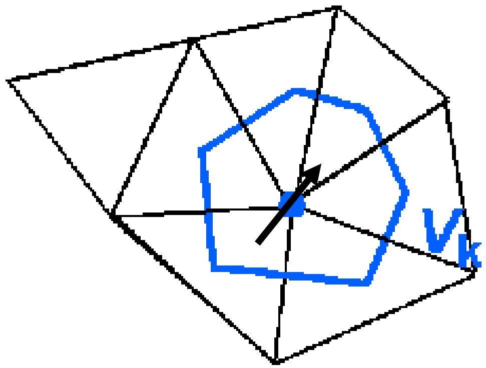


- divide particles into finite elements
  ⇨ triangles, tetrahedrons
- expand **J** with basis function φ

$$\vec{J}(\vec{x}) = \sum_{i=1}^{nodes} \vec{J}_i \varphi_i(\vec{x})$$

- energy as a function of $\mathbf{J}_1$, $\mathbf{J}_1$ … $\mathbf{J}_N$

$$E(\vec{J}_1,\vec{J}_2....\vec{J}_N)$$



- effective field

$$\vec{H}_k = -\frac{1}{V_k}\frac{\partial E(\vec{J}_1,\vec{J}_2....\vec{J}_N)}{\partial \vec{J}_k}$$

⇨ effective field on irregular grids
⇨ rigid magnetic moment at the **nodes**

Anisotropy energy 1

The magnetocrystalline anisotropy energy for uniaxial anisotropy is given by

$$E_{\mathrm{ani}} = \int_\Omega \sum_j K_1 (1 - (\boldsymbol{a} \cdot \boldsymbol{u}_j \eta_j)^2)\, dv \quad . \tag{3.23}$$

The gradient is given by

$$\frac{\partial E_{\mathrm{ani}}}{\partial u_{i,l}} = \int_\Omega \sum_j K_1 \frac{\partial}{\partial u_{i,l}} \left(1 - \left(\sum_k^{\{x,y,z\}} (a_k \cdot u_{j,k} \eta_j) \right)^2 \right) dv \tag{3.24}$$

$$\begin{aligned} \frac{\partial}{\partial u_{i,l}} \left(\sum_k^{\{x,y,z\}} (a_k \cdot u_{j,k} \eta_j) \right)^2 &= 2 \sum_k^{\{x,y,z\}} (a_k \cdot u_{j,k} \eta_j) \cdot \sum_m^{\{x,y,z\}} (a_m \delta_{ij} \delta_{lm} \eta_j) = \\ &= 2 \sum_k^{\{x,y,z\}} (a_k \cdot u_{j,k} \eta_j) \cdot a_l \eta_i \end{aligned} \tag{3.25}$$

Anisotropy energy 2

and we get the result

$$\frac{\partial E_{\text{ani}}}{\partial u_{i,l}} = -2K_1 a_l \int_\Omega \sum_j \sum_k^{\{x,y,z\}} a_k u_{j,k} \eta_j \cdot \eta_i \, dv \quad . \tag{3.26}$$

This can be rewritten in matrix notation as

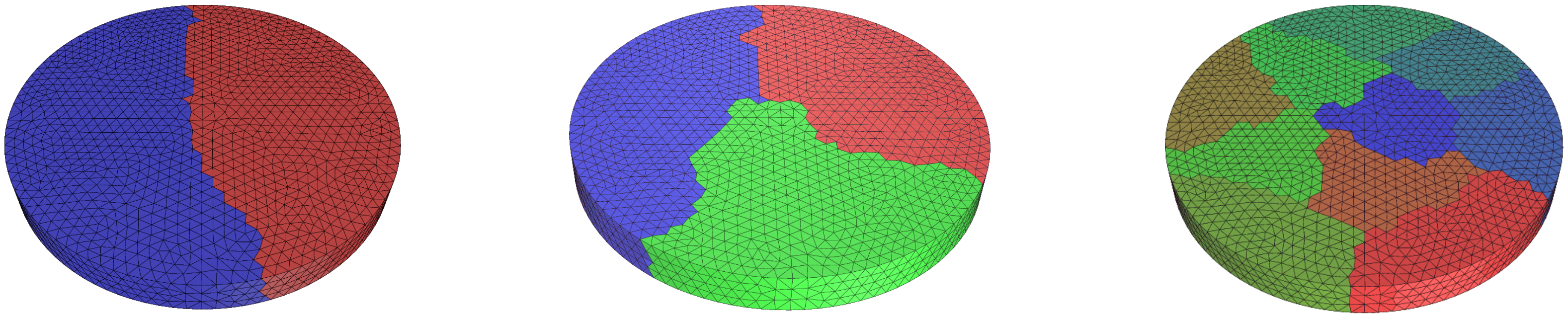
$$\boldsymbol{g}_{\text{ani}} = G_{\text{ani}} \cdot \boldsymbol{u} \tag{3.27}$$

with

$$G_{\text{ani},i,l} = -2K_1 a_l \int_\Omega \sum_k^{\{x,y,z\}} a_k \eta_j \cdot \eta_i \, dv \quad . \tag{3.28}$$

