A fast parallel Poisson solver on irregular domains

Peter Arbenz[*]
Yves Ineichen[*,**]
Andreas Adelmann[**]

[*] ETH Zurich
Chair of Computational Science

[**] Paul Scherrer Institute
Accelerator Modelling and Advanced Simulations

Woudschouten Conference
Utrecht, October 6–8, 2010

Outline

Outline

Motivation from beam dynamics

Vlasov-Poisson formulation for particle evolution

- In physical devices like accelerators $10^9 \dots 10^{14}$ (or more) charged particles are accelerated in electric fields.
- Instead of computing with individual particles one considers a particle density $f(\mathbf{x}, \mathbf{v}, t)$ in phase space (position-velocity $(\mathbf{x}, \mathbf{v})$ space).
- The *Vlasov equation* describes the evolving particle density

$$\frac{df}{dt} = \partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q}{m_0}(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = 0,$$

  where $\mathbf{E}$ and $\mathbf{B}$ are electric and magnetic fields, respectively.
- The charged particles are 'pushed' by Newton's law

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{v}, \quad \frac{d\mathbf{v}(t)}{dt} = \frac{q}{m_0}\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right).$$

Motivation from beam dynamics (cont.)

Vlasov-Poisson formulation for particle evolution (cont.)

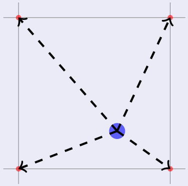
- The determination of $\mathbf{E}$ and $\mathbf{B}$ is done in the co-moving Lorentz frame where $\hat{\mathbf{B}} \approx \mathbf{0}$ and
$$\hat{\mathbf{E}} = -\nabla \hat{\phi},$$
where the electrostatic potential $\hat{\phi}$ is the solution of the *Poisson problem*
$$-\Delta \hat{\phi}(\mathbf{x}) = \frac{\hat{\rho}(\mathbf{x})}{\varepsilon_0}, \tag{1}$$
equipped with appropriate boundary conditions.
- The charge densities ρ is proportional to the particle density.

Motivation from beam dynamics (cont.)

Particle-in-cell (PIC) method in N-body Simulations

- Interpolate individual particle charges to a rectangular grid
- Discretize the Poisson equation by finite differences on the rectangular grid
- This leads to a system of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{2}$$



  $\mathbf{b}$ denotes the interpolated charge densities at the mesh points.
- Solve the Poisson equation on the mesh in a Lorentz frame
- $\mathcal{O}(n \log n)$ operations needed provided that the domain is rectangular (FFT based solvers).

Real beam-pipes are not rectangular

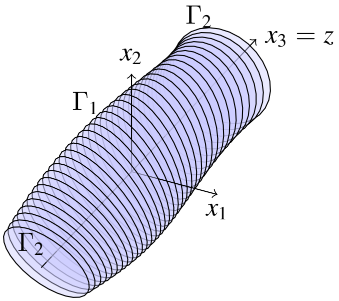
Boundary value problem

$$\nabla^2 \phi = -\frac{\rho}{\epsilon_0} \quad \text{in } \Omega \subset \mathbb{R}^3,$$

$$\phi = 0 \quad \text{on } \Gamma_1$$

$$\frac{\partial \phi}{\partial \vec{n}} + \frac{1}{d}\phi = 0 \quad \text{on } \Gamma_2$$

- $\Omega \subset \mathbb{R}^3$: simply connected computational domain
- ϵ_0: the dielectric constant
- $\Gamma = \Gamma_1 \cup \Gamma_2$: boundary of Ω
- d: distance of bunch centroid to the boundary

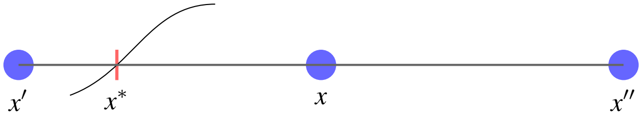


Γ_1 is the surface of an

1. Elliptical-shaped beam-pipe
2. Arbitrary beam-pipe element

Finite-difference discretization

1. Second order finite difference scheme: standard 7-point stencil on 3D Cartesian grid.
2. Boundary requires special treatment.



