Solving large-scale inverse electromagnetic scattering problems: A parallel AD-based approach

Andrea Walther
Institut für Mathematik
Universität Paderborn

Outline

Motivation: The Rosetta-Project

The Direct Problem: A FDTD-Discretization of Maxwell's equations

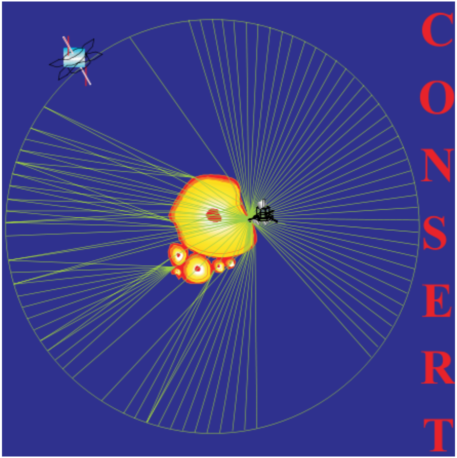
The Inverse Problem: An AD-based Solution Approach

Conclusion and Outlook

Current cooperation with
F. Hoffeins, U. Markwardt, W. Nagel (ZIH, TU Dresden)
D. Plettemeier (Electrical Engineering , TU Dresden)

at Uni Paderborn: Maria Brune

Identification of Material Parameters



The Consert-Mission

In 2004: Launch of spacecraft Rosetta
In 2014: Arrival at the comet 67P/Churyumov-Gerasimenko

On board: Lander Philae with 10 instruments, one is CONSERT

Goal: Determine internal structure (permittivity!) of the comet

The Consert-Mission

In 2004: Launch of spacecraft Rosetta
In 2014: Arrival at the comet 67P/Churyumov-Gerasimenko

On board: Lander Philae with 10 instruments, one is CONSERT

Goal: Determine internal structure (permittivity!) of the comet

For that:

- Send electromagnetic waves through the comet nucleus
- Measure the outcome using the orbiter Rosetta

The Consert-Mission

In 2004: Launch of spacecraft Rosetta
In 2014: Arrival at the comet 67P/Churyumov-Gerasimenko

On board: Lander Philae with 10 instruments, one is CONSERT

Goal: Determine internal structure (permittivity!) of the comet

For that:

- Send electromagnetic waves through the comet nucleus
- Measure the outcome using the orbiter Rosetta

One challenge:
Size of the comet is estimated by $2 \times 2 \times 2\ km$
resulting in a desired resolution of at least 700^3 grid cells

Maxwell Equations for Electromagnetic Field

$$\frac{\partial B}{\partial t} = -\nabla \times E \qquad\qquad \frac{\partial D}{\partial t} = \nabla \times H - \tilde{J}$$

$$\nabla \cdot B = 0 \qquad\qquad \nabla \cdot D = \rho$$

B and D = magnetic respectively electric flux density,
H and E = magnetic and electric field strength,

Maxwell Equations for Electromagnetic Field

$$\frac{\partial B}{\partial t} = -\nabla \times E \qquad \frac{\partial D}{\partial t} = \nabla \times H - \tilde{J}$$

$$\nabla \cdot B = 0 \qquad \nabla \cdot D = \rho$$

B and *D* = magnetic respectively electric flux density,
H and *E* = magnetic and electric field strength,

Can be substituted by

$$-\mu \frac{\partial H}{\partial t} = \nabla \times E \qquad \varepsilon \frac{\partial E}{\partial t} = \nabla \times H - \sigma E + J$$

Maxwell Equations for Electromagnetic Field

$$\frac{\partial B}{\partial t} = -\nabla \times E \qquad \frac{\partial D}{\partial t} = \nabla \times H - \tilde{J}$$

$$\nabla \cdot B = 0 \qquad \nabla \cdot D = \rho$$

B and D = magnetic respectively electric flux density,
H and E = magnetic and electric field strength,

Can be substituted by

$$-\mu \frac{\partial H}{\partial t} = \nabla \times E \qquad \varepsilon \frac{\partial E}{\partial t} = \nabla \times H - \sigma E + J$$

Goal: Want to estimate ε from measurements at the boundary

Theoretical Aspects

Formulation as time-dependent problem?

- Radon transform based reconstruction methods
  (Benna, Barriot, Kofman '02)
  requires statistical a priori information

Theoretical Aspects

Formulation as time-dependent problem?

- Radon transform based reconstruction methods
  (Benna, Barriot, Kofman '02)
  requires statistical a priori information

  $\Rightarrow$ keep time-dependence!!

Theoretical Aspects

Formulation as time-dependent problem?

