Matrix-equation-based strategies for convection-diffusion equations

Davide Palitta and Valeria Simoncini

Dipartimento di Matematica, Università di Bologna
davide.palitta3@unibo.it

CIME-EMS Summer School 2015
Exploiting Hidden Structure in Matrix Computations. Algorithms and Applications
Cetraro, 22-26 June 2015

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The convection-diffusion equation

Convection-diffusion equation:

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = f, \qquad \text{in} \quad \Omega \subset \mathbb{R}^d$$

with $d = 2, 3$, $\epsilon > 0$ (*viscosity parameter*), $\mathbf{w} \in \mathbb{R}^d$ (*convection vector*)

The convection-diffusion equation

Convection-diffusion equation:

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = f, \qquad \text{in} \quad \Omega \subset \mathbb{R}^d$$

with $d = 2, 3$, $\epsilon > 0$ (*viscosity parameter*), $\mathbf{w} \in \mathbb{R}^d$ (*convection vector*)

- Dominant Convection $\rightarrow \|\mathbf{w}\| \gg \epsilon$
- Incompressible flow $\rightarrow \text{div}(\mathbf{w}) = 0$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The convection-diffusion equation

Convection-diffusion equation:

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = f, \qquad \text{in} \quad \Omega\subset\mathbb{R}^d$$

with $d = 2,3$, $\epsilon > 0$ (*viscosity parameter*), $\mathbf{w}\in\mathbb{R}^d$ (*convection vector*)

- Dominant Convection $\rightarrow \|\mathbf{w}\| \gg \epsilon$
- Incompressible flow $\rightarrow \text{div}(\mathbf{w}) = 0$

Further assumptions:

- $\mathbf{w}$ separable coefficients, e.g., in 2D

$$\mathbf{w} = (w_1, w_2) = (\phi_1(x)\psi_1(y), \phi_2(x)\psi_2(y))$$

- Ω is a rectangle (parallelepipedal) domain

Discretization phase

Discretizing by FEM or FD, we get the nonsymmetric linear system

$$A\mathbf{u} = \mathbf{f}, \qquad \text{where} \quad A \in \mathbb{R}^{N \times N}$$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Discretization phase

Discretizing by FEM or FD, we get the nonsymmetric linear system

$$A\mathbf{u} = \mathbf{f}, \qquad \text{where} \quad A \in \mathbb{R}^{N \times N}$$

REMARK: The discretization step is crucial to avoid *spurious oscillations* in the numerical solution: Refine the meshsize $h \Rightarrow$ huge increase of N

Different strategies:

- Artificial diffusion
- SUPG
- ...

A matrix-equation-based strategy

AIM: Preconditioning the very large and sparse nonsymmetric linear system

$$A\mathbf{u} = \mathbf{f}, \qquad \text{where} \quad A \in \mathbb{R}^{N \times N}$$

with a "simplified" matrix version of the discretized differential operator

A matrix oriented formulation

Poisson equation

$$-u_{xx} - u_{yy} = f, \quad \text{in } \Omega = (0,1)^2 \quad \text{(w/ zero Dirichlet b.c.)}$$

A matrix oriented formulation

Poisson equation

$$-u_{xx} - u_{yy} = f, \quad \text{in } \Omega = (0,1)^2 \quad \text{(w/ zero Dirichlet b.c.)}$$

Discretize Ω with a uniform mesh Ω_h with nodes (x_i, y_j), $i, j = 1, \dots, n-1$.
Define $T, F \in \mathbb{R}^{(n-1)\times(n-1)}$ such that

$$T = \frac{1}{h^2}\text{tridiag}(-1, \underline{2}, -1) \quad \text{and} \quad F_{i,j} = f(x_i, y_j)$$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A matrix oriented formulation

Poisson equation

$$-u_{xx} - u_{yy} = f, \quad \text{in } \Omega = (0,1)^2 \quad \text{(w/ zero Dirichlet b.c.)}$$

Discretize Ω with a uniform mesh Ω_h with nodes (x_i, y_j), $i, j = 1, \ldots, n-1$.
Define $T, F \in \mathbb{R}^{(n-1)\times(n-1)}$ such that

$$T = \frac{1}{h^2}\text{tridiag}(-1, \underline{2}, -1) \quad \text{and} \quad F_{i,j} = f(x_i, y_j)$$