1. *Constant extrapolation*: $u(x') = u(x^*)$ and $x^* \in \Gamma_1$
2. *Linear extrapolation*: $u(x')$ is obtained by linear interpolation of $u(x)$ and $u(x^*)$
   System matrix $\mathbf{A}$ is symmetric positive definite.
3. *Quadratic extrapolation* (Shortley-Weller approximation): $u(x')$ is obtained by quadratic interpolation of $u(x)$, $u(x'')$, and $u(x^*)$
   $\rightarrow$ non-symmetric stencil
   System matrix $\mathbf{A}$ is positive definite but not symmetric. ($\mathbf{A}$ is still an M-matrix)

Goal

Divise an efficient iterative solver for the Poisson equation on irregular domains

- Solve anisotropic electrostatic Poisson equation with an iterative solver
- Easy to specify boundary surface
- **Irregular** domain imbedded in a rectangular grid.
- “Exact” (Dirichlet) boundary conditions
- Achieving good parallel efficiency
- Reuse information available from previous time steps
- Ref: Adelmann/Arbenz/Ineichen, J. Comp. Phys., 229, 4554–4566 (2010).
- Similar to McCorquodale/Colella/Grote/Vay, J. Comp. Phys., 201, 34–60, 2004

Outline

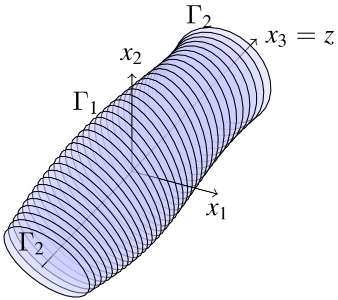
Outline

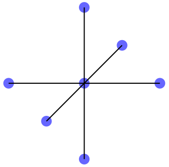
Solver

- Large sparse linear system

  $$A\mathbf{x} = \mathbf{b}$$

  with A (non-)symmetric positive definite.
- Conjugate gradient algorithm is iterative solver method of choice (?)
- Preconditioned by smoothed aggregation-based algebraic multigrid





Implementation

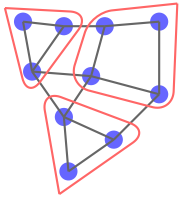
For preconditioner setup and iterative solver we used TRILINOS (see http://trilinos.sandia.gov)

- EPETRA: distributed matrices and vectors
- AMESOS: direct coarse level solver
- AZTECOO: iterative solver
- ML: smoothed aggregation based AMG preconditioner
- ISORROPIA: partitioning and load balancing

The Object Oriented Parallel Accelerator Library Framework (OPAL) provides a partitioning of the data based on the underlying rectangular grid.
(See http://amas.web.psi.ch/docs/opal)

AMG parameters

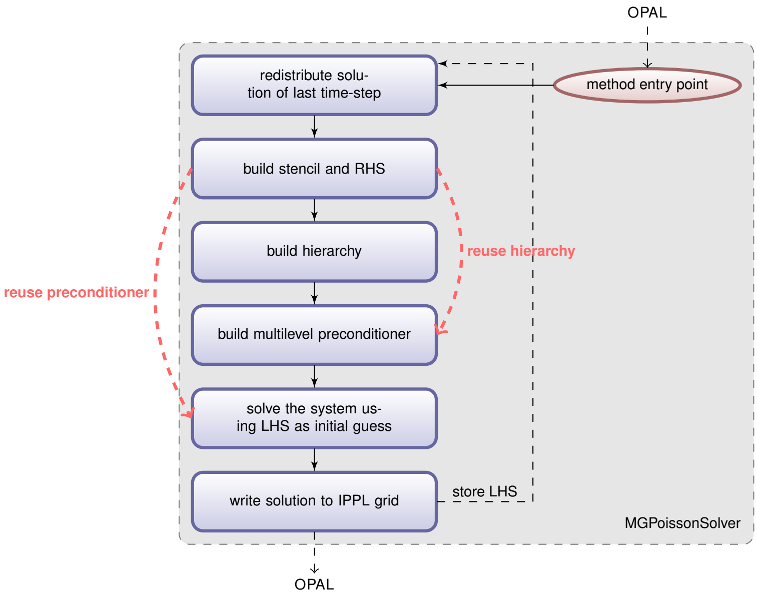
- “Decoupled” aggregation scheme: aggregates of size $3 \times 3 \times 3$
  - Each processor aggregates its portion of the grid
  - Many aggregates near inter-processor boundaries with non-optimal size
  - Number of vertices is substantially reduced in every coarsening step


