Towards a comparison of h- and p- Refinement in a Finite Element Maxwell Time Domain Solver

Christof Kraus, Andreas Adelmann (PSI), Peter Arbenz (ETHZ), Marcus Wittberger (PSI)

October 4, 2006

Motivation and Requirements

$\therefore$ track particles in complicated accelerator structures

$\therefore$ calculate the fields self-consistantly i.e. taking into account the field induced by the particles themselves

Maxwell-solver

$$\begin{aligned}
\frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t} - \nabla \wedge \mathbf{B} &= -\mu_0 \mathbf{J} \\
\frac{\partial \mathbf{B}}{\partial t} + \nabla \wedge \mathbf{E} &= 0 \\
\nabla \cdot \mathbf{E} &= \frac{\rho}{\varepsilon_0} \\
\nabla \cdot \mathbf{B} &= 0
\end{aligned}$$

& boundary conditions.

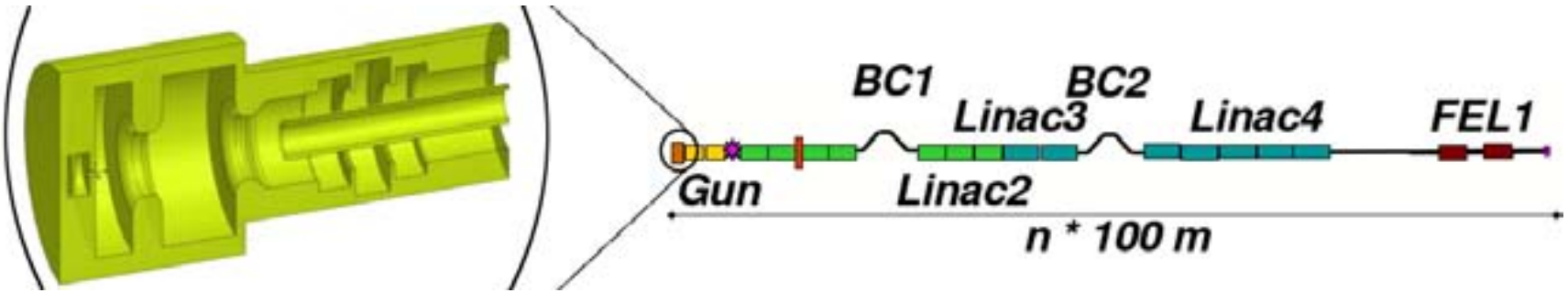
$\leftarrow$

$\rightarrow$

PIC-code

$$\begin{aligned}
\mathbf{a} &= \frac{q}{m}(\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \\
\mathbf{x} &= \mathbf{x}_0 + \mathbf{v}_0(t - t_0) \\
&+ \int_{t_0}^{t}\int_{t_0}^{t'} \mathbf{a}(t'')dt'dt''
\end{aligned}$$

In accelerator structures we have complicated geometries and large scale differences



$\Rightarrow$ a Finite Element Method will be used

$\Rightarrow$ a large linear system has to be solved iteratively at every time step.

Therefore:

