Assessing OpenFOAM discretization schemes

João Romero, Ricardo Costa, J. Miguel Nóbrega

i3N/Institute for Polymers and Composites
University of Minho, Portugal

FOAM@IBERIA 2018

Goals and Motivation

- Evaluate the accuracy and performance of various Laplacian ($\triangle$) discretization schemes
  - Order of convergence
  - Execution time
- Implicit vs Explicit approach

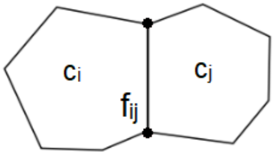
$$\underbrace{\frac{\partial}{\partial t}\rho c T + \nabla\cdot(\rho c \boldsymbol{U} T) - \nabla\cdot(k\nabla T)}_{Implicit} = \underbrace{\tau : \nabla \boldsymbol{U}}_{Explicit}$$

- Solvers in OpenFOAM 5 that explicitly calculate the Laplacian:
  - PDRFoam
  - XiFOAM
  - solidDisplacementFoam

Outline

Methodology - Notation

- c_i Cell i
- $|c_i|$ Volume of c_i
- f_{ij} Shared face between c_i and c_j
- N Number of cells in the mesh
- E_i Error of the solution at cell c_i
- $|\Omega|$ Volume of the computational domain ($|\Omega| = \sum_i |c_i|$)



Methodology - Error Analysis

$$\varepsilon_1 = \frac{\sum_i |E_i|.|c_i|}{|\Omega|} \qquad \varepsilon_\infty = \max_i |E_i|$$

For both ε_1 and ε_∞, for a 2D problem, we expect

$$\varepsilon \sim C \left(\frac{1}{\sqrt{N}} \right)^{\alpha}$$

If you apply log to both sides:

$$\log \varepsilon \sim \log C - \frac{\alpha}{2} \log N$$

1. For each scheme and each mesh type calculate ε_1 and ε_∞ using different number of cells N
2. Through linear regression calculate α_1 and α_∞

α_1 and α_∞ represent the scheme's **order of convergence**

Methodology - Implicit vs Explicit

$$\triangle \phi(x,y) = S(x,y)$$

Implicit approach (IMP) Calculate $\phi(x,y)$, given $S(x,y)$ - fvm namespace in OpenFOAM

Explicit approach (EXP) Calculate $S(x,y)$, given $\phi(x,y)$ - fvc namespace in OpenFOAM

```
fvScalarMatrix Eqn
(
    fvm::ddt(rho*c, T)
  + fvm::div(rho*c, T)
  - fvm::laplacian(k, T)
  ==
    tau && fvc::grad(U)
);
```

Methodology - Implicit vs Explicit

$$\triangle \phi(x, y) = S(x, y)$$

Implicit (Manufactured Solution Method)

1. Choose a (generally) non-trivial analytical solution ϕ_{ex} from which BCs and S are defined
2. Initialize field S
3. Solve differential equation $\triangle \phi = S$ for ϕ^1
4. Calculate the error between the numerical results (ϕ) and the analytical solution (ϕ_{ex}) - $E_i = |\phi_i - \phi_{ex,i}|$

Methodology - Implicit vs Explicit

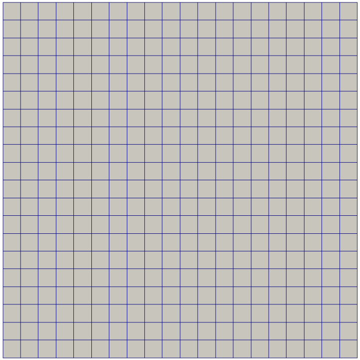
$$\triangle\phi(x,y) = S(x,y)$$

Explicit

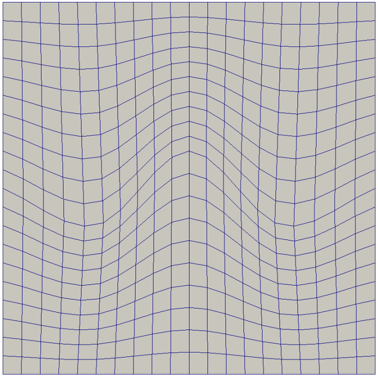
1. Choose a (generally) non-trivial analytical solution ϕ_{ex} from which BCs and S are defined
2. Initialize field ϕ with its analytical solution ϕ_{ex}
3. Explicitly calculate $\triangle\phi$
4. Calculate the error between the numerical results $(\triangle\phi)$ and the analytical solution (S) - $E_i = |\triangle\phi_i - S|$

Methodology - Domain and Meshes

Square $1m \times 1m$ computational domain



Orthogonal (OM)

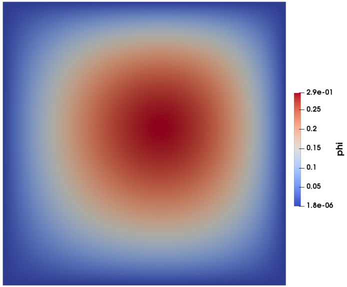


Non-orthogonal (NM)

Meshes considered

Methodology - Field ϕ

- We will be using $\phi(x,y) = xy(e - e^x)(e - e^y)$
  - It is not polynomial
  - $\phi = 0$ at the boundaries of the chosen domain

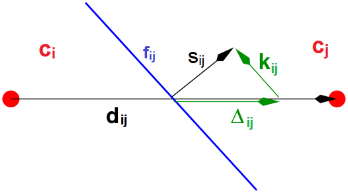


Laplacian - Numerical Calculation

The Laplacian discretization is based on Gauss's theorem. For a generic cell c_i

$$\int_{c_i} \triangle\phi ds = \oint_{\partial c_i} \nabla\phi \boldsymbol{ds} \simeq \sum_j \boldsymbol{s}_{ij} \cdot \nabla\phi_{ij}$$

For each face f_{ij}

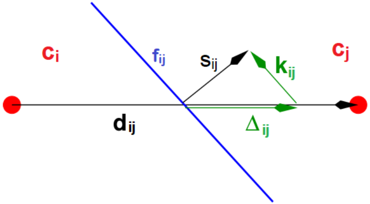


$$\triangle_{ij} = \frac{\boldsymbol{d}_{ij}}{\boldsymbol{d}_{ij} \cdot \boldsymbol{s}_{ij}} |\boldsymbol{s}_{ij}|^2$$

$$\boldsymbol{k}_{ij} = \boldsymbol{s}_{ij} - \triangle_{ij}$$

$\boldsymbol{s}_{ij} \cdot \nabla\phi_{ij}$ is calculated according to the selected scheme

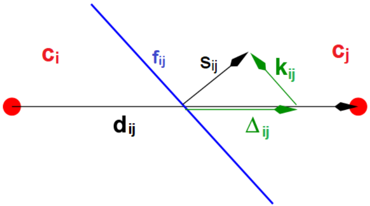
Laplacian - Numerical Calculation



Orthogonal scheme

$$\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\boldsymbol{s}_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d}_{ij}|}$$

Laplacian - Numerical Calculation

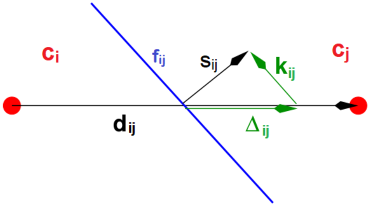


Corrected scheme - non-orthogonal correction (with linear interpolation)

$$\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\triangle_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d_{ij}}|} + \boldsymbol{k}_{ij} \cdot \nabla \phi'_{ij}$$

$\phi' =$ Field from previous iteration

$\nabla \phi'_{ij}$ - linear interpolation of $\nabla \phi'_i$ and $\nabla \phi'_j$ where field $\nabla \phi'$ is explicitly calculated, using the gradient scheme selected (e.g. Gauss **linear** corrected selects a linear scheme for $\nabla \phi'$)

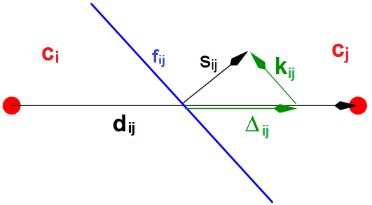


Limited scheme

Implicit (orthogonal) term forced to be larger than explicit (non-orthogonal) term for stability purposes[1]. λ is selected by user.

$$\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\triangle_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d_{ij}}|} + \min\left(\lambda \, |\triangle_{ij}| \frac{\phi_j' - \phi_i'}{|\boldsymbol{d}_{ij}|}, \boldsymbol{k_{ij}} \cdot \nabla \phi_{ij}' \right)$$

Laplacian - Numerical Calculation



Uncorrected scheme

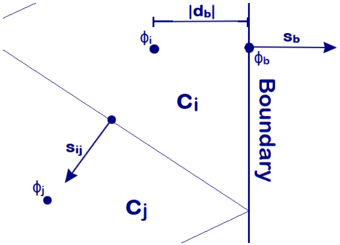
Equivalent to Limited scheme with $\lambda = 0$

$$\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = \left| \triangle_{ij} \right| \frac{\phi_j - \phi_i}{\left| \boldsymbol{d}_{ij} \right|}$$

Laplacian - Boundary Conditions

Neumann (NEU) $\mathbf{s_b} \cdot \nabla \phi_b = \mathbf{s_b} \cdot \nabla \phi_{ex,b}$

Dirichlet (DIR) $\phi_b = \phi_{ex,b}$, $\mathbf{s_b} \cdot \nabla \phi_b = |\mathbf{s_b}| \frac{\phi_b - \phi_i}{|\mathbf{d_b}|}$



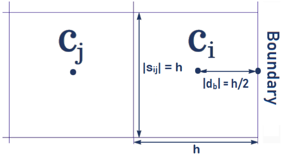
Implicit approach

Dirichlet conditions must be used in at least one boundary face, otherwise the solution is not unique

Explicit approach

Both conditions can be used

Laplacian - Boundary Conditions



Consider cell c_i at the right boundary of the domain.

$$\triangle\phi_i = \left(\frac{\partial^2\phi}{\partial x^2}\right)_i + \left(\frac{\partial^2\phi}{\partial y^2}\right)_i$$

Using Gauss's theorem

$$\left(\frac{\partial^2\phi}{\partial x^2}\right)_i = \frac{\boldsymbol{s_b}\cdot\nabla\phi_b + \boldsymbol{s}_{ij}\cdot\nabla\phi_{ij}}{|c_i|} = \frac{|\boldsymbol{s_b}|\,\frac{\phi_b-\phi_i}{|\boldsymbol{d_b}|} + |\boldsymbol{s}_{ij}|\,\frac{\phi_j-\phi_i}{|\boldsymbol{d_{ij}}|}}{|c_i|}$$

In an orthogonal uniform squared mesh:

$$|\boldsymbol{s}| = |\boldsymbol{s_{ij}}| = |\boldsymbol{d_{ij}}| = h,\ |\boldsymbol{d}| = \frac{h}{2},\ |c_i| = h^2$$

Laplacian - Boundary Conditions

$$\left(\frac{\partial^2\phi}{\partial x^2}\right)_i = \frac{2\phi_b - 3\phi_i + \phi_j}{h^2}$$

Simplified notation

$$\phi(x) = \phi(x, y_i),\ \phi'(x) = \frac{\partial\phi(x,y_i)}{\partial x},\ \phi''(x) = \frac{\partial^2\phi(x,y_i)}{\partial x^2}.$$

Then, $\phi_i = \phi(x_i)$, $\phi_b = \phi\left(x_i + \frac{h}{2}\right)$, $\phi_j = \phi(x_i - h)$.
Thus, the numerical evaluation of $\left(\frac{\partial^2\phi}{\partial x^2}\right)_i = \phi''(x_i)$ yields

$$\phi''_{num}(x_i) = \frac{2\phi\left(x_i + \frac{h}{2}\right) - 3\phi(x_i) + \phi(x_i - h)}{h^2}$$

Laplacian - Boundary Conditions

The Taylor expansions of $\phi\left(x_i - h\right)$ and $\phi\left(x_i + \frac{h}{2}\right)$ of order 3 are:

$$\phi\left(x_i - h\right) = \phi\left(x_i\right) - \phi'\left(x_i\right)h + \frac{\phi''\left(x_i\right)}{2!}h^2 - \frac{\phi^{(3)}\left(x_i\right)}{3!}h^3$$

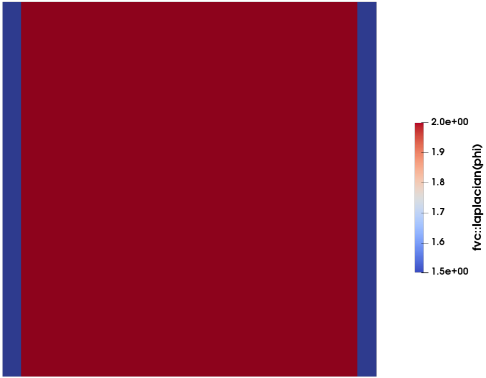
$$\phi\left(x_i + \frac{h}{2}\right) = \phi\left(x_i\right) + \phi'\left(x_i\right)\frac{h}{2} + \frac{\phi''\left(x_i\right)}{2!}\left(\frac{h}{2}\right)^2 + \frac{\phi^{(3)}\left(x_i\right)}{3!}\left(\frac{h}{2}\right)^3$$

Plugging this into the formula from the previous slide yields

$$\phi''_{num}\left(x_i\right) = \frac{3}{4}\phi''\left(x_i\right) + \boldsymbol{O}\left(h\right)$$

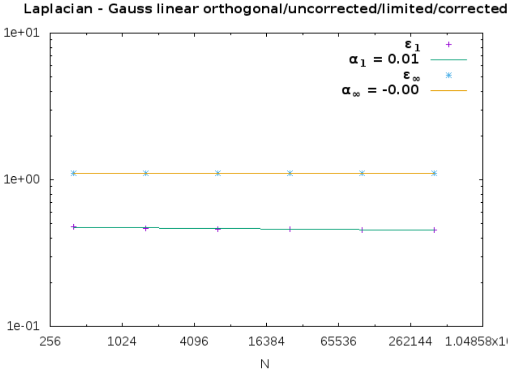
Dirichlet conditions **lead to the calculation of the only $\frac{3}{4}\triangle\phi$ instead of $\triangle\phi$ in the direction perpendicular to the boundary.** In the tested function this direction is dominant.

Laplacian - Results (EXP-OM-DIR)

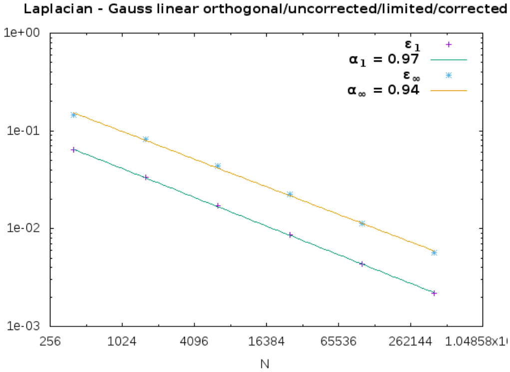


$$\phi(x,y) = x^2$$

Laplacian - Results on Boundary Cells (EXP-OM-DIR)

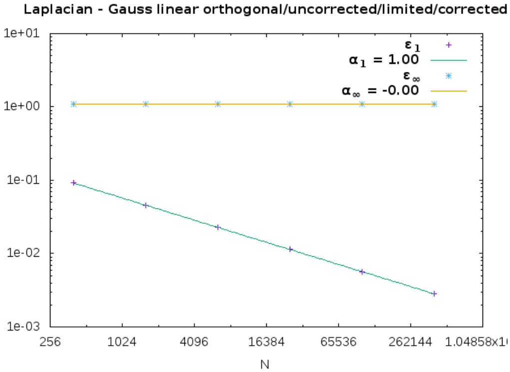


$$E_i = \left| \triangle \phi_{ex,i} - \triangle \phi_i \right|$$

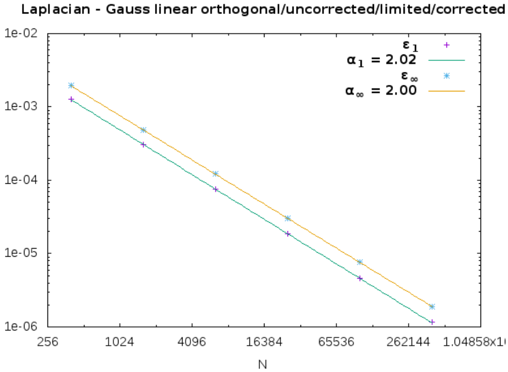


$$E_i = \left| \triangle \phi_{ex,i} - \frac{4}{3} \triangle \phi_i \right|$$

Laplacian - Results (EXP-OM-DIR)

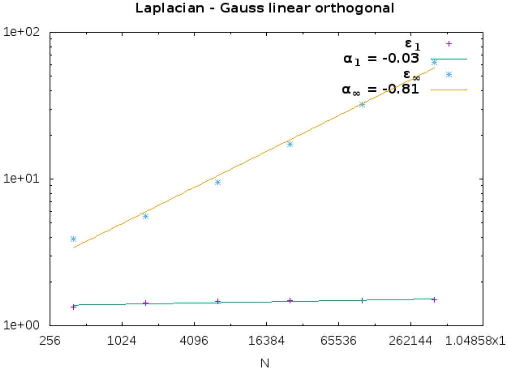


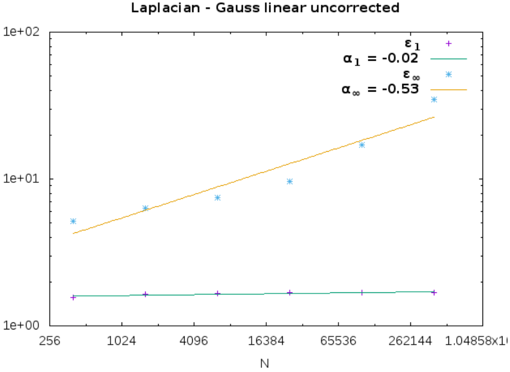
All domain



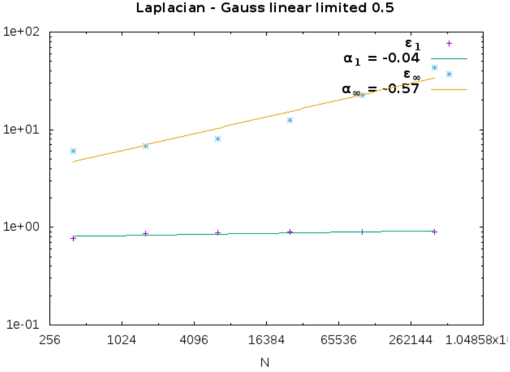
Inner cells

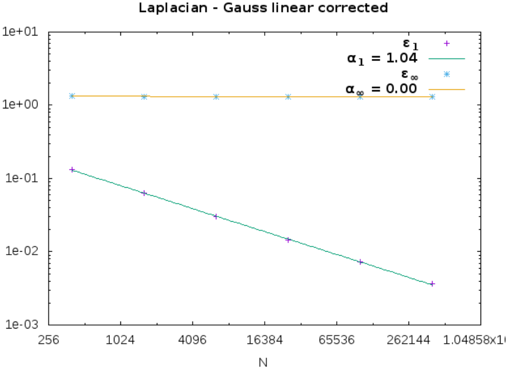
Laplacian - Results (EXP-NM-DIR)



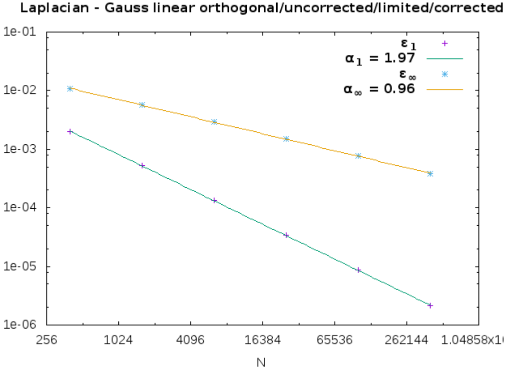


Laplacian - Results (EXP-NM-DIR)

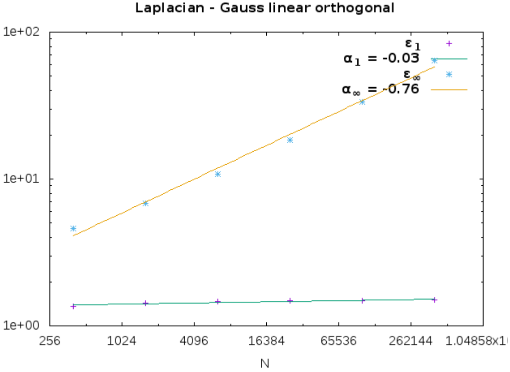


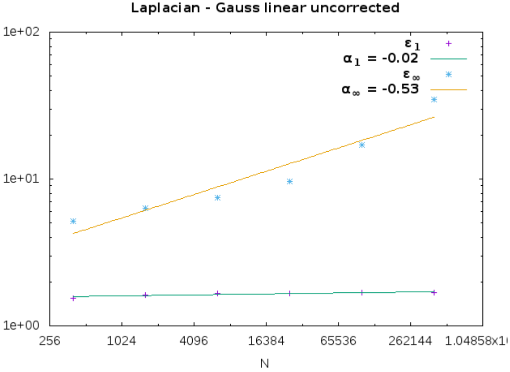


Laplacian - Results (EXP-OM-NEU)

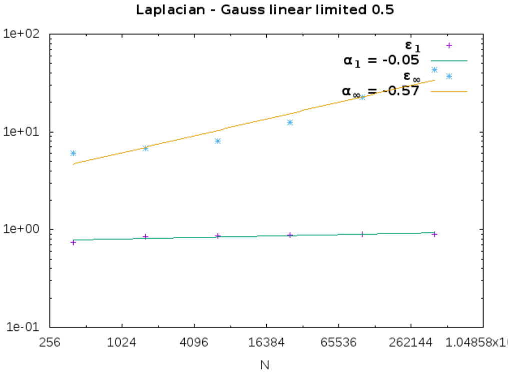


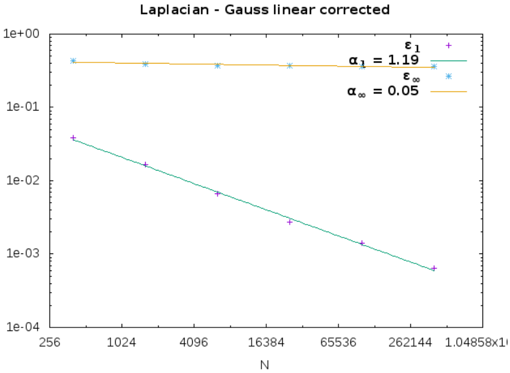
Laplacian - Results (EXP-NM-NEU)





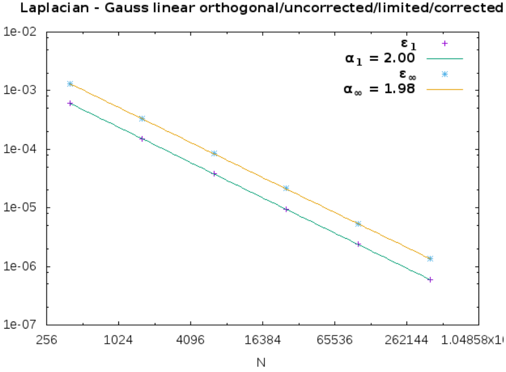
Laplacian - Results (EXP-NM-NEU)



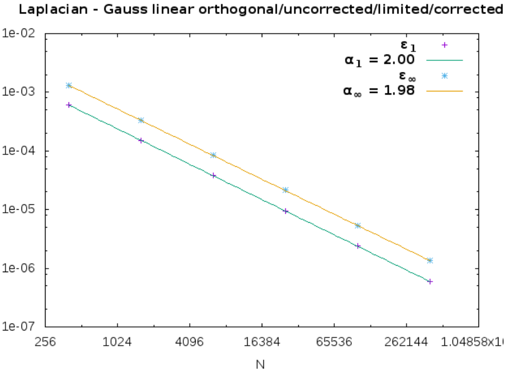


Neumann boundary conditions are advised

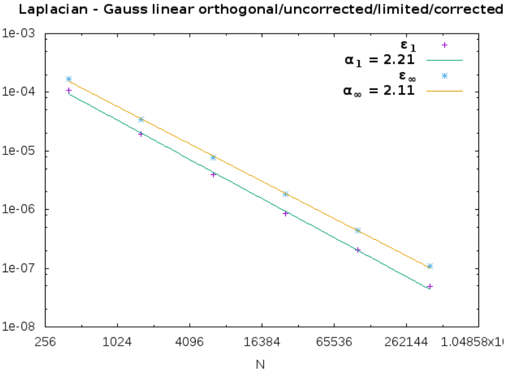
Laplacian - Results (IMP-OM)



Laplacian - Results (IMP-OM)



$S(x,y) = \triangle\phi_{ex}$ for all domain

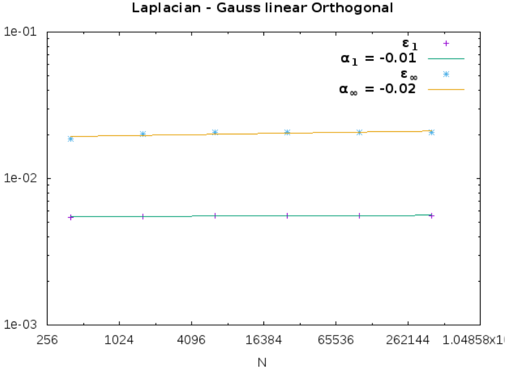


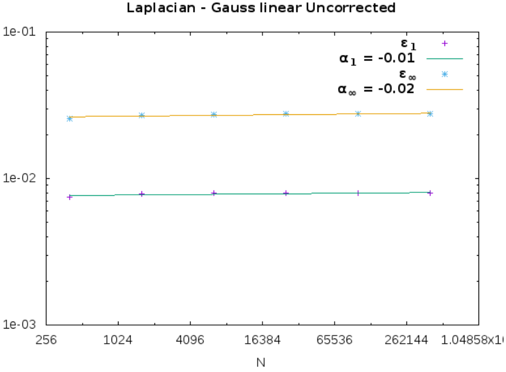
$S(x,y) = \frac{3}{4}\triangle\phi_{ex}$ for boundary cells

$S(x,y) = \triangle\phi_{ex}$ for inner cells

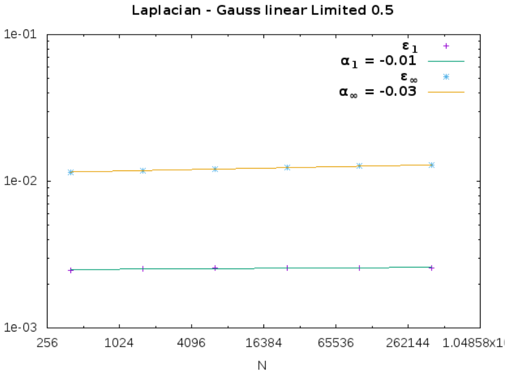
$$\triangle\phi(x,y) = S(x,y)$$

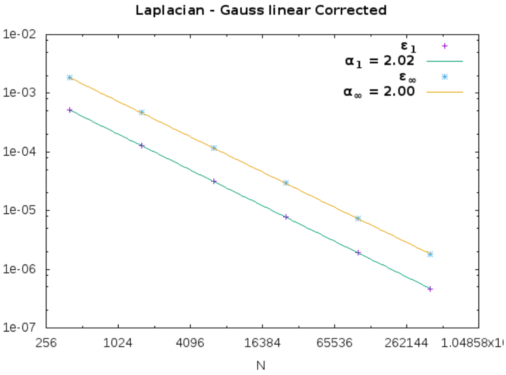
Laplacian - Results (IMP-NM)





Laplacian - Results (IMP-NM)





Laplacian - Execution Time

Othogonal $\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\boldsymbol{s}_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d}_{ij}|}$

Uncorrected $\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\triangle_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d}_{ij}|}$

Corrected $\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\triangle_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d}_{ij}|} + \boldsymbol{k}_{ij} \cdot \nabla \phi'_{ij}$

Limited $\boldsymbol{s}_{ij} \cdot \nabla \phi_{ij} = |\triangle_{ij}| \frac{\phi_j - \phi_i}{|\boldsymbol{d}_{ij}|} + \min\left(\lambda\, |\triangle_{ij}| \frac{\phi'_j - \phi'_i}{|\boldsymbol{d}_{ij}|}, \boldsymbol{k_{ij}} \cdot \nabla \phi'_{ij}\right)$

Execution time relative to Limited 0.5 scheme.

Scheme	Orthogonal	Uncorrected	Corrected	Limited 0.5
Ex. Time (s)	6.19%	16.19%	85.71%	100%

Conclusions

- A methodology to quantify the order of convergence of OpenFOAM discretization schemes was proposed, using Explicit and Implicit
- The methodology was tested for the Laplacian operator in orthogonal and non-orthogonal meshes, using Neumann and Dirichlet boundary conditions
- The schemes used lead to an error in the calculation of the Laplacian at boundary cells
- The order of convergence of discretization schemes is different
- Laplacian schemes with no non-orthogonal corrections (and even partially corrected) should be avoided in non-orthogonal meshes
- Non-orthogonal corrections have a significant impact on the scheme's execution time
- **Future work**:
  - Apply the same methodology to other differential operators
  - Extend the analysis

Acknowledgments

The authors acknowledge the funding from COMPETE/FEDER and Fundação para a Ciência e a Tecnologia (FCT), Portugal, through project PTDC/ENS-ENE/3362/2014 and FEDER, under PEst-C/CTM/LA0025/2013.