- Radon transform based reconstruction methods
  (Benna, Barriot, Kofman '02)
  requires statistical a priori information

  $\Rightarrow$ keep time-dependence!!
- Then:
  Maxwell's equations do have a unique solution for one source and observations at a closed surface that comprises the reconstruction domain (Kirsch 1998)

Theoretical Aspects

Formulation as time-dependent problem?

- Radon transform based reconstruction methods
  (Benna, Barriot, Kofman '02)
  requires statistical a priori information

  $\Rightarrow$ keep time-dependence!!
- Then:
  Maxwell's equations do have a unique solution for one source and observations at a closed surface that comprises the reconstruction domain (Kirsch 1998)

Discretization?

Discretization: FDTD

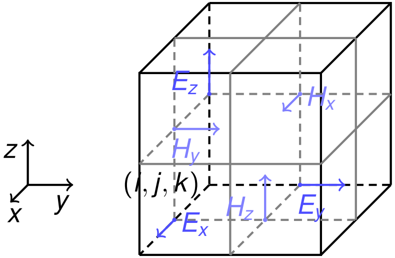
The **F**inite **D**ifferences in **T**ime **D**omain method

- proposed by Yee in 1966
- discretization of the curl equations by centered finite in space
- discretization of the curl equations by leapfrog scheme in time
- widely-used for simulations
- E and H components are staggered in time and space

Discretization: FDTD

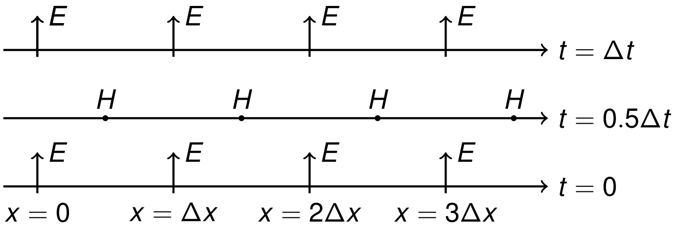
The **F**inite **D**ifferences in **T**ime **D**omain method

- proposed by Yee in 1966
- discretization of the curl equations by centered finite in space
- discretization of the curl equations by leapfrog scheme in time
- widely-used for simulations
- *E* and *H* components are staggered in time and space



Layout of a Yee-cell

The Staggered Computation



Leapfrog scheme in 1D

and the Formulas in 3D

$$H_x\Big|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = H_x\Big|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{\Delta t}{\mu}\left(\frac{E_y\big|_{i,j+\frac{1}{2},k+1}^{n} - E_y\big|_{i,j+\frac{1}{2},k}^{n}}{\Delta z} - \frac{E_z\big|_{i,j+1,k+\frac{1}{2}}^{n} - E_z\big|_{i,j,k+\frac{1}{2}}^{n}}{\Delta y}\right)$$

H_y and H_z similar.

and the Formulas in 3D

$$H_x\Big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} = H_x\Big|^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}} + \frac{\Delta t}{\mu}\left(\frac{E_y\big|^{n}_{i,j+\frac{1}{2},k+1} - E_y\big|^{n}_{i,j+\frac{1}{2},k}}{\Delta z} - \frac{E_z\big|^{n}_{i,j+1,k+\frac{1}{2}} - E_z\big|^{n}_{i,j,k+\frac{1}{2}}}{\Delta y}\right)$$

H_y and H_z similar.

$$E_x\Big|^{n}_{i+\frac{1}{2},j,k} = \frac{1-\frac{\sigma\Delta t}{2\varepsilon}}{1+\frac{\sigma\Delta t}{2\varepsilon}}\, E_x\Big|^{n-1}_{i+\frac{1}{2},j,k} - \frac{\frac{\Delta t}{\varepsilon}}{1+\frac{\sigma\Delta t}{2\varepsilon}}\left(J_{source_x}\Big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k}\right) + \frac{\frac{\Delta t}{\varepsilon}}{1+\frac{\sigma\Delta t}{2\varepsilon}}\left(\frac{H_z\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k} - H_z\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},k}}{\Delta y} - \frac{H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}} - H_y\big|^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\Delta z}\right)$$

E_y and E_z similar,

and the Formulas in 3D

$$H_x\Big|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n+\frac{1}{2}} = H_x\Big|_{i,j+\frac{1}{2},k+\frac{1}{2}}^{n-\frac{1}{2}} + \frac{\Delta t}{\mu}\left(\frac{E_y\big|_{i,j+\frac{1}{2},k+1}^{n} - E_y\big|_{i,j+\frac{1}{2},k}^{n}}{\Delta z} - \frac{E_z\big|_{i,j+1,k+\frac{1}{2}}^{n} - E_z\big|_{i,j,k+\frac{1}{2}}^{n}}{\Delta y}\right)$$