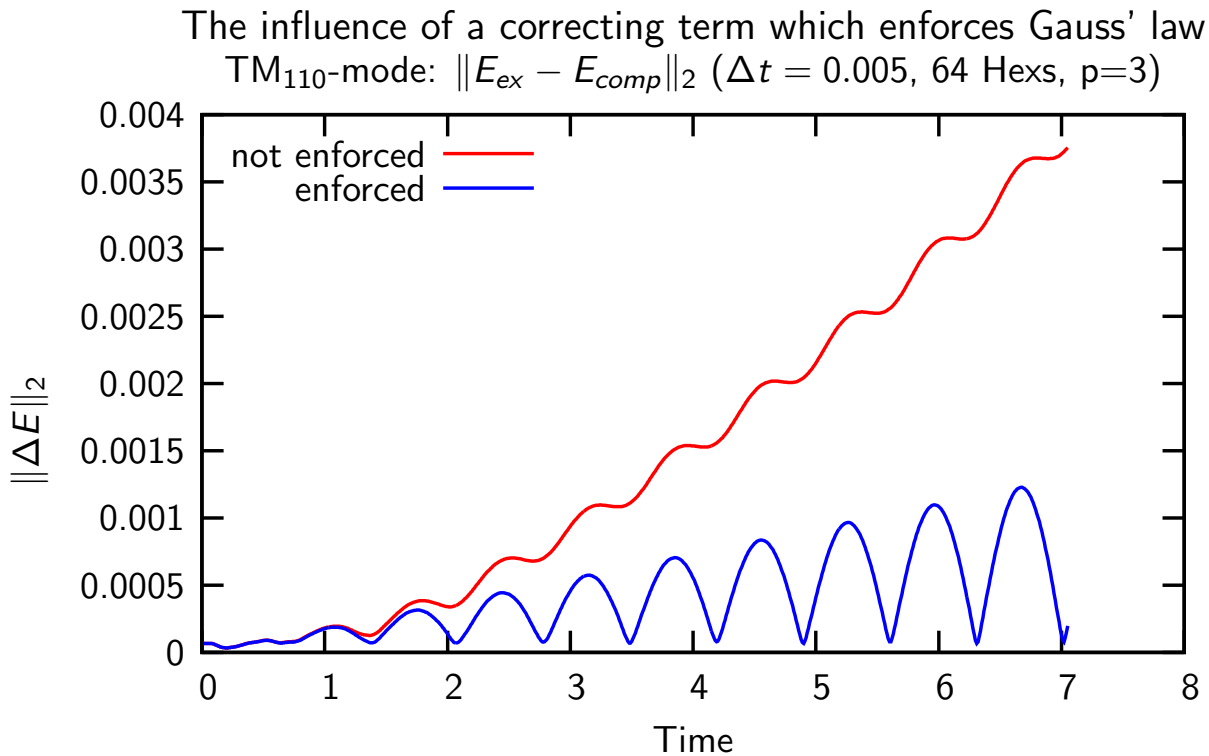
- $\therefore$ good choice of basis functions is important
- $\therefore$ good preconditioners are needed
- $\therefore$ code has to run on distributed memory clusters due to the sizes of the problems in focus $\Rightarrow$ scalability w.r.t. the number of processors is an issue

Charge conservation combined with the Maxwell equations implies that Gauss' Law is satisfied at any time step provided it holds at $t = t_0$.

In discrete space this is not necessarily true!

Charge conservation combined with the Maxwell equations implies that Gauss' Law is satisfied at any time step provided it holds at $t = t_0$.

In discrete space this is not necessarily true!



$\Rightarrow$ Gauss' Law has to be enforced explicitly.
An ansatz with Lagrange multipliers is used (see Assous et al, J. Comput. Phys., 1993). This yields a saddle point problem:

$$\begin{aligned} \frac{\partial^2 \mathbf{E}}{\partial t^2} \quad + c^2 \nabla \wedge \nabla \wedge \mathbf{E} - \nabla \phi \quad &= -\frac{1}{\varepsilon_0} \frac{\partial \mathbf{J}}{\partial t} \\ \nabla \cdot \mathbf{E} \qquad\qquad\qquad\qquad\qquad\qquad\quad &= \frac{\rho}{\varepsilon_0} \end{aligned}$$

With PEC boundary conditions:
$\Rightarrow$ Find $(\mathbf{E}(t), \Phi(t)) \in \mathrm{H}(\mathrm{curl}, \Omega) \times \mathrm{H}_0^1(\Omega)$ such that

$$\int_\Omega \frac{\partial^2 \mathbf{E}}{\partial t^2} \cdot \mathbf{F} d\mathbf{x} \quad + c^2 \int_\Omega (\nabla \wedge \mathbf{E}) \cdot (\nabla \wedge \mathbf{F}) d\mathbf{x} \qquad - \int_\Omega \nabla \Phi \cdot \mathbf{F} d\mathbf{x} \quad = -\frac{1}{\varepsilon_0} \int_\Omega \frac{\partial \mathbf{J}}{\partial t} \cdot \mathbf{F} d\mathbf{x}$$

$$\forall \mathbf{F} \in \mathrm{H}(\mathrm{curl}, \Omega),$$

$$\int_\Omega \mathbf{E} \, \nabla \varphi \, d\mathbf{x} \qquad\qquad\qquad\qquad\qquad\qquad = \frac{1}{\varepsilon_0} \int_\Omega \rho \, \varphi \, d\mathbf{x}$$

$$\forall \varphi \in \mathrm{H}_0^1(\Omega).$$

Written as matrices:

$$\mathbb{M}\mathbf{u} = \mathbf{f} \quad \Rightarrow \quad \begin{pmatrix} \mathbb{M} & \mathbb{C} \\ \mathbb{C}^T & \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

