AMSI VACATION RESEARCH SCHOLARSHIPS 2020–21

Get a Thirst for Research this Summer

The Performance of Multiscale Finite Element Methods

Jacob Vandenberg

Supervised by Sandiago Badia & Alberto Martin

Monash University

Vacation Research Scholarships are funded jointly by the Department of Education, Skills and Employment and the Australian Mathematical Sciences Institute.

AMSI Australian Mathematical Sciences Institute

Australian Government
Department of Education, Skills and Employment

Contents

1 Abstract

In this work we present and implement the primal hybrid method as a numerical solver for Poisson's equation. The convergence rate of the L^2 error was found to be two when using linear functions on the bulk of the domain and constant functions on the boundaries. This work serves as the precursor to analysing multiscale finite element solvers in the finite element fraamework Gridap. Further research into developing this method, and applying it to Darcy's equation is needed.

2 Introduction

Some physical modelling problems, such as the extraction of oil through rocks, involve multiscale features. Multiscale features occur when different areas of the domain have significantly different properties, particularly when one area has much smaller features than the other, often called subscale features. When modelling the extraction of oil, this might occur at the join between different types of rock. If one type is very porous, and the other is not, then we are going to have significantly different properties in each area. In some cases, the scale of the solution will be much smaller in one area when compared with the other. So it is desirable to have a solver which can adapt to these scenarios, in order to provide an accurate and reliable numerical estimate.

Porous flow problems are usually modelled using the Darcy equation, which is a partial differential equation (PDE). A modern approach to numerically solving a PDE is using the Finite Element Method (FEM). In traditional finite element methods, we require that shape functions be continuous across cell boundaries. If this requirement can be relaxed, we will have much more flexibility in the choice of shape functions, allowing us to adapt the solver to multiscale features. In this work we will present the primal hybrid method, which is a precursor to a multiscale method.

3 Statement of Authorship

Santiago Badia developed the outline of the project, and the theory underpinning the research. Jacob Vandenberg and Alberto Martin developed the Julia implementation of the primal hybrid method. Jacob Vandenberg wrote this report and conducted the numerical analysis. Santiago Badia and Alberto Martin supervised the project and proofread the report.

4 Problem statement

4.1 Strong Form

We will consider the problem:
Given $f, u_D \in C^0(\Omega)$, find $u \in C^2(\Omega)$ such that

$$-\Delta u = f$$

$$u|_{\partial\Omega} = u_D$$

4.2 Defining L^2 and H^1 spaces.

We will define the L^2 norm $\|\cdot\|_{L^2}$.

$$\|v\|_{L^2} = \left(\int_\Omega |v(x)|^2 \, \mathrm{d}x \right)^{1/2}$$

Then we can define the L^2 Sobolev space.

$$L^2(\Omega) = \{\|v\|_{L^2} < \infty\}$$

The H^1 norm $\|\cdot\|_{H^1}$ is defined as:

$$\|v\|_{H^1} = \left(\int_\Omega |\boldsymbol{\nabla} v(x)|^2 \, \mathrm{d}x + \int_\Omega |v(x)|^2 \, \mathrm{d}x \right)^{1/2}$$

Likewise we can define the Sobolev space H^1.

$$H^1(\Omega) = \{\|v\|_{H^1} < \infty\}$$

The space H^1_0 is the H^1 space where all functions vanish on the boundary.

$$H^1_0(\Omega) = \{\|v\|_{H^1} < \infty;\, v|_{\partial\Omega} = 0\}$$

Similarly, $H^1_{u_D}$ is the H^1 space where all functions which are equal to u_D when restricted to the boundary.

4.3 Weak Form

To find an approximate solution, we will consider the weak form of the problem:
Given $f \in L^2(\Omega)$, find $u \in H^1_{u_D}(\Omega)$ such that:

$$\int_\Omega \nabla u \cdot \nabla v \, d\Omega = \int_\Omega f v \, d\Omega, \forall v \in H^1_0(\Omega). \tag{1}$$

5 The Primal Hybrid Formulation

Decompose the domain Ω into R disjoint subdomains $\Omega = \bigcup_{r=1}^R \Omega_r$. We can define the facets ∂T as the set of boundaries of the subdomains $\partial\Omega_i$, $i \in \{1, \ldots, R\}$.

The primal hybrid formulation divides the previous problem into solving for both *primal* and *dual* unknowns. The primal unknown u is defined on the bulk of the domain, and represents the solution to the problem, as before. The dual unknown λ acts as a Lagrange multiplier to weakly enforce the continuity of the solution, and is defined on the facets of the subdomains ∂T. Given the outwards unit normal $\mathbf{n}$, λ can be expressed as

$$\lambda = \boldsymbol{\nabla} u \cdot \mathbf{n}.$$

Let M be the *broken* space of $H^{-1/2}$ functions which exist on the boundaries of these subdomains. M is broken in that elements of M only have to exist in $H^{-1/2}$ when restricted to any facet $\partial\Omega_i$, $i \in \{1, \ldots, R\}$. This allows the functions to be discontinuous between cells. We will also define X as the broken space of H^1 functions on the bulk of the domain Ω.

As shown by Raviart & Thomas (1977), (1) can be expressed as follows.

Given $f \in L^2(\Omega)$, find $\mathbf{u} \in X$, and $\boldsymbol{\lambda} \in M$ such that

$$\begin{aligned} \sum_{r=1}^R \int_{\Omega_r} \boldsymbol{\nabla}\mathbf{u} \cdot \boldsymbol{\nabla} v \, \mathrm{d}\Omega - \sum_{r=1}^R \int_{\partial\Omega_r} v\boldsymbol{\lambda} \, \mathrm{d}\partial\Omega_r &= \int_\Omega f v \, \mathrm{d}\Omega \\ - \sum_{r=1}^R \int_{\partial\Omega_r} \mathbf{u}\mu \, \mathrm{d}\partial\Omega_r &= \int_{\partial\Omega} u_D \mu \, \mathrm{d}\partial\Omega, \end{aligned}$$

for all $v \in X$, $\mu \in M$.

6 Galerkin Method

6.1 Weak formulation

For any face E, we can assign the two adjacent cells to be K^+ and K^-. We can then define $\mathbf{n}^+$ to be the outward normal of E with respect to K^+. $\mathbf{n}^-$ can be defined similarly. We can then define the jump operator $[\cdot]$.

$$[v] = \mathbf{n}^+ v^+ + \mathbf{n}^- v^-,$$

where v^+ the value of v when approached from the $\mathbf{n}^+$ direction, and analogously for v^-

The weak formulation is as follows:

Given $f \in L^2(\Omega)$, find $\mathbf{u} \in X$, and $\boldsymbol{\lambda} \in M$ such that

$$\int_\Omega \boldsymbol{\nabla}\mathbf{u} \cdot \boldsymbol{\nabla} v \, \mathrm{d}\Omega + \int_{\partial T} [v] \cdot \boldsymbol{\lambda}\boldsymbol{n} \, \mathrm{d}\partial T = \int_\Omega f v \, \mathrm{d}\Omega, \ \forall v \in X$$
$$\int_{\partial T} [\mathbf{u}] \cdot \mu \boldsymbol{n} \, \mathrm{d}\partial T = \int_{\partial\Omega} u_D \mu \, \mathrm{d}\partial\Omega, \ \forall \mu \in M$$

for all $v \in X$, $\mu \in M$.

6.2 Dimension Reduction

We will replace the infinite dimensional spaces X and M with the finite vector spaces U and Λ.

$$U = \operatorname{span}\{u_1, u_2, \ldots, u_n\}$$
$$\Lambda = \operatorname{span}\{\lambda_1, \lambda_2, \ldots, \lambda_k\},$$

where n and k are the dimensions of U and Λ respectively.

6.3 Linear System

We can express our solution as

$$\mathbf{u} = \sum_{i=1}^{n} \alpha_i u_i$$
$$\boldsymbol{\lambda} = \sum_{i=1}^{k} \beta_i \lambda_i.$$

Substituting this into the weak formulation results in the following block form linear system.

$$\left[\begin{array}{c|c} A & B \\ \hline B^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline \boldsymbol{\beta} \end{array}\right] = \left[\begin{array}{c} \mathbf{f} \\ \hline \mathbf{g} \end{array}\right],$$

where the entries are defined as follows:

$$A_{ij} = \int_\Omega \boldsymbol{\nabla} u_i \cdot \boldsymbol{\nabla} u_j \, \mathrm{d}\Omega, \tag{2}$$

$$B_{ij} = \int_S [u_i] \cdot \lambda_j \boldsymbol{n} \, \mathrm{d}S, \tag{3}$$

$$\mathbf{f}_i = \int_\Omega f u_i \, \mathrm{d}\Omega. \tag{4}$$

$$\mathbf{g}_i = \int_{\partial\Omega} u_D \lambda_i \, \mathrm{d}\partial\Omega.$$

7 Finite Element Method

7.1 Finite elements on the bulk

We can define a simplicial mesh $\mathcal{M}$ of our domain Ω, and let our subdomains Ω_r be the cells of $\mathcal{M}$. We will define our finite elements as done by Ciarlet (2002) as a set $\{K, P, \Sigma\}$. $K = \Omega_r$ is a subset of $\mathbb{R}^D$, and is a cell of the mesh. P is a vector space of functions defined on K. Σ is a set of linear functionals ϕ_i defined over P.

For the primal hybrid method P will be the space of order k polynomials. We will define a basis $\{p_1, \ldots, p_N\}$ such that $\phi_i(p_j) = \delta_{ij}$.

7.2 Finite elements on the trace

We will also define a simplicial mesh $\mathcal{S}$ of the set facets ∂T. The finite elements are defined as a set $\{E, P, \Sigma\}$. E is a subset of $\mathbb{R}^D$, and a facet of the mesh $\mathcal{M}$. E is a subset of $\mathbb{R}^D$, and a facet of the mesh $\mathcal{M}$. Q is a vector space of functions defined on E. Ξ is a set of linear functionals ψ_i defined over P.

For stability, the order of polynomials defined on the facets should be $k-1$ (Raviart & Thomas 1977). So a basis of order $k-1$ polynomials $q_i : \mathbb{R}^{D-1} \to \mathbb{R}^D$ will also be chosen such that $\psi_i(q_j) = \delta_{ij}$.

7.3 Local-global mapping

Each shape function p_i, will have a local ID $1 \leq i \leq N$, where N refers to the dimension of P. We will also assign each shape function a unique global ID $1 \leq i_{glob} \leq N_{glob}$, where N_{glob} is the total number of shape functions from all finite elements. We will assign the global IDs such that the global ID of any shape function λ_i is larger than the global ID any shape function u_j, to ensure that equations (2) and (3) still hold for the global matrix. Given a finite element $\{K, P, \Sigma\}$, let I be a local to global mapping, such that $p_i = u_{I(i)}$. This mapping is defined analogously for the trace finite elements, with $q_i = \lambda_{I(i)}$.

7.4 Reference Element

Instead of evaluating the shape functions on the physical domain Ω, we can define a *reference* finite element. They will be defined as $\left\{\hat{K}, \hat{P}, \hat{\Sigma}\right\}$ and $\left\{\hat{E}, \hat{Q}, \hat{\Xi}\right\}$ for the finite elements on the bulk and on the trace respectively.

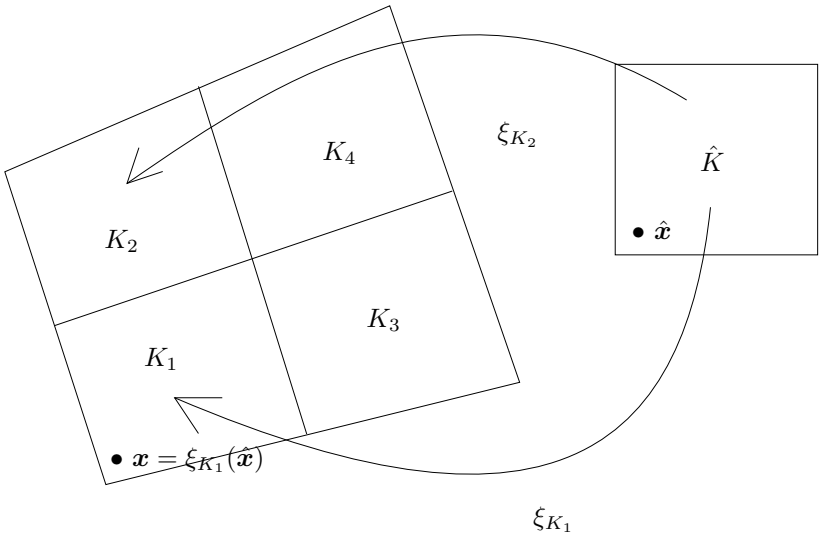

Figure 1: Figure depicting the mapping from a square reference element (right) to the physical finite elements, using their respective cell maps ξ_{K_1} and ξ_{K_2}

For each finite element in the physical space we will define a *cell map*. The cell map ξ is a geometrical mapping, such that $K = \xi(\hat{K})$. Now we can use the mapping to construct the element on the physical domain as follows.

$$
\begin{aligned}
K &= \xi\left(\hat{K}\right) \\
P &= \left\{\hat{p}_i \circ \xi^{-1} \mid \hat{p}_i \in \hat{P}\right\} \\
\Sigma &= \left\{\hat{\phi}_i \circ \xi^{-1} \mid \hat{\phi}_i \in \hat{\Sigma}\right\}
\end{aligned}
$$

Similarly, we can construct all physical finite elements defined on the model facets.

8 Integration

Since shape functions vanish outside of their respective cells, we can integrate cell-wise, rather than over the whole domain. Furthermore, by using the cell maps, the number of times the shape functions have to be evaluated is significantly reduced. The new integration rules are as follows, where $\mathbf{J}$ is the Jacobian of the cell map. The geometrical map $\zeta_E : \hat{E} \to \hat{K}$ takes the reference face $\hat{E}$ to the reference cell $\hat{K}$, and depends on the face E.

$$\int_\Omega \nabla u_{I(i)} \cdot \nabla u_{I(j)} \,\mathrm{d}\Omega = \int_{\hat{K}} \left(\mathbf{J}^{-T}\nabla\hat{\phi}_i\right) \cdot \left(\mathbf{J}^{-T}\nabla\hat{\phi}_j\right) |\mathbf{J}| \,\mathrm{d}\hat{K}$$
$$\int_S \left[u_{I(i)}\right] \cdot \lambda_{I(j)-n}\boldsymbol{n} \,\mathrm{d}S = \int_{\hat{E}} \left[\hat{\phi}_i \circ \zeta_E\right] \cdot \hat{\lambda}_j \mathbf{n} \,|\mathbf{J}| \,\mathrm{d}\hat{E},$$
$$\int_\Omega f u_{I(i)} \,\mathrm{d}\Omega = \int_{\hat{K}} (f \circ \xi)\, \hat{\phi}_i \,|\mathbf{J}| \,\mathrm{d}\hat{K}.$$

We can utilise Gaussian quadrature of order $2k$ for each of these transformed integrals which will be exact for polynomials. The integral (4) will be approximate for most choices of f, however an exact integration scheme could be considered.

9 Results

This method was implemented in Gridap, an open source finite element framework written in Julia (Badia & Verdugo 2020). We will analyse the error convergence using a two dimensional manufactured solution u, where

$$u(x, y) = \sin(\pi x)\sin(\pi y).$$

A comparison of the actual solution and the hybrid method approximation can be seen in figures 2 and 3. Using $k = 1$, we obtain a convergence rate of 2 in the L^2 norm of the error. This can be seen in figure 4

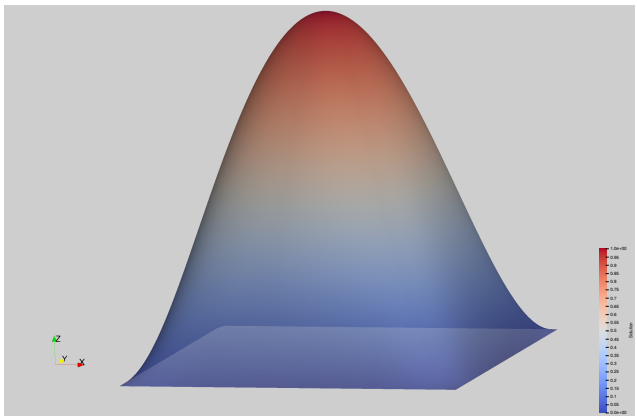

Figure 2: The exact solution $u(x, y) = \sin(\pi x)\sin(\pi y)$

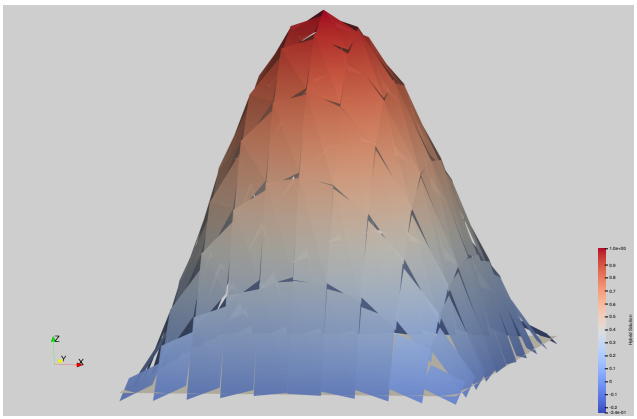

Figure 3: Numerical approximation of the solution using the hybrid finite element method

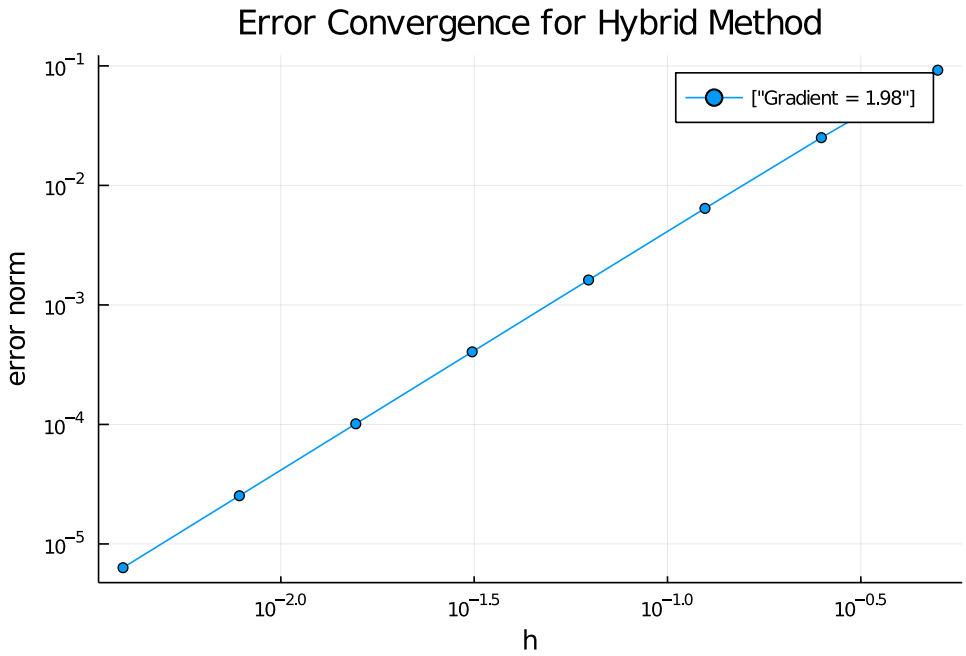

Figure 4: L^2 error norm as the diameter of cells h approaches zero.

10 Extension to multiscale methods

Previously every finite element was constructed from the singular reference finite element. Because the continuity of the solution is only weakly enforced with the test function μ, we can modify the primal hybrid method by using different vector spaces in each finite element. We could even define a submesh within coarser cells, making the vector space a finite element space. Not only would this allow for the solver to adapt to subscale features, it would also allow for parts of the algorithm to be parallelised.

11 Conclusion

In this paper we implemented the primal hybrid method and analysed its performance. This solver differs from the standard finite element method by allowing for discontinuities in the shape functions across the cell boundaries, and allowing the shape functions to be non-zero at the boundary. The continuity of the solution is only weakly enforced by the dual variable λ, which is defined on the boundaries of the cells. Relaxing the constraints on the functions that can be defined in the cells means that this method directly leads to a multiscale method. Future research should focus on implementing a multiscale method, which would utilise different shape functions in different areas of the domain. This method would adapt to multiscale problems, such as oil extraction, and also parallelise to a large extent.

References

Badia, S. & Verdugo, F. (2020), 'Gridap: An extensible finite element toolbox in julia', *Journal of Open Source Software* **5**(52), 2520.
URL: *https://doi.org/10.21105/joss.02520*

Ciarlet, P. G. (2002), *The Finite Element Method for Elliptic Problems*, Society for Industrial and Applied Mathematics.
URL: *https://epubs.siam.org/doi/abs/10.1137/1.9780898719208*

Raviart, P. A. & Thomas, J. M. (1977), 'Primal hybrid finite element methods for 2nd order elliptic equations', *Mathematics of Computation* **31**(138), 391.
URL: *https://doi.org/10.2307/2006423*