Centered FD leads to the symmetric linear system

$$A\mathbf{u} = \mathbf{f}, \qquad \mathbf{f} = \text{vec}(F)$$

where $A = T \otimes I_{n-1} + I_{n-1} \otimes T \in \mathbb{R}^{(n-1)^2\times(n-1)^2}$

A matrix oriented formulation

$U_{i,j} \approx u(x_i, y_j)$ at interior node (x_i, y_j)
For each $i, j = 1, \ldots, n-1$,

$$u_{xx}(x_i, y_j) \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} = \frac{1}{h^2}[1, -2, 1] \begin{bmatrix} U_{i-1,j} \\ U_{i,j} \\ U_{i+1,j} \end{bmatrix}$$

and

$$u_{yy}(x_i, y_j) \approx \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h^2} = \frac{1}{h^2}[U_{i,j-1},\ U_{i,j},\ U_{i,j+1}] \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

A matrix oriented formulation

$U_{i,j} \approx u(x_i, y_j)$ at interior node (x_i, y_j)
For each $i, j = 1, \ldots, n-1$,

$$u_{xx}(x_i, y_j) \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h^2} = \frac{1}{h^2}[1, -2, 1] \begin{bmatrix} U_{i-1,j} \\ U_{i,j} \\ U_{i+1,j} \end{bmatrix}$$

and

$$u_{yy}(x_i, y_j) \approx \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h^2} = \frac{1}{h^2}[U_{i,j-1},\ U_{i,j},\ U_{i,j+1}] \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

$$TU + UT = F$$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

A matrix oriented formulation, $-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = f$

Proposition. (2D case)

Assume $\mathbf{w} = (w_1, w_2) = (\phi_1(x)\psi_1(y), \phi_2(x)\psi_2(y))$.
Let $(x_i, y_j) \in \Omega_h$, $i, j = 1, \ldots, n-1$, and set, for $k = 1, 2$,

- $\Phi_k = \text{diag}(\phi_k(x_1), \ldots, \phi_k(x_{n-1}))$
- $\Psi_k = \text{diag}(\psi_k(y_1), \ldots, \psi_k(y_{n-1}))$

Then, the centered FD discretization of the continuous operator

$$\mathcal{L} : u \mapsto -\epsilon\Delta u + \phi_1(x)\psi_1(y)u_x + \phi_2(x)\psi_2(y)u_y$$

leads to the following operator:

$$\mathcal{L}_h : \mathbf{U} \mapsto \epsilon T\mathbf{U} + \epsilon\mathbf{U}T + (\Phi_1 B)\mathbf{U}\Psi_1 + \Phi_2\mathbf{U}(B^T\Psi_2)$$

where

$$B = \frac{1}{2h}\text{tridiag}(-1, \underline{0}, 1) \in \mathbb{R}^{(n-1)\times(n-1)}$$

The "easy" case: $-\epsilon\Delta u + \phi_1(x)u_x + \psi_2(y)u_y = f$

$\mathbf{w} = (\phi_1(x), \psi_2(y))$ (common in academic examples!)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The "easy" case: $-\epsilon\Delta u + \phi_1(x)u_x + \psi_2(y)u_y = f$

$$\mathbf{w} = (\phi_1(x), \psi_2(y)) \text{ (common in academic examples!)}$$
$$\Downarrow$$
$$\Psi_1 = \Phi_2 = I$$

The "easy" case: $-\epsilon\Delta u + \phi_1(x)u_x + \psi_2(y)u_y = f$

$\mathbf{w} = (\phi_1(x), \psi_2(y))$ (common in academic examples!)

$$\Downarrow$$

$$\Psi_1 = \Phi_2 = I$$

$$\Downarrow$$

$$\epsilon T\mathbf{U} + \epsilon\mathbf{U}T + (\Phi_1 B)\mathbf{U} + \mathbf{U}(B^T\Psi_2) = F$$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

The "easy" case: $-\epsilon\Delta u + \phi_1(x)u_x + \psi_2(y)u_y = f$

$$\mathbf{w} = (\phi_1(x), \psi_2(y)) \text{ (common in academic examples!)}$$
$$\Downarrow$$
$$\Psi_1 = \Phi_2 = I$$
$$\Downarrow$$
$$\epsilon T\mathbf{U} + \epsilon\mathbf{U}T + (\Phi_1 B)\mathbf{U} + \mathbf{U}(B^T\Psi_2) = F$$
$$\Downarrow$$
$$(\epsilon T + \Phi_1 B)\mathbf{U} + \mathbf{U}(\epsilon T + B^T\Psi_2) = F$$

This is a Sylvester equation that can be explicitly and efficiently solved!

The general case: preconditioning strategy

$$\mathbf{w} = (w_1, w_2) = (\phi_1(x)\psi_1(y), \phi_2(x)\psi_2(y))$$

$$\mathcal{L}_h : \mathbf{U} \mapsto \epsilon T\mathbf{U} + \epsilon \mathbf{U} T + (\Phi_1 B)\mathbf{U}\Psi_1 + \Phi_2 \mathbf{U}(B^T \Psi_2)$$

The general case: preconditioning strategy

$$\mathbf{w} = (w_1, w_2) = (\phi_1(x)\psi_1(y), \phi_2(x)\psi_2(y))$$

$$\mathcal{L}_h : \mathbf{U} \mapsto \epsilon T\mathbf{U} + \epsilon \mathbf{U} T + (\Phi_1 B)\mathbf{U}\Psi_1 + \Phi_2\mathbf{U}(B^T\Psi_2)$$

Approximating

$$\Psi_1 \approx \bar{\psi}_1 I, \qquad \Phi_2 \approx \bar{\phi}_2 I, \quad \text{where, e.g., } \bar{\psi}_1, \bar{\phi}_2 \text{ mean values}$$

The general case: preconditioning strategy

$$\mathbf{w} = (w_1, w_2) = (\phi_1(x)\psi_1(y), \phi_2(x)\psi_2(y))$$

$$\mathcal{L}_h : \mathbf{U} \mapsto \epsilon T\mathbf{U} + \epsilon \mathbf{U} T + (\Phi_1 B)\mathbf{U}\Psi_1 + \Phi_2 \mathbf{U}(B^T \Psi_2)$$

Approximating

$$\Psi_1 \approx \bar{\psi}_1 I, \qquad \Phi_2 \approx \bar{\phi}_2 I, \quad \text{where, e.g., } \bar{\psi}_1, \bar{\phi}_2 \text{ mean values}$$

$$\mathcal{P} : \mathbf{Y} \mapsto (\epsilon T + \bar{\psi}_1 \Phi_1 B)\mathbf{Y} + \mathbf{Y}(\epsilon T + \bar{\phi}_2 B^T \Psi_2)$$

Implementation details

Nonsymmetric linear system:

$$A\mathbf{u} = \mathbf{f}, \qquad A \in \mathbb{R}^{(n-1)^2 \times (n-1)^2}$$

Solver: GMRES with right preconditioning.

Implementation details

Nonsymmetric linear system:

$$A\mathbf{u} = \mathbf{f}, \qquad A \in \mathbb{R}^{(n-1)^2 \times (n-1)^2}$$

Solver: GMRES with right preconditioning.
At each iteration k:

$$\widetilde{\mathbf{v}}_k = \mathcal{P}^{-1}\mathbf{v}_k \in \mathbb{R}^{(n-1)^2}$$

Implementation details

Nonsymmetric linear system:

$$A\mathbf{u} = \mathbf{f}, \qquad A \in \mathbb{R}^{(n-1)^2 \times (n-1)^2}$$

Solver: GMRES with right preconditioning.
At each iteration k:

$$\widetilde{\mathbf{v}}_k = \mathcal{P}^{-1}\mathbf{v}_k \in \mathbb{R}^{(n-1)^2}$$

How to compute it?

Implementation details

Nonsymmetric linear system:

$$A\mathbf{u} = \mathbf{f}, \qquad A \in \mathbb{R}^{(n-1)^2 \times (n-1)^2}$$

Solver: GMRES with right preconditioning.
At each iteration k:

$$\widetilde{\mathbf{v}}_k = \mathcal{P}^{-1}\mathbf{v}_k \in \mathbb{R}^{(n-1)^2}$$

How to compute it?