clustering contiguous vertices into aggregates

- Chebyshev polynomial pre- and postsmoothers perform well for parallel solvers (Adams/Brezina/Hu/Tuminaro, J. Comp. Phys., 2003)
  - Estimates for the spectrum of the matrices $\mathbf{A}_k$ are needed.
- LU based direct coarse level solver or a few steps of Gauss-Seidel iteration
  - Matrices $\mathbf{A}_k$ tend to get dense with increasing level.
- V-cycle.
- Starting vector.

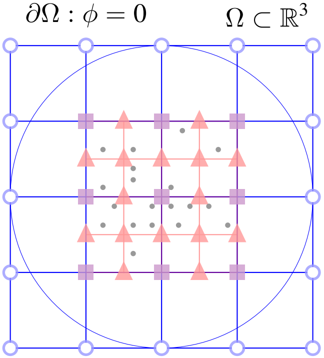
AMG performance critically depends on choice of parameters!

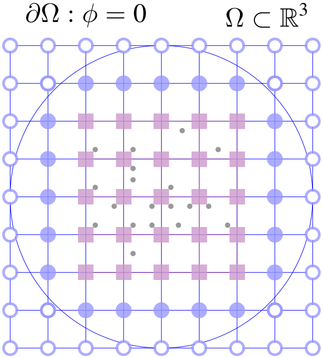
Integration of the solver in OPAL


OPAL
method entry point
redistribute solution of last time-step
build stencil and RHS
build hierarchy
build multilevel preconditioner
solve the system using LHS as initial guess
write solution to IPPL grid
store LHS
reuse hierarchy
reuse preconditioner
MGPoissonSolver
OPAL

Outline

FFT-based Fast Poisson solvers





Sketch of the test cases with equal number of mesh points (left), and equal mesh resolution (right), respectively. Displayed are the shared (square), FFT only (triangle), and AMG only (filled circle) mesh points on a cross section of the grid plane. Illustrative particles (gray) inside the FFT domain denote the charge density.

The Poisson equation

$$-\Delta\phi = \rho$$

discretized on a rectangular m-by-n grid can be written in block-tridiagonal form

$$A\mathbf{x} = (T_m \oplus I_n + I_m \oplus T_n)\mathbf{x} = \mathbf{b} \qquad (*)$$

provided that the boundary conditions along an edge (face in 3D) are constant. Here, with homogeneous Dirichlet boundary conditions,

$$T_k := \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{k\times k},$$

Equation $(*)$ can be written in the form

$$T_m X + X T_n = Y, \qquad X, Y \in \mathbb{R}^{m\times n}. \qquad (**)$$

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$.

Let

$$T_n = C_n \Lambda_n C_n^T, \qquad \Lambda = \mathrm{diag}(\lambda_1^{(n)}, \ldots, \lambda_n^{(n)})$$

be the *spectral decomposition* of T_n. C_n is orthogonal, $C_n^{-1} = C_n^T$, and, most importantly, operating with C or C^T can be implemented by means of the fast Fourier transform (FFT), i.e. in $\mathcal{O}(n \log n)$ floating point operations provided that n is a power of two.

Eq. ($**$) can be rewritten as

$$\Lambda_m(C_m^T X C_n) + (C_m^T X C_n)\Lambda_n = (C_m^T Y C_n). \qquad (***)$$

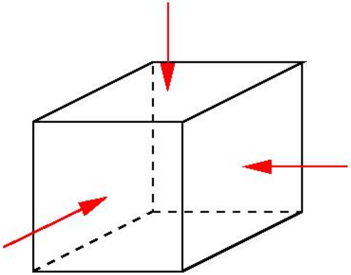
Notice that this is a diagonal system!

The procedure to solve ($**$) is now as follows

1. Compute $C_m^T Y C_n$: apply (inverse) FFT's from left and right
2. Solve $(\Lambda_m \oplus I_n + I_m \oplus \Lambda_n)Z(:) = (C_m^T Y C_n)(:)$.
3. Compute $X = C_m Z C_n^T$: apply (inverse) FFT's from left and right.

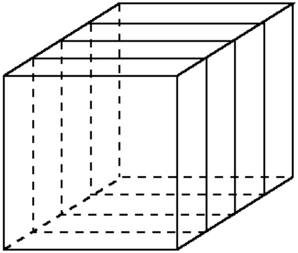
3-dimensional FFT

Let X be a 3-dimensional $n \times n \times n$ array.



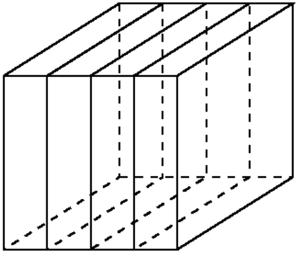
We want to apply n^2 FFTs of length n in each direction.

Let us assume that we have a $\sqrt{p} \times \sqrt{p}$ processor grid.



Step 1. n/p 2D FFTs in slices.

Step 2. “all-to-all personalized” communication with blocks of size $\left(\frac{n}{\sqrt{p}}\right)^2 \times n$.



Step 3. n^2/p 1D FFTs in last direction.

Outline

Performance and scalability of parallel algorithms

Speedup: gain in time that is obtained by parallel execution of a program.

$$S(p) = \frac{T(1)}{T(p)}, \qquad S(p) \le p.$$

Efficiency: obtained speedup relative to ideal speedup.

$$E(p) = \frac{S(p)}{p}, \qquad E(p) \le 1.$$

Amdahl's law

Algorithm consists of two portions

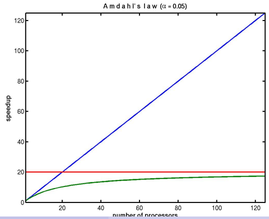
- sequential (fraction α)
- ideally parallelizable (fraction $1-\alpha$)

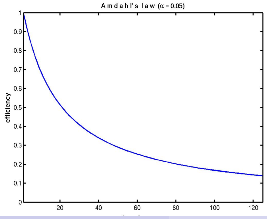
Then,

$$T(p) = \left(\alpha + \frac{1-\alpha}{p}\right) T(1),$$

By consequence,

$$S(p) = \frac{1}{\alpha + \frac{1-\alpha}{p}} < \frac{1}{\alpha}, \qquad E(p) = \frac{1}{p\alpha + 1 - \alpha},$$





Gustafson's law

Increase the problem size proportional to the processor number.

$$T(1) = \alpha + (1-\alpha)p, \qquad T(p) = \alpha + \frac{(1-\alpha)p}{p} = 1,$$

Then,

$$S(p) = \alpha + (1-\alpha)p \longrightarrow (1-\alpha)p, \qquad E(p) = 1 - \alpha + \frac{\alpha}{p} \longrightarrow 1 - \alpha,$$

Strong scalability: Algorithm (code) speeds up (almost) linearly in Amdahl's sense

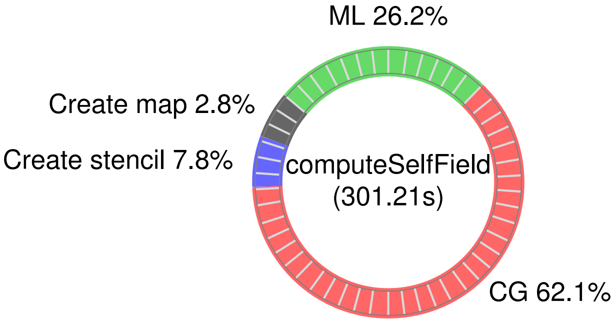
Weak scalability: Algorithm (code) speeds up (almost) linearly in Gustafson's sense

Outline

Environment

Buin: Cray XT4 cluster at the CSCS in Manno (Switzerland)

- 468 AMD dual core Opteron at 2.6 GHz
- 936 GB DDR RAM
- 30 TB Disk
- 7.6 GB/s interconnect bandwith



Validation of the Solver

For validation purposes we investigated an axi-symmetric problem with known analytical solution.

h	$\|\|e_h\|\|_2$	r	$\|\|e_h\|\|_\infty$	r
$1/64$	$2.162 \cdot 10^{-3}$	—	$7.647 \cdot 10^{-3}$	—
$1/128$	$1.240 \cdot 10^{-3}$	0.80	$4.153 \cdot 10^{-3}$	0.88
$1/64$	$2.460 \cdot 10^{-5}$	—	$6.020 \cdot 10^{-5}$	—
$1/128$	$6.226 \cdot 10^{-6}$	1.98	$1.437 \cdot 10^{-5}$	2.07
$1/64$	$5.581 \cdot 10^{-6}$	—	$1.689 \cdot 10^{-5}$	—
$1/128$	$1.384 \cdot 10^{-7}$	2.01	$4.550 \cdot 10^{-6}$	1.89

Solution error for constant (top), linear (middle), quadratic (bottom) extrapolation.

The convergence rate r is defined by

$$r = \log_2 \left(\frac{||e_{2h}||}{||e_h||} \right)$$

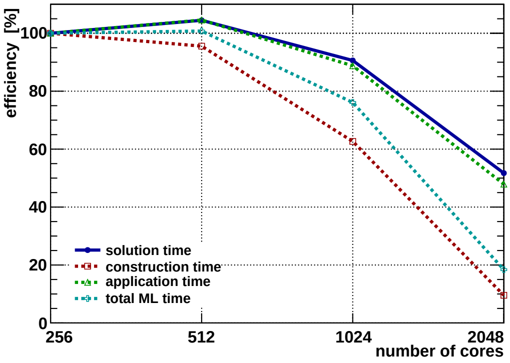
Comparison with FFT-based Poisson solver

Simulation timings of one solve in the first and second time step, respectively.

solver	reusing	mesh size	mesh points	first [s]	second [s]
FFT	—	128×128×256	4,194,304	12.3	—
AMG	—	128×128×256	3,236,864	49.9	42.2
AMG	hierarchy	128×128×256	3,236,864	—	35.5
AMG	preconditioner	128×128×256	3,236,864	—	28.2
AMG	—	166×166×256	5,462,016	81.8	71.2
AMG	hierarchy	166×166×256	5,462,016	—	60.4
AMG	preconditioner	166×166×256	5,462,016	—	43.8

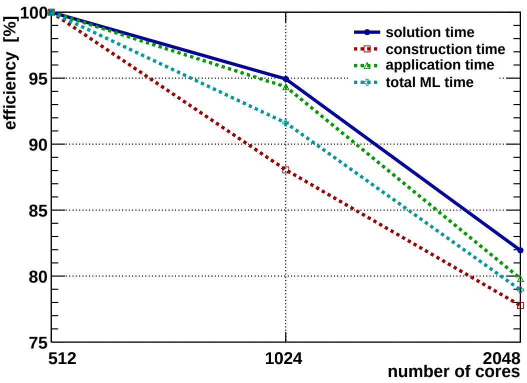
Equal number of mesh points (above) and equal mesh spacings (below) for FFT and AMG.

Parallel efficiency



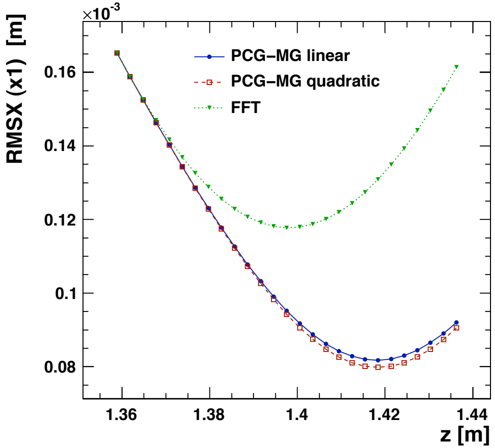
- Obtained for a tube embedded in a $512 \times 512 \times 512$ grid
- ML construction phase is performing poorly
- Influence of problem size on the low performance of the aggregation in ML.
- Problem is too small for the number of processors used.

Parallel efficiency (cont.)



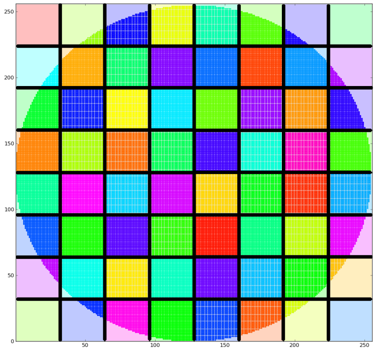
- Obtained for a tube embedded in a $1024 \times 1024 \times 1024$ grid
- Construction phase is performing the worst with an efficiency of 73%
- Problem size is appropriate for the number of processors.

Impact on Physics of OPAL Simulations



- Statistics of the distance of the particles to the z-axis.
- Shift of the beam size minimum (beam waist) towards larger z values
- A smaller minimum $\rightarrow$ self forces are larger when considering the beam pipe
- Beam pipe radius is an important optimization quantity

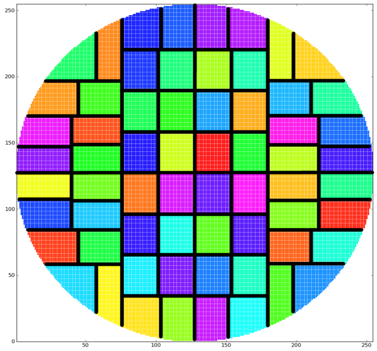
Load balance issues



Cross section of data distribution on 512 cores on a $8\times8\times8$ processor grid (colors indicate data owned by a processor).

- Data is distributed according to the underlying rectangular grid (induced by the particle code OPAL).
- Severe load imbalance.
- But speedup looks quite good! How can this be?
- Look at the work of the most heavily loaded processor: it decreases linearly with the number of processors!

Load balance issues (cont.)



Cross section of data redistributed by RCB on 512 cores on a $8{\times}8{\times}8$ processor grid.

- Isorropia's recursive coordinate bisection (RCB) algorithm distributes the data perfectly balanced.
- Maximal number of nodes per processor decreased
  $\Longrightarrow$ less work / processor!
- Shape of subdomains not rectangular anymore.
- Number of neighbors increases
  $\Longrightarrow$ # of messages increased!
- Lower / higher execution times??

Load balance issues (cont.)

Results for $1024 \times 1024 \times 1024$ grid

cores	solution	construction	application	total ML	iterations
512	63.12 [1.00]	32.09 [1.00]	52.73 [1.00]	84.80 [1.00]	20
1024	33.54 [0.94]	16.31 [0.98]	28.04 [0.94]	44.35 [0.96]	20
2048	18.56 [0.85]	8.10 [0.99]	15.66 [0.84]	23.76 [0.89]	21

Times in seconds and relative parallel efficiencies. The original data distribution is used, and the coarsest AMG level is solved iteratively.

cores	solution	construction	application	total ML	iterations
512	51.08 [1.00]	25.65 [1.00]	44.89 [1.00]	70.55 [1.00]	20
1024	27.38 [0.93]	12.96 [0.99]	24.51 [0.92]	37.07 [0.95]	20
2048	14.76 [0.87]	6.69 [0.96]	13.10 [0.86]	19.79 [0.89]	19

Times in seconds and relative parallel efficiencies. Data is distributed by RCB. The coarsest AMG level is solved iteratively.

Outline

Summary

- Conjugate gradient solver for Poisson equation on rectangular grid with special treatment of irregular boundary.
- Elliptic and arbitrary domains based on real geometries.
- Smoothed aggregation based algebraic Multigrid preconditioning
- Non-symmetric equations resulting from quadratic boundary treatment converge well with PCG.
- Reduced time to solution (20 and 40%) by reusing hierarchy or preconditioner.
- Reduced time to solution (20%) by balancing data among processors.
- Good parallel efficiency if data per processors is reasonably large.
- Compared to FFT more flexibilities for only a small performance loss.
- Considerable impact on physics (in particular, for narrow beam pipes).
- **Future work**:
  - Introduce adaptive mesh refinement (AMR).
  - Overcome Trilinos' global index 32-bit integer size limitation.

References

- A. Adelmann, P. Arbenz, Y. Ineichen. *A fast parallel Poisson solver on irregular domains applied to beam dynamics simulations.* J. Comp. Phys., 229, 4554–4566 (2010).
- A. Adelmann, P. Arbenz, Y. Ineichen. *Improvements of a Fast Parallel Poisson Solver on Irregular Domains.* To (hopefully) appear in the proceedings of the PARA'10 conference. Reykjavik, Iceland, June 6-9, 2010.