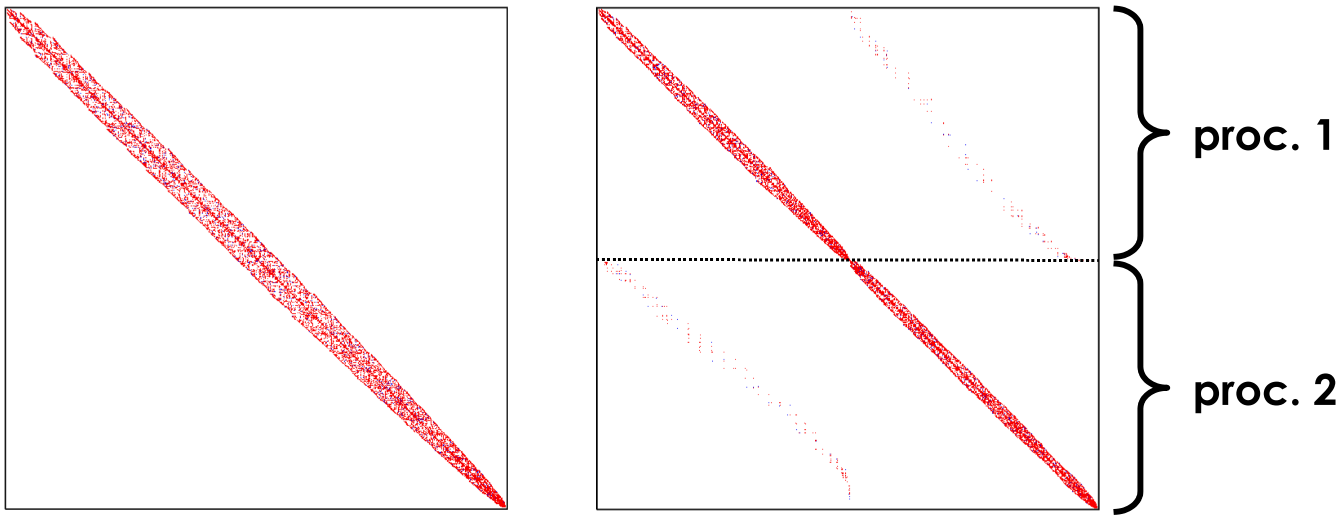
Mesh Partitioning

Partitioning pattern for 2, 4, and 10 processors



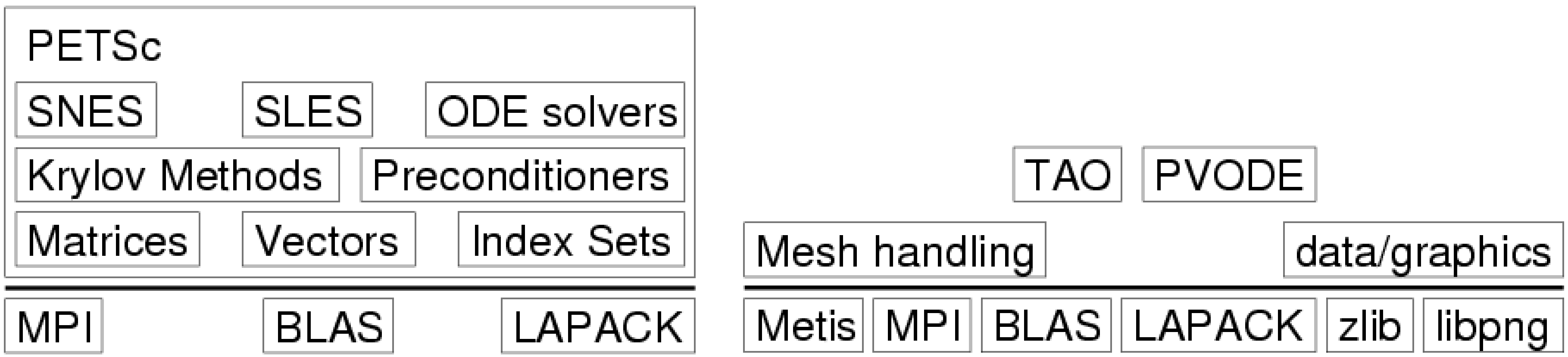
Sparsity pattern of the stiffness matrix

