High-order Adaptive Mesh Refinement Multigrid Poisson Solver in any Dimension

Erwan Deriaz

Équipe Plasmas Chauds

Institut Jean Lamour (Nancy), CNRS / Université de Lorraine

Séminaire MoCo – IRMA

Strasbourg – 5 Avril 2022

Overview

Poisson solvers in Physics simulation

Poisson Equation in dimension d:

$$\Delta u = v, \quad \text{with} \quad \Delta = \sum_{i=1}^{d} \partial_i^2 \tag{1}$$

Ubiquitous equation, appears:

- in the computation of potential forces which derive from gradients of potentials of the kind $\mathbf{a} = \nabla\varphi$ with $\Delta\varphi = \pm\rho$ where ρ represents a density of mass or of charges and φ a gravitational field (plus case) or an electric field (minus case) for instances,
- to numerically solve the Incompressible Navier-Stokes Equation where the pressure is obtained through $\Delta P = -\rho \operatorname{div}((\mathbf{u}\cdot\nabla)\mathbf{u})$,
- to model dispersive effects as in the Heat Equation $\partial_t u = \Delta u$ which needs to be solved with implicit numerical methods of the kind $(Id - \nu\Delta)u_{n+1} = u_n$ with $\nu > 0$, related to Poisson Equation.

reduced MHD equations

- scalar functions $F, \psi, J, U, \varphi : \mathbb{T}^2 \mapsto \mathbb{R}$ satisfying:

$$\frac{\partial F}{\partial t} + [\varphi\,,\,F] = \rho_s^2[U\,,\,\psi] + \eta(\nabla^2\psi - \nabla^2\psi_{eq}), \qquad (2)$$

$$\frac{\partial U}{\partial t} + [\varphi\,,\,U] = [\psi\,,\,\nabla^2\psi] + \nu(\nabla^2 U - \nabla^2 U_{eq}), \qquad (3)$$

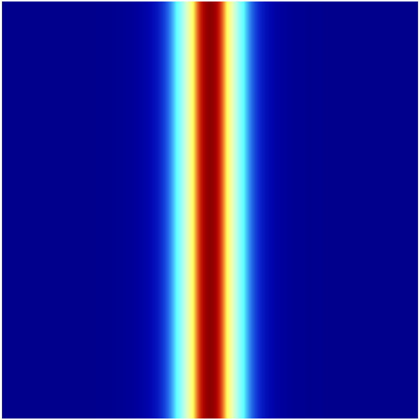
$$F \equiv \psi - d_e^2\nabla^2\psi \qquad\qquad U \equiv \nabla^2\varphi, \qquad (4)$$

  with the ion skin depth $d_e \geq 0$, the ion-sound Larmor radius $\rho_s \geq 0$, the resistivity $\eta \geq 0$ and the viscosity $\nu \geq 0$.
- Periodic boundary conditions.
- Initial conditions: $\varphi_0 = 0$, $\psi_0 = -\frac{1}{2}\cosh^{-2}(2x) + \varepsilon\sin(y)$, $(x, y) \in [-\pi, +\pi]$.

Poisson brackets notation:

$$[a, b] = \partial_x a\,\partial_y b - \partial_y a\,\partial_x b$$

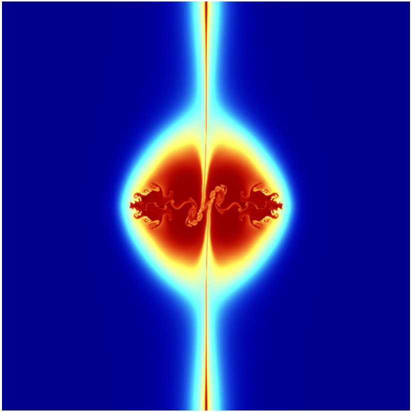
2D Magnetic Reconnection



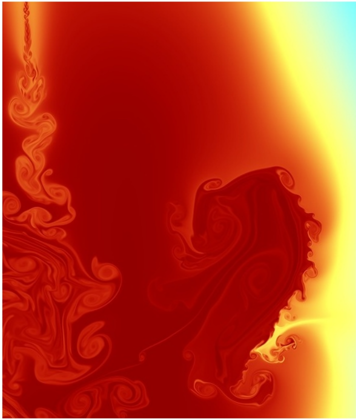
Magnetic stream F

refinement level of the grid

Turbulence



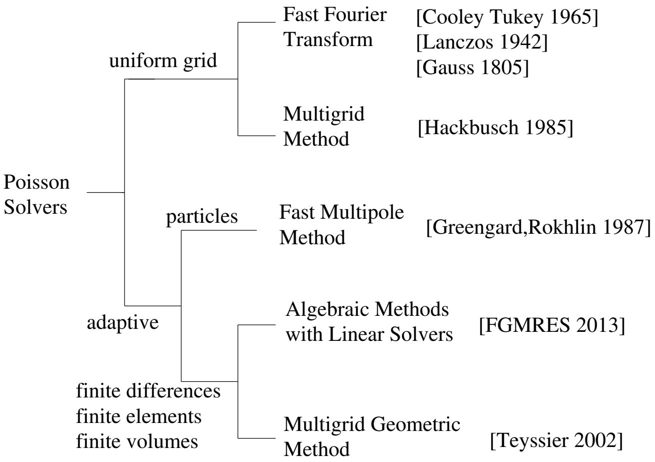
caring about the symmetry



multiscaled turbulence

Overview

Poisson Solvers



Overview

Fast Fourier Transform

The Fourier Transform diagonalizes the Δ operator:

$$\Delta u = v \qquad \Longleftrightarrow \qquad -\xi^2 \widehat{u}(\xi) = \widehat{v}(\xi)$$

with

$$\widehat{u}(\xi) = \int_{\mathbb{R}^d} u(x) e^{-i\xi \cdot x}\, dx$$

Fast Fourier Transform: computation of

$$f_j = \sum_{k=0}^{n-1} u_k e^{-\frac{2\pi i}{n} jk}, \quad 0 \leq j \leq n-1$$

in $O(n \log(n))$ operations thanks to foldings.

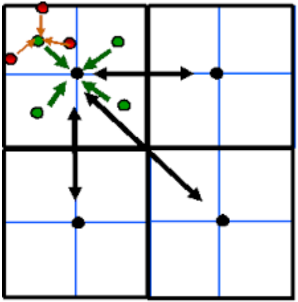
Overview

Fast Multipole Method

Green Kernel:

$$\Delta u = v \qquad \Longleftrightarrow \qquad u(x) = \int_{\mathbb{R}^d} K(x' - x) v(x')\, dx'$$

with $\Delta K = \delta$ the Dirac operator: $\int_{\mathbb{R}^d} \delta(x) f(x)\, dx = f(0)$



ack. 3DS.COM/SIMULIA

Overview

Tree structure

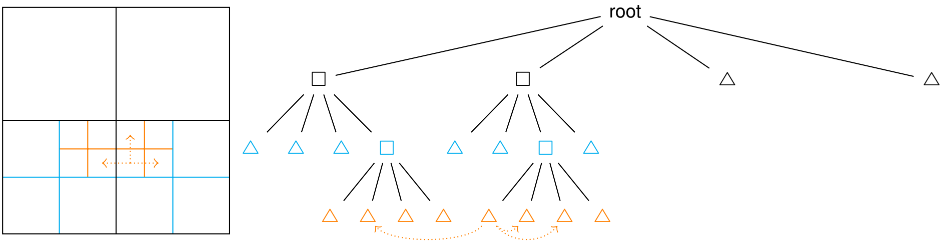
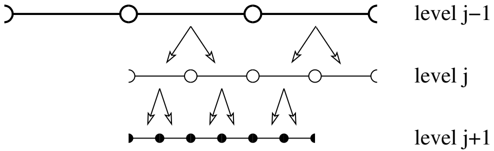


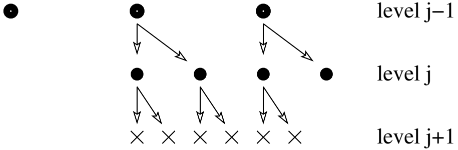
Figure: Fully-threaded tree (right) and its associated non-uniform grid (left).

