Politecnico di Milano
School of Industrial and Information Engineering
M.Sc. in High Performance Computing

Hybrid thread-MPI parallelization for ADR equation

Peng Rao, Jiali Claudio Huang, Ruiying Jiao

Repo: https://github.com/Peng-Rao/HybridADRSolver

Contents

1 Problem statement

1.1 Strong formulation

Consider the following **Advection-Diffusion-Reaction** equation with mixed Dirichlet-Neumann boundary conditions:

$$\begin{cases} -\nabla \cdot (\mu \nabla u) + \nabla \cdot (\boldsymbol{\beta} u) + \gamma u = f & \text{in } \Omega, \\ u = g & \text{on } \Gamma_D \subset \partial\Omega, \\ \nabla u \cdot \boldsymbol{n} = h & \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{cases}$$

where:

- $\Omega \subset \mathbb{R}^d$ (with $d = 1, 2, 3$) is an open bounded domain with boundary $\partial\Omega$;
- $\mu > 0$ is the diffusion coefficient;
- $\boldsymbol{\beta} \in [L^\infty(\Omega)]^d$ is the advection velocity field;
- $\gamma \geq 0$ is the reaction coefficient;
- $f \in L^2(\Omega)$ is a source term;
- $g \in H^{1/2}(\Gamma_D)$ is the Dirichlet boundary data;
- $h \in L^2(\Gamma_N)$ is the Neumann boundary data;
- $\boldsymbol{n}$ is the outward unit normal vector on the boundary $\partial\Omega$.
- u is the unknown scalar function to be solved for.
- Γ_D and Γ_N are the Dirichlet and Neumann parts of the boundary, respectively.

1.2 Weak formulation

We begin by defining the trial and test function spaces. To accommodate the non-homogeneous Dirichlet boundary condition, we introduce a *lifting function* $u_g \in H^1(\Omega)$ such that $u_g = g$ on Γ_D:

$$V_g := \{v \in H^1(\Omega) : v = g \quad \text{on } \Gamma_D\}$$

The test space is the linear subspace of $H^1(\Omega)$ with homogeneous Dirichlet boundary conditions:

$$V_0 := \{v \in H^1(\Omega) : v = 0 \quad \text{on } \Gamma_D\}$$

Consequently, we decompose the solution as $u = u_0 + u_g$, where the unknown $u_0 \in V_0$. Multiply the equation by a test function $v \in V_0$ and integrate over the domain Ω:

$$\int\limits_{\Omega} (-\nabla \cdot (\mu \nabla u) + \nabla \cdot (\boldsymbol{\beta} u) + \gamma u)\, v \, d\Omega = \int\limits_{\Omega} f v \, d\Omega$$

Using the linearity of the integral, we separate the terms:

$$-\int\limits_{\Omega} \nabla \cdot (\mu \nabla u) v \, d\Omega + \int\limits_{\Omega} \nabla \cdot (\boldsymbol{\beta} u) v \, d\Omega + \int\limits_{\Omega} \gamma u v \, d\Omega = \int\limits_{\Omega} f v \, d\Omega$$

We apply Green's first identity to the diffusion term to reduce the order of differentiation:

$$-\int\limits_{\Omega} \nabla \cdot (\mu \nabla u) v \, d\Omega = \int\limits_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega - \int\limits_{\partial\Omega} (\mu \nabla u \cdot \mathbf{n}) v \, d\Gamma$$

Substituting this back into the integral equation:

$$\int\limits_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega - \int\limits_{\partial\Omega} (\mu \nabla u \cdot \mathbf{n}) v \, d\Gamma + \int\limits_{\Omega} \nabla \cdot (\boldsymbol{\beta} u) v \, d\Omega + \int\limits_{\Omega} \gamma u v \, d\Omega = \int\limits_{\Omega} f v \, d\Omega$$

Split the boundary integral into contributions from Γ_D and Γ_N:

$$\int\limits_{\partial\Omega} (\mu \nabla u \cdot \mathbf{n}) v \, d\Gamma = \int\limits_{\Gamma_D} (\mu \nabla u \cdot \mathbf{n}) v \, d\Gamma + \int\limits_{\Gamma_N} (\mu \nabla u \cdot \mathbf{n}) v \, d\Gamma$$