H_y and H_z similar.

$$E_x\Big|_{i+\frac{1}{2},j,k}^{n} = \frac{1-\frac{\sigma\Delta t}{2\varepsilon}}{1+\frac{\sigma\Delta t}{2\varepsilon}}\, E_x\Big|_{i+\frac{1}{2},j,k}^{n-1} - \frac{\frac{\Delta t}{\varepsilon}}{1+\frac{\sigma\Delta t}{2\varepsilon}}\left(J_{source_x}\Big|_{i+\frac{1}{2},j,k}^{n-\frac{1}{2}}\right) + \frac{\frac{\Delta t}{\varepsilon}}{1+\frac{\sigma\Delta t}{2\varepsilon}}\left(\frac{H_z\big|_{i+\frac{1}{2},j+\frac{1}{2},k}^{n-\frac{1}{2}} - H_z\big|_{i+\frac{1}{2},j-\frac{1}{2},k}^{n-\frac{1}{2}}}{\Delta y} - \frac{H_y\big|_{i+\frac{1}{2},j,k+\frac{1}{2}}^{n-\frac{1}{2}} - H_y\big|_{i+\frac{1}{2},j,k-\frac{1}{2}}^{n-\frac{1}{2}}}{\Delta z}\right)$$

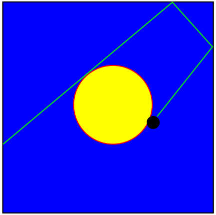
E_y and E_z similar, material parameter ε enters nonlinearly!!

Handling of Boundary

No boundary in space

Handling of Boundary

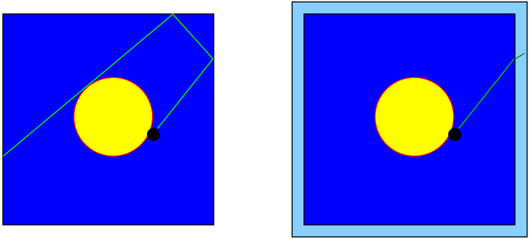
No boundary in space but simulation performed on bounded domain.



Handling of Boundary

No boundary in space but simulation performed on bounded domain.

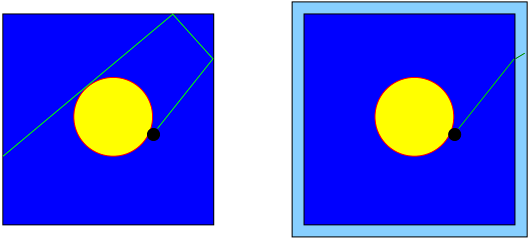
Therefore: Waves have to be damped at the boundary



Handling of Boundary

No boundary in space but simulation performed on bounded domain.

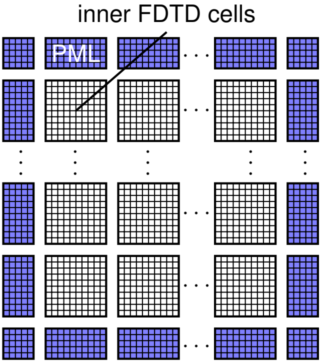
Therefore: Waves have to be damped at the boundary



We use perfectly matched layers (PML) in a variant proposed by Gedney in 1996 consisting of uniaxial absorbing material.

Computational Domain in 2D

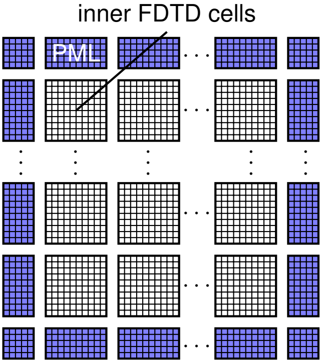
Due to size of the problem: Parallelization is indispensable!



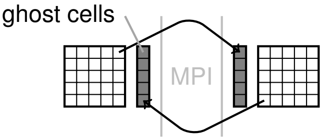
(a) 2D domain decomposition

Computational Domain in 2D

Due to size of the problem: Parallelization is indispensable!