1. Compute $G_k \in \mathbb{R}^{(n-1)\times(n-1)}$ s.t. $\mathbf{v}_k = \text{vec}(G_k)$
2. Solve $(\epsilon T + \bar{\psi}_1 \Phi_1 B)\mathbf{Y} + \mathbf{Y}(\epsilon T + B^T \bar{\phi}_2 \Psi_2) = G_k$
3. Compute $\widetilde{\mathbf{v}}_k = \text{vec}(\mathbf{Y})$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Implementation details

Sylvester equation:

$$(\epsilon T + \bar{\psi}_1 \Phi_1 B)\mathbf{Y} + \mathbf{Y}(\epsilon T + B^T \bar{\phi}_2 \Psi_2) = G_k$$

Solver: KPIK **[V. Simoncini, 2007], [T. Breiten, V. Simoncini, M. Stoll, 2014].**

Implementation details

Sylvester equation:

$$(\epsilon T + \bar{\psi}_1 \Phi_1 B)\mathbf{Y} + \mathbf{Y}(\epsilon T + B^T \bar{\phi}_2 \Psi_2) = G_k$$

Solver: KPIK **[V. Simoncini, 2007], [T. Breiten, V. Simoncini, M. Stoll, 2014].**

Observations:

- Inner: Inexact method ⇒ Outer: FGMRES
- rhs must be low rank: Truncated SVD of G_k

Implementation details

Sylvester equation:

$$(\epsilon T + \bar{\psi}_1 \Phi_1 B)\mathbf{Y} + \mathbf{Y}(\epsilon T + B^T \bar{\phi}_2 \Psi_2) = G_k$$

Solver: KPIK **[V. Simoncini, 2007], [T. Breiten, V. Simoncini, M. Stoll, 2014].**

Observations:

- Inner: Inexact method $\Rightarrow$ Outer: FGMRES
- rhs must be low rank: Truncated SVD of G_k

Three parameters:

1. inner tolerance: `tol_inner`$= 10^{-4}$
2. truncation tolerance: `tol_truncation`$= 10^{-2}$
3. maximum rank allowed: `r_max`$= 10$

3D case

To fix ideas: $\Omega = (0,1)^3$.

$U_{i,j}^{(k)} \approx u(x_i, y_j, z_k)$, and define the tall matrix*

$$\mathfrak{U} := \begin{bmatrix} U^{(1)} \\ \vdots \\ U^{(n-1)} \end{bmatrix} = \sum_{k=1}^{n-1} (e_k \otimes U^{(k)}) \in \mathbb{R}^{(n-1)^2 \times (n-1)}$$

where $e_j = I(:,j)$.

*Other orderings are possible.

3D case: $\mathcal{L} : u \mapsto -\epsilon\Delta u + w_1 u_x + w_2 u_y + w_3 u_z$

Proposition. (3D case)

Assume $\mathbf{w} = (w_1, w_2, w_3)$ s.t.

- $w_1 = \phi_1(x)\psi_1(y)\upsilon_1(z)$
- $w_2 = \phi_2(x)\psi_2(y)\upsilon_2(z)$
- $w_3 = \phi_3(x)\psi_3(y)\upsilon_3(z)$

Let $(x_i, y_j, z_k) \in \Omega_h$, $i, j, k = 1, \dots, n-1$, and set, for $\ell = 1, 2, 3$,

- $\Phi_\ell = \text{diag}(\phi_\ell(x_1), \dots, \phi_\ell(x_{n-1}))$
- $\Psi_\ell = \text{diag}(\psi_\ell(y_1), \dots, \psi_\ell(y_{n-1}))$
- $\Upsilon_\ell = \text{diag}(\upsilon_\ell(z_1), \dots, \upsilon_\ell(z_{n-1}))$

Then, the centered FD discretization leads to the following operator:

$$\mathcal{L}_h : \mathfrak{U} \mapsto (I \otimes \epsilon T)\mathfrak{U} + \epsilon\mathfrak{U}T + (\epsilon T \otimes I)\mathfrak{U} + (\Upsilon_1 \otimes \Phi_1 B)\mathfrak{U}\Psi_1 + (\Upsilon_2 \otimes \Phi_2)\mathfrak{U}B^T\Psi_2 + [(\Upsilon_3 B) \otimes \Phi_3]\mathfrak{U}\Psi_3$$

Numerical experiments

Competitors on $A\mathbf{u} = \mathbf{f}$:

- GMRES+MI20[†]
- FGMRES+AGMG[‡]

REMARK: our preconditioner/solver is implemented in interpreted Matlab functions while both MI20 and AGMG are fortran90 compiled codes provided with mex files

[†] `control.one_pass_coarsen`$= 1$, [J. Boyle, M.D. Mihajlovič, J.A. Scott, 2010]

[‡] all default parameters, [Y. Notay, 2010]

Test 1. Explicit matrix solution (2D)