The corresponding matrix is symmetric and indefinite due to the large zero block $\Rightarrow$ causes some troubles when applying f.e. a cg-solver to it. But the problem can be reformulated with help of the gradient matrix $\mathbb{Y}$ and the Poisson matrix $\mathbb{H}$. Since $\mathbb{C} = \mathbb{M}\mathbb{Y}$ and $\mathbb{H} = \mathbb{Y}^T\mathbb{M}\mathbb{Y}$

$$\begin{aligned} \mathbb{M}\tilde{\mathbf{u}} &= \mathbf{f} \\ \mathbb{H}\mathbf{v} &= -\mathbf{g} + \mathbb{Y}^T\mathbf{f} \\ \mathbf{u} &= \tilde{\mathbf{u}} - \mathbb{Y}\mathbf{v}. \end{aligned}$$

Two different frameworks for high order edge and face elements are used:

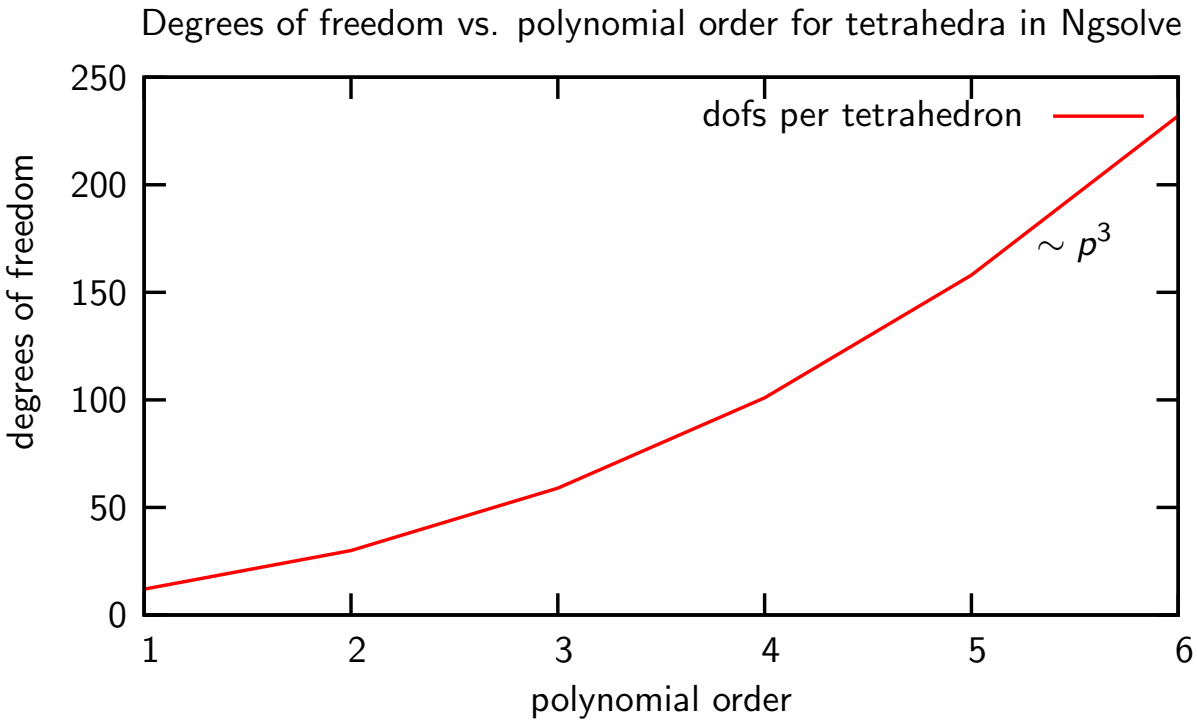
$\therefore$ Femster (P. Castillo et al., CMES 2002) uses interpolatory bases of Silvester-Lagrange or alternatively spectral type; makes use of discrete differential forms

$\therefore$ Ngsolve uses hierarchical bases which provide a local complete sequence (J. Schöberl, S. Zaglmayr, Compel 2005). As a consequence they yield a very simple gradient matrix $\mathbb{Y}$.

Use of high order basis functions

High order basis functions are well adapted to approximate smooth electric and magnetic fields: higher approximation rate is expected
$\Rightarrow$ less degrees of freedom needed to get the same error
Though

- $\therefore$ they lead to denser matrices
- $\therefore$ the condition numbers get worse
- $\therefore$ fields around corners and edges are not approximated as well
- $\therefore$ more degrees of freedom per element



- ∴ calculations in frequency domain show reduction in computation time
- ∴ polynomial order can be raised/lowered elementwise where needed (adaptively)
- ∴ mesh can still be refined (adaptively)

Methods and Benchmarks

For the comparison of h- and p-refinement we used the unit cube and a cylinder with PEC boundary conditions:

Both problems have analytic solution and known eigenfrequencies. But they do not have singularities at edges or corners!

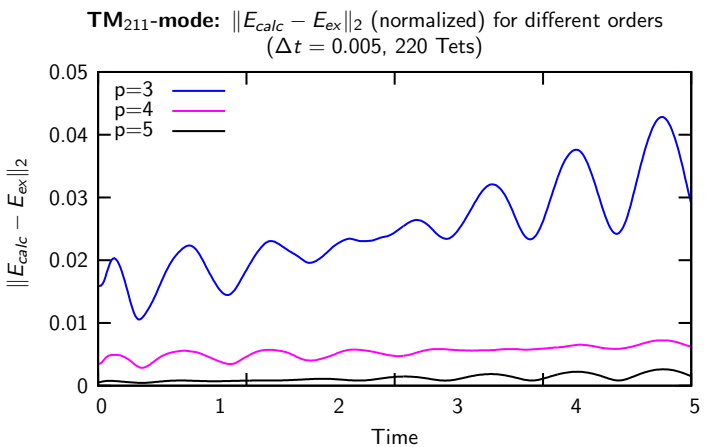
Two methods were used to validate our codes:

$\therefore$ initialize the fields with the exact solution and compare the computed with the exact solution after every timestep

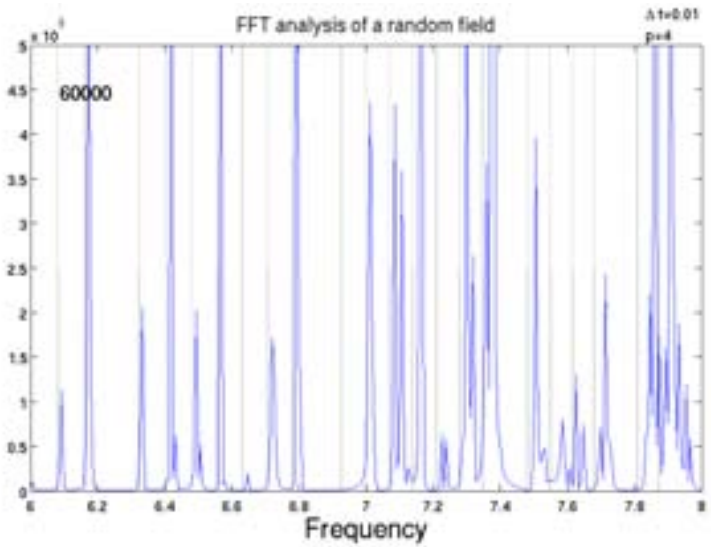
$\therefore$ initialize the field randomly and compare the spectrum with the known eigenfrequencies after a few thousand timesteps

The output of the two is:

an error plot

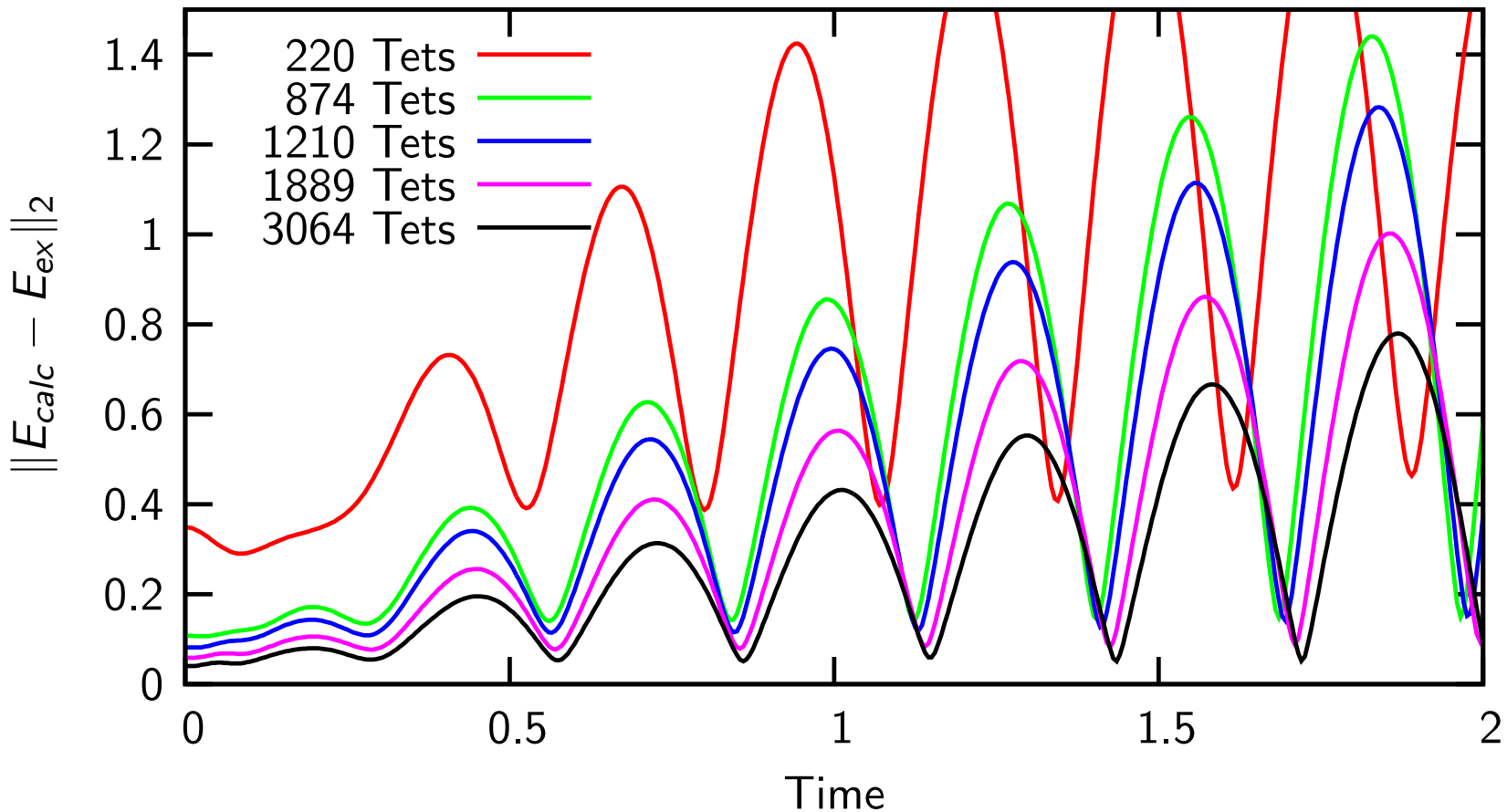


a spectrum

