A null-space method based algebraic iterative scheme for saddle point problems

Murat Manguoğlu [1] and Volker Mehrmann [2]

[1] Department of Computer Engineering, Middle East Technical University, Ankara, Turkey

[2] Institut für Mathematik, Technische Universität Berlin, Germany

Sparse Days Meeting, Tolouse, France 2024

Part of this work was done when the first author was funded by Alexander von Humboldt Foundation for a research stay at TU-Berlin.

Outline

Motivation

- Most applications (for example: time-discretization of dissipative Hamiltonian differential equations) naturally give rise to linear systems with coefficient matrices which consist of a symmetric part ($M = M^T$) that is positive semi-definite ($x^T Mx \geq 0, \forall x \in \mathbb{R}^n$) and skew-symmetric parts ($J = -J^T$), often in a blocked structured form (Güdücü et al. 2022)
- We have recently proposed an Approximate Shifted Skew-Symmetrizer (ASSS) preconditioner for systems in which the coefficient matrices are not yet given in this form (most scientists and engineers do that when they form these systems even though most applications come in this form) and also assuming M is just symmetric (not even positive semi-definite) by transforming the coefficient into shifted skew-symmetric form ($\tilde{I} + \tilde{J}$), approximately and obtain a more robust general solver (Manguoğlu and Mehrmann 2021)
- <u>In this work</u>, we propose a new method for solving saddle point problems via the Null-Space Method in which the solution of shifted skew-symmetric systems plays a crucial role

Why shifted skew-symmetric systems?

Given a coefficient matrix in the form: $A = \alpha I + J$ where J is skew-symmetric ($S = -S^T$) and $\alpha \in \mathbb{R}$ is a scalar, we can use a general method (such as GMRES and BiCGStab) or a method that is based on the skew-Lancsoz process with short recurrences (such as Concus, Golub, Widlund (CGW) method and MRS (Jiang 2007; Idema and Vuik 2007)

Some Krylov subspace methods that can be used for solving shifted skew-symmetric systems and their properties (k is the current iteration number and "Opt." stands for the optimality property)

Method	#Matvecs	#Inner prod.	Mem.	Short Recur.	Opt.
GMRES	1	$\mathcal{O}(k)$	$\mathcal{O}(k)$	No	Yes
BiCGStab	2	4	7	Yes	No
CGW	1	2	4	Yes	No
MRS	1	1	5	Yes	Yes

Solving saddle point problems: a new general framework via the null space method

Given a system of equations

$$\begin{bmatrix} A & B \\ -B^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsymmetric and $B \in \mathbb{R}^{n\times m}$ with $m < n$ has full rank.

- Assume we are able to compute easily $Z \in \mathbb{R}^{n\times(n-m)}$ whose columns forms a basis for the null-space of B^T (i.e. $B^T Z = 0$)
- Second block equation is an underdetermined linear least squares problem: $-B^T x = g$ which has infinitely many solutions. Suppose a particular solution is $\hat{x}$ s.t. $-B^T \hat{x} = g$

Saddle point problems: null space method

Then, we can solve an equivalent system

$$\begin{bmatrix} A & B \\ -B^T & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ y \end{bmatrix} = \begin{bmatrix} f - A\hat{x} \\ 0 \end{bmatrix}$$

where $x = \hat{x} + \bar{x}$, solving the equality of the second block-row means finding z such that $\bar{x} = Zz$, then substituting it into the equation in the first block-row,

$$AZz + By = f - A\hat{x}$$

Multiplying both sides of the equality with Z^T, we obtain

$$Z^T AZz = Z^T(f - A\hat{x})$$

Solving the "reduced system" for z , we can recover $\bar{x} = Zz$, then $x = \bar{x} + \hat{x}$, and finally recover y by solving an overdetermined LLS problem.

Saddle point problems: null space method

1: Construct $Z \in \mathbb{R}^{n \times (n-m)}$ whose columns forms a basis for the null-space of B^T
2: Find a particular solution, $\hat{x} \in \mathbb{R}^n$, of $-B^T x = g$
3: Solve $Z^T AZz = Z^T(f - A\hat{x})$
4: Recover $x = \hat{x} + Zz$
5: Solve $By = f - Ax$

Some notes:

- Instead of the exact Z, a sparse approximation can be used (in this case an outer iterative scheme is needed)
- LLS problems in steps 2 and 5 can be solved iteratively
- Linear system in step 3 can be solved iteratively
- An extension of null-space method for saddle point problems with nonzero (2,2)-block is also available (Scott and Tuma 2022)

Saddle point problems: proposed scheme†

† A Preliminary variant with exact null-space basis and exact M-orthogonalization

1: Split $A = M + J$ where $M = \frac{A+A^T}{2}$, and $J = \frac{A-A^T}{2}$

2: Construct $Z \in \mathbb{R}^{n\times(n-m)}$ whose columns form a basis for the null-space of B^T

3: Given Z, obtain a Q (s.t. $Q^TMQ = I$ and $B^TQ = 0$ via MGS procedure)

4: Find a particular solution, $\hat{x} \in \mathbb{R}^n$, of $-B^Tx = g$

5: Solve $Q^TAQz = Q^T(f - A\hat{x})$ via MRS (since $Q^TAQ = I + Q^TJQ$)

6: Recover $x = \hat{x} + Zz$

7: Solve $By = f - Ax$

Computing a sparse approximate basis for the null-space

- There are several possible options for computing a sparse basis for the null-space: Sparse incomplete QR factorization, Sparse incomplete/complete LDU factorization, and Right Oblique Conjugation(Scott and Tuma 2022) , etc.
- We choose, Right Oblique Conjugation (ROC) due to its flexibility. Given $B \in \mathbb{R}^{n\times m}$ (with $rank(B^T) = k \leq m$),

$$B^T V = L$$

  where $V \in \mathbb{R}^{n\times n}$ and $L \in \mathbb{R}^{k\times n}$ is a lower trapezoidal matrix. Then, the rightmost $n-k$ columns of V form a basis for the null-space
- In our case:
  - Since B is sparse, to avoid division by zero (or a small number) we use column pivoting
  - We are not interested in L, so we do not store it
  - We apply numerical dropping and partial conjugation based on the numerical values of the coefficients, and propose a sparse "approximate" ROC (explained in the next slide)

Sparse **Approximate** Right Oblique Conjugation (SAROC)

Input: $B \in \mathbb{R}^{n \times m}$ with $rank(B^T) = k \leq m$, ρ (threshold) and τ (dropping tolerance) ▷ b_i: is the i^{th} column of B

Output: $\tilde{Z} \in \mathbb{R}^{n \times (n-k)}$ ▷ z_i: is the i^{th} column of $\tilde{Z}$

$v_i = e_i$ for $(1 \leq i \leq n)$
for $i = 1, \ldots, m$ **do**
    $\sigma = b_i^T [v_i, v_{i+1}, \ldots, v_n]$ ▷ compute the coefficient: $\sigma \in \mathbb{R}^{n-i+1}$
    $l^* = \arg\max_l (|\sigma_l|)$ ▷ find a good pivot
    $v_i \leftrightarrow v_{i-l^*+1}$ ▷ Swap the corresponding columns of V
    $\sigma_1 \leftrightarrow \sigma_{l^*}$ ▷ Swap the corresponding coefficients
    for $j = i+1, \ldots, n$ **do**
        if $|\sigma_{j-i+1}/\sigma_1| > \rho$ **then**
            $v_j \xleftarrow{|v_j(.)| \geq \tau ||v_j||_2} v_j - (\sigma_{j-i+1}/\sigma_1) v_i$
        end if
    end for
end for
$\tilde{Z} = [v_{k+1}, v_{k+2}, \ldots, v_n]$

Sparse Approximate M-Orthogonalization (SAMO) via MGS

Input: $\tilde{Z} \in \mathbb{R}^{n\times(n-k)}$, $M \in \mathbb{R}^{n\times n}$, w (window size) and τ (dropping tolerance) ▷ z_i: is the i^{th} column of $\tilde{Z}$

Output: $\tilde{Q} \in \mathbb{R}^{n\times(n-k)}$ ▷ q_i: is the i^{th} column of $\tilde{Q}$

```
q_1 = z_1 / sqrt(z_1^T M z_1)
for i = 2, ..., n − k do
    q_i = z_i
    for j = i − w, ..., i − 1 do
        q_i ←[|q_i(.)| ≥ τ||q_i||_2]— q_i − (q_j^T M q_i) q_j
    end for
    q_i = q_i / sqrt(q_i^T M q_i)
end for
```

$$q_1 = z_1/\sqrt{z_1^T M z_1}$$

for $i = 2, \ldots, n-k$ **do**

$\quad q_i = z_i$

$\quad$ **for** $j = i-w, \ldots, i-1$ **do**

$$\quad\quad q_i \xleftarrow{|q_i(.)| \geq \tau ||q_i||_2} q_i - (q_j^T M q_i) q_j$$

$\quad$ **end for**

$$\quad q_i = q_i/\sqrt{q_i^T M q_i}$$

end for

Saddle point problems: proposed scheme

1: Split $A = M + J$ where $M = \frac{A+A^T}{2}$, and $J = \frac{A-A^T}{2}$
2: Construct $\tilde{Z} \in \mathbb{R}^{n\times(n-m)}$ whose columns form an approximate basis for the null-space of B^T via SAROC
3: Given $\tilde{Z}$, obtain a $\tilde{Q}$ (s.t. $\tilde{Q}^T M\tilde{Q} \approx I$ and $B^T\tilde{Q} \approx 0$ via SAMO)
4: Obtain the factorized sparse approximate inverse (FSAI) of $\tilde{Q}^T M\tilde{Q}$ s.t. $\tilde{W}^T\tilde{Q}^T M\tilde{Q}\tilde{W} \approx I$
5: Solve $\begin{bmatrix} A & B \\ -B^T & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via a preconditioned flexible Krylov subspace method (fGMRES(10)) where the systems involving the preconditioner $\mathrm{M}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$ are solved via the null-space method as follows:
   - Find a particular solution ,$\hat{z}_1 \in \mathbb{R}^n$, of $-B^T z_1 = t_2$ via a Krylov subsace method (LSQR)
   - Solve $\tilde{W}^T\tilde{Q}^T A\tilde{Q}\tilde{W}(\tilde{W}^{-1}u) = \tilde{W}^T\tilde{Q}^T(t_1 - A\hat{z}_1)$ via fGMRES(10) with a preconditioner $N = I + \tilde{W}^T\tilde{Q}^T J\tilde{Q}\tilde{W}$ (systems involving the preconditioner are solved via MRS)
   - Recover $z_1 = \hat{z}_1 + \tilde{Z}u$
   - Solve $Bz_2 = t_1 - Az_1$ via LSQR

Factorized Sparse Approximate Inverse (FSAI) of $\tilde{Q}^T M \tilde{Q}$

Input: $\tilde{Q} \in \mathbb{R}^{n\times(n-k)}$, $M \in \mathbb{R}^{n\times n}$, ρ (threshold) and τ (dropping tolerance) ▷ q_i: is the i^{th} column of $\tilde{Q}$

Output: $\tilde{W} \in \mathbb{R}^{n\times(n-k)}$ ▷ w_i: is the i^{th} column of $\tilde{W}$

$w_i = e_i$ for $(1 \le i \le n-k)$

for $k = 1, \ldots, n-k$ **do**

  $[\sigma_k, \sigma_{k+1}, \ldots, \sigma_{n-k}]^T = [w_k, w_{k+1}, \ldots, w_{n-k}]^T(\tilde{Q}^T(Mq_k))$

  for $i = k+1, \ldots, n-k$ **do**

    if $|\sigma_{j-i+1}/\sigma_1| > \rho$ **then**

      $w_i \xleftarrow{|w_i(.)| \ge \tau ||w_i||_2} w_i - (\sigma_i/\sigma_k)w_k$

    end if

  end for

end for

$\sigma_i = 1/\sqrt{\sigma_i}$ for $(1 \le i \le n-k)$

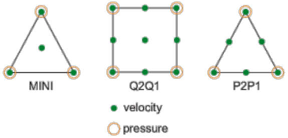
$\tilde{W} = \tilde{W}P$ where P is a diagonal matrix with entries $P_{(i,i)} = \sigma_i$

The parameters of the proposed method

- $\tau_{SAROC}, \rho_{SAROC}, \tau_{SAMO}, w_{SAMO}, \tau_{FSAI}, \rho_{FSAI}$
- stopping tolerances of outer fGMRES, inner fGMRES, LSQR, MRS
- Since they belong to the same iteration layer we consider fGMRES and LSQR stopping tolerances to be the same
- Outer fGMRES stopping tolerance is defined by the user.
- Let us also fix the restart values are fixed at 10, maximum number of iterations are fixed at $10,000$
- **Objective**: We still have 8 parameters to determine. Then, the question is: Given a problem and a target relative residual what are the best parameters? : $\rightarrow$ we propose a machine learning model to achieve this objective

Training Dataset: Lid-driven Cavity Problem

- Linear systems arising in the linearized Navier-Stokes equations are obtained from the IFISS software (Elman et al. 2014). We solve the lid-driven cavity problem with a spacial discretization based on Q_2/Q_1 elements and a uniform grid (steady state solution).



Some finite element discretizations that satisfy "Inf-Sup" condition of mixed finite element discretizations (image source:https://ww2.lacan.upc.edu/huerta/exercises/Incompressible/Incompressible_Ex1.htm)

- The linear systems have the following form:

$$\begin{bmatrix} A & B \\ -B^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

where $A \in \mathbb{R}^{n \times n}$ is nonsymmetric and $B \in \mathbb{R}^{n \times m}$ (discrete gradient) with $m < n$, here u and p corresponds to velocity and pressure components of the solution.

Training Dataset: Lid-driven Cavity Problem

We ran the proposed method for different mesh sizes and Reynold's numbers (1/viscosity) , and varying the parameters (within a reasonable range):

- 16×16 mesh $\rightarrow 281,600$ runs
- 32×32 mesh $\rightarrow 32,076$ runs
- 64×64 mesh $\rightarrow 160$ runs
  - Total $\rightarrow 313,836$ runs. For each run, we have the result of the total number of iterations (its) and the total number of nonzeros of $\tilde{Z}$, $\tilde{Q}$, and $\tilde{W}$ (nnz)
  - Define cost $\leftarrow nnz + its$

$\rightarrow$ **Obtain a performance prediction (cost) model using Regression Learner Toolbox of Matlab 2023b with 11 parameters (8 + user inputs (outer tolerance, mesh size, viscosity)), i.e.**

$$f(outer\_tol, mesh\_size, viscosity, inner\_fgmres\_lsqr\_tol, mrs\_tol, \ldots$$
$$\tau_{SAROC}, \rho_{SAROC}, \tau_{SAMO}, w_{SAMO}, \tau_{FSAI}, \rho_{FSAI}) \rightarrow cost$$

Performance (cost) prediction model training

Tree and Neural Network based models are trained, each model has some additional parameters and they are determined using at most 15 iterations of Bayesian optimization as follows and "5-fold" validated:

Training Results

RMSE (Validation)	10246
R-Squared (Validation)	1.00
MSE (Validation)	1.0498e+08
MAE (Validation)	3736.8
Prediction speed	~1700000 obs/sec
Training time	1.7032e+05 sec
Model size (Compact)	~30 kB

Model Hyperparameters

Preset: Optimizable Neural Network
Number of fully connected layers: 2
Activation: ReLU
Iteration limit: 10000
Standardize data: Yes

Optