Comparison of iterative methods for discretized nonsymmetric elliptic problems

Student: Lados Bálint István **Supervisor:** Karátson János

Introduction

Elliptic partial differential equations can be solved numerically with finite element or finite difference discretization. These methods involve introducing a set of discrete grid points in the domain of the studied boundary value problem, where the exact solution is approximated. This process results in a system of linear equations that can be solved by an iterative method.

In the last semester, my work focused on the finite difference method. Various nonsymmetric elliptic problems have been solved with two commonly used iterative methods: the preconditioned CGN and GCR algorithms. The basis of comparison was the size of the norm of the vector field in the differential equation. I observed that the CGN method could solve the resulting problems with less iterative steps when the norm of the vector field was small, while the GCR method performed significantly better when the norm was larger.

The aim of this project is to compare the performance of the preconditioned CGN and GCR algorithms when solving convection-dominated elliptic problems, for which the discretization is carried out with the streamline diffusion finite element method. I examined how the results depend on the coefficients of the PDE, and compared it with my previous work.

Convection-dominated elliptic problems

Let us consider the following boundary value problem:

$$\begin{cases} -\varepsilon\Delta u + \mathbf{w}\cdot\nabla u = f \\ u|_{\partial\Omega} = 0 \end{cases}$$

where $\Omega = [0,1]^2$, $\varepsilon > 0$ is a constant, $\mathbf{w} \in C^1(\Omega, \mathbb{R}^2)$ and $\operatorname{div}(\mathbf{w}) = 0$. These conditions guarantee that the PDE has a unique weak solution for any function $f \in L^2(\Omega)$. The second-order term $-\varepsilon\Delta u$ models diffusion, while the nonsymmetric term $\mathbf{w}\cdot\nabla u$ models convection. The constant ε is typically chosen to be close to zero, which makes the convection term more dominant.

In this particular problem, the standard finite element method (FEM) does not perform well. It can be improved by modifying the bilinear form in a special way and splitting it up elementwise. This is called streamline diffusion finite element method (SDFEM).

First of all, let us see how the standard FEM would look like in the case of convection-dominated elliptic problems. We are going to use uniformly spaced grid points of length h and Courant elements. The construction starts with the weak form of the problem:

$$\int_\Omega (\varepsilon\nabla u_h \cdot \nabla v_h + (\mathbf{w}\cdot\nabla u_h)v_h) = \int_\Omega f v_h, \quad \forall v_h \in V_h,$$

where the finite dimensional subspace V_h is given by the Courant elements (piecewise linear functions on a uniform triangular mesh).

One of the major problems with this approach is that the bilinear form

$$a(u,v) := \int_\Omega (\varepsilon \nabla u \cdot \nabla v + (\mathbf{w} \cdot \nabla u)v)$$

is coercive with the lower bound $\varepsilon \approx 0$, and by this, the ratio of the upper and lower bounds $\frac{M}{\varepsilon}$ in Céa's lemma becomes very large. However, if we choose the test functions in the form $w_h := v_h + \delta \mathbf{w} \cdot \nabla v_h$, where $v_h \in V_h$ and $\delta > 0$ is constant, and we use the Petrov–Galerkin weak form of the problem, then we obtain the following bilinear form, provided that $\mathbf{w}$ is piecewise constant:

$$a_{SD}(u,v) := \int_\Omega (\varepsilon \nabla u \cdot \nabla v + (\mathbf{w} \cdot \nabla u)v + \delta (\mathbf{w} \cdot \nabla u)(\mathbf{w} \cdot \nabla v))$$

Therefore, the discretized problem has the form

$$a_{SD}(u_h, v_h) = \int_\Omega f(v_h + \delta \mathbf{w} \cdot \nabla v_h) =: l(v_h), \quad \forall v_h \in V_h.$$

By this extension of the bilinear form, the lower estimate in coercivity becomes independent of ε, which stabilizes the convergence of the finite element method.

Implementation of the SDFEM

The implementation was based on the work in [3], but it needed to be extended because it worked originally only for constant vector fields that highly limited its applicability.