Since $v = 0$ on Γ_D, the first term vanishes. On Γ_N, we use the Neumann condition $\nabla u \cdot \mathbf{n} = h$:

$$\int\limits_{\Gamma_N} (\mu \nabla u \cdot \mathbf{n}) v \, d\Gamma = \int\limits_{\Gamma_N} \mu h v \, d\Gamma$$

Substituting back, we have:

$$\int\limits_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int\limits_{\Omega} \nabla \cdot (\boldsymbol{\beta} u) v \, d\Omega + \int\limits_{\Omega} \gamma u v \, d\Omega = \int\limits_{\Omega} f v \, d\Omega + \int\limits_{\Gamma_N} \mu h v \, d\Gamma$$

Substituting $u = u_0 + u_g$ into the equation, we get:
Find $u_0 \in V_0$ such that

$$a(u_0, v) = F(v), \quad \forall v \in V_0$$

where the *bilinear form* $a : V_0 \times V_0 \to \mathbb{R}$ is defined as:

$$a(u, v) = \int\limits_{\Omega} \mu \nabla u \cdot \nabla v \, d\Omega + \int\limits_{\Omega} \nabla \cdot (\boldsymbol{\beta} u) v \, d\Omega + \int\limits_{\Omega} \gamma u v \, d\Omega$$

and the *linear functional* $F : V_0 \to \mathbb{R}$ is given by:

$$F(v) = \int\limits_{\Omega} f v \, d\Omega + \int\limits_{\Gamma_N} \mu h v \, d\Gamma - a(u_g, v)$$

1.3 Manufactured solution

We define the exact solution $u_{\text{ex}} : \Omega \to \mathbb{R}$ on the unit hypercube domain $\Omega = [0, 1]^d$ (where $d = 2, 3$) as the product of sine functions:

$$u_{\text{ex}}(\mathbf{x}) = \prod_{i=1}^{d} \sin(\pi x_i). \tag{1}$$

This function vanishes on the boundary hyperplanes where $x_i = 0$ or $x_i = 1$, making it naturally suitable for homogeneous Dirichlet boundary conditions.

The physical coefficients for the benchmark problem are chosen as follows:

- **Diffusion:** A constant isotropic diffusion coefficient $\mu = 1.0$.
- **Reaction:** A constant reaction coefficient $\gamma = 0.1$.
- **Advection:** A rotational velocity field $\boldsymbol{\beta}(\mathbf{x})$, defined to make the problem non-symmetric:

$$\boldsymbol{\beta}(\mathbf{x}) = \begin{cases} \begin{bmatrix} -x_2 \\ x_1 \end{bmatrix} & \text{if } d = 2, \\ \begin{bmatrix} -x_2 \\ x_1 \\ 0.1 \end{bmatrix} & \text{if } d = 3. \end{cases} \tag{2}$$

Substituting u_{ex} into the governing equation $-\nabla \cdot (\mu \nabla u) + \nabla \cdot (\boldsymbol{\beta} u) + \gamma u = f$, we compute the source term f.

First, we observe that the Laplacian of the chosen exact solution is:

$$\Delta u_{\text{ex}} = \sum_{i=1}^{d} \frac{\partial^2 u_{\text{ex}}}{\partial x_i^2} = \sum_{i=1}^{d} (-\pi^2 u_{\text{ex}}) = -d\pi^2 u_{\text{ex}}. \tag{3}$$

Assuming $\boldsymbol{\beta}$ is divergence-free ($\nabla \cdot \boldsymbol{\beta} = 0$, which holds for the rotational field defined above), the advection term simplifies to $\boldsymbol{\beta} \cdot \nabla u_{\text{ex}}$. The source term f is therefore implemented as:

$$f(\mathbf{x}) = \mu d\pi^2 u_{\text{ex}}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x}) \cdot \nabla u_{\text{ex}}(\mathbf{x}) + \gamma u_{\text{ex}}(\mathbf{x}). \tag{4}$$

The problem domain boundary $\partial\Omega$ is split into Dirichlet (Γ_D) and Neumann (Γ_N) portions to test mixed boundary conditions. Figure 1 illustrates the boundary conditions on a 2D unit square domain.

Neumann Boundary (Γ_N) We apply a Neumann condition on the "Right" face of the hypercube, defined as the plane $x_1 = 1$. The outward unit normal is $\mathbf{n} = (1, 0, \dots)^T$. The required flux h is derived from the exact solution:

$$h(\mathbf{x}) = \nabla u_{\text{ex}} \cdot \mathbf{n}\Big|_{x_1=1} = \frac{\partial u_{\text{ex}}}{\partial x_1}\Big|_{x_1=1}. \tag{5}$$

Computing the partial derivative:

$$\frac{\partial u_{\text{ex}}}{\partial x_1} = \pi \cos(\pi x_1) \prod_{j=2}^{d} \sin(\pi x_j). \tag{6}$$

Evaluated at $x_1 = 1$, where $\cos(\pi) = -1$, the Neumann data imposed is:

$$h(\mathbf{x}) = -\pi \prod_{j=2}^{d} \sin(\pi x_j). \tag{7}$$

Dirichlet Boundary (Γ_D) On all other boundaries ($\partial\Omega \backslash \Gamma_N$), we enforce a homogeneous Dirichlet condition:

$$u = 0 \quad \text{on } \Gamma_D. \tag{8}$$

This is consistent with the exact solution, as $\sin(\pi x_i) = 0$ when $x_i \in \{0, 1\}$.

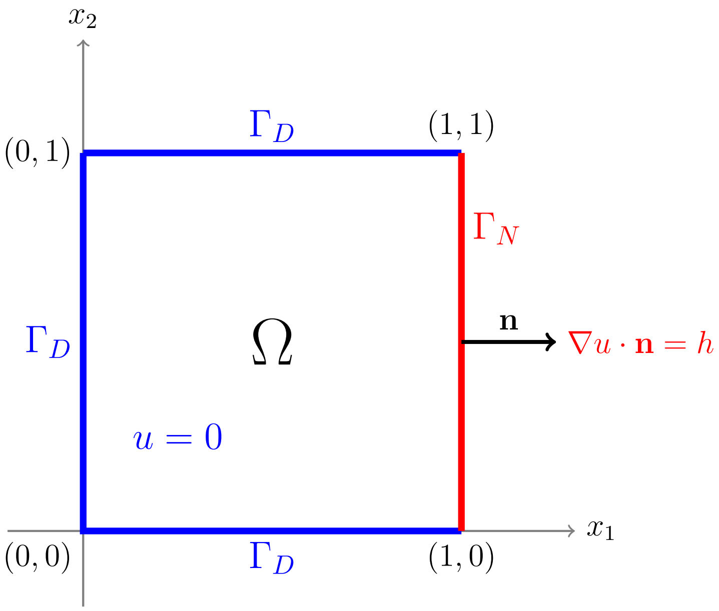

Figure 1: 2D Domain with Mixed Boundary Conditions

2 Finite Element Discretization

To solve the weak formulation numerically, we employ the Finite Element Method (FEM). This involves approximating the infinite-dimensional function spaces V_g and V_0 with finite-dimensional subspaces defined on a computational mesh.

2.1 Triangulation and Finite Element Space

We consider a triangulation $\mathcal{T}_h = \{K\}$ of the domain Ω, consisting of non-overlapping hexahedral (or quadrilateral in 2D) cells K such that $\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} \overline{K}$. The parameter h denotes the characteristic mesh size, $h = \max_{K \in \mathcal{T}_h} \operatorname{diam}(K)$.

We introduce the finite-dimensional space $V_h^k \subset H^1(\Omega)$ consisting of continuous piecewise polynomial functions of degree k. In the context of the `deal.II` library, we utilize Lagrangian finite elements (tensor product polynomials of degree k, denoted as Q_k). The discrete trial and test spaces are defined as:

$$V_{h,g} = \{u_h \in V_h^k : u_h|_{\Gamma_D} = I_h(g)\}, \tag{9}$$

$$V_{h,0} = \{v_h \in V_h^k : v_h|_{\Gamma_D} = 0\}, \tag{10}$$

where $I_h(g)$ is the nodal interpolation of the Dirichlet boundary data onto the mesh nodes on Γ_D.

2.2 Galerkin Approximation

The discrete problem is obtained by restricting the weak form to these subspaces. We seek $u_h \in V_{h,g}$ such that:

$$a(u_h, v_h) = L(v_h) \quad \forall v_h \in V_{h,0}. \tag{11}$$

We expand the approximate solution u_h in terms of the standard nodal basis functions $\{\varphi_j\}_{j=1}^{N_{dof}}$. Let u_h be decomposed into a part satisfying the homogeneous boundary conditions and a lifting of the Dirichlet data:

$$u_h(\mathbf{x}) = \sum_{j \in \mathcal{I}_{free}} U_j \varphi_j(\mathbf{x}) + \sum_{j \in \mathcal{I}_{dir}} g_j \varphi_j(\mathbf{x}), \tag{12}$$

where U_j are the unknown coefficients (degrees of freedom), $\mathcal{I}_{free}$ is the set of indices for nodes not on Γ_D, and $\mathcal{I}_{dir}$ contains indices for nodes on the Dirichlet boundary with known values g_j.

2.3 Algebraic System

Substituting the basis expansion into Eq. (11) and testing with each basis function φ_i (for $i \in \mathcal{I}_{free}$), we obtain the linear system of equations:

$$\mathbf{A}\mathbf{U} = \mathbf{F}, \tag{13}$$

where $\mathbf{U}$ is the vector of unknown coefficients. The entries of the global stiffness matrix $\mathbf{A}$ and the right-hand side vector $\mathbf{F}$ are computed by assembling contributions from each cell $K \in \mathcal{T}_h$.

The matrix entries A_{ij} correspond to the bilinear form $a(\varphi_j, \varphi_i)$:

$$A_{ij} = \int\limits_{\Omega} (\mu \nabla \varphi_j \cdot \nabla \varphi_i + (\boldsymbol{\beta} \cdot \nabla \varphi_j)\varphi_i + \gamma \varphi_j \varphi_i)\ dx. \tag{14}$$

Using numerical quadrature, the integral over Ω is computed as the sum of integrals over cells K. For a specific cell K, the local matrix contributions are:

$$A_{ij}^K = \sum_{q=1}^{N_q} (\mu \nabla \varphi_j(\mathbf{x}_q) \cdot \nabla \varphi_i(\mathbf{x}_q) + (\boldsymbol{\beta}(\mathbf{x}_q) \cdot \nabla \varphi_j(\mathbf{x}_q))\varphi_i(\mathbf{x}_q) + \gamma \varphi_j(\mathbf{x}_q)\varphi_i(\mathbf{x}_q))\, w_q |J_K(\mathbf{x}_q)|, \tag{15}$$

where $\{\mathbf{x}_q\}$ and $\{w_q\}$ are the quadrature points and weights defined on the reference element, mapped to physical space via the Jacobian determinant $|J_K|$.

The right-hand side vector $\mathbf{F}$ includes the source term, the Neumann boundary contributions, and the modifications due to the Dirichlet lifting:

$$F_i = \int\limits_{\Omega} f \varphi_i \, dx + \int\limits_{\Gamma_N} \mu h \varphi_i \, ds - \sum_{j \in \mathcal{I}_{dir}} g_j A_{ij}. \tag{16}$$

The Neumann term is only non-zero if the support of φ_i intersects with Γ_N.

2.4 Convergence and Error Estimation

The finite element solution u_h converges to the exact solution u as the mesh is refined ($h \to 0$). Under appropriate regularity assumptions on the solution and the domain, classical a priori error estimates provide bounds on the discretization error.

Error Norms We measure the approximation error in the standard Sobolev norms:

- **L^2-norm (energy norm for the solution):**

$$\|u - u_h\|_{L^2(\Omega)} = \left(\int\limits_{\Omega} |u - u_h|^2 \, dx \right)^{1/2}. \tag{17}$$

- **H^1-seminorm (energy norm for the gradient):**

$$|u - u_h|_{H^1(\Omega)} = \left(\int\limits_{\Omega} |\nabla(u - u_h)|^2 \, dx \right)^{1/2}. \tag{18}$$

A Priori Error Estimates Assuming the exact solution $u \in H^{k+1}(\Omega)$ and the bilinear form $a(\cdot, \cdot)$ is coercive and continuous, the following error estimates hold for Lagrangian finite elements of polynomial degree k:

- **H^1-error estimate:** By Céa's lemma and standard interpolation theory,

$$|u - u_h|_{H^1(\Omega)} \leq C h^k |u|_{H^{k+1}(\Omega)}, \tag{19}$$

where $C > 0$ is a constant independent of h.

- L^2**-error estimate:** Using the Aubin-Nitsche duality argument,

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^{k+1}|u|_{H^{k+1}(\Omega)}. \tag{20}$$

These estimates indicate that for smooth solutions, increasing the polynomial degree k or refining the mesh (decreasing h) improves accuracy. Specifically:

- The H^1-error converges at rate $\mathcal{O}(h^k)$.
- The L^2-error converges at rate $\mathcal{O}(h^{k+1})$.

3 Experiments

3.1 Convergence

To verify the theoretical convergence rates, we conduct a series of numerical experiments using the manufactured solution defined in Eq. (1). We define dimensions $d = 2$ and use polynomial degree $k = 2$ for the finite element space. The resulting convergence plots are shown in Figure 2.

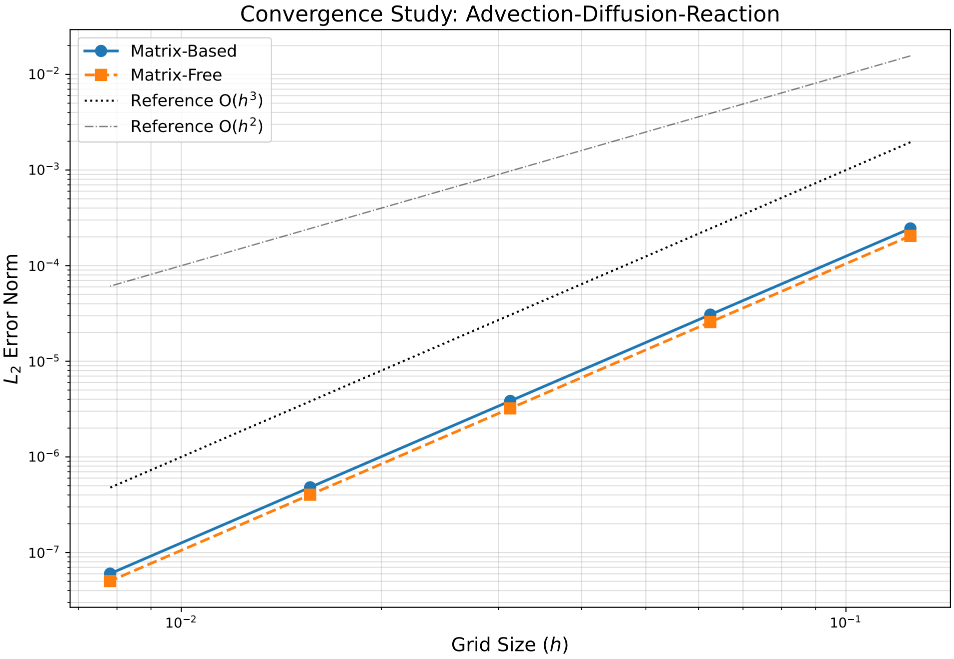

Figure 2: Convergence plot showing L^2 errors for Matrix-Based and Matrix-Free implementations.

The results confirm the expected convergence rates in section 2.4 for both implementations, demonstrating the correctness of the finite element discretization and the solver implementations.

3.2 Time Complexity

We compare the performance of Algebraic Multigrid (AMG) matrix-based solver and Geometric Multigrid (GMG) matrix-free solver for solving the linear system arising from the finite element discretization of the ADR equation. The experiments are conducted on a series of uniformly refined meshes in 2D, with polynomial degree $k = 2$. The timing results for both setup and assembly, as well as the solve phase, are summarized in Table 1.

3.3 Scalability

Table 1: Performance comparison of Multigrid Matrix-based and Matrix-free solvers.

	AMG Matrix-based		GMG Matrix-free	
n_{dofs}	Setup+Assem. (s)	Solve (s)	Setup+Assem. (s)	Solve (s)
81	0.0279	0.0121	0.0549	0.0091
289	0.0200	0.0039	0.0529	0.0097
1,089	0.0260	0.0080	0.0845	0.0134
4,225	0.0412	0.0213	0.1002	0.0220
16,641	0.1233	0.0718	0.1573	0.1218
66,049	0.3952	0.3021	0.3382	0.3880
263,169	1.5418	1.0441	0.8482	1.2239
1,050,625	6.1596	4.2531	3.1118	3.6356
4,198,401	25.3630	18.7690	12.0978	11.6747

4 Conclusion