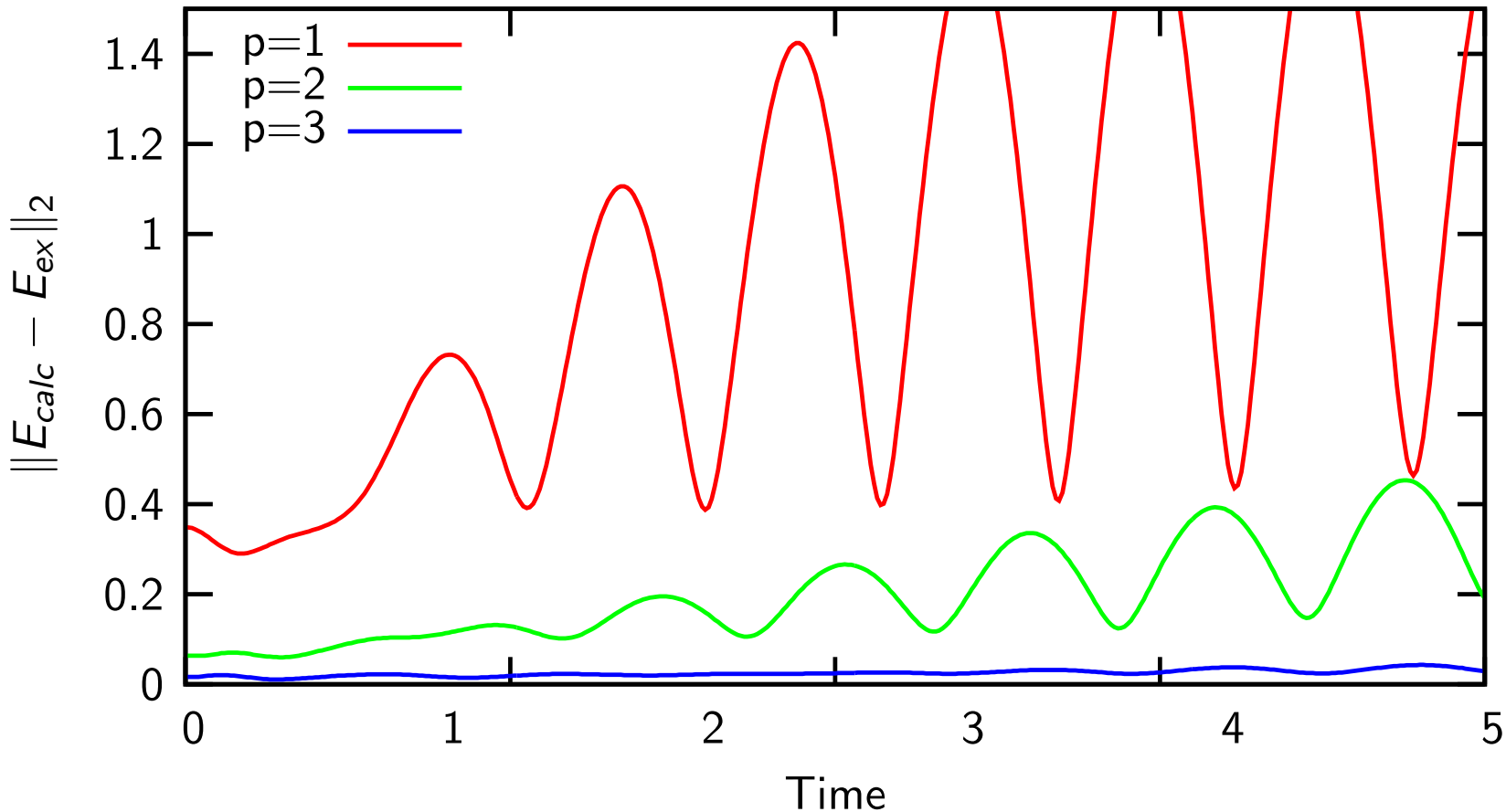


Results

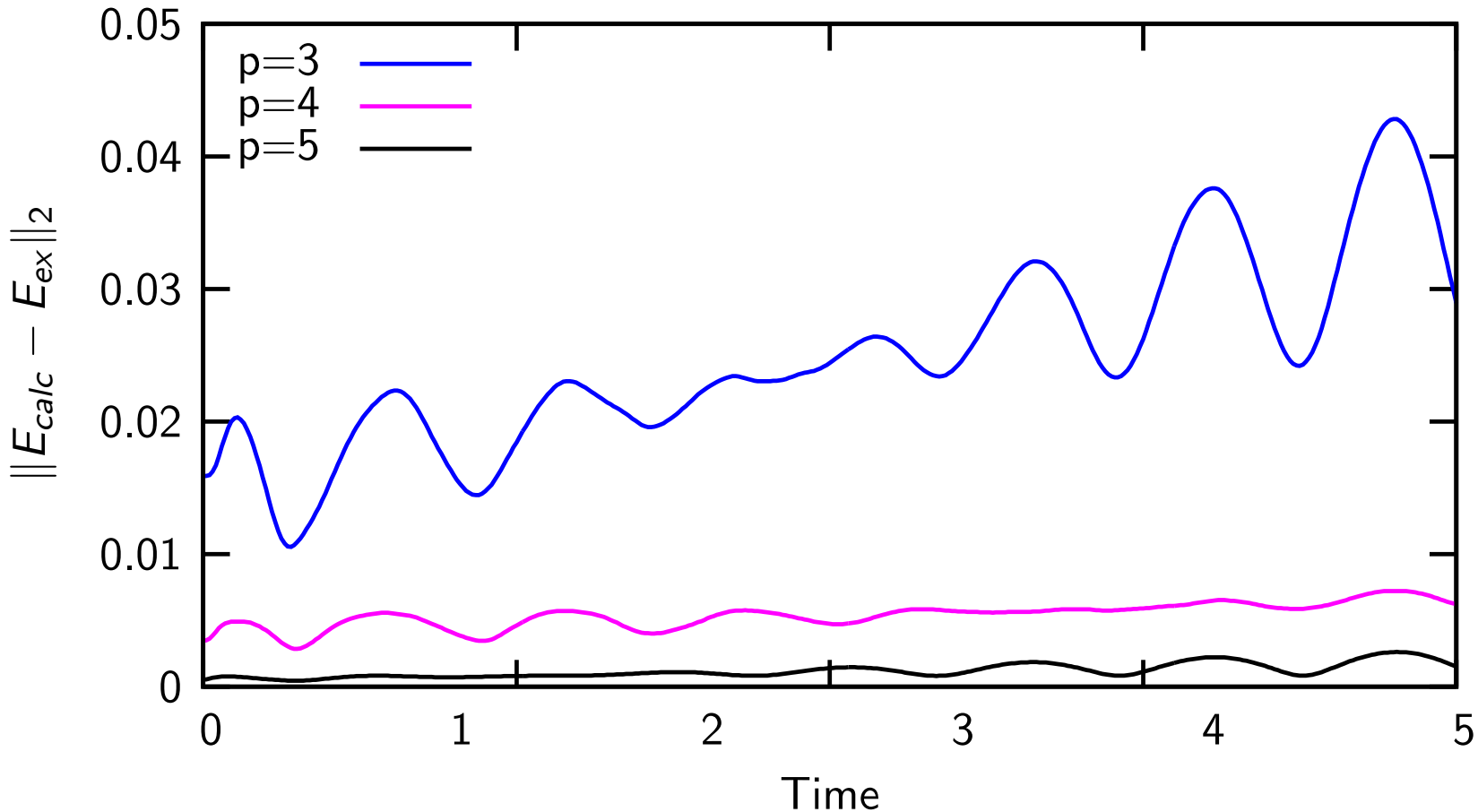
TM$_{211}$-mode: $\|E_{calc} - E_{ex}\|_2$ (normalized) for different meshes ($\Delta t = 0.005$, p=1)



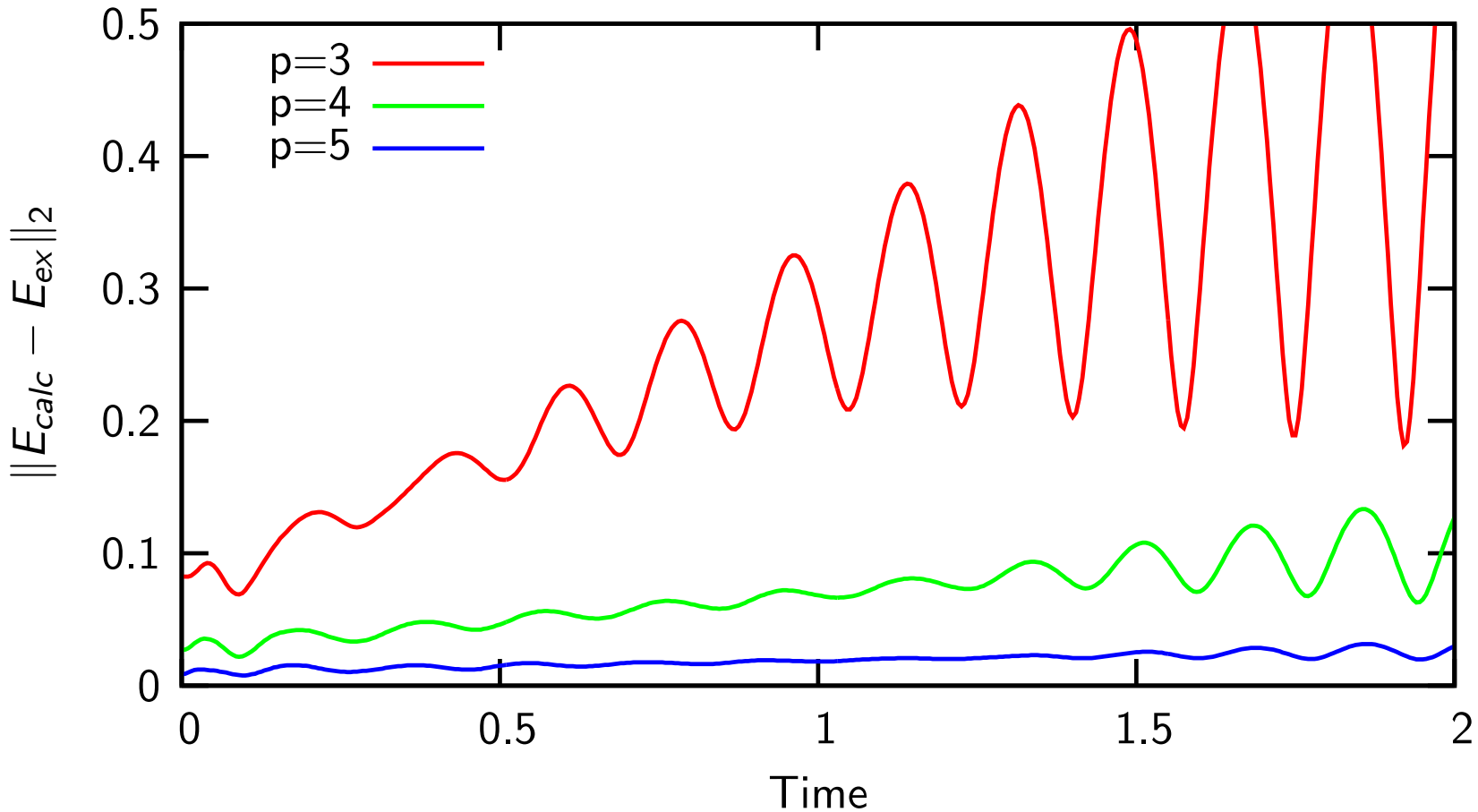
TM$_{211}$-mode: $\|E_{calc} - E_{ex}\|_2$ (normalized) for different orders
($\Delta t = 0.005$, 220 Tets)



TM$_{211}$-mode: $\|E_{calc} - E_{ex}\|_2$ (normalized) for different orders
($\Delta t = 0.005$, 220 Tets)



TM_{511}-mode: $\|E_{calc} - E_{ex}\|_2$ (normalized) for different orders
($\Delta t = 0.005$, 220 Tets)



For both, cubic and cylindrical geometry, we found that p-refinement performes very well. In the table below we show this on the example of a cubic TM_{220}-mode, $\Delta t = 0.001$, $t \in [0, 1]$.

p	Tets	aver. error	cpu-time
2	4856	$1.95 \cdot 10^{-2}$	1127 s
3	607	$1.10 \cdot 10^{-2}$	788 s

p	Tets	aver. error	cpu-time
4	384	$5.29 \cdot 10^{-3}$	1050 s
6	48	$4.22 \cdot 10^{-3}$	419 s

Brief summary

∴ we could show that one can improve accuracy and computation time by applying higher order basis functions to a FETD code in simple geometries.

Future work

∴ benchmarks for geometries with singular edges and corners

∴ adaptive hp-refinement

∴ parallelization of code to increase problem sizes

∴ combine Maxwell solver with a PIC-code (IPPL)