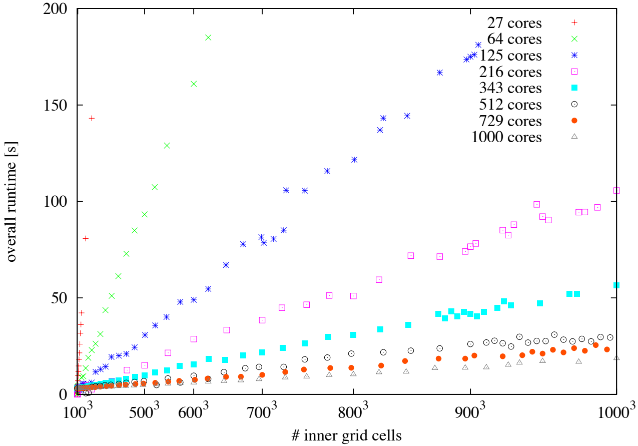


(a) 2D domain decomposition



(b) Synchronization of ghost cells

Runtime: Function Evaluation



The Inverse Problem

$$J(\varepsilon) = \sum_{(i,j,k)\in\mathcal{M}} \sum_{n=0}^{N} \frac{1}{2} \left(\| u(\varepsilon)|_{i,j,k}^{n} - u^{obs}|_{i,j,k}^{n} \|^2 \right) + \beta \|\varepsilon\|_*^2$$

with

$\mathcal{M}$ . . . set of indices of observed cells

N . . . number of observed time steps

$u(\varepsilon)$ . . . simulated state

u^{obs} . . . observed state

β . . . Regularization parameter

The Inverse Problem

$$J(\varepsilon) = \sum_{(i,j,k)\in\mathcal{M}} \sum_{n=0}^{N} \frac{1}{2} \left(\| u(\varepsilon)|_{i,j,k}^{n} - u^{obs}|_{i,j,k}^{n} \|^2 \right) + \beta \|\varepsilon\|_*^2$$

with

$\mathcal{M}$ ... set of indices of observed cells

N ... number of observed time steps

$u(\varepsilon)$ ... simulated state

u^{obs} ... observed state

β ... Regularization parameter

All solution approaches need derivatives.

The Inverse Problem

$$J(\varepsilon) = \sum_{(i,j,k)\in\mathcal{M}} \sum_{n=0}^{N} \frac{1}{2} \left(\| u(\varepsilon)|_{i,j,k}^{n} - u^{obs}|_{i,j,k}^{n} \|^2 \right) + \beta \|\varepsilon\|_*^2$$

with

$\mathcal{M}$ ... set of indices of observed cells

N ... number of observed time steps

$u(\varepsilon)$ ... simulated state

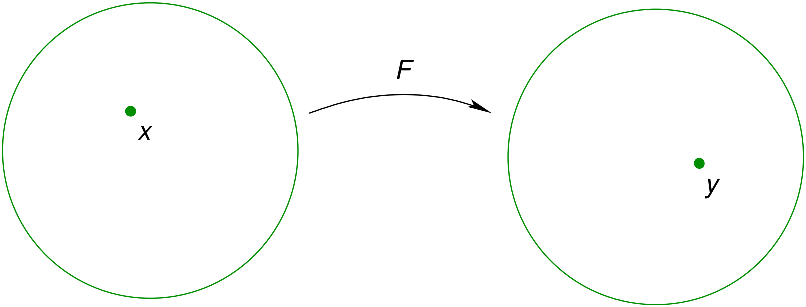
u^{obs} ... observed state

β ... Regularization parameter

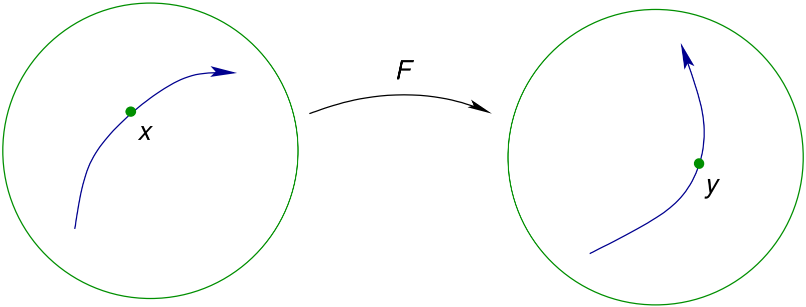
All solution approaches need derivatives.

Consistent derivatives for discrete version? (Abenius '04) provided by **algorithmic differentiation (AD)**.

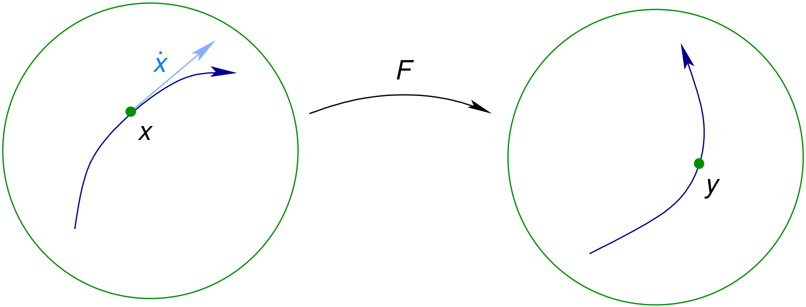
Forward mode AD = Tangents/Sensitivities



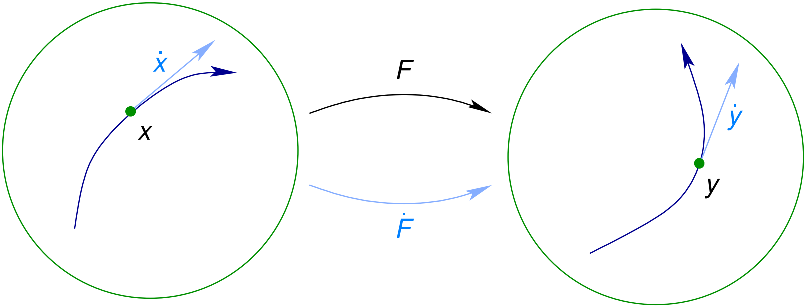
Forward mode AD = Tangents/Sensitivities



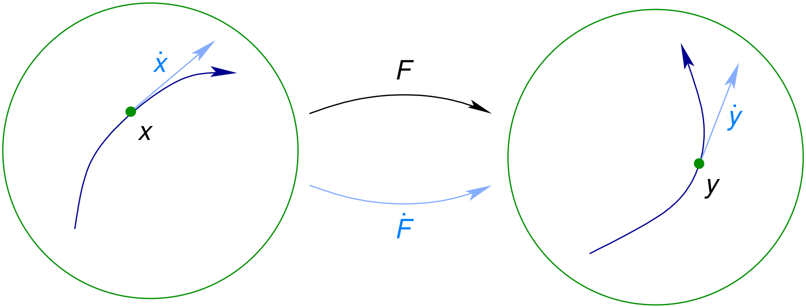
Forward mode AD = Tangents/Sensitivities



Forward mode AD = Tangents/Sensitivities

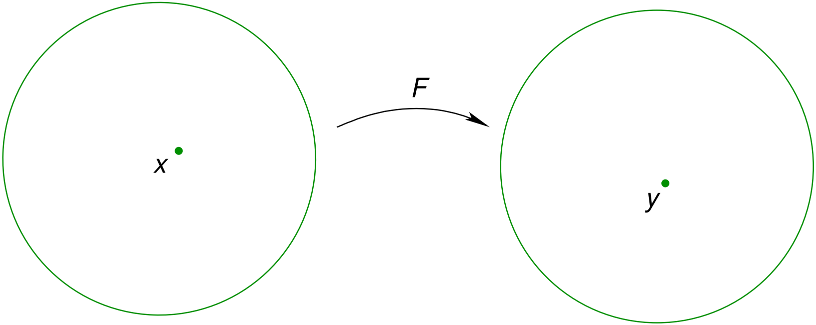


Forward mode AD = Tangents/Sensitivities

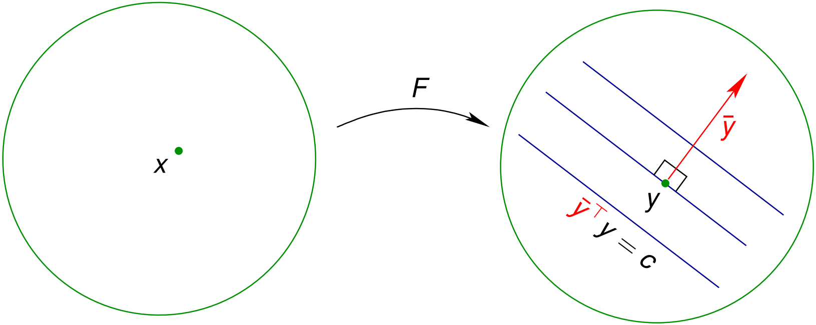


$$\dot{y}(t) \;=\; \frac{\partial}{\partial t} F(x(t)) \;=\; F'(x(t))\,\dot{x}(t) \equiv \;\dot{F}(x, \dot{x})$$

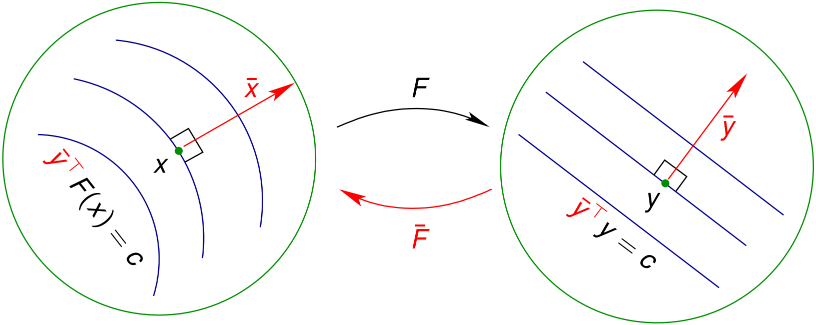
Reverse Mode AD = Discrete Adjoints



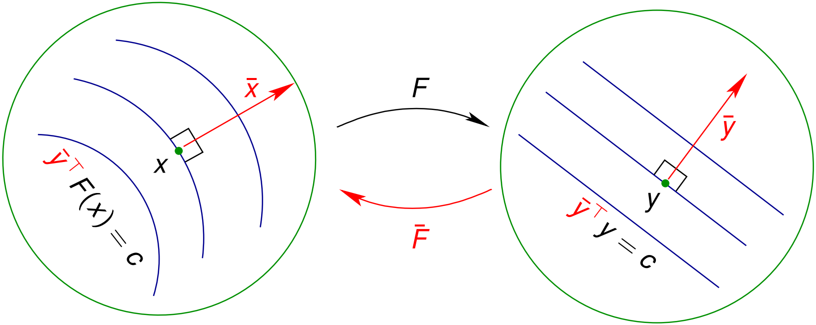
Reverse Mode AD = Discrete Adjoints



Reverse Mode AD = Discrete Adjoints



Reverse Mode AD = Discrete Adjoints



$$\bar{x} \equiv \bar{y}^\top F'(x) = \nabla_x \langle \bar{y}^\top F(x) \rangle \equiv \bar{F}(x, \bar{y})$$

Algorithmic Differentiation (AD)

- Differentiation of “computer programs” within machine precision
- Evaluation of derivatives with working accuracy
- Forward mode: $\text{OPS}(F'(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [2, 5/2]$
  Reverse mode: $\text{OPS}(\bar{y}^\top F'(x)) \leq c\,\text{OPS}(F), \quad c \in [3, 4]$
  $\text{MEM}(\bar{y}^\top F'(x)) \sim \text{OPS}(F),$
  Combination: $\text{OPS}(\bar{y}^\top F''(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [7, 10]$
- Tools: ADOL-C, CppAD, Tapenade, . . .
- www.autodiff.org, (Griewank, Walther 08)

Algorithmic Differentiation (AD)

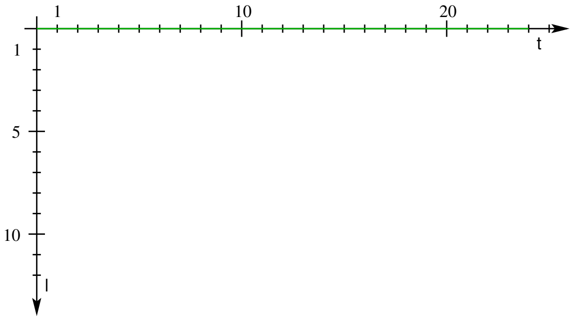
- Differentiation of “computer programs” within machine precision
- Evaluation of derivatives with working accuracy
- Forward mode: $\text{OPS}(F'(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [2, 5/2]$
  Reverse mode: $\text{OPS}(\bar{y}^\top F'(x)) \leq c\,\text{OPS}(F), \quad c \in [3, 4]$
  $\text{MEM}(\bar{y}^\top F'(x)) \sim \text{OPS}(F),$
  Combination: $\text{OPS}(\bar{y}^\top F''(x)\dot{x}) \leq c\,\text{OPS}(F), \quad c \in [7, 10]$
- Tools: ADOL-C, CppAD, Tapenade, ...
- www.autodiff.org, (Griewank, Walther 08)

Remarks:

- Cost for gradient calculation independent of # variables
- Memory requirement may cause problem! $\Rightarrow$ Checkpointing

