Filomat 37:30 (2023), 10395–10413
https://doi.org/10.2298/FIL2330395M

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Algorithms for computing the optimal Geršgorin-type localizations

S. Milićević[a], V. R. Kostić[b,c]

[a]*Department for Applied fundamental disciplines, Faculty of Technical Sciences, University of Novi Sad, Trg D. Obradovića 6, 21000 Novi Sad, Serbia*
[b]*Istituto Italiano di Tecnologia, Via Melen, 83, 16152 Genova, Italy*
[c]*Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Trg D. Obradovića 4, 21000 Novi Sad, Serbia*

Abstract. In this paper we provide novel algorithms for computing the minimal Geršgorin set for the localizations of eigenvalues. Two strategies for curve tracing are considered: predictor-corrector and triangular grid approximation. We combine these two strategies with two characterizations (explicit and implicit) of the Minimal Geršgorin set to obtain four new numerical algorithms. We show that these algorithms significantly decrease computational complexity, especially for matrices of large size, and compare them on matrices that arise in practically important eigenvalue problems.

1. Introduction

There are numerous ways to localize eigenvalues. One of the best known results in numerical linear algebra is that the spectrum of a given square complex matrix is a subset of a union of circles centered at diagonal elements of the matrix whose radii equal to the sum of the moduli of the off-diagonal elements of a corresponding row in the matrix (Geršgorin's theorem, 1931). Among all Geršgorin-type sets, the minimal Geršgorin set (MGS) gives the sharpest and the most precise localization of the spectrum ([6]). While the research on the minimal Geršgorin set provided several interesting theoretical results, its practical computation remain the bottleneck for its wide use. Unlike the Geršgorin set, it is not easy to numerically determine MGS, ([11, 12]), since it is defined as an intersection of infinitely many sets. Luckily, as we will see in the paper, using different approaches, it is possible to overcome this problem even for large matrices.

The paper consists of five sections. In Section 2 we provide some preliminary results. Sections 3 and 4 contains the main contribution, while in Section 5 numerical tests of new algorithms are performed and their comparison with existing algorithms is provided. Finally, we summarize all advantages of new results in a brief conclusion in Section 6.

2020 *Mathematics Subject Classification*. Primary 65F15; Secondary 15A18
Keywords. eigenvalue localization, minimal Geršgorin set, predictor-corrector method, triangular grid
Received: 05 June 2021; Accepted: 10 June 2023
Communicated by Marko Petković
Research supported the Ministry of Science, Research Grant 174019, Provincial Secretariat for Science of Vojvodina, Research Grants 1136, 1850 and 2010.
Email addresses: srdjan88@uns.ac.rs (S. Milićević), vkostic@dmi.uns.ac.rs, vladimir.kostic@iit.it (V. R. Kostić)

2. Preliminaries

The spectrum $\sigma(A)$ of a given matrix $A = [a_{ij}] \in \mathbb{C}^{n,n}, i, j \in N := \{1, 2, \dots, n\}$ is

$$\sigma(A) := \{\lambda \in \mathbb{C} : \det(\lambda I - A) = 0\}, \tag{1}$$

where I is the identity matrix of a size n, $n \in \mathbb{N}$. The spectral abscissa $\alpha(A)$ of $A \in \mathbb{C}^{n,n}$ is defined by

$$\alpha(A) := \{\max(Re(\lambda)) : \lambda \in \sigma(A)\}. \tag{2}$$

The following well-known eigenvalue perturbation result is at the basis of our algorithms.

Theorem 2.1. *([8], Theorem 2) Let λ_0 be a simple eigenvalue of a matrix $A_0 \in \mathbb{C}^{n,n}$, and let v_0 be an associated eigenvector, so that $A_0 v_0 = \lambda_0 v_0$. Then a (complex) function λ and a (complex) vector function v are defined for all A in some neighborhood $\mathcal{O}(A_0) \in \mathbb{C}^{n,n}$ of A_0, such that*

$$\lambda(A_0) = \lambda_0,\ v(A_0) = v_0$$

and

$$Av = \lambda v,\ v_0^* v = 1,\ A \in \mathcal{O}(A_0).$$

Moreover, the functions λ and v are smooth on $\mathcal{O}(A_0)$ and the differentials at A_0 are

$$d\lambda = u_0^*(dA)v_0 / u_0^* v_0 \tag{3}$$

and

$$dv = (\lambda_0 I - A_0)^+ (I - v_0 u_0^* / u_0^* v_0)(dA) v_0, \tag{4}$$

where u_0 is the left eigenvector of A_0 associated with the eigenvalue λ_0.

The set

$$\Gamma(A) := \Big\{ z \in \mathbb{C} : \bigcup_{i=1}^{n} |z - a_{ii}| \leq \sum_{j \in N \setminus \{i\}} |a_{ij}| \Big\}. \tag{5}$$

is called *the Geršgorin set*. It is a well-known result that the spectrum of matrix A is a subset of its Geršgorin set, i.e., $\sigma(A) \subset \Gamma(A)$.
Given a positive vector $x = [x_1, x_2, \dots, x_n] > 0$ and a diagonal matrix $X := diag(x) \in \mathbb{R}^{n,n}$, the associated Geršgorin set for matrix $X^{-1}AX$ is

$$\Gamma^{r^x}(A) := \bigcup_{i=1}^{n} \Big\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \in N \setminus \{i\}} \frac{|a_{ij}| x_j}{x_i} \Big\}. \tag{6}$$

The set

$$\Gamma^{\mathcal{R}}(A) := \bigcap_{x \in \mathbb{R}^n,\, x > 0} \Gamma^{r^x}(A) \tag{7}$$

is called *the minimal Geršgorin set* and it is obvious that $\sigma(A) \subseteq \Gamma^{\mathcal{R}}(A) \subseteq \Gamma(A)$. Moreover, in a certain sense, it gives the sharpest inclusion set for $\sigma(A)$ among all Geršgorin-type sets [6].
The abscissa $\mu(A)$ of the minimal Geršgorin set $\Gamma^{\mathcal{R}}(A)$ is

$$\mu(A) := \{\max(Re(z)) : z \in \Gamma^{\mathcal{R}}(A)\}. \tag{8}$$

Given any matrix $A = [a_{ij}] \in \mathbb{C}^{n,n}$ and the complex number $z \in \mathbb{C}$, define the matrix $Q_A(z) = [q_{ij}(z)]$ by

$$q_{ii}(z) := -|z - a_{ii}| \text{ and } q_{ij}(z) := |a_{ij}|, \text{ for } i, j \in N, \ i \neq j. \tag{9}$$

The right-most eigenvalue of the essentially non-negative $Q_A(z) = [q_{ij}(z)]$ is real and it can be computed by

$$\nu_A(z) := \inf_{x>0} \max_{i\in N} (r_i^x(A) - |z - a_{ii}|). \tag{10}$$

Using this notation, one obtains the following characterization of the minimal Geršgorin set in the complex plane.

Theorem 2.2. *([11, Proposition 4.3]) For any $A = [a_{ij}] \in \mathbb{C}^{n,n}$, $n \geq 2$, then $z \in \Gamma^{\mathcal{R}}(A)$ if and only if $\nu_A(z) \geq 0$. If $z \in \partial\Gamma^{\mathcal{R}}(A)$, then $\nu_A(z) = 0$.*

Theorem 2.3. *([11, Theorem 4.6]) For any irreducible matrix $A = [a_{ij}] \in \mathbb{C}^{n,n}$, $n \geq 2$, then $\nu_A(a_{ii}) > 0$, for every $i \in N$. Moreover, for each a_{ii} and each real θ, $0 \leq \theta \leq 2\pi$, let $\widehat{\rho_i}(\theta)$ be the smallest positive number for which*

$$\nu_A(a_{ii} + \widehat{\rho_i}(\theta)e^{i\theta}) = 0 \tag{11}$$

and there is a sequence of complex numbers $\{z_j\}_{j=1}^{\infty}$ with $\lim_{j\to\infty} z_j = a_{ii} + \widehat{\rho_i}(\theta)e^{i\theta}$, such that $\nu_A(z_j) < 0$, $j \in \mathbb{N}$. Then, the complex interval $[a_{ii} + te^{i\theta}]_{t=0}^{\widehat{\rho_i}(\theta)}$ is contained in $\Gamma^{\mathcal{R}}(A)$, i.e.,

$$\bigcup_{\theta=0}^{2\pi} [a_{ii} + te^{i\theta}]_{t=0}^{\widehat{\varrho_i}(\theta)} \tag{12}$$

is a star-shaped *subset of $\Gamma^{\mathcal{R}}(A)$ with respect to a_{ii} and*

$$a_{ii} + \widehat{\varrho_i}(\theta)e^{i\theta} \in \partial\Gamma^{\mathcal{R}}(A). \tag{13}$$

As the boundary points of $\Gamma^{\mathcal{R}}(A)$ are of the form $a_{ii} + \widehat{\varrho_i}(\theta)e^{i\theta}$, here, for a given matrix $A \in \mathbb{C}^{n,n}$, $\xi \in \mathbb{C}$ and $\theta \in [0, 2\pi)$, we define a function $f_A^{\xi,\theta} : [0, \infty) \to \mathbb{R}$ by

$$f_A^{\xi,\theta}(t) = \nu_A(\xi + te^{\mathrm{i}\theta}). \tag{14}$$

Using the function (14), we get the *explicit* characterization of MGS ([7]).

Next, we present a different characterization of the minimal Geršgorin set constructed in ([10]). Given an arbitrary irreducible matrix $A \in \mathbb{C}^{n,n}$, a complex number ξ and a real number $\theta \in [0, 2\pi)$, let us fix a vector $c \in \mathbb{R}^n$, $c > 0$, and for every $t \geq 0$ construct a system of linear equations

$$\underbrace{\begin{bmatrix} -Q_A(\xi + t\,e^{\mathrm{i}\theta}) & -c \\ -c^T & 0 \end{bmatrix}}_{M_A^{\xi,\theta}(t)} \begin{bmatrix} w_A^{\xi,\theta}(t) \\ g_A^{\xi,\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}. \tag{15}$$

For a fixed $\xi \in \mathbb{C}$ and $0 \leq \theta < 2\pi$, for $t \geq 0$, define the functions:

$$\chi_A^{\xi,\theta}(t) := \min\left\{(w_A^{\xi,\theta}(t))_i \ : \ 1 \leq i \leq n\right\} \text{ and } h_A^{\xi,\theta}(t) := \min\left\{g_A^{\xi,\theta}(t), \chi_A^{\xi,\theta}(t)\right\}. \tag{16}$$

Theorem 2.4. *([10, Theorem 3.4]) Let be given an arbitrary irreducible matrix $A \in \mathbb{C}^{n,n}$ and its arbitrary diagonal entry $\xi = a_{kk}$, $k \in N$ and $z = \xi + t\,e^{\mathrm{i}\theta}$, where $t \geq 0$ and $0 \leq \theta < 2\pi$. Then,*

- *$z \notin \Gamma^{\mathcal{R}}(A)$ if and only if $h_A^{\xi,\theta}(t) > 0$;*

- *$z \in \partial\Gamma^{\mathcal{R}}(A)$ such that $t = \widehat{\rho}_k(\theta)$ if and only if*

    i) $h_A^{\xi,\theta}(t) = g_A^{\xi,\theta}(t) = 0$,

    ii) $h_A^{\xi,\theta}(s_1) \leq 0$ *holds for all* $0 \leq s_1 \leq t$, *and*

    iii) for every $\varepsilon > 0$ *there exists* $s_2 \geq t$ *such that* $s_2 - t < \varepsilon$ *and* $h_A^{\xi,\theta}(s_2) > 0$.

We refer to the previous theorem as the *implicit* characterization of MGS ([10]).

Finally, to simplify notation, by $\mathrm{diag}(a_i)_{i\in N}$ we will denote a diagonal matrix $\mathrm{diag}(a_1, \ldots, a_n)$.

3. Predictor-corrector framework : algorithms eMGSp and iMGSp

One of the typical path following methods to numerically trace the curve C in the complex plane is a generic predictor-corrector method. It uses a combination of two different steps.
Let C be a solution curve of the equation $H(\omega) = 0$, where $H : \mathbb{C} \to \mathbb{R}$ is a smooth map and $0 \in \mathrm{range}(H)$. In the first step (predictor step), an approximation along the curve is used, usually in the direction of the tangent of the curve. In the second step (corrector step), iterations for solving $H(\omega) = 0$ are used. In that way, the predicted point "brings back" to the curve.

Generic predictor-corrector method

Input: $\omega_0 \in \mathbb{C}$, $H(\omega_0) \approx 0$ (initial point), $h > 0$ (initial step length)
for $k = 1 : m$ **do**
  (**Predictor step)** Predict $z_i \in \mathbb{C}$ such that $\|z_i - \omega_{i-1}\| \approx h$ in the direction of tracing;
  (**Corrector step)** Let $\omega_i \in \mathbb{C}$ approximately solve
  $\min\limits_{\omega}\{\|z_i - \omega\| : H(\omega) = 0\}$ and choose a new step-length $h > 0$;
end for
Output: $\omega_i \in \mathbb{C}$, $i \in \{1, 2, ..., m\}$

Since we will repeatedly work on construction of mappings with a complex argument $f : \mathbb{C} \to \mathbb{C}^{m,n}$, where $m, n \in \mathbb{N}$, without possible confusion to simplify notations, we will use abbreviations:

$$f = f(z), \text{ for } z \in \mathbb{C};$$

$$f^{\xi,\theta}(t) = f(\xi + te^{\mathrm{i}\theta}), \text{ for } \xi \in \mathbb{C},\ \theta \in [0, 2\pi),\ t \in \mathbb{R};$$

$$f(x, y) = f(x + \mathrm{i}y), \text{ for } x, y \in \mathbb{R}.$$

In that context, derivatives in the corresponding arguments are denoted as:

$$f_x = \frac{\partial}{\partial x} f(x, y),\ f_y = \frac{\partial}{\partial y} f(x, y),\ f_{xx} = \frac{\partial^2}{\partial x^2} f(x, y),\ f_{xy} = \frac{\partial^2}{\partial x \partial y} f(x, y),\ f_{yy} = \frac{\partial^2}{\partial y^2} f(x, y).$$

First, we consider the explicit characterization of the minimal Geršgorin set. In the following theorem, we

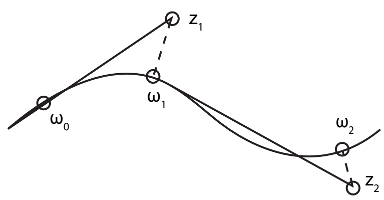


Figure 1: Predictor-corrector step.

present derivatives of the first and second order of f_A, with respect to x and y.

Theorem 3.1. *For a given an irreducible matrix $A \in \mathbb{C}^{n,n}$ and $z = x + \mathrm{i}y \in \mathbb{C}$, let's $v(x + \mathrm{i}y)$ and $u(x + \mathrm{i}y)$ be right and left eigenvector of $Q_A(x + \mathrm{i}y)$, corresponding to $f_A(x + \mathrm{i}y)$, where $Q_A(x + \mathrm{i}y)$ and $f_A(x + \mathrm{i}y)$ are defined by (9) and (14), respectively. Then, the first and second derivatives of f_A are defined by:*

$$f_x = -\frac{u^T D_x v}{u^T v}, \tag{17}$$

$$f_y = -\frac{u^T D_y v}{u^T v}, \tag{18}$$

$$f_{xx} = -\frac{u^T D_{xx} v + 2u^T D_x v_x + 2 f_x u^T v_x}{u^T v}, \tag{19}$$

$$f_{xy} = -\frac{u^T D_{xy} v + u^T D_x v_y + u^T D_y v_x + f_x u^T v_y + f_y u^T v_x}{u^T v}, \text{ and} \tag{20}$$

$$f_{yy} = -\frac{u^T D_{yy} v + 2u^T D_y v_y + 2 f_y u^T v_y}{u^T v}, \tag{21}$$

where for $i \in N$:

$$D_x = \operatorname{diag}\Big(\frac{Re(x + \mathrm{i}y - a_{ii})}{|x + \mathrm{i}y - a_{ii}|}\Big)_{i\in N},$$

$$D_y = \operatorname{diag}\Big(\frac{Im(x + \mathrm{i}y - a_{ii})}{|x + \mathrm{i}y - a_{ii}|}\Big)_{i\in N},$$

$$D_{xx} = \operatorname{diag}\Big(\frac{(Im(x + \mathrm{i}y - a_{ii}))^2}{|x + \mathrm{i}y - a_{ii}|^3}\Big)_{i\in N},$$

$$D_{xy} = \operatorname{diag}\Big(\frac{-Re(x + \mathrm{i}y - a_{ii})Im(x + \mathrm{i}y - a_{ii})}{|x + \mathrm{i}y - a_{ii}|^3}\Big)_{i\in N}, \text{ and}$$

$$D_{yy} = \operatorname{diag}\Big(\frac{(Re(x + \mathrm{i}y - a_{ii}))^2}{|x + \mathrm{i}y - a_{ii}|^3}\Big)_{i\in N}, \text{ for } z = x + \mathrm{i}y \neq a_{ii}.$$

Proof. From the definition of f_A, we have

$$Q_A(x + \mathrm{i}y)v(x, y) = f_A(x + \mathrm{i}y)v(x, y), \tag{22}$$

$$(u(x, y))^T Q_A(x + \mathrm{i}y) = f_A(x + \mathrm{i}y)(u(x, y))^T. \tag{23}$$

Differentiating the equation (22) with respect to x and y, we obtain

$$-D_x v + Q_A v_x = f_x v + f v_x, \tag{24}$$

$$-D_y v + Q_A v_y = f_y v + f v_y. \tag{25}$$

Multiplying the equations (24) and (25) by u^T and using (22) and (23), we obtain (17) and (18).

Using Theorem 2.1, we have

$$v_x = -(f_A I - Q_A)^+ (I - \frac{vu^T}{u^T v}) D_x v \tag{26}$$

$$v_y = -(f_A I - Q_A)^+ (I - \frac{vu^T}{u^T v}) D_y v. \tag{27}$$

Differentiating the equation (24) with respect to x and y, and the equation (25) with respect to y, we obtain equations:

$$-D_{xx} v - 2D_x v_x + Q_A v_{xx} = f_{xx} v + 2 f_x v_x + f v_{xx}, \tag{28}$$

$$-D_{xy} v - D_x v_y - D_y v_x + Q_A v_{xy} = f_{xy} v + f_x v_y + f_y v_x + f v_{xy}, \tag{29}$$

$$-D_{yy} v - 2D_y v_y + Q_A v_{yy} = f_{yy} v + 2 f_y v_y + f v_{yy}, \tag{30}$$

respectively.

Multiplying the equations (28), (29) and (30) by u^T and using (22) and (23), we obtain expressions (19), (20) and (21). $\square$

Now, let's consider the implicit characterization of the minimal Geršgorin set given by the system:

$$\begin{bmatrix} -Q_A(x+\mathrm{i}y) & -c \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} w_A(x,y) \\ g_A(x,y) \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}. \tag{31}$$

Theorem 3.2. *Given an irreducible matrix $A \in \mathbb{C}^{n,n}$, a vector $c > 0$, $c \in \mathbb{R}^n$, and w_A and g_A defined by the system (31). Then, the first and second derivatives of g_A are given by systems:*

$$\begin{bmatrix} -Q_A & -c \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} w_x & w_y \\ g_x & g_y \end{bmatrix} = \begin{bmatrix} -D_x w & -D_y w \\ 0 & 0 \end{bmatrix} \tag{32}$$

and

$$\begin{bmatrix} -Q_A & -c \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} w_{xx} & w_{xy} & w_{yy} \\ g_{xx} & g_{xy} & g_{yy} \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} D_1 & D_2 & D_3 \\ 0 & 0 & 0 \end{bmatrix}, \tag{33}$$

where $D_1 = -2D_x w_x - D_{xx} w$, $D_2 = -D_x w_y - D_y w_x - D_{xy} w$ and $D_3 = -2D_y w_y - D_{yy} w$.

Proof. Differentiating the system of equations

$$\begin{aligned} -Q_A w - cg &= 0 \\ -c^T w &= -1 \end{aligned} \tag{34}$$

with respect to x and y, we obtain systems:

$$\begin{aligned} -Q_A w_x - c g_x &= -D_x w \\ -c^T w_x &= 0 \end{aligned} \tag{35}$$

and

$$\begin{aligned} -Q_A w_y - c g_y &= -D_y w \\ -c^T w_y &= 0, \end{aligned} \tag{36}$$

respectively.
Writing (35) and (36) in a matrix form, we obtain (32).

Differentiating the system (35) with respect to x and y, and the system (36) with respect to y, we obtain systems:

$$\begin{aligned} -Q_A w_{xx} - c g_{xx} &= -2D_x w_x - D_{xx} w \\ -c^T w_{xx} &= 0 \end{aligned}, \tag{37}$$

$$\begin{aligned} -Q_A w_{xy} - c g_{xy} &= -D_x w_y - D_y w_x - D_{xy} w \\ -c^T w_{xy} &= 0 \end{aligned} \tag{38}$$

and

$$\begin{aligned} -Q_A w_{yy} - c g_{yy} &= -2D_y w_y - D_{yy} w \\ -c^T w_{yy} &= 0 \end{aligned}, \tag{39}$$

respectively. Finally, using (37), (38) and (39), it follows (33). $\square$

Now, we construct the algorithm **eMGSp** given in detail in Appendix A.1. The boundary of the minimal Geršgorin set is given by $\partial\Gamma^{\mathcal{R}}(A) = \{z = x + \mathrm{i}y \in \mathbb{C} : f_A(x + \mathrm{i}y) = 0\}$. Starting with the point $\omega_0 \in \partial\Gamma^{\mathcal{R}}$, which we can obtain by the procedure **eSearch** from [7], we want to find the next point on the boundary of $\Gamma^{\mathcal{R}}(A)$, named ω_1.

Firstly, in the predictor step, we obtain the point

$$z_1 := \omega_0 + hdl, \tag{40}$$

where h is a given length of a step and $dl := \pm\dfrac{-f_y + \mathrm{i}f_x}{|-f_y + \mathrm{i}f_x|}$ (we choose the sign in the direction of the curve tracing), where f_x and f_y are computed in ω_0.

Then, in the corrector step, we want to find the point $\omega_1 \in \partial\Gamma^{\mathcal{R}}$, which is the nearest to z_1. To that end, we solve the problem:

$$\|\omega - z_1\|_2^2 \to \min,\ f_A(\omega) = 0.$$

Forming a function

$$L(x, y, \lambda) := (x - \mathrm{Re}(z_1))^2 + (y - \mathrm{Im}(z_1))^2 + \lambda f_A(x, y)$$

and differentiating it with respect to x and y, we obtain the following iterations:

$$\begin{bmatrix} x_1^{(k)} \\ y_1^{(k)} \\ \lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \mathrm{Re}(\omega_1^{(k)}) \\ \mathrm{Im}(\omega_1^{(k)}) \\ 1 \end{bmatrix} - \begin{bmatrix} 2 + \lambda f_{xx} & \lambda f_{xy} & f_x \\ \lambda f_{xy} & 2 + \lambda f_{yy} & f_y \\ f_x & f_y & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\mathrm{Re}(\omega_1^{(k)} - z_1) + \lambda f_x \\ 2\mathrm{Im}(\omega_1^{(k)} - z_1) + \lambda f_y \\ f \end{bmatrix}, \tag{41}$$

where $\omega_1^{(0)} := z_1,\ \omega_1^{(k)} := x_1^{(k-1)} + \mathrm{i}y_1^{(k-1)},\ k \in \mathbb{N}$, and $f,\ f_x,\ f_y,\ f_{xx},\ f_{xy}$ and f_{yy} are computed in $\omega_1^{(k)}$.

Computation of these iterations will stop when $|f_A| \leq tol$, for some $l \in \mathbb{N}$ and a given accuracy $tol > 0$. In practice, as z_1 is near the border of $\Gamma^{\mathcal{R}}(A)$, it is sufficient to compute just a few iterations. In that way, we get $\omega_1 := \omega_1^{(l)}$.

Analogously, we find a sequence of points $\{\omega_j\}_{j=0}^{m}$, which approximate the boundary of the one component of minimal Geršgorin set. In the same way, we can find approximation of all other components of $\Gamma^{\mathcal{R}}(A)$.

Finally, using the function g_A instead of f_A for the characterization of the boundary of the minimal Geršgorin set and the procedure **iSearch** from [10] instead of the procedure **eSearch**, we construct the implicit predictor-corrector method for computing the minimal Geršgorin set- **iMGSp**, see Appendix A.1. In that case, we use the characterizations of derivatives of g_A given in Theorem 3.2.

4. Triangular grid framework: algorithms eMGSt and iMGSt

In this section, two new algorithms for computing the minimal Geršgorin set are constructed. For a given matrix $A \in \mathbb{C}^{n,n}$, we combine the triangular grid approach presented in [9] with the characterizations of $\Gamma^{\mathcal{R}}(A)$ by the functions f_A and h_A to develop algorithms **eMGSt** and **iMGSt**, respectively.

Given any $(z_i, z_e) \in \mathbb{C}^2$ such that $z_i \neq z_e$, for $k \neq l$, define the following points:

$$\mathcal{L}_{k,l} = z_i + k(z_e - z_i) + l(z_e - z_i)e^{\frac{\mathrm{i}\pi}{3}},$$

to obtain a uniform lattice of vertices

$$\mathcal{L}(z_i, z_e) = \{\mathcal{L}_{k,l} : (k, l) \in \mathbb{Z}^2\},$$

satisfying

$$|\mathcal{L}_{k,l+1} - \mathcal{L}_{k,l}| = |\mathcal{L}_{k+1,l} - \mathcal{L}_{k,l}| = |z_i - z_e|.$$

Next, we define a triangular mesh, see Figure 2, as:

$$\Omega(z_i, z_e) = \Psi(z_i, z_e) \cup \widetilde{\Psi}(z_i, z_e),$$

where

$$\Psi(z_i, z_e) = \{T_{kl} = \{\mathcal{L}_{k,l}, \mathcal{L}_{k+1,l}, \mathcal{L}_{k,l+1}\} : (k, l) \in \mathbb{Z}^2\},$$

and

$$\widetilde{\Psi}(z_i, z_e) = \{\tilde{T}_{kl} = \{\mathcal{L}_{k,l}, \mathcal{L}_{k+1,l}, \mathcal{L}_{k+1,l-1}\} : (k, l) \in \mathbb{Z}^2\}.$$

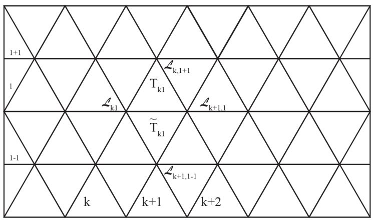


Figure 2: Triangular grid.

For a given matrix $A \in \mathbb{C}^{n,n}$, let us denote by $\mathcal{T}$ the subset of $\Omega(z_i, z_e)$, where $T \in \mathcal{T}$ if and only if T has at least one vertex in $\Gamma^{\mathcal{R}}(A)$ and at least one outside of $\Gamma^{\mathcal{R}}(A)$.
Let the pivot $p(T)$ be the vertex of a triangle $T \in \mathcal{T}$ which is situated on the opposite side of the border of $\Gamma^{\mathcal{R}}(A)$ to other two vertices, e.g., vertex $\tilde{z}_{i,0}$ in the triangle $\{\tilde{z}_{i,0}, \tilde{z}_{i,1}, \tilde{z}_{i,2}\}$ in Figure 3. Define a transformation:

$$F(T) = \rho(p(T), \operatorname{sgn}(\nu_A(p(T))) \cdot \frac{\pi}{3})(T),$$

where $\operatorname{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$ and $\rho(z, \theta)(\omega)$ denotes the rotation of $\omega \in \mathbb{C}$ centered at $z \in \mathbb{C}$ with angle θ, i.e.,

$$\rho(z, \theta)(\omega) := z + (\omega - z)e^{\mathrm{i}\theta}.$$

Now, we state some useful properties of triangular grids and mapping F defined on them. The proofs of the following prepositions are given in [9].

Proposition 4.1. *For $z_i \neq z_e$, $\mathcal{T}$ is a finite set.*

Proposition 4.2. *For a given triangle $T \in \mathcal{T}$, the following statements hold:*

1. *$F(T) \neq T$;*
2. *$p(T)$ is a vertex of $F(T)$;*
3. *T and $F(T)$ are adjacent;*
4. *$F(T) \in \mathcal{T}$;*
5. *$p(F(T))$ is a vertex of T;*
6. *$F^2(T) \neq T$;*
7. *if $T \in \Psi(z_i, z_e)$, then $F(T) \in \tilde{\Psi}(z_i, z_e)$ and if $T \in \tilde{\Psi}(z_i, z_e)$, then $F(T) \in \Psi(z_i, z_e)$.*

Proposition 4.3. *F is a bijection from $\mathcal{T}$ onto $\mathcal{T}$.*

For any given $T \in \mathcal{T}$ define $T_k := F^k(T), k \in \mathbb{N}$, and set $O(T) := \{T_k,\ k \in \mathbb{N}\}$, where $T_0 := T$.

Proposition 4.4. *For a given triangle $T \in \mathcal{T}$, the following statements hold:*

1. *the set $O(T)$ is finite;*
2. *if $n = card(O(T))$, then n is even and the smallest positive integer such that $T_n = T$;*
3. $\sum_{i=0}^{n-1} \theta_i = 2\pi m,\ m \in \mathbb{N}_0$, *where θ_i is the rotation angle of F for the triangle T_i;*
4. *for a given triangle T', either $O(T) = O(T')$ or $O(T) \cap O(T') = \emptyset$.*

Using the prepositions above and the function f_A, we can construct an algorithm **eMGSt**, see Appendix A.2. As before, given irreducible matrix $A \in \mathbb{C}^{n,n}$, the set of its different diagonal elements is $\mathcal{D} = \{a_{i_1i_1}, a_{i_2i_2}, ..., a_{i_{\tilde{n}}i_{\tilde{n}}}\}$, where $\mathrm{Re}(a_{i_1i_1}) \leq \mathrm{Re}(a_{i_2i_2}) \leq ... \leq \mathrm{Re}(a_{i_{\tilde{n}}i_{\tilde{n}}})$, $\tilde{n} \in N$, and let s be the number of disjoint components of $\Gamma^{\mathcal{R}}(A)$. We denote m_i points representing the approximation of the boundary of the i^{th} component of the minimal Geršgorin set by $\{z_{i,j}\}_{j=1}^{m_i}$, $i \in \{1, 2, ..., s\}$. Starting with $\xi = a_{i_1i_1}$, $\varphi = -\pi$ and given accuracy $\epsilon > 0$ (e.g., $\epsilon = 10^{-12}$), we use the procedure **eSearch** from [7] to obtain $\omega_1 := \xi + t_1 e^{\mathrm{i}\varphi}$, with $t_1 :=$ **eSearch**$(A, \xi, \varphi, \epsilon)$.

Then, we get the points $\tilde{z}_{1,0} := \omega_1 - \frac{\tilde{\tau}}{2}$ and $\tilde{z}_{1,1} := \omega_1 + \frac{\tilde{\tau}}{2}$, where $\tilde{\tau}$ is a given length of edge of equilateral triangles which form triangular grid and $z_{1,1} := \tilde{z}_{1,1}$. Furthermore, we obtain the point $\tilde{z}_{1,2} := \tilde{z}_{1,0} + (\tilde{z}_{1,1} - \tilde{z}_{1,0})e^{\frac{\mathrm{i}\pi}{3}}$. As a result, the triangle $T_1 = \{\tilde{z}_{1,0}, \tilde{z}_{1,1}, \tilde{z}_{1,2}\}$ is an element of $\mathcal{T}$ which generates the set $O(T_1)$. If $f_A(\tilde{z}_{1,2}) < 0$, we define $z_{1,2} := \tilde{z}_{1,2}$ and choose as pivot $\tilde{z}_{piv}$ the point $\tilde{z}_{1,0}$ to get $\tilde{z}_{1,3} := \tilde{z}_{piv} + (\tilde{z}_{1,2} - \tilde{z}_{piv})e^{\frac{\mathrm{i}\pi}{3}}$. Otherwise, we choose $\tilde{z}_{piv} := \tilde{z}_{1,1}$ and $\tilde{z}_{1,3} := \tilde{z}_{piv} + (\tilde{z}_{1,2} - \tilde{z}_{piv})e^{-\frac{\mathrm{i}\pi}{3}}$. Analogously, we construct a sequence of points $\{\tilde{z}_{1,l}\}$, $l \in \mathbb{N}_0$, as long as $\tilde{z}_{1,l} = \tilde{z}_{1,0}$. From that set of points, we choose a subset $\{z_{1,j}\}_{j=1}^{m_1}$, such that $f_A(z_{1,j}) < 0$. The obtained polygon $\{z_{1,j}\}_{j=1}^{m_1}$ contains one component of the set minimal Geršgorin set $\Gamma^{\mathcal{R}}(A)$, see Figure 3, and $\mathrm{dist}(z_{i,j}, \partial\Gamma^{\mathcal{R}}(A)) \leq \tilde{\tau}$. Notice, that we could give also an inner approximation of the boundary $\partial\Gamma^{\mathcal{R}}(A)$ simply by taking a subset of points with non-negative values of f_A.

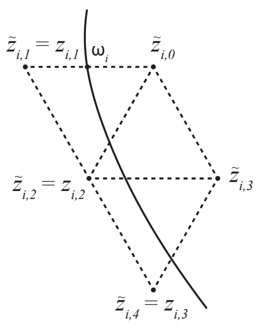


Figure 3: Construction of the polygon $\{z_{i,j}\}_{j=1}^{m_i}$.

After completing the construction of the first component of the minimal Geršgorin set, we check which entries from $\mathcal{D}$ are in that component and denote the set of these diagonal entries by $\mathcal{S}_1$. If $\mathcal{S}_1 \neq \mathcal{D}$, choosing for ξ the leftmost element of $\mathcal{D}\backslash\mathcal{S}_1$, we construct a new polygon $\{z_{2,j}\}_{j=1}^{m_2}$ that represents the approximation of next disjoint component of the minimal Geršgorin set. Then, we again test which entries from the set $\mathcal{D}\backslash\mathcal{S}_1$ are in that component and denote the set of these entries by $\mathcal{S}_2$. We stop with that procedure when all elements of $\mathcal{D}$ are included in some components of the minimal Geršgorin set.

Finally, we can simply construct the implicit version **iMGSt** by replacing the function f_A with the function h_A, see Appendix A.2. Doing so, using the idea of the implicit determinant method [5], we achieve to reduce significantly the overall number of expensive eigenvalue computations. This algorithm gives excellent results, especially for matrices of large sizes, which will be shown through examples in the next section.

Moreover, remark that both algorithms **eMGSt** and **iMGSt** produce polygons that include the minimal Geršgorin set in their interiors (interior is where diagonal entries lie). Indeed, for every point $z_{i,j}$ of the constructed polygons by the algorithm **eMGSt** / **iMGSt**, it holds that $f_A(z_{i,j}) < 0$ / $h_A(z_{i,j}) < 0$. But this, according to Theorems 2.2 and 2.4, guaranties that $z_{i,j} \in \mathbb{C} \setminus \Gamma^{\mathcal{R}}(A)$ and, hence, polygons $\{z_{i,j}\}_{j=1}^{m_i}$ surround connected components of MGS.

5. Numerical examples

In this section, we test new ({**e,i**}**MGS**{**s,p,t**}) on three examples and compare their performances with the performances of **eMGS** and **iMGS** algorithms that were the state of art. We notice that new approaches significantly accelerate convergence. To adapt notation, the abbreviations **eMGSs** and **iMGSs** are used instead of **eMGS** and **iMGS**, respectively, and parameters $\epsilon_1 = tol$ and $\epsilon_2 = \dfrac{2d(A)\sqrt{3}}{3N_s}$. All algorithms are implemented in MATLAB version R2018b and tested on 2.7 GHz Intel® Core™ i7 machine.

Example 5.1. *In the first example we test algorithms on the cyclic matrix of a size $n = 4$:*

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & i & 1 \\ 1 & 0 & 0 & -i \end{bmatrix},$$

setting the parameters of the algorithms to be: $tol = 10^{-12}$, $\tau = 2$, $h = 0.0254$, $N_s = 40$ and $N_t = 500$. CPU times for all algorithms are presented in Table 1. The number of computed points for **eMGSs** *and* **iMGSs** *is 430, for* **eMGSp** *and* **iMGSp** *436 and for* **eMGSt** *and* **iMGSt** *it is 2086. Figure 4 shows the minimal Geršgorin set of A using all three approaches. Also, their corresponding zoomed versions around the orgin are presented. Comparing them, we notice that the algorithms* **eMGSt** *and* **iMGSt** *give more reliable approximation (zero belongs to the minimal Geršgorin set of A).*

MGS	s	p	t
e	1.4667s	0.2728s	0.1102s
i	0.3721s	0.1203s	**0.0282**s

Table 1: CPU times for Example 5.1.

Example 5.2. *In this example we implement the algorithms on the Leslie matrix: $L = diag(b \cdot (1:n-1).\hat{}(-1), -1) + a \cdot [\xi.\hat{}(1:n); zeros(n-1,n)], L(1,1) = 0$, for values $a = 0.1$, $b = 0.2$, $\xi = 0.95$ and $n = 70$. The results obtained with parameters $tol = 10^{-12}$, $\tau = 2$, $h = 0.0036, N_s = 100$ and $N_t = 200$ are presented in Table 2. Figure 5 represents the approximation of the minimal Geršgorin set of the Leslie matrix obtained by (a)* ***eMGSs/iMGSs*** *algorithm (315 points), (b)* ***iMGSp*** *algorithm (315 points) and (c)* ***eMGSt/iMGSt*** *algorithm (602 points).*

MGS	s	p	t
e	38.9456s	/	2.3955s
i	1.0306s	0.2796s	**0.2576**s

Table 2: CPU times for Example 5.2.

Example 5.3. *In this example, we test all algorithms on the Orr-Sommerfeld matrix of a size $n = 1000$ from the Matrix Market respiratory ([4]). For $tol = 10^{-12}$ and $N_t = 400$, the CPU time for **iMGSt** is* 51.41335*s* (546 *points, see Figure* 6). *Other algorithms do not give results in the observed period.*

Example 5.4. *Finally, in the last example, the performance of the algorithms w.r.t. computational time is measured for randomly perturbed the Grcar matrices ([4]). Namely, for different problem sizes $n \in \{10, 25, 50, 100, 150, 200\}$, we run all six algorithms for the matrix $A + x^T y$, where A is the Grcar matrix and x and y are normed standard Gaussian vectors of a size n. For each size experiment is repeated ten times and mean CPU times are shown in Table* 3 *together with standard deviations in the brackets. Additionally, this is illustrated in Figure* 8 *and the typical shape of MGS is shown in Figure* 7 *for one instance of perturbation. The parameters for the algorithms are chosen as $tol = 10^{-12}$, $\tau = 2$ and $N_s = 50$ for ***MGSs**, $tol = 10^{-12}$ and $h = 0.08$ for ***MGSp**, and $tol = 10^{-12}$ and $N_t = 200$ for ***MGSt**, where $* \in \{e, i\}$. We observe that as the size of the problem grows, implicit algorithms perform much better, and among them specially **iMGSt** outperforms others by a safe margin.*

MGS	**s**	**p**	**t**
		n=10	
e	1.8192*s* (0.0361*s*)	0.2053*s* (0.0287*s*)	**0.0295*s* (0.0029*s*)**
i	0.3243*s* (0.0116*s*)	0.11373*s* (0.0167*s*)	0.0405*s* (0.0023*s*)
		n=25	
e	11.6757*s* (0.4536*s*)	0.8182*s* (0.0132*s*)	0.1424*s* (0.0126*s*)
i	0.7342*s* (0.0239*s*)	0.1031*s* (0.0196*s*)	**0.0869*s* (0.0125*s*)**
		n=50	
e	12.5082*s* (0.3717*s*)	2.7947*s* (0.0415*s*)	0.5593*s* (0.0218*s*)
i	2.2096*s* (0.1085*s*)	0.2564*s* (0.0238*s*)	**0.1941*s* (0.0158*s*)**
		n=100	
e	31.3543*s* (4.3844*s*)	10.7534*s* (0.2943*s*)	2.5751*s* (0.1344*s*)
i	6.1163*s* (0.4178*s*)	0.9892*s* (0.0807*s*)	**0.4161*s* (0.0243*s*)**
		n=150	
e	61.4527*s* (12.2141*s*)	31.4344*s* (0.5817*s*)	8.8938*s* (0.1544*s*)
i	13.9478*s* (0.2921*s*)	2.2617*s* (0.0351*s*)	**0.7827*s* (0.0273*s*)**
		n=200	
e	101.2427*s* (13.1459*s*)	56.9199*s* (0.9823*s*)	14.9982*s* (0.4493*s*)
i	22.2831*s* (1.2684*s*)	4.7791*s* (0.4092*s*)	**1.0643*s* (0.0302*s*)**

Table 3: CPU times for Example 5.4.

6. Conclusion

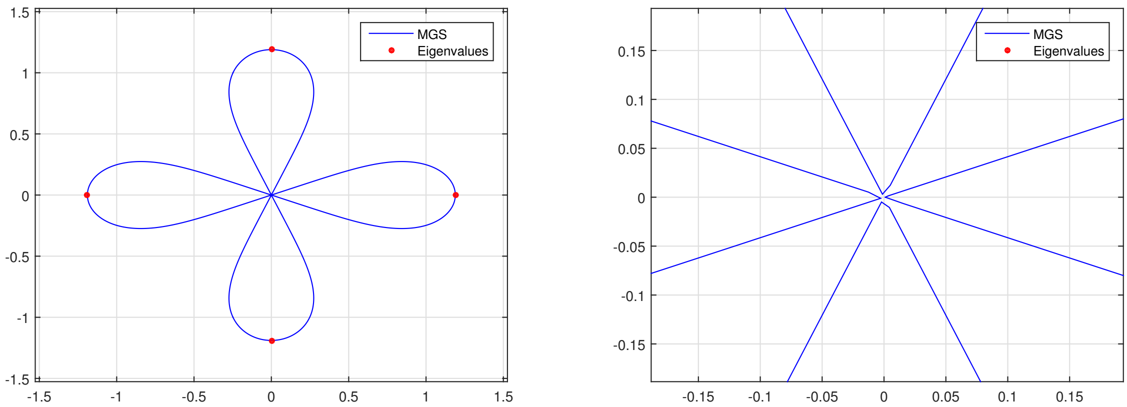
In this paper, we have developed new algorithms for computing the minimal Geršgorin set that have several important advantages. Firstly, new methods are significantly faster. As it is presented in the examples, the run time of new algorithms outperforms the existing ones. Furthermore, for some test matrices of large sizes, the previously known algorithms did not produce any result in the observed period of time. Secondly, new algorithms are simpler for implementation. For example, the algorithms which use the triangular approach for curve tracing are straightforward since they do not depend on many parameters (the only required information is accuracy and the number of triangular grid points). All other necessary information is computed automatically. Third, new approaches are more reliable. The algorithms **eMGSt** and **iMGSt** produce the polygons that always contain the desired localization set.

7. Acknowledgements

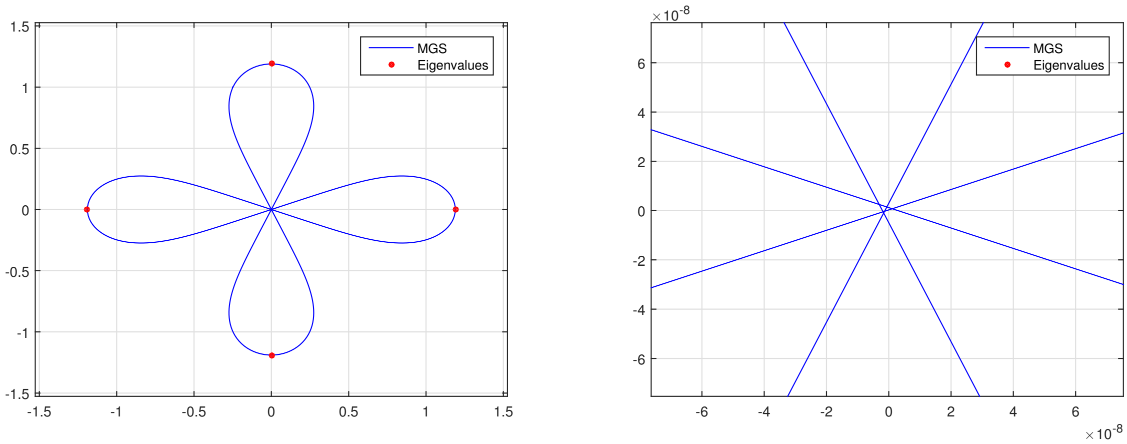
The work of S. Milicević and V. R. Kostić has been partially supported by the Ministry of Science, Research Grant 174019, Provincial Secretariat for Science of Vojvodina, Research Grants 1136, 1850 and 2010.

References

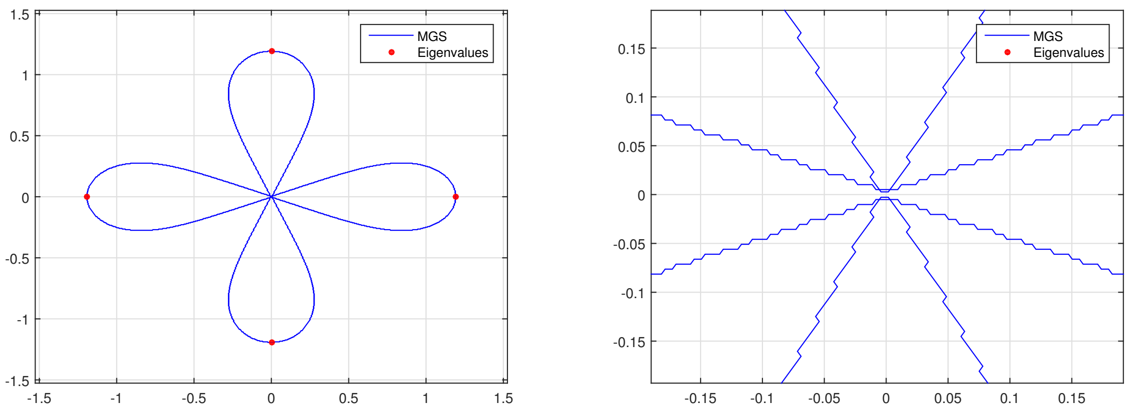
[1] E. L. Allgower and K. Georg, Continuation and path following, Acta numerica, 1-64, 1993.
[2] E. L. Allgower and K. Georg, Numerical path following, Colorado State University, 1994.
[3] A. Berman and R. Plemmons, Nonnegative Matrices in the Mathematical Sciences, 2014.
[4] B. Boisvert, R. Pozzo, K. Remington, B. Miller and R. Lipman, Matrix Market repository, http://math.nist.gov/MatrixMarket/.
[5] M. A. Freitag and A. Spence, A Newton-based method for the calculation of the distance to instability, Linear Algebra Appl., 435(12):3189-3205, 2011.
[6] V. Kostić, On general principles of eigenvalue localizations via diagonal dominance, Advances in Computational Mathematics, 41:55-75, 2015.
[7] V. Kostić, A. Miedlar and Lj. Cvetković, An algorithm for computing minimal Geršgorin sets. Numerical linear algebra with applications, 00:1-19, 2015.
[8] J. R. Magnus, On differentiating eigenvalues and eigenvectors, Econometric Theory, 1 : 179-191, 1985.
[9] D. Mehzer and B. Philippe, PAT-a Reliable Path Following Algorithm, Numerical Algorithms, 2002.
[10] S. Milićević, V. Kostić, Lj. Cvetković and A. Miedlar, An implicit algorithm for computing the minimal Geršgorin set, FILOMAT, 13 : 4229-4238, 2019.
[11] R. S. Varga, Geršgorin na His Circles. Springer-Verlag, New York, 2004.
[12] R. S. Varga, Minimal Gershgorin sets, Pac. J. Math., 15:719-729, 1965.
[13] R. S. Varga, Lj. Cvetković and V. Kostić, Approximation of the minimal Geršgorin set of a square complex matrix, ETNA (Electronic Transactions on Numerical Analysis), 40:308-405, 2008.



(a) ***MGSs**

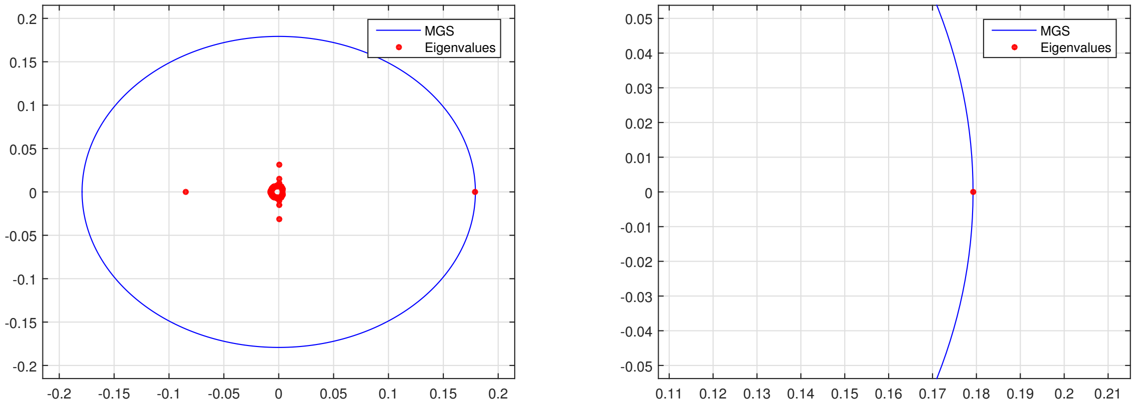


(b) ***MGSp**

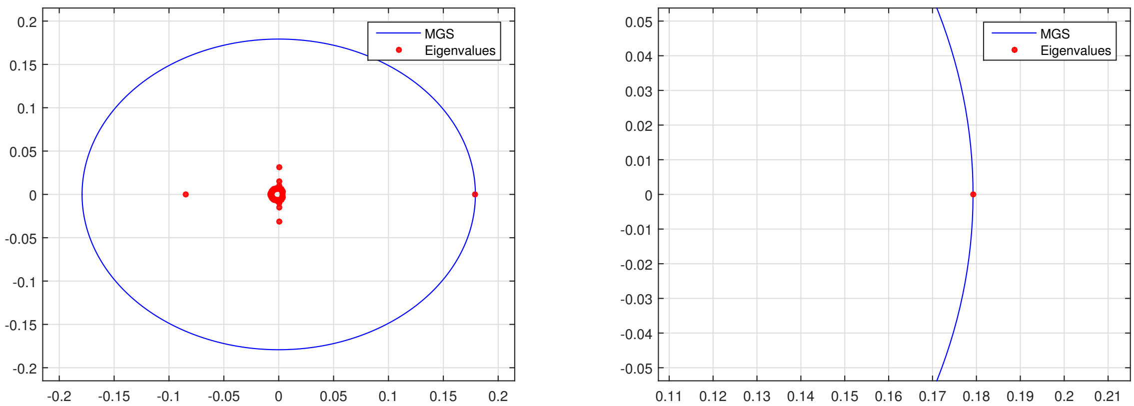


(c) ***MGSt**

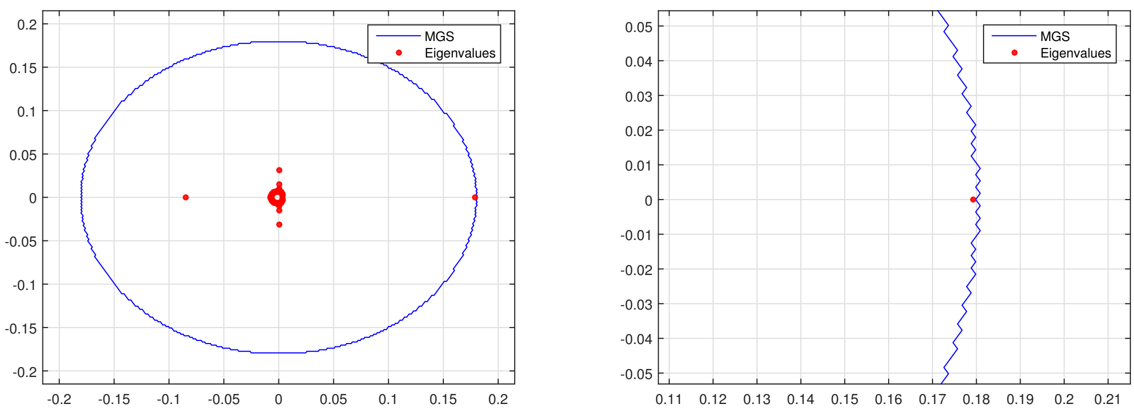
Figure 4: The results of the algorithms for the the cyclic matrix A from Example 5.1: complete plot and plot zoomed around the origin.



(a) ***MGSs**



(b) **iMGSp**



(c) ***MGSt**

Figure 5: The results of the algorithms for the Leslie matrix of a size $n = 70$: complete plot and plot zoomed around the rightmost eigenvalue.

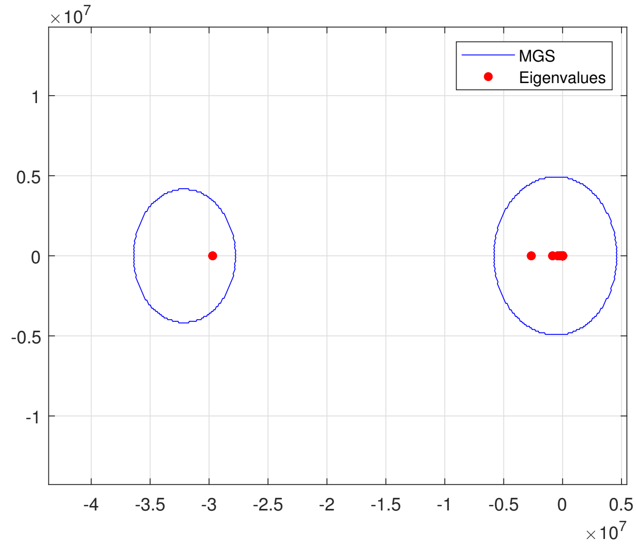
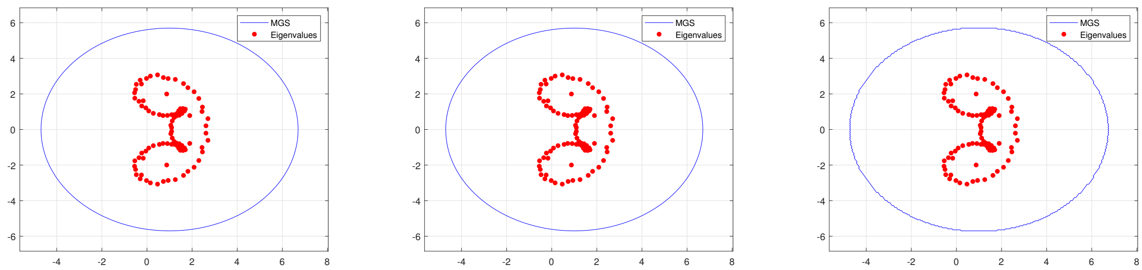


Figure 6: The result of the algorithm **iMGSt** for the Orr-Sommerfeld matrix of a size $n = 1000$.



(a) ***MGSs** (b) ***MGSp** (c) ***MGSt**

Figure 7: The results of the algorithms for the Grcar matrix of a size $n = 100$.

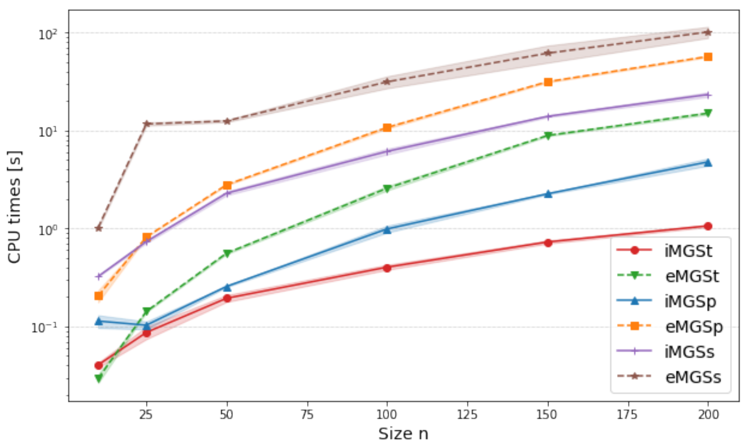


Figure 8: The comparison of CPU times of the algorithms for the Grcar matrices.

Appendix A. Algorithms in psedocode

For reader's connivance we provide pseudocodes of the proposed algorithms.

Appendix A.1. Algorithms based on predictor-corrector method

eMGSp

Input: A, h, tol

Set $\mathcal{D} = \{a_{i_1 i_1}, a_{i_2 i_2}, ..., a_{i_{\tilde{n}} i_{\tilde{n}}}\}$ and initialize $i = 1$;
while $\mathcal{D} \neq \emptyset$ **do**
    Initialize $\xi = \mathcal{D}(1),\ \theta = -\pi,\ \theta_1 = -3\pi$ and $j = 0$;
    Set $\omega =$ **eSearch** $(A, \xi, -\pi, tol)$, and $\omega_{i,0} := \omega$;
    while $\theta - \theta_1 > -\pi$ **do**
        Compute f_x and f_y in $\omega_{i,j}$ by (17) and (18);
        Compute $z_{i,j+1}$ using (40)
        Set $w = z_{i,j+1}$ and compute $f = f(w)$ as the Perron-Frobenius eigenvalue of $Q_A(w)$;
        while $|f| > tol$ **do**
            Compute $f_{xx},\ f_{xy}$ and f_{yy} in w by (19), (20) and (21);
            Compute w by solving the system (41);
            Compute $f = f(w)$ as the Perron-Frobenius eigenvalue of $Q_A(w)$;
        end while
        Update $j \leftarrow j + 1$ and $\omega_{i,j} \leftarrow w$;
        Set $\theta_1 := \theta,\ \theta := -\mathrm{i} \ln \frac{\omega_{i,j} - \xi}{|\omega_{i,j} - \xi|}$;
    end while
    Update $i \leftarrow i + 1$;
    Update $\mathcal{D}$ to exclude all elements inside of the polygon $\{\omega_{i,j}\}_{0 \le j \le m_i}$;
end while

Output: $\{\{\omega_{1,j}\}_{0 \le j \le m_1}, \{\omega_{2,j}\}_{0 \le j \le m_2}, ..., \{\omega_{s,j}\}_{0 \le j \le m_s}\}$

iMGSp

Input: A, h, tol

Set $\mathcal{D} = \{a_{i_1 i_1}, a_{i_2 i_2}, ..., a_{i_{\hat{n}} i_{\hat{n}}}\}$ and initialize $i = 1$;
while $\mathcal{D} \neq \emptyset$ **do**
  Initialize $\xi = \mathcal{D}(1)$, $\theta = -\pi$, $\theta_1 = -3\pi$ and $j = 0$;
  Set $\omega =$ **iSearch** $(A, \xi, -\pi, tol)$, $\omega_{i,0} := \omega$;
  while $\theta - \theta_1 > -\pi$ **do**
    Compute g_x and g_y in $\omega_{i,j}$ by solving the system (32);
    Compute $z_{i,j+1}$ using (40);
    Set $w = z_{i,j+1}$ and compute $g = g(w)$ by solving the system (31);
    while $|g| > tol$ **do**
      Compute g_{xx}, g_{xy} and g_{yy} in w by solving the system (33);
      Compute w by solving the system (41);
      Compute $g = g(w)$ by solving the system (31);
    end while
    Update $j \leftarrow j + 1$ and $\omega_{i,j} \leftarrow w$;
    Set $\theta_1 := \theta$, $\theta := -\mathrm{i} \ln \frac{\omega_{i,j} - \xi}{|\omega_{i,j} - \xi|}$;
  end while
  Update $i \leftarrow i + 1$;
  Update $\mathcal{D}$ to exclude all elements inside of the polygon $\{\omega_{i,j}\}_{0 \le j \le m_i}$;
end while

Output: $\{\{\omega_{1,j}\}_{0\le j\le m_1}, \{\omega_{2,j}\}_{0\le j\le m_2}, ..., \{\omega_{s,j}\}_{0\le j\le m_s}\}$

Appendix A.2. Algorithms based on triangular grid

In the following algorithms, we use the notation:

$$u_{re} = \max_{i\in N}\{Re(a_{ii}) + \sum_{j\in N\setminus\{i\}} |a_{ij}|\},\ l_{re} = \min_{i\in N}\{Re(a_{ii}) - \sum_{j\in N\setminus\{i\}} |a_{ij}|\},$$
$$u_{im} = \max_{i\in N}\{Im(a_{ii}) + \sum_{j\in N\setminus\{i\}} |a_{ij}|\},\ l_{im} = \min_{i\in N}\{Im(a_{ii}) - \sum_{j\in N\setminus\{i\}} |a_{ij}|\}.$$

eMGSt

Input: A, N_t, tol

Set $\tilde{\tau} = \dfrac{2d(A)\sqrt{3}}{3N_t}$, where $d(A) = \max\{u_{re} - l_{re}, u_{im} - l_{im}\}$;
Set $\mathcal{D} = \{a_{i_1 i_1}, a_{i_2 i_2}, ..., a_{i_{\tilde{n}} i_{\tilde{n}}}\}$ and initialize $i = 1$;
while $\mathcal{D} \neq \emptyset$ **do**
  Set $\xi = \mathcal{D}(1)$ and $\theta = -\pi$;
  Run **eSearch**(A, ξ, θ, tol) to compute $\omega_i \in \mathbb{C}$;
  Compute $\tilde{z}_{i,0} = \omega_i - \frac{\tau}{2}$ and $\tilde{z}_{i,1} = \omega_i + \frac{\tau}{2}$;
  Compute $\tilde{z}_{i,2} = \tilde{z}_{i,0} + (\tilde{z}_{i,1} - \tilde{z}_{i,0})e^{\frac{i\pi}{3}}$;
  Set $z_{i,start} = \tilde{z}_{i,0}$ and $z_{i,1} = \tilde{z}_{i,1}$;
  Initialize $j = 2$;
  while $\tilde{z}_{i,2} \neq z_{i,start}$ **do**
    if $f_A(\tilde{z}_{i,2}) < 0$ **then**
      $z_{i,j} = \tilde{z}_{i,2}$;
      $\tilde{z}_{i,0} = \tilde{z}_{i,0}$;
      $\tilde{z}_{i,1} = \tilde{z}_{i,2}$;
      $\tilde{z}_{i,2} = \tilde{z}_{i,0} + (\tilde{z}_{i,1} - \tilde{z}_{i,0})e^{\frac{i\pi}{3}}$;
      Update $j \leftarrow j + 1$;
    else
      $\tilde{z}_{i,0} = \tilde{z}_{i,2}$;
      $\tilde{z}_{i,1} = \tilde{z}_{i,1}$;
      $\tilde{z}_{i,2} = \tilde{z}_{i,1} + (\tilde{z}_{i,0} - \tilde{z}_{i,1})e^{-\frac{i\pi}{3}}$;
    end if
  end while
  Update $i \leftarrow i + 1$;
  Update $\mathcal{D}$ to exclude all elements inside of the polygon $\{z_{i,j}\}_{1\leq j\leq m_i}$;
end while

Output: $\{\{z_{1,j}\}_{1\leq j\leq m_1}, \{z_{2,j}\}_{1\leq j\leq m_2}, ..., \{z_{s,j}\}_{1\leq j\leq m_s}\}$

iMGSt

Input: A, N_t, tol

Set $\tilde{\tau} = \dfrac{2d(A)\sqrt{3}}{3N_t}$, where where $d(A) = \max\{u_{re} - l_{re}, u_{im} - l_{im}\}$;
Set $\mathcal{D} = \{a_{i_1 i_1}, a_{i_2 i_2}, ..., a_{i_{\tilde{n}} i_{\tilde{n}}}\}$ and initialize $i = 1$;
while $\mathcal{D} \neq \emptyset$ **do**
  Set $\xi = \mathcal{D}(1)$ and $\theta = -\pi$;
  Run **iSearch**(A, ξ, θ, tol) to compute $\omega_i \in \mathbb{C}$;
  Compute $\tilde{z}_{i,0} = \omega_i - \frac{\tau}{2}$ and $\tilde{z}_{i,1} = \omega_i + \frac{\tau}{2}$;
  Compute $\tilde{z}_{i,2} = \tilde{z}_{i,0} + (\tilde{z}_{i,1} - \tilde{z}_{i,0})e^{\frac{i\pi}{3}}$;
  Set $z_{i,start} = \tilde{z}_{i,0}$ and $z_{i,1} = \tilde{z}_{i,1}$;
  Initialize $j = 2$;
  while $\tilde{z}_{i,2} \neq z_{i,start}$ **do**
    if $h_A(\tilde{z}_{i,2}) < 0$ **then**
      $z_{i,j} = \tilde{z}_{i,2}$;
      $\tilde{z}_{i,0} = \tilde{z}_{i,0}$;
      $\tilde{z}_{i,1} = \tilde{z}_{i,2}$;
      $\tilde{z}_{i,2} = \tilde{z}_{i,0} + (\tilde{z}_{i,1} - \tilde{z}_{i,0})e^{\frac{i\pi}{3}}$;
      Update $j \leftarrow j + 1$;
    else
      $\tilde{z}_{i,0} = \tilde{z}_{i,2}$;
      $\tilde{z}_{i,1} = \tilde{z}_{i,1}$;
      $\tilde{z}_{i,2} = \tilde{z}_{i,1} + (\tilde{z}_{i,0} - \tilde{z}_{i,1})e^{-\frac{i\pi}{3}}$;
    end if
  end while
  Update $i \leftarrow i + 1$;
  Update $\mathcal{D}$ to exclude all elements inside of the polygon $\{z_{i,j}\}_{1 \leq j \leq m_i}$;
end while

Output: $\{\{z_{1,j}\}_{1 \leq j \leq m_1}, \{z_{2,j}\}_{1 \leq j \leq m_2}, ..., \{z_{s,j}\}_{1 \leq j \leq m_s}\}$