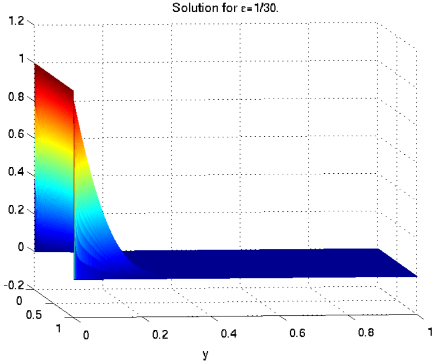
Consider

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = 0, \quad \text{in} \quad \Omega = (0,1)^2$$

$$\mathbf{w} = \left(1 + \frac{1}{4}(x+1)^2, 0\right)$$

Dirichlet b.c.:

$$\begin{cases} u(x,0) = 1 & x \in [0,1], \\ u(x,1) = 0 & x \in [0,1]. \end{cases}$$



Test 1. Explicit matrix solution (2D)

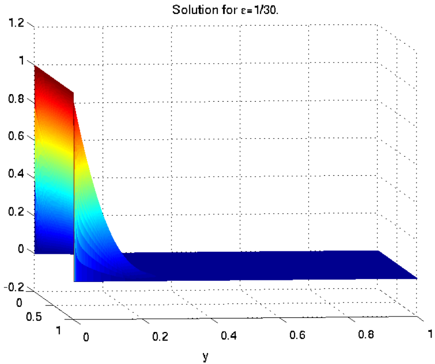
Consider

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = 0, \quad \text{in} \quad \Omega = (0,1)^2$$

$$\mathbf{w} = \left(1 + \frac{1}{4}(x+1)^2, 0\right)$$

Dirichlet b.c.:

$$\begin{cases} u(x,0) = 1 & x \in [0,1], \\ u(x,1) = 0 & x \in [0,1]. \end{cases}$$



$$(\epsilon T + \Phi_1 B)\mathbf{U} + \epsilon \mathbf{U} T = F$$

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Test 1. Explicit matrix solution (2D)

ϵ	n_x	FGMRES+AGMG time (# its)	GMRES+MI20 time (# its)	KPIK time (# its)
0.0333	129	**0.1649** (13)	0.2843 (7)	0.2131 (24)
0.0333	257	0.3874 (15)	0.5715 (8)	**0.2817** (32)
0.0333	1025	11.4918 (20)	8.4540 (8)	**1.5002** (44)
0.0333	1200	12.5441 (17)	8.7843 (7)	**1.9722** (46)
0.0167	129	0.2150 (14)	0.2750 (7)	**0.1638** (22)
0.0167	257	0.4356 (14)	0.6533 (9)	**0.3628** (32)
0.0167	513	2.0712 (15)	2.2171 (9)	**0.6324** (38)
0.0167	1025	11.5428 (18)	8.0454 (8)	**2.8454** (64)
0.0167	1200	13.2109 (18)	9.5501 (8)	**2.1961** (52)
0.0083	129	0.1501 (14)	0.2685 (10)	**0.1394** (22)
0.0083	257	0.3651 (15)	0.5871 (8)	**0.2885** (34)
0.0083	513	1.6615 (14)	2.1814 (10)	**0.5439** (42)
0.0083	1025	10.0859 (18)	10.6729 (11)	**2.7800** (66)
0.0083	1200	14.4866 (18)	11.0856 (9)	**2.8459** (58)

Table : Test 1 (2D). Performance achieved as the viscosity and mesh parameter vary. `tol_outer`$= 10^{-8}$.

Test 2[†] . Preconditioning strategy (2D)

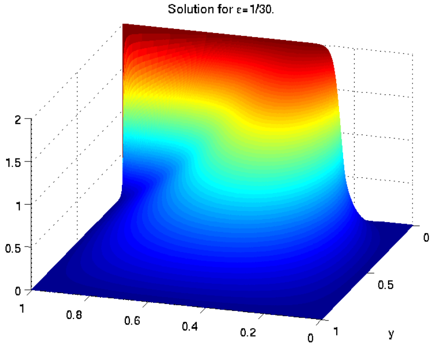
Consider

$$-\epsilon \Delta u + \mathbf{w} \cdot \nabla u = 0, \quad \text{in} \quad \Omega = (0,1)^2$$

$$\mathbf{w} = (y(1-(2x+1)^2), -2(2x+1)(1-y^2))$$

Zero Dirichlet b.c. except for the side $y = 0$:

$$\begin{cases} u(x,0) = 1 + \tanh[10 + 20(2x-1)], & 0 \le x \le 0.5 \\ u(x,0) = 2, & 0.5 < x \le 1 \end{cases}$$



[†] slightly modification of Example V in [H.C. Elman, A. Ramage, 2002].