Cell-based/vertex-based

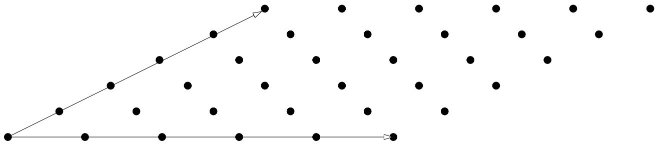
- refinement centered on the cell



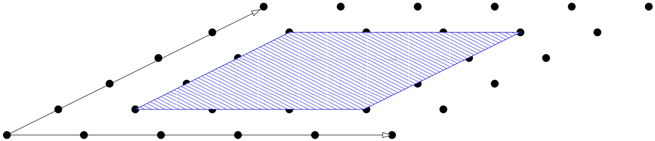
- refinement centered on the point



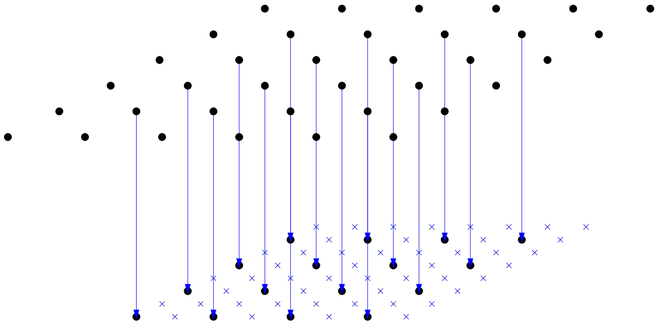
Point activation



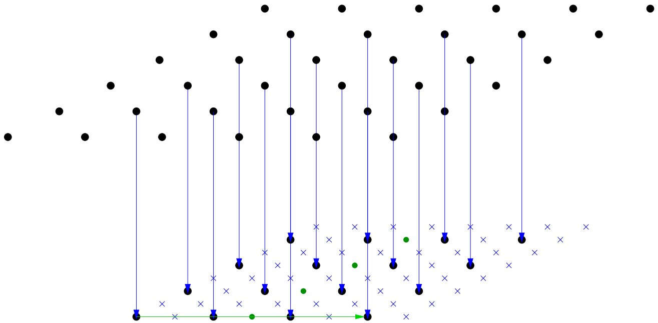
Point activation



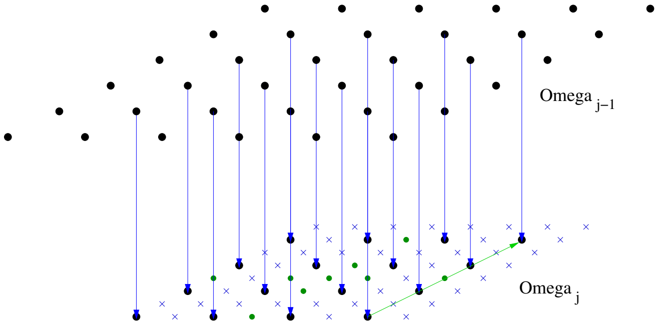
Point activation



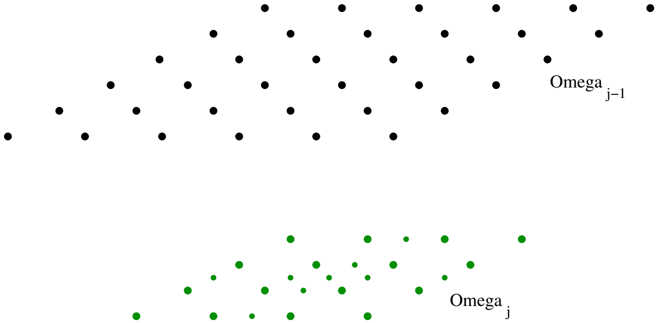
Point activation



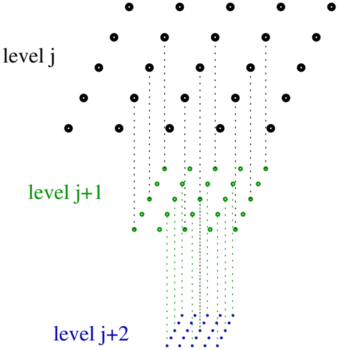
Point activation



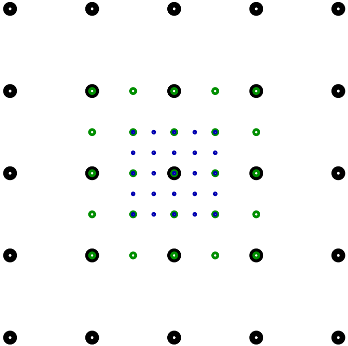
Point activation



Instance of mapping



embedded grids



resulting grid

Boundaries

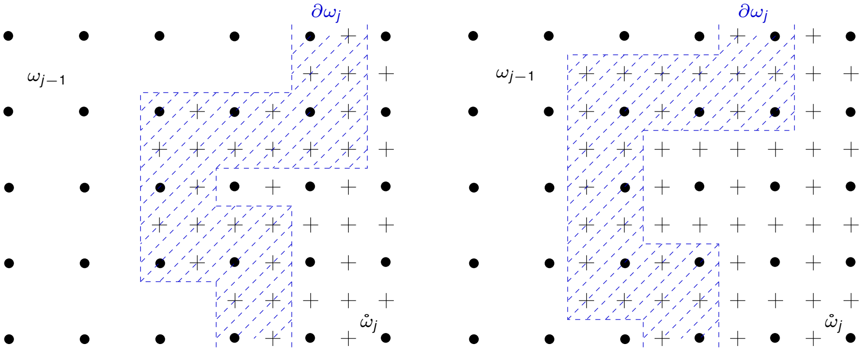


Figure: Two ways of fixing the boundary with respect to the interior domain $\mathring{\omega}_j$: setting the limit just after the middle points (on the left) or just after the pivotal points (on the right). The first case is more efficient with respect to memory management when r_A is even, the second case converges much faster.

Overview

Principle

We call *stencil* a discrete function $A : \mathbb{Z}^d \to \mathbb{R}^d, \mathbf{k} \mapsto a_{\mathbf{k}}$. From this function we form an approximation $\mathcal{A}$ of the Laplacian Δ. For any function u sufficiently regular at 0,

$$\lim_{h \to 0} \frac{1}{h^2} \sum_{\mathbf{k} \in \mathbb{Z}^d} a_{\mathbf{k}} u(\mathbf{k}h) = \Delta u(0) = \sum_{i=1}^{d} \partial_i^2 u(0).$$

By extension the A stencil may also denote the discrete operator $\mathcal{A}$