single processor — two processors



magpar - Software Modules

- **Preprocessing: AutoCAD, Patran, GiD**
- **PETSc library**
  Portable, Extensible Toolkit for Scientific Computation
- **MPI library**
- **METIS: Multilevel partitioning**
- **TAO library – energy minimization**
- **SUNDIALS (PVODE) – LLG time integration**
  SUite of Nonlinear and DIfferential/ALgebraic equation Solvers
- **Postprocessing: PNG snapshots, GeomView, (Micro)AVS**



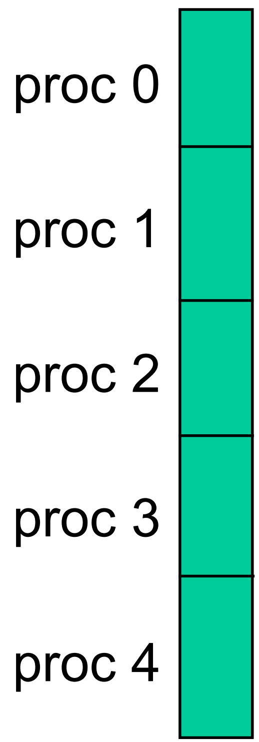
Parallel linear algebra

We need...

- **Data structures for**
  - **Vectors**
  - **Matrices**
- **Assembly of vectors and global matrices**
- **Mathematical operations**
  - **Vectors (BLAS 1)**
  - **Matrix-Vector (BLAS 2)**
  - **Matrix-Matrix (not supported by PETSc!)**

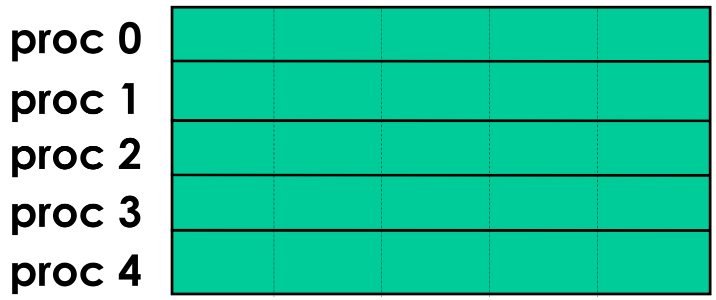
Vectors

- **PETSc vectors**
  - Objects for storing field solutions, right-hand sides, etc.
  - Each process locally owns a subvector of contiguously numbered global indices
  - VecCreate
    - MPI_Comm - processors that share the vector
    - number of elements local to this processor
    - or total number of elements
  - VecSetType(Vec,VecType)
    - Where VecType is
      - VEC_SEQ, VEC_MPI, or VEC_SHARED



Matrices

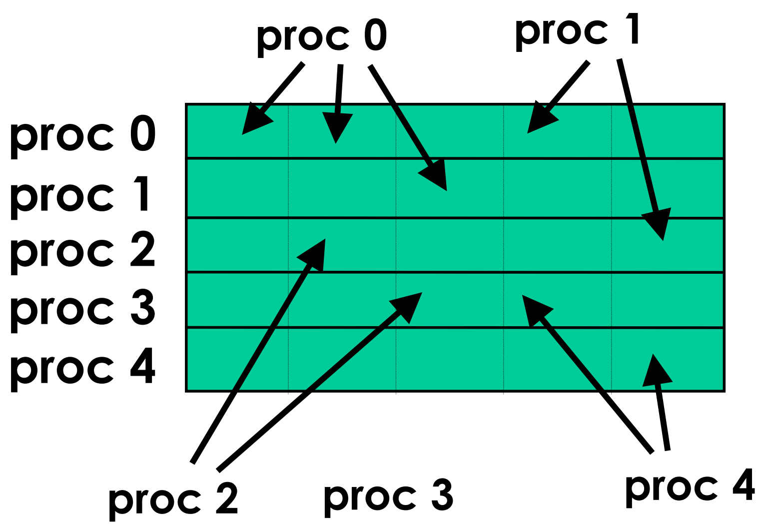
- **PETSc matrices**
  - Objects for storing linear operators



  - MatCreate
    - MPI_Comm - processors that share the matrix
    - number of local/global rows
  - MatSetType(Mat,MatType)
    - where MatType is one of
      - default sparse AIJ: MPIAIJ, SEQAIJ (no sparsity pattern required)
      - block sparse AIJ (for multi-component PDEs): MPIAIJ, SEQAIJ
      - symmetric block sparse AIJ: MPISBAIJ, SAEQSBAIJ
      - block diagonal: MPIBDIAG, SEQBDIAG
      - dense: MPIDENSE, SEQDENSE
      - matrix-free
      - etc.

Parallel Matrix and Vector Assembly

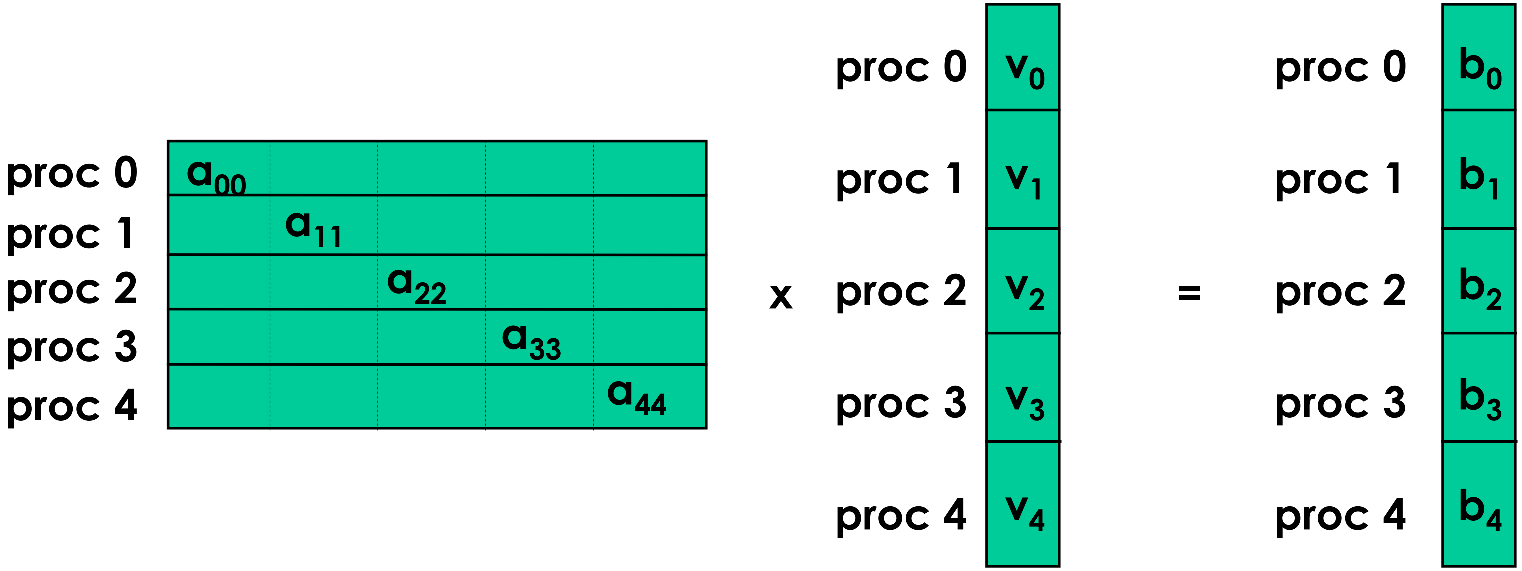
- Any processor may generate any entries in vectors and matrices
- Entries generated on one processor may be (ultimately) stored on another
- PETSc automatically moves data during the assembly process if necessary



Matrix-Vector Multiplication (1)

Matrix x Vector = Vector

A x v = b

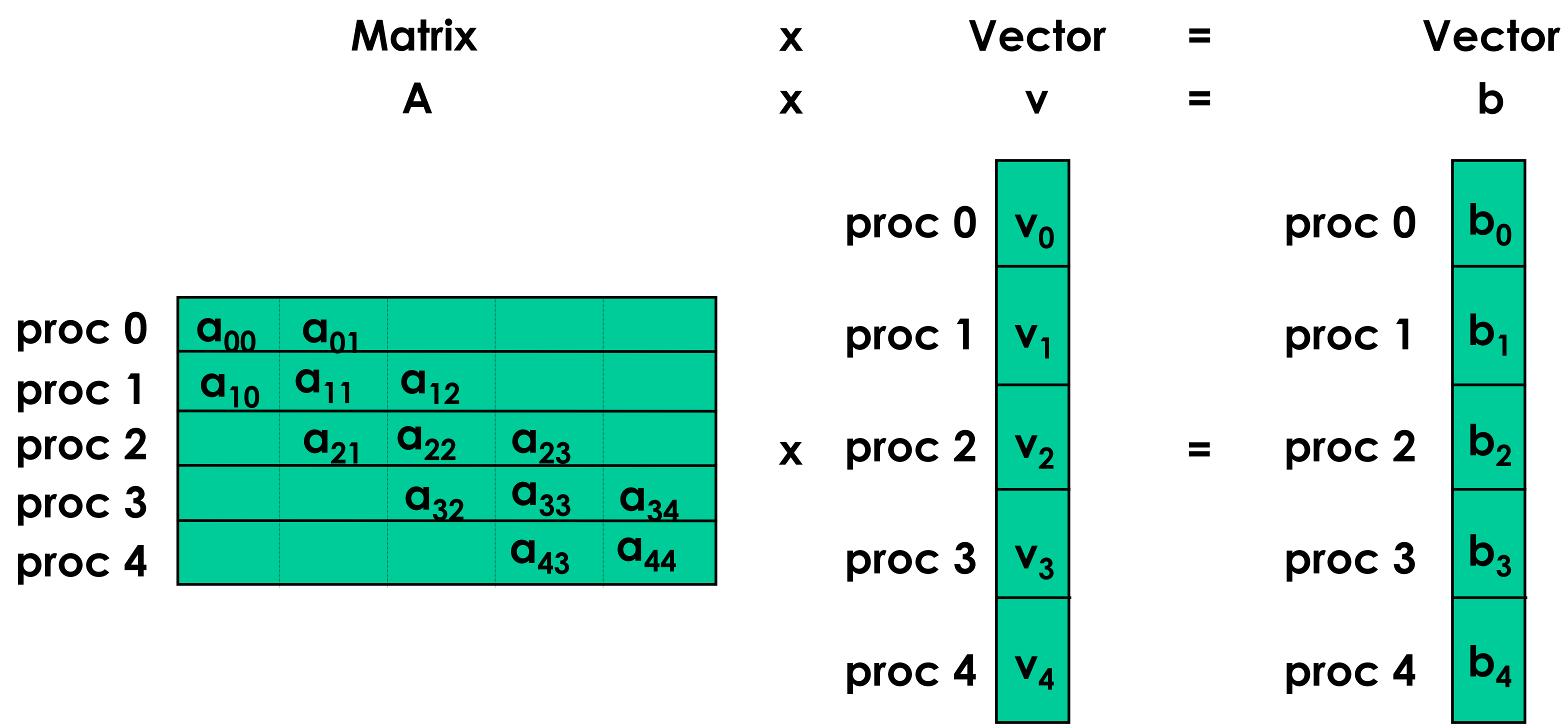


proc 0 $a_{00} * v_0 = b_0$ ...no communication required

proc 1 $a_{11} * v_1 = b_1$ ...no communication required

...

Matrix-Vector Multiplication (2)



proc 0 $a_{00}*v_0 + a_{01}*v_1 = b_0$ ...proc 0 requires v_1 from proc 1

proc 1 $a_{10}*v_0 + a_{11}*v_1 + a_{12}*v_2 = b_1$ ...proc 1 requires v_0 from proc 0 and v_2 from proc 2

...

Energy Minimization using TAO

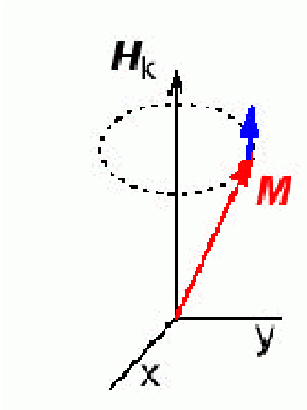
$$E_{tot} = E_{exch} + E_{ani} + E_{mag} + E_{Zee}$$

- **Magnetization given in Cartesian coordinates**
- **Energy gradient calculated in spherical coordinates**
- **TAO provides several solvers**
- **LMVM (limited memory variable metric) quasi-Newton method**
- **requires only energy and gradient (no Hessian)**
- **second-order information approx. by FD**

LLG Integration using PVODE

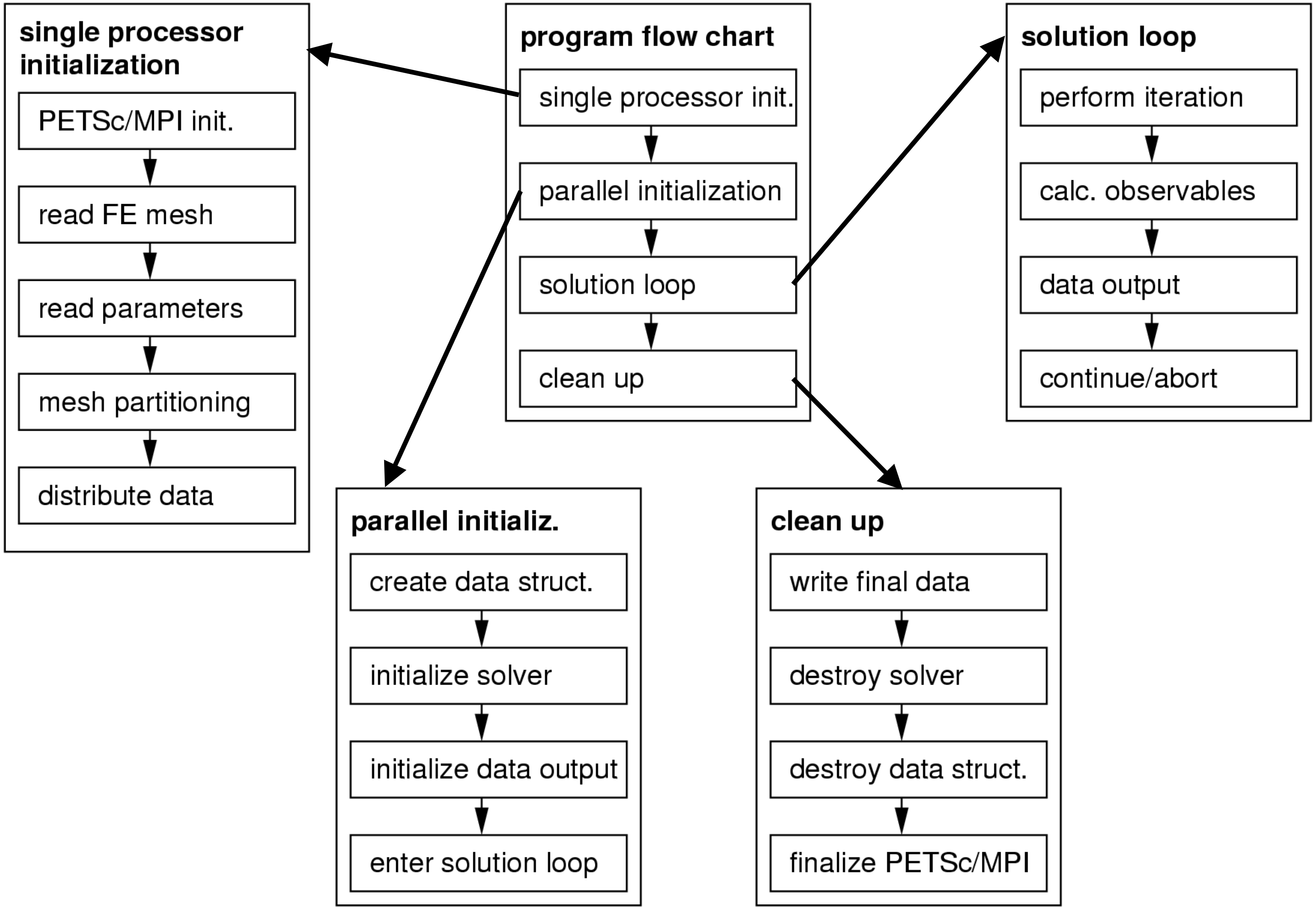
$$\frac{\partial \mathbf{J}}{\partial t} = -|\gamma| \mathbf{J} \times \mathbf{H}_{\mathrm{eff}} + \frac{\alpha}{J_s} \mathbf{J} \times \frac{\partial \mathbf{J}}{\partial t}$$

$$\mathbf{H}_{\mathrm{eff}} = -\frac{\delta E_{tot}}{\delta \mathbf{m}}$$



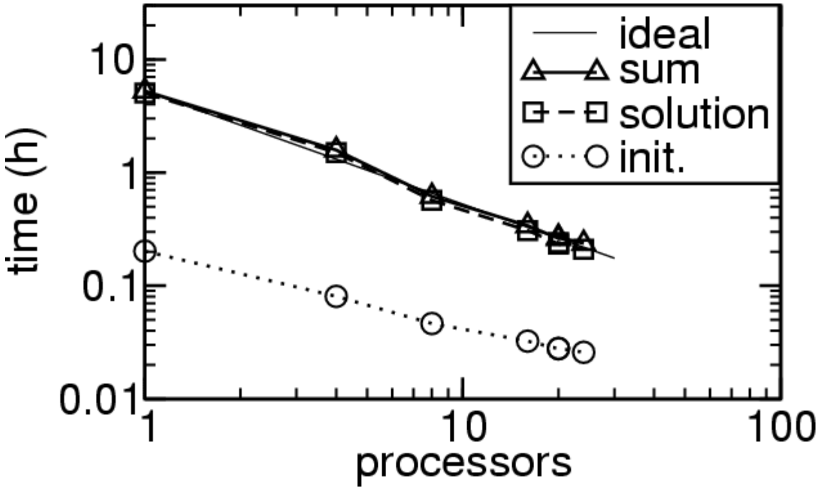
- Integration using ODE solver PVODE
- BDF and Adams-Moulton formulas
- variable step size, variable order
- varied automatically and dynamically
- preconditioning

Program Overview



Parallel Efficiency

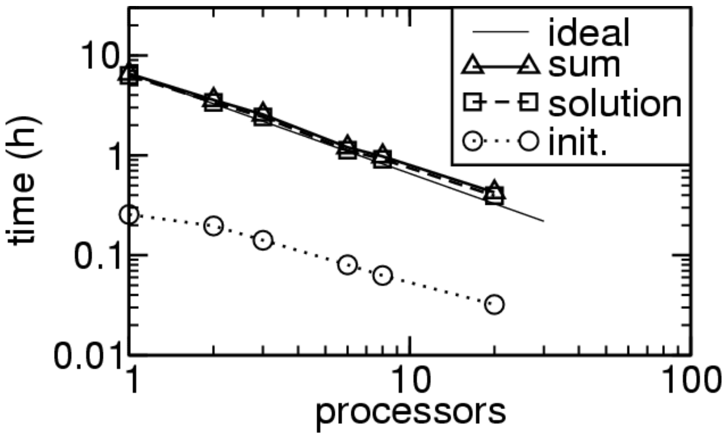
TAO (energy minimization)

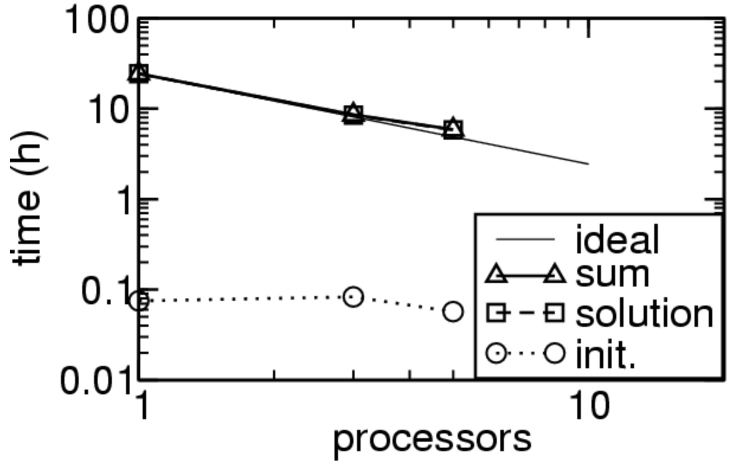


- TAO:
  speedup 23.9 on 24 processors
  superlinear behavior on 8 processors
  (caching effects)

- PVDODE:
  speedup 16 on 20 processors

PVODE (dynamic LLG integration)





Implemented features 1

- **Debugging and optimized compilation**
- **Easy activation of optional components**
- **Consistency checking**
  assert statements
- **Memory allocation tracking**
  PetscMalloc, PetscFree, memory usage statistics
- **C++ compatible**
  required by TAO
- **Problem independent parallelization**
- **Profiling**
  timing in every subroutine
- **Performance evaluation**
  timing, FLOP count (PVODE missed!)

Implemented features 2

- **Mesh import**
  - MSC/Patran neutral file (no surface triangles)
  - AVS inp file (Patran neutral file <u>not</u> required)
  - GiD
- **Mesh analysis**
  - element and node volumes (max,min,avg)
  - edge lengths (max,min,avg)
  - element quality check
  - model bounding box
  - volume by property id
- **Mesh distortion**
  mimic surface/interface roughness
- **Mesh refinement**
  full regular refinement before partitioning:
  x 8^n number of nodes and elements

Implemented features 3

- Micromagnetics
  - Uniaxial anisotropy
  - Exchange
  - Magnetostatic field (hybrid FEM/BEM)
  - External field (quasistatic, sweeping, rotating)
- Dynamic LLG integration using PVODE
- Static energy minimization using TAO
- Data output
  - Geomview output
  - Log file
  - PNG files
  - “sampling line”

System/software requirements

- **Hardware/Software platform, which is supported by PETSc**
  http://www-fp.mcs.anl.gov/petsc/docs/machines.html
  IBM RS6000 including IBM SP, SGI running IRIX, 64 bit SGI including Origin and PowerChallenge, Convex Exemplar running HPUX, HP running HPUX, Sun Sparcstations running Solaris, Cray T3D/E, DEC Alpha OSF (Tru64), Intel processors running Linux, FreeBSD, Windows, Mac OS X, PC Running BeOS
- **MPI library**
  The Message Passing Interface (MPI) standard
  http://www-unix.mcs.anl.gov/mpi/mpich/
- **PETSc library**
  Portable, Extensible Toolkit for Scientific Computation
  http://www-fp.mcs.anl.gov/petsc/
- **GNU make**
  http://www.gnu.org/software/make/make.html
- **C/C++ compiler**
  http://www.gnu.org/software/gcc/
- **METIS**
  Multilevel partitioning
  http://www-users.cs.umn.edu/~karypis/metis/

Optional components

- **TAO library – energy minimization**
  Toolkit for Advanced Optimization
  http://www-fp.mcs.anl.gov/tao/
- **SUNDIALS (PVODE) – LLG time integration**
  SUite of Nonlinear and DIfferential/ALgebraic equation Solvers
  http://acts.nersc.gov/pvode/main.html
  http://www.llnl.gov/CASC/sundials/
- **zlib – compression of output data**
  A Massively Spiffy Yet Delicately Unobtrusive Compression Library
  http://www.gzip.org/zlib/
- **libpng – output of graphics files**
  The official PNG reference library
  http://www.libpng.org/pub/png/libpng.html

Licenses

- **MPICH:** free
  http://www-unix.mcs.anl.gov/mpi/mpich/mpich-license.txt
- **PETSc library:** free
  http://www-fp.mcs.anl.gov/petsc/docs/copyright.html
- **GNU make, GNU C:** free: GPL
  http://www.gnu.org/licenses/licenses.html
- **METIS:** free
  http://www-users.cs.umn.edu/~karypis/metis/metis/faq.html#distribute
- **TAO:** unknown (same as PETSc ? – also ANL tool)
- **SUNDIALS:** free (BSD style license)
  http://www.llnl.gov/CASC/sundials/download/cvode_par_agree.html
- **zlib:** free (OSI approved license)
  http://www.gzip.org/zlib/zlib_license.html
- **libpng:** free (OSI approved license)
  http://www.libpng.org/pub/png/src/libpng-LICENSE.txt

Summary

- Finite element micromagnetics: easy to formulate with matrix-vector operations
- Suitable for parallelization
- Efficient implementation and parallelization using free open source software packages
- PETSc library compiles and runs on a variety of hardware platforms ranging from simple clusters of PCs to massively parallel supercomputers
- Excellent speed up measured
- Universal tool: soft and hard magnets, inhomogeneous microstructures, static and dynamic solvers
- Free open source package

Acknowledgement

- Josef Fidler
- Thomas Schrefl
- Dieter Suess, Rok Dittrich, Vassilios Tsiantos, Hermann Forster
- Roy Chantrell

References

- **WWW (download, documentation): http://magnet.atp.tuwien.ac.at/scholz/magpar/**
- **email: magpar@magnet.atp.tuwien.ac.at**
- **mailing-lists: majordomo@magnet.atp.tuwien.ac.at**

- **Papers:**
  - dissertation: Werner Scholz, "Scalable Parallel Micromagnetic Solver for Magnetic Nanostructures"
  - W. Scholz, J. Fidler, T. Schrefl, D. Suess, R. Dittrich, H. Forster, V. Tsiantos, Comp. Mat. Sci 28 (2003) 366-383.
  - 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. (2003), submitted.