Test 2. Preconditioning strategy (2D)

ϵ	n_x	FGMRES+AGMG time (# its)	GMRES+MI20 time (# its)	FGMRES+KPIK time (# its)
0.1000	128	**0.1081** (11)	0.1989 (4)	0.5743 (8)
0.1000	256	**0.3335** (14)	0.3812 (4)	0.5351 (7)
0.1000	512	1.0773 (11)	1.1731 (4)	**1.0543** (7)
0.1000	1024	9.2493 (17)	4.3287 (4)	**2.6372** (6)
0.1000	2048	52.0430 (15)	19.6757 (4)	**16.7394** (5)
0.0500	128	**0.0936** (9)	0.2269 (4)	0.5168 (10)
0.0500	256	**0.2897** (12)	0.3862 (4)	0.6455 (9)
0.0500	512	1.2603 (11)	1.2380 (4)	**1.1769** (8)
0.0500	1024	10.3623 (18)	4.3345 (4)	**3.0812** (7)
0.0500	2048	60.5041 (17)	20.4056 (4)	**14.9237** (6)
0.0333	128	**0.0882** (8)	0.2368 (4)	0.6428 (11)
0.0333	256	**0.2181** (9)	0.4218 (4)	0.8149 (11)
0.0333	512	1.5849 (14)	**1.1977** (4)	1.3786 (9)
0.0333	1024	5.4624 (12)	4.4130 (4)	**3.7214** (8)
0.0333	2048	120.9686 (23)	20.1120 (4)	**17.9188** (7)

Table : Test 2 (2D). Performance achieved as the viscosity and mesh parameter vary. tol_outer$=10^{-6}$

Test 3. Explicit matrix solution (3D)

Consider

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = 1, \quad \text{in} \quad \Omega = (0,1)^3$$

$\mathbf{w} = (x \sin x, y \cos y, e^{z^2-1})$

Zero Dirichlet b.c.

Test 3. Explicit matrix solution (3D)

Consider

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = 1, \quad \text{in} \quad \Omega = (0,1)^3$$

$\mathbf{w} = (x\sin x, y\cos y, e^{z^2-1})$

Zero Dirichlet b.c.

$$\left[I\otimes(\epsilon T + \Phi_1 B) + (\epsilon T + \Psi_2 B)^T \otimes I\right]\mathcal{U} + \mathcal{U}\,(\epsilon T + B\Upsilon_3) = \mathbf{1}\mathbf{1}^T$$

where $\mathbf{1}$ is the vector of all ones

REMARK: low rank rhs in the matrix equation is crucial for explicit solution!

Test 3. Explicit matrix solution (3D)

ϵ	n_x	FGMRES+AGMG time (# its)	GMRES+MI20 time (# its)	KPIK time (# its)
0.0050	60	1.1250 (14)	1.8022 (7)	**0.1734** (18)
0.0050	70	2.0385 (14)	3.4253 (7)	**0.2326** (20)
0.0050	80	3.4803 (14)	4.4297 (7)	**0.3583** (20)
0.0050	90	5.7324 (15)	6.8705 (7)	**0.4999** (22)
0.0050	100	8.0207 (15)	9.7207 (7)	**0.5677** (22)
0.0010	60	1.3011 (14)	1.7854 (7)	**0.2386** (18)
0.0010	70	1.9509 (14)	2.7829 (7)	**0.2346** (20)
0.0010	80	3.5291 (14)	4.6576 (7)	**0.4096** (20)
0.0010	90	5.1344 (14)	6.8176 (7)	**0.4253** (22)
0.0010	100	7.6815 (14)	9.4935 (7)	**0.5446** (22)
0.0005	60	1.2560 (14)	1.7341 (6)	**0.2314** (18)
0.0005	70	2.2242 (14)	2.9667 (7)	**0.2301** (20)
0.0005	80	3.4558 (14)	4.5964 (7)	**0.3472** (22)
0.0005	90	4.8076 (14)	6.4841 (7)	**0.4257** (22)
0.0005	100	7.3914 (14)	9.6274 (7)	**0.5927** (24)

Table : Test 3 (3D). Performance achieved as the viscosity and mesh parameter vary. tol_outer$=10^{-9}$

Test 4. Preconditioning strategy (3D)

Consider

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = 1, \quad \text{in} \quad \Omega = (0,1)^3$$

$\mathbf{w} = (yz(1-x^2), 0, e^z)$

Zero Dirichlet b.c.