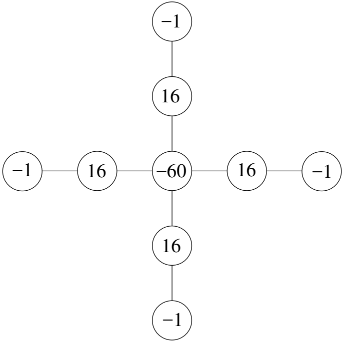
$$\mathcal{A} : \left(\mathbb{R}^d\right)^{\left(\mathbb{Z}^d\right)} \to \left(\mathbb{R}^d\right)^{\left(\mathbb{Z}^d\right)}, (u_{\mathbf{k}})_{\mathbf{k} \in \mathbb{Z}^d} \mapsto \left(\frac{1}{h^2} \sum_{\mathbf{k}' \in \mathbb{Z}^d} a_{\mathbf{k}'} u_{\mathbf{k}+\mathbf{k}'} \right)_{\mathbf{k} \in \mathbb{Z}^d}.$$

To define a compact finite difference scheme we need two stencils A and B with A an approximation of the Laplace operator and B an approximation of the Identity. The compact scheme is $2p$th-order if for all discretized function $u_h = (u(\mathbf{k}h))_{\mathbf{k}}$ with u C^{2p+2}-regular,

$$\frac{1}{h^2} A u_h = B(\Delta u)_h + O(h^{2p}). \tag{5}$$

2D usual non-compact stencil

$\frac{1}{12h^2}$



$u = v$ (6)

“Mehrstellenverfahren” scheme

$$\frac{1}{6h^2}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u = \frac{1}{12}\begin{bmatrix} & 1 & \\ 1 & 8 & 1 \\ & 1 & \end{bmatrix} v \qquad (7)$$

6h order any-dimensional case

$$A = \alpha[p \;\; 1 \;\; p]^d + \begin{array}{ccccc} & & \gamma & & \\ & & | & & \\ \gamma & - & \beta & - & \gamma \\ & & | & & \\ & & \gamma & & \end{array} \tag{8}$$

namely

$$a_{\mathbf{0}} = \alpha + \beta, \quad a_{(1,0,\ldots,0)} = \alpha\, p + \gamma, \quad \text{and} \quad a_{(\underbrace{1,\ldots,1}_{k \text{ times}},0,\ldots,0)} = \alpha\, p^k \quad \text{for} \quad k \geq 2.$$

To reach the sixth-order accuracy, computations lead to:

$$p = \frac{1}{3}, \quad \alpha = \frac{3}{2}\left(\frac{3}{5}\right)^{d-2}, \quad \beta = -\frac{25+2d}{6} \quad \text{and} \quad \gamma = \frac{1}{6}.$$

6h order any-dimensional case

$$B = \omega[q \;\; 1 \;\; q]^d + \begin{array}{ccccccccc} & & & & b_{01} & & & & \\ & & & & | & & & & \\ & & & & 0 & & & & \\ & & & & | & & & & \\ b_{01} & - & 0 & - & \lambda & - & 0 & - & b_{01} \\ & & & & | & & & & \\ & & & & 0 & & & & \\ & & & & | & & & & \\ & & & & b_{01} & & & & \end{array}$$

namely

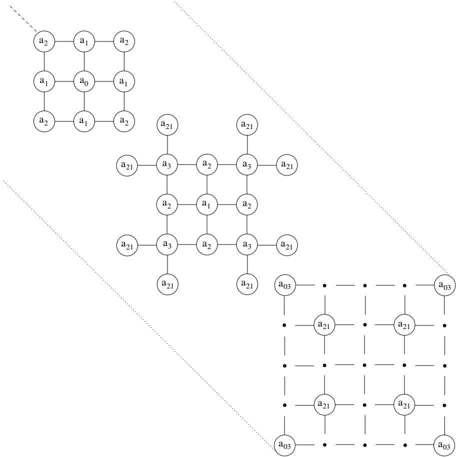
$$b_{\mathbf{0}} = \omega + \lambda, \quad b_{(\underbrace{1,\ldots,1}_{k \text{ times}},0,\ldots,0)} = \omega\, q^k \quad \text{for} \quad k \geq 1, \quad \text{and} \quad b_{(2,0,\ldots,0)} = b_{01},$$

with

$$q = \frac{1}{7}, \quad \omega = \frac{9}{10}\left(\frac{7}{9}\right)^d, \quad \lambda = \frac{1}{10} + \frac{d}{120} \quad \text{and} \quad b_{01} = -\frac{1}{240}.$$

10th order 3-dimensional case

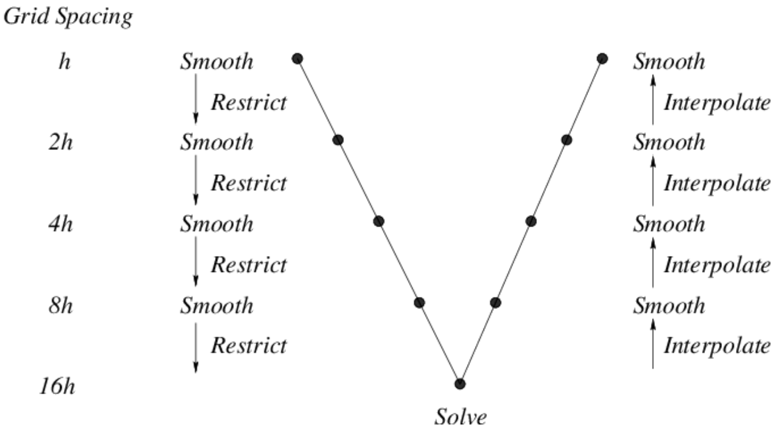
Bottom half of the symmetric three-dimensional stencil:



There are 113 points for the B stencil: b_0, b_1, b_2, b_3, b_{01}, b_{11}, b_{02}, b_{03}, b_{001}, b_{101}, b_{0001}.

Overview

In the uniform case, the V–cycle



[Colin Fox]

Gauss-Seidel iterations

Gauss-Seidel iteration (Smoother): for $x_\lambda \in \mathring{\omega}_j$ do

$$tmp(x_\lambda) = tmp(x_\lambda) + \frac{h^2}{a_{\mathbf{0}}}\left(res(x_\lambda) - \frac{1}{h^2}A\ tmp(x_\lambda)\right)$$

Interpolation of the boundary

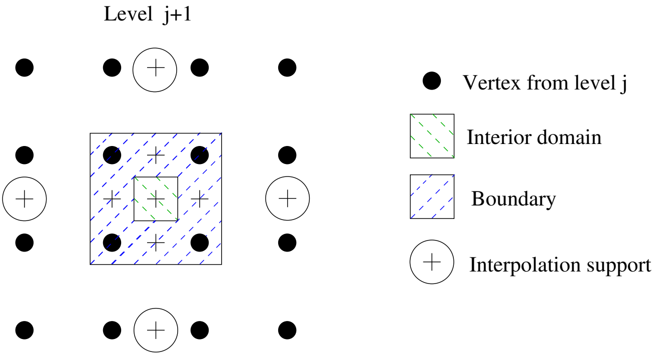


Figure: Instance of fourth-order consistent distribution of vertices with their types. Only one among the horizontal couple of interpolatory points or the vertical one is necessary.

Interpolets

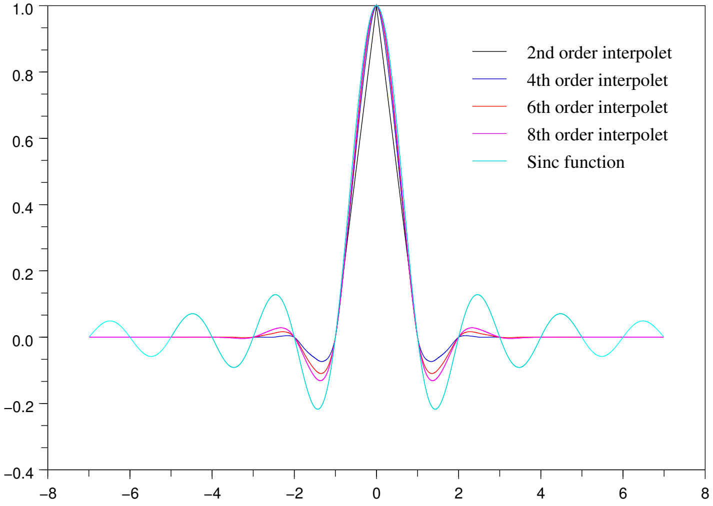


Figure: Interpolets of 2nd, 4th, 6th and 8th orders tending to the sinc function.

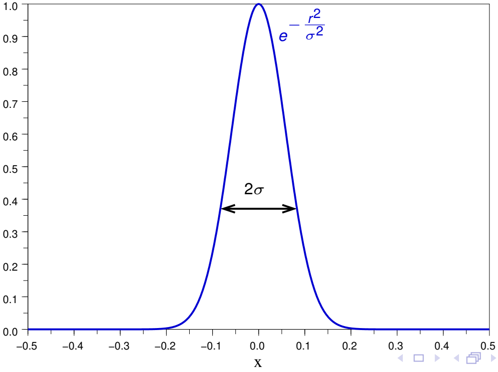
Overview

Test case: the Gaussian

$$u(\mathbf{x}) = \exp\left(-\frac{r^2}{\sigma^2}\right) \tag{9}$$

with $r = |\mathbf{x}|$ and the parameter σ small enough.

$$v(\mathbf{x}) = \left(\frac{4|\mathbf{x}|^2}{\sigma^4} - \frac{2d}{\sigma^2}\right)\exp\left(-\frac{|\mathbf{x}|^2}{\sigma^2}\right)$$



Convergence of the algorithm

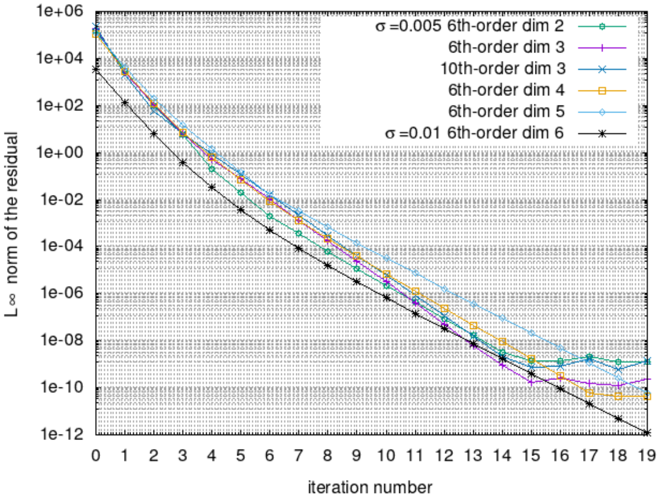


Figure: Convergence process of the multigrid algorithm. The first iterations achieve the largest reductions of the residual.

Accuracy 6th order stencils

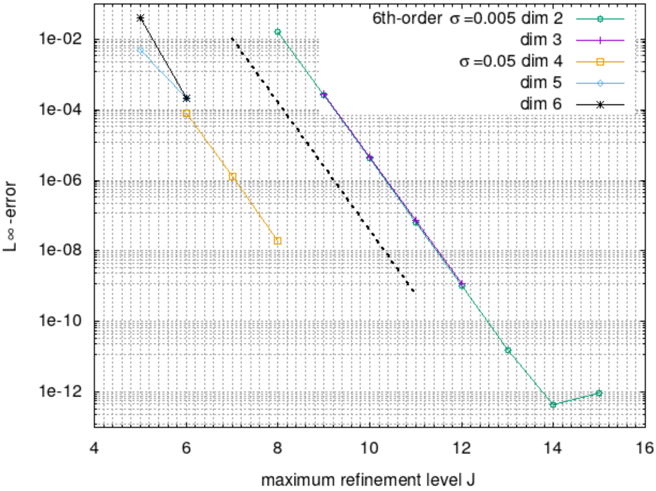


Figure: Order of convergence for the 6th-order scheme in various dimensions. The dashed line represents the 6th-order slope.

Accuracy 10th order stencil

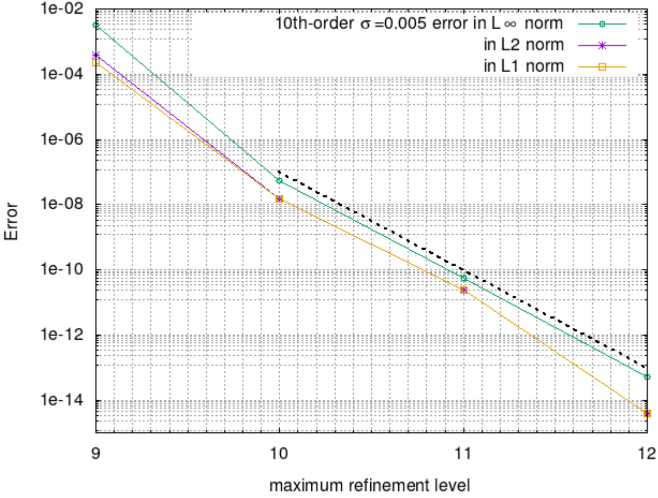


Figure: Convergence for the 3-dimensional 10th-order scheme in various norms. The dashed line represents the 10th-order slope.

Convergence with a continuous refinement

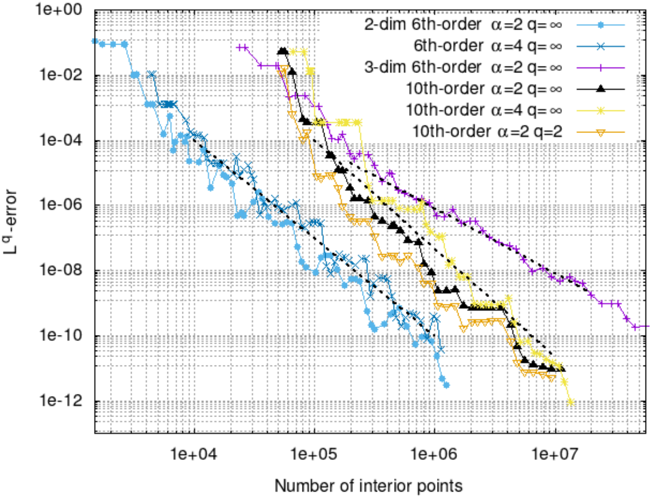


Figure: Experimental L^q-error as a function of the number of points. The theoretical slopes $\varepsilon_{L^q} = KN_{pt}^{-p/d}$ ($\frac{p}{d} = \frac{6}{2}, \frac{10}{3}, \frac{6}{3}$ from left to right) are represented by dashed lines.

Overview

Cost comparison with FMM

Total cost C of the multigrid Poisson solver:

$$C = (\#B + N_{it}(4\#A + 2 \times 3 + 2(2p + 2)))\, N_{pt} \quad (10)$$

$\#B$ and $\#A$ stand for the numbers of non zero elements of the stencils B and A, N_{it} the number of V-cycle iterations, N_{pt} the number of points and $2p$ the order of the method.
For instance, the 3-dimensional 6th-order HOC stencil verifies $\#A = 27$, $\#B = 25$ and only needs 12 iterations to converge to the computer rounding error so its cost is given by $C = 1585N_{pt}$. Of course it is possible to decrease this cost by fixing a larger error tolerance and taking fewer iterations. For instance converging to 10^{-6} only takes 5 iterations then the cost is given by $C = 675N_{pt}$.

This compares advantageously to the costs given for the pth-order Fast Multipole Method whose optimal implementation in 3 dimensions yields

$$C = 200N_{pt}p + 3.5N_{pt}p^2.$$

Which means $C = 1326N_{pt}$ for the 6th-order $p = 6$ case.

Experimental computational time with OpenMP

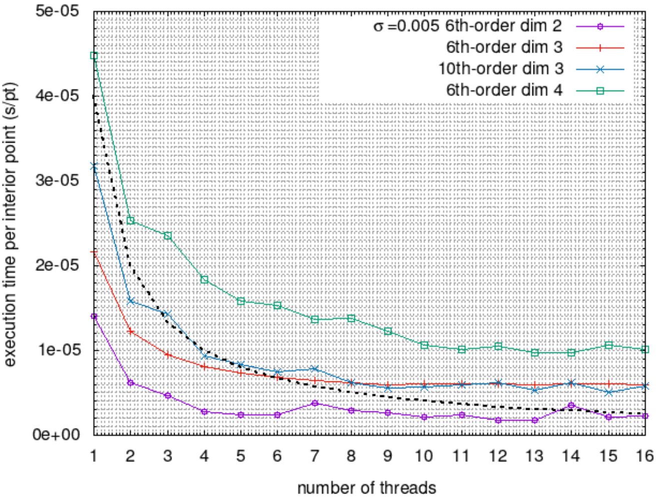


Figure: Time per interior point to converge to $\|\Delta u_n - \Delta u\| < 10^{-10}\|\Delta u\|$ when varying the number of threads.

Overview

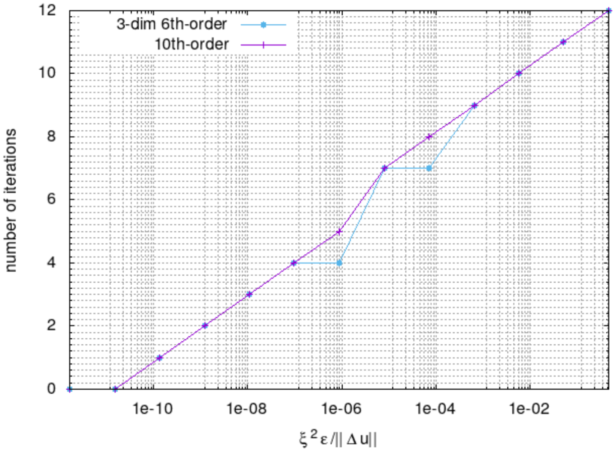
Starting from an initial guess

Starting from an initial guess u_G close to solution u, in the sense $\|\Delta u_G - \Delta u\|$ small compared to $\|\Delta u\|$, allows to decrease the number of iterations necessary to reach convergence. To test this concept and prove its efficiency, we take

$$u_G(\mathbf{x}) = u(\mathbf{x}) + \varepsilon \cos(\xi x_1) \quad \text{so} \quad \Delta u_G(\mathbf{x}) = \Delta u(\mathbf{x}) - \xi^2 \varepsilon \cos(\xi x_1), \tag{11}$$

and we look at the number of iterations needed to reach a residual equal to $10^{-10}\|\Delta u\|_\infty$. Starting without any initial guess *i.e.* $u_G = 0$, the algorithm needs exactly 10 iterations with the 6th-order and 10th-order stencils to reach this ending condition.

Number of iterations with an initial guess



Number of iterations needed to reach a residual equal to $10^{-10}\|\Delta u\|_\infty$ ($\|\Delta u\|_\infty = 2.4 \times 10^5$) and stop the iterative process, starting from an initial guess. We generate a new point by multiplying ξ by 3. At $\xi^2\varepsilon = 10^{-5}$ we shift ξ from $2187 \times 2\pi$ back to 2π and ε from 10^{-8} to 4.8×10^{-2}.

Overview

Conclusion – Perspectives

Original vertex-centered AMR multigrid Poisson solver applying to any orders and dimensions.

Perspectives:

- introduction of immersed boundary to provide an AMR immersed boundary scheme,
- MPI-parallelization,
- direct computation of the gradient of the potential from the density function as it is done in Fast Multipole Methods.