We define a basis on the inner grid points of the triangulation with the usual tent functions. The resulting basis functions $\{\phi_j\}_{j=1}^N$ are used to determine the system of linear equations whose solution will yield the coefficients for the linear combination of the basis functions producing the numerical solution. The system has the form $F\mathbf{c} = \mathbf{f}$, where $[F]_{i,j} = a_{SD}(\phi_j, \phi_i)$ and $\mathbf{f}_i = l(\phi_i)$. Vector $\mathbf{f}$ is approximated with the one-point Gauss quadrature:

$$\mathbf{f}_i = \int_\Omega f(\phi_i + \delta \mathbf{w} \cdot \nabla \phi_i) = \int_\Omega f\phi_i + \delta \int_\Omega f \mathbf{w} \cdot \nabla \phi_i \approx f(x_i, y_i) \int_\Omega \phi_i = f(x_i, y_i) \cdot h^2$$

Here the second term vanishes with the one-point approximation because $\int_\Omega \partial_x \phi_i = \int_\Omega \partial_y \phi_i = 0$ in our construction of the uniform triangular mesh. The elements of matrix F can be calculated according to [3] when $\mathbf{w} \equiv (w_x, w_y)$ is constant. When $\mathbf{w}$ is not constant, we are still able to reduce the problem to the constant case with the one-point quadrature:

$$\int_\Omega (\mathbf{w} \cdot \nabla \phi_j)\phi_i = \int_\Omega (w_x \partial_x \phi_j)\phi_i + \int_\Omega (w_y \partial_y \phi_j)\phi_i \approx$$

$$w_x(x_i, y_j) \int_\Omega ((1,0) \cdot \nabla \phi_j)\phi_i + w_y(x_i, y_j) \int_\Omega ((0,1) \cdot \nabla \phi_j)\phi_i$$

$$\delta \int_\Omega (\mathbf{w} \cdot \nabla \phi_j)(\mathbf{w} \cdot \nabla \phi_i) = \delta \int_\Omega w_x^2 \partial_x \phi_j \partial_x \phi_i + \delta \int_\Omega w_y^2 \partial_y \phi_j \partial_y \phi_i + \delta \int_\Omega w_x w_y (\partial_x \phi_j \partial_y \phi_i + \partial_y \phi_j \partial_x \phi_i) \approx$$

$$\delta w_x^2(x_i, y_j) \int_\Omega ((1,0) \cdot \nabla \phi_j)((1,0) \cdot \nabla \phi_i) + \delta w_y^2(x_i, y_j) \int_\Omega ((0,1) \cdot \nabla \phi_j)((0,1) \cdot \nabla \phi_i) +$$

$$\delta w_x(x_i, y_j) w_y(x_i, y_j) \int_\Omega (((1,1) \cdot \nabla \phi_j)((1,1) \cdot \nabla \phi_i) - ((1,0) \cdot \nabla \phi_j)((1,0) \cdot \nabla \phi_i) - ((0,1) \cdot \nabla \phi_j)((0,1) \cdot \nabla \phi_i))$$

A further generalization of the problem is when the differential equation is in the form of

$$-\varepsilon \cdot \operatorname{div}(p\nabla u) + \mathbf{w} \cdot \nabla u = f,$$

where $p \in L^\infty(\Omega)$ and $p(x) \geq m > 0$ (a.e. $x \in \Omega$). The bilinear form corresponding to this new problem differs only in its first term from the previous one:

$$\int_\Omega -\varepsilon \cdot \operatorname{div}(p\nabla\phi_j)\phi_i = \int_\Omega \varepsilon p \nabla\phi_j \cdot \nabla\phi_i \approx p(x_i, y_j) \int_\Omega \varepsilon \nabla\phi_j \cdot \nabla\phi_i.$$

I implemented in Matlab the finite element method for this general version of the convection-dominated elliptic problems according to the previous calculations. Parameter δ in the method usually has a magnitude of $\mathrm{O}(h)$. In my construction it is always $\delta := h$ unless otherwise specified. The obtained system of linear equations can be solved with the preconditioned CGN and GCR methods that I have already implemented in Matlab in the previous semester. These iterative algorithms run until the norm of the residual error vector r_n gets below 10^{-5}. The preconditioner matrix was chosen to be the symmetric part of the matrix, i.e. $S := \frac{F+F^T}{2}$.