‡Other variable aggregations are possible.

Test 4. Preconditioning strategy (3D)

Consider

$$-\epsilon\Delta u + \mathbf{w}\cdot\nabla u = 1, \quad \text{in} \quad \Omega = (0,1)^3$$

$\mathbf{w} = (yz(1-x^2), 0, e^z)$

Zero Dirichlet b.c.

The discretization yields[‡]

$$\epsilon T\mathfrak{U} + \mathfrak{U}(I\otimes\epsilon T + \epsilon T\otimes I) + \Upsilon_3 B\mathfrak{U} + \Upsilon_1\mathfrak{U}(\Psi_1\otimes\Phi_1 B) = \mathbf{1}\mathbf{1}^T$$

where $\mathbf{1}$ is the vector of all ones

[‡] Other variable aggregations are possible.

Test 4. Preconditioning strategy (3D)

Consider

$$-\epsilon \Delta u + \mathbf{w} \cdot \nabla u = 1, \quad \text{in} \quad \Omega = (0,1)^3$$

$\mathbf{w} = (yz(1-x^2), 0, e^z)$

Zero Dirichlet b.c.

The discretization yields[‡]

$$\epsilon T \mathfrak{U} + \mathfrak{U}(I \otimes \epsilon T + \epsilon T \otimes I) + \Upsilon_3 B \mathfrak{U} + \Upsilon_1 \mathfrak{U}(\Psi_1 \otimes \Phi_1 B) = \mathbf{1}\mathbf{1}^T$$

where $\mathbf{1}$ is the vector of all ones

$\Upsilon_1 \approx \bar{\upsilon}_1 I$ and obtain the preconditioning operator

$$\mathcal{P} : \mathbf{Y} \mapsto (\epsilon T + \Upsilon_3 B)\mathbf{Y} + \mathbf{Y}(I \otimes \epsilon T + \epsilon T \otimes I + \Psi_1 \otimes \bar{\upsilon}_1 \Phi_1 B)$$

[‡]Other variable aggregations are possible.

Test 4. Preconditioning strategy (3D)

ϵ	n_x	FGMRES+AGMG	GMRES+MI20	FGMRES+KPIK
0.5000	60	1.2095 (15)	1.8236 (7)	**1.2027** (6)
0.5000	70	2.1041 (15)	3.1649 (7)	**1.6585** (6)
0.5000	80	3.7370 (16)	4.9765 (7)	**2.4943** (6)
0.5000	90	7.5874 (16)	9.2040 (8)	**3.2513** (6)
0.5000	100	7.7626 (16)	11.9912 (8)	**4.7548** (6)
0.1000	60	2.1310 (18)	- (-)	**1.5299** (8)
0.1000	70	2.8043 (18)	- (-)	**1.8926** (8)
0.1000	80	5.1219 (19)	- (-)	**3.2928** (9)
0.1000	90	7.3179 (19)	- (-)	**4.6429** (9)
0.1000	100	9.5759 (19)	- (-)	**6.5590** (9)
0.0500	60	1.4318 (18)	- (-)	**1.7296** (10)
0.0500	70	2.8427 (19)	- (-)	**2.5215** (10)
0.0500	80	4.9616 (20)	- (-)	**3.6615** (10)
0.0500	90	7.1038 (20)	- (-)	**5.0098** (10)
0.0500	100	10.7181 (21)	- (-)	**6.8661** (10)

Table : Test 4 (3D). Performance achieved as the viscosity and mesh parameter vary. "-" stands for excessive time in building the preconditioner. `tol_outer`$= 10^{-9}$.

Conclusions and outlook

- Preliminary numerical experiments show that the new approach performs comparably well with respect to state-of-the-art approaches.
- Generalize the approach to more general settings overcoming the limitation to the use of uniform mesh on rectangle (parallelepipedal) domains.
- Modify the approach to handle different discretization strategies, e.g., SUPG (early attempts in [D.P., 2014, Master's thesis]).

Reference:

Matrix-equation-based strategies for convection-diffusion equations

D. Palitta and V. Simoncini

ArXiv: 1501.02920

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA