High-performance multiphysics FEM simulations in the Sparselizard open source library

Alexandre Halbach, Tampere University
This work is supported by Superconducting Magnets Beyond 20 T, Academy of Finland, under Grant 324887
Thanks

Kindly provided the Feather M2 magnet dimensions.
Oh, yet another open source FEM software!

Yes, but unlike others!

- High-performance
- Strongly multiphysics
- Concise to write

sparselizard.org
Feather M2 magnet geometry

Created by J. Ruuskanen

Air
Feather M2 magnet geometry

Magnetic shell
Feather M2 magnet geometry

Top and bottom coils
Feather M2 magnet geometry

Irons
Feather M2 magnet geometry

Electric supply
Feather M2 magnet geometry

Mesh it with gmsh (1.1 million tetrahedra, few mins on my laptop)
Sparselizard demos - Region tags

- Download the static library or compile it
- Create a text file main.cpp
- Define region tags

```cpp
#include "sparselizard.h"
using namespace sl;

void createmesh(void);

int main(void)
{
    int air = 3000, topcoil = 3100, botcoil = 3200, topiron = 4100, botiron = 4200, lefttopiron = 4300;
    int leftbotiron = 4400, coreiron = 5000, bnd = 1, vintop = 2200, vouttop = 2300, vinbot = 2400, voutbot = 2500;

    mesh mymesh;
    mymesh.selectskin(bnd);
    mymesh.load("fm2.msh");

    int coils = selectunion({topcoil, botcoil});
    int irons = selectunion({topiron, botiron, lefttopiron, leftbotiron, coreiron});
    int vins = selectunion({vintop, vinbot});
    int vouts = selectunion({vouttop, voutbot});
    int solid = selectunion({coils, irons});
    int notmagnetic = selectunion({air, coils});
    int all = selectall();
}
```
➡️ **Ports feature added for KiCAD**

➡️ **Solve** \( \nabla \cdot (\sigma \nabla v) = 0 \) with 8 kA supply

➡️ **Takes 1.6 sec on my laptop**

```cpp
field v("h1");
v.setorder(all, 1);

port Vt, It, Vb, Ib;

v.setport(vintop, Vt, It);
v.setport(vinbot, Vb, Ib);

v.setconstraint(vouts);

double sigma = 1e8;
expression j = sigma * -grad(v);

formulation elec;

elec += It * 8000.0;
elec += Ib * 8000.0;

elec += integral(coils, -sigma * grad(dof(v)) * grad(tf(v)));

elec.solve();

//v.write(coils, "v.vtu", 1);
//j.write(coils, "j.vtu", 1);
```
Unsurprisingly this is the electric potential profile:
- Saturation is taken into account (see online)
- Use A-v formulation (for a H-phi example see online)
- Takes 57 sec on my laptop (+- 1.2 M unknowns)

```
# Code snippet

double mu0 = 4*getpi() * 1e-7;

parameter mu;

mu.all = mu0;
mu.iron = 1000 * mu0;

spanningtree spantree({bnd});

field a("hcurl", spantree);

a.setgauge(all);
a.setorder(all, 0);
a.setconstraint(bnd);

formulation magnetostatics;
magnetostatics += integral(all, 1/mu * curl(dof(a)) * curl(tf(a)));
magnetostatics += integral(coils, -j * tf(a));
magnetostatics.solve();

std::cout << "B center is " << norm(curl(a)).interpolate(all, {0,0,0})[0] << " T" << std::endl;
```
Sparselizard demos - Static magnetic field

- With saturation ($B_{\text{center}} = 4.1$ T): 3 min run time (3 NL its)
- $B_{\text{center}}$ is already very accurate here!
Sparselizard demos - Static magnetic field

- Now use a.setorder(all, 2): 10+ million unknowns
- 30 min for 3 NL its on 16 core CPU
Sparselizard demos - Static magnetic field
Sparselizard demos - Static magnetic field
Sparselizard demos - Mechanical stresses

- See magnetostriction and mag. force sandbox examples online
- Include prestress + geometric nonlinearity in one line change
- Need plasticity? See H. Milanchian’s talk
- In the unsaturated case simply:

```python
field u("h1xyz");
setorder(coils, 2);

setconstraint(vins);
setconstraint(vouts);

expression b = curl(a);

formulation elasticity;

elasticity += integral(coils, predefinedelasticity(dof(u), tf(u), 150e9, 0.3));
// Magnetostatic Maxwell stresses:
elasticity += integral(all, predefinedmagnetostaticforce(tf(u, coils), b/mu, mu ));

elasticity.solve();
u.write(coils, "u.vtu", 2);
```
Sparselizard demos - hp-adaptive 3D tape quench

▶ See video
Need to run larger problems? DDM is available for any physic.
100 M dofs 3D acoustic freq. analysis: 40 min/100 CPUs
Thank you for your attention! See you at www.sparselizard.org