Convergence of the SDFEM

I wanted to verify that the implemented method approximates the exact solution with an error of $\mathrm{O}(h)$. For this numerical experiment, let us consider the following functions and coefficients:

$$\varepsilon := 10^{-2}; \qquad p(x,y) := 1 + \frac{x^2+y^2}{2}; \qquad w(x,y) := (-y - \frac{1}{2},\, x - \frac{1}{2});$$

$$f(x,y) := -\varepsilon(p(x,y)(-2x(1-x) - 2y(1-y)) + x(1-2x)y(1-y) + y(1-2y)x(1-x)) + \\ + w_x(x,y)(1-2x)y(1-y) + w_y(x,y)(1-2y)x(1-x).$$

Function f was chosen in a way that the exact solution is $u(x,y) = x(1-x)y(1-y)$. It is clearly visible from table 1 that doubling the grid density roughly halves the error, indeed.

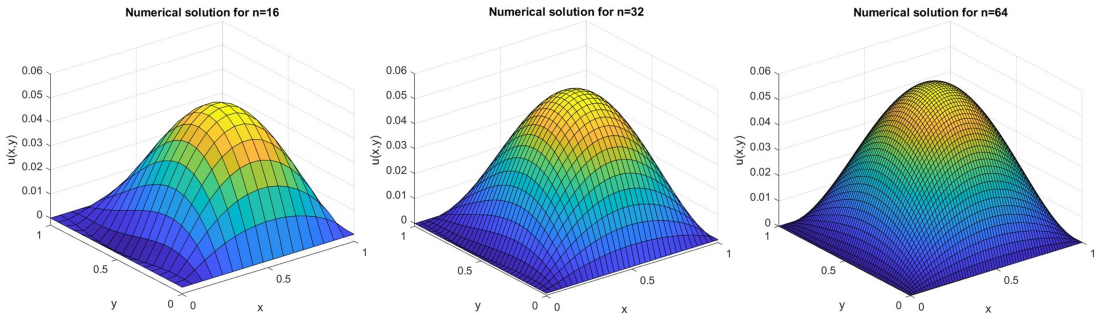

Figure 1: Numerical solution of the test problem for grid density $n = 16,\ 32,\ 64$.

n	8	16	32	64	128	256
e	0.0371	0.0240	0.0136	0.0071	0.0034	0.0015

Table 1: The largest difference between the numerical and exact solution in the grid points (**e**) with respect to the grid density (**n**).

Comparison of the iterative methods

Let us consider the following two sets of boundary value problems depending on k and m:

$$\begin{cases} -10^{-m}\Delta u + k(1,\,0)\cdot\nabla u = 1 \\ u|_{\partial\Omega} = 0 \end{cases} \qquad \begin{cases} -10^{-m}\Delta u + k(-y-\frac{1}{2},\,x-\frac{1}{2})\cdot\nabla u = 1 \\ u|_{\partial\Omega} = 0 \end{cases}$$

Question: How does the number of iterative steps change for the preconditioned CGN and GCR algorithms as we decrease ε and increase the norm of the vector field $\mathbf{w}$?

We can see in the graphs of figures 2 and 3 that in case of the SDFEM discretization (in contrast to the finite difference method), the preconditioned CGN algorithm always takes less steps than the GCR. Comparing the two sets of boundary value problems, the graphs look similar, but their asymptotic behaviour is different.

In figure 2 when the vector field is constant (left), after the first peak at around $k = 1$, the two curves are getting closer to each other while approaching zero. On the other hand, in case of the rotating vector field (right), the two curves maintain the same distance and they settle at a value near the peak.

In figure 3, both curves are ascending. However, their growth is bounded above and they reach their peak at around $m = 6$. After the peak, the number of iterations slowly decreases for the constant vector field, while the other one maintains the same iteration number onwards.

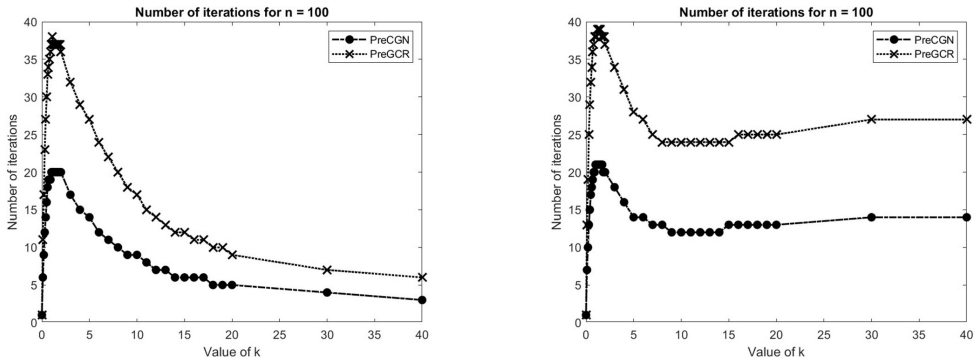

Figure 2: Number of iterative steps taken by the preconditioned CGN and GCR algorithms to solve the system of linear equations with tolerance 10^{-5} when $m = 2$ is fixed and k is varied.

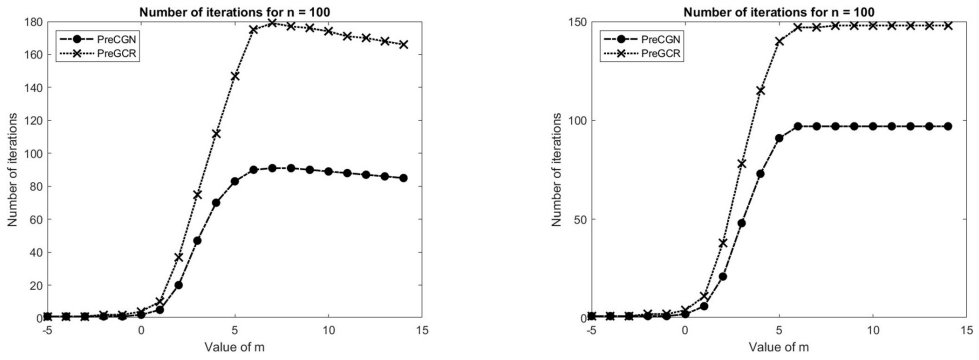

Figure 3: Number of iterative steps taken by the preconditioned CGN and GCR algorithms to solve the system of linear equations with tolerance 10^{-5} when $k = 1$ is fixed and m is varied.

The boundedness of the curves in figure 3 has been studied in [1]. The characteristic curves of the vector field $\mathbf{w} = (1,0)$ can be parameterized with $\gamma_s(t) := (t,s)$, where $(s,t) \in [0,1]^2 = \Omega$. The absolute value of the determinant of its Jacobian matrix can be calculated as follows:

$$J_{\mathbf{w}}(s,t) = \left|\det \begin{pmatrix} \partial_s(t) & \partial_t(t) \\ \partial_s(s) & \partial_t(s) \end{pmatrix}\right| = \left|\det \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\right| = |-1| = 1$$

This is bounded from below and above by $\mu = \tilde{\mu} := 1$. From this, we can calculate

$$C_{\mathbf{w}} = \operatorname{diam}(\Omega) \cdot \sqrt{\tilde{\mu}/\mu} = \operatorname{diam}(\Omega) = \sqrt{1^2+1^2} = \sqrt{2}.$$

The streamline Poincaré–Friedrichs inequality described in [1] gives an upper bound for the number of iterations independent of ε. In case of the CGN method, the bound is the following:

$$\left(\frac{\|r_k\|_S}{\|r_0\|_S}\right)^{\frac{1}{k}} \leq 2^{\frac{1}{k}} \cdot \frac{C_{\mathbf{w}}}{C_{\mathbf{w}} + 2\delta}$$

According to the previous calculations, when $C_{\mathbf{w}} = \sqrt{2}$ and $\delta := h = 0.01$, then $\|r_k\|_S = 10^{-5}$ should be reached after at most around 954 iterative steps. In the graph of figure 3 we can see that actually less than 100 steps were enough for the algorithm to reach the desired accuracy.

Connection with my previous results

In my earlier work when I studied the finite difference method, the CGN algorithm was better for small numbers k, and the GCR algorithm proved to be faster for sufficiently large numbers k. A similar relation can be observed now for the standard FEM when $\delta = 0$, and the transition to the graph in figure 2 can be seen if we gradually increase parameter δ from 0 to $h = 10^{-2}$.

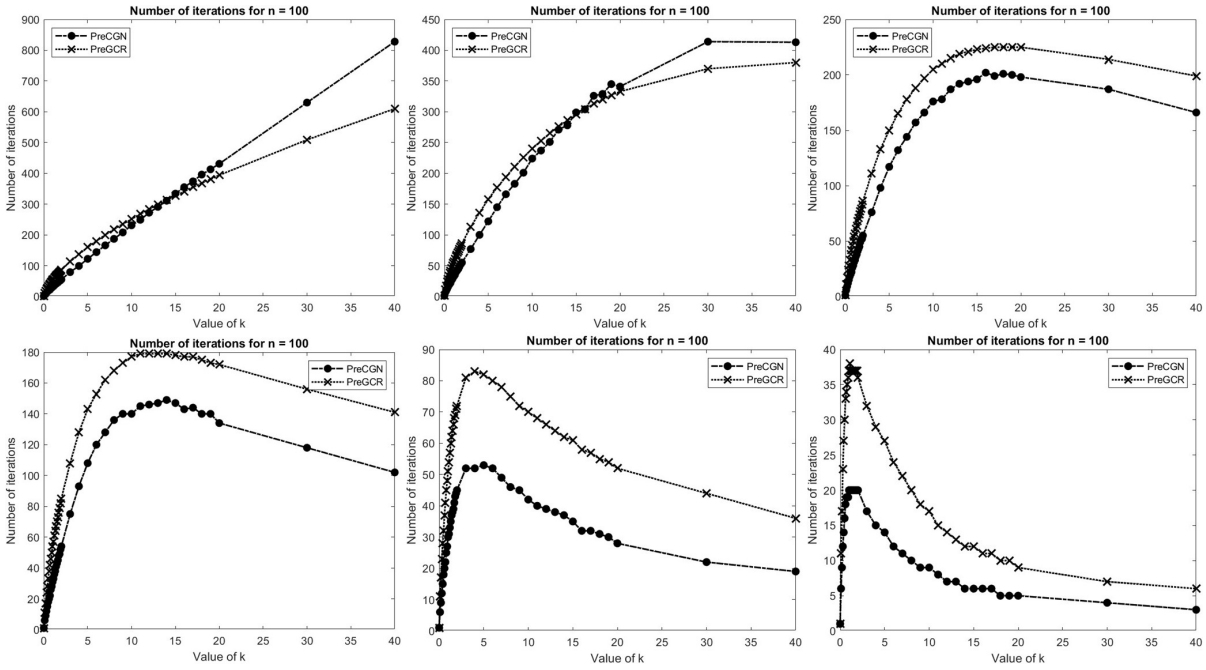

Figure 4: Number of iterative steps taken by the preconditioned CGN and GCR algorithms for $\delta = 0,\ 10^{-5},\ 5\cdot 10^{-5},\ 10^{-4},\ 10^{-3},\ 10^{-2}$ when $m = 2$ is fixed and k is varied.

References

[1] AXELSSON, O.; KARÁTSON, J.; KOVÁCS B.: *Robust Preconditioning Estimates for Convection-Dominated Elliptic Problems via a Streamline Poincaré–Friedrichs Inequality.* SIAM Journal on Numerical Analysis, Vol. 52, Iss. 6, 2014.

[2] KARÁTSON, J.; HORVÁTH, R.: *Numerical Methods for Elliptic Partial Differential Equations.* URL: `https://kajkaat.web.elte.hu/pdnmell-ang-2022.pdf`

[3] BAKOS, I.: *Konvekció-diffúziós egyenletek.* BSc thesis, BME, 2014.

[4] THE MATHWORKS, INC.: *MATLAB R2023a Update 5*. Natick, Massachusetts, 2023. URL: `https://www.mathworks.com/help/matlab/`