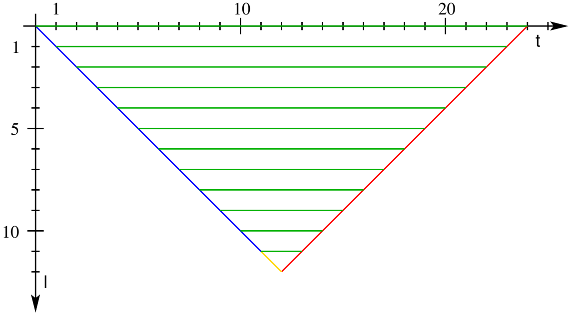
Store-Everything Approach

Example: 12 time steps



Store-Everything Approach

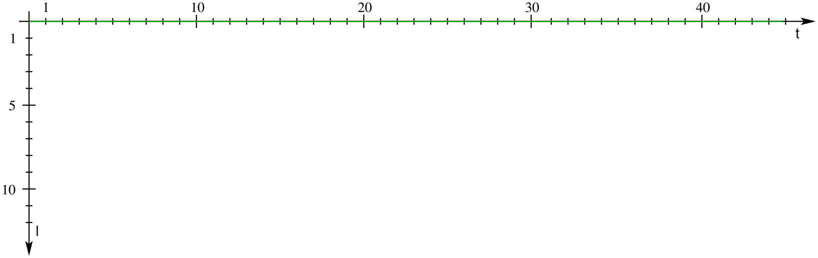
Example: 12 time steps



➡ MEM = $O(l)$, TIME = 2l,
might cause problems, even if it fits in memory

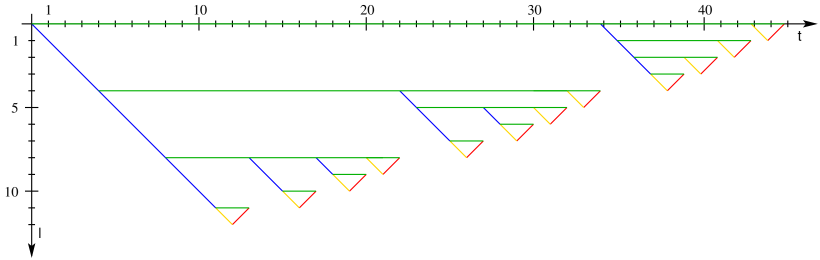
Binomial Checkpointing

Example: 12 time steps, 4 checkpoints, reusage of all checkpoints!



Binomial Checkpointing

Example: 12 time steps, 4 checkpoints, reusage of all checkpoints!



➔ MEM = c, TIME $=$?

Checkpointing Theory

Goal: Minimal number of recomputations for c checkpoints

Available results:

- l known, constant step costs
  (Griewank '92), (Griewank, Walther '00), (Kowarz, Walther '07)
- l known, variable step costs
  (Walther '00), (Hinze, Sternberg '05)
- l unknown, constant step costs
  (Hinze, Walther, Sternberg '06), (Stumm, Walther '10),
  (Moin, Wang '10)
- l known, variable access cost
  (Stumm, Walther '09)

Checkpointing Theory

Goal: Minimal number of recomputations for *c* checkpoints

Available results:

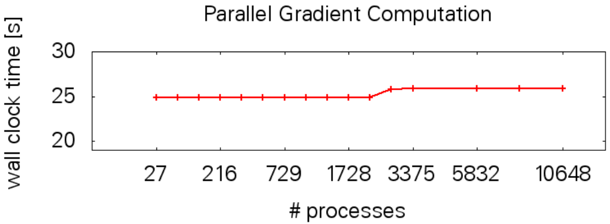
- *l* known, constant step costs
  (Griewank '92), (Griewank, Walther '00), (Kowarz, Walther '07)
- *l* known, variable step costs
  (Walther '00), (Hinze, Sternberg '05)
- *l* unknown, constant step costs
  (Hinze, Walther, Sternberg '06), (Stumm, Walther '10), (Moin, Wang '10)
- *l* known, variable access cost
  (Stumm, Walther '09)

Also applicable for continuous adjoints!
Implemented in software driver revolve

Software Components for Inverse Problem

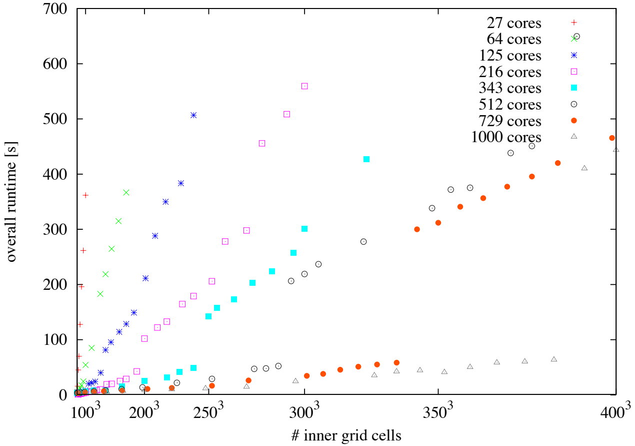
- own simulation code in C++ using MPI
- ADOL-C for the computation of adjoints for one time step
- revolve for checkpointing of time loop
- coupling with L-BFGS
- coupling with Ipopt (goal: also optimizer in parallel)

Scaling of Gradient Computation

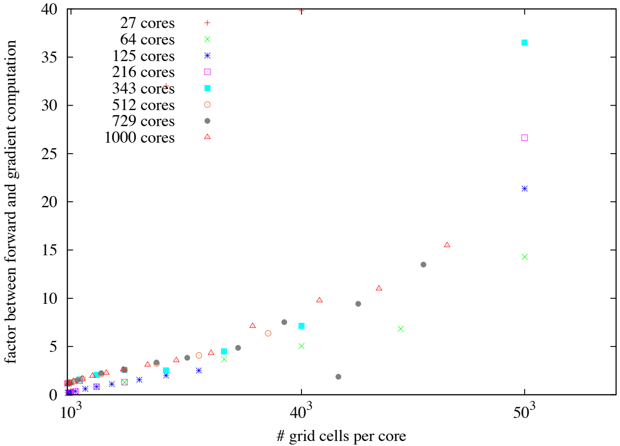


weak scaling: same amount of work per processor

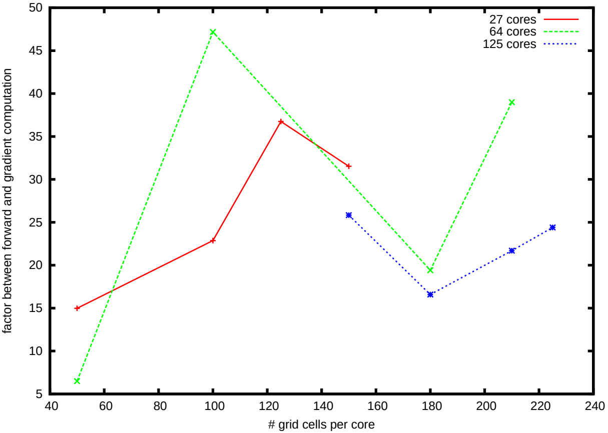
Runtime Gradient I



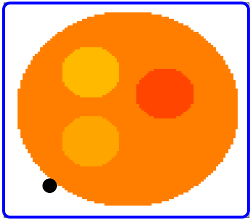
Runtime Quotient I



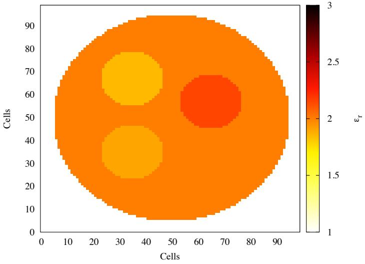
Runtime Quotient II



Test Problem

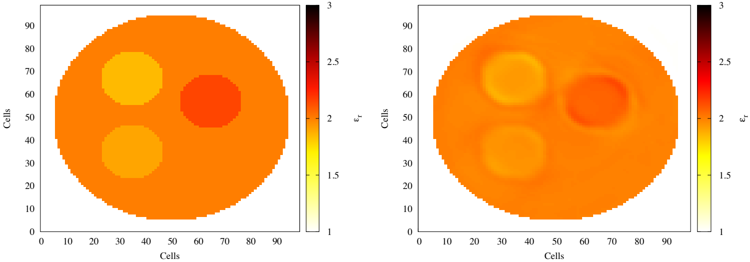


Test Problem



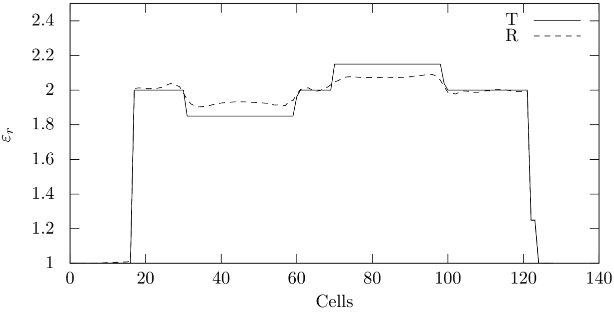
reference and starting point

Test Problem



reference and final point
after 60 iterations for 729000 unknowns

Test Problem



target and reconstructed permittivity

First Tests of Regularizations

So far: Treated as PDE constrained optimization problem

Now: Add appropriate regularisation

Implemented: $\|.\|_* = \|.\|_2$ and $\|.\|_* = \|.\|_{TV}$

	iter	function value	β
no reg	70	7.5084540e-03	0.0
L_2	70	3.9867834e-03	2.0
TV	70	3.9868059e-03	8.0

Conclusions

- Simulation in parallel for 3D $\Rightarrow$ large-scale discretizations
- Gradient in parallel for 3D $\Rightarrow$ large-scale discretizations
- Coupling with L-BFGS and recently with Ipopt
- First tests with respect to regularisations
- Next steps:
  - infinite dimensional setting ?
  - appropriate regularization techniques ?
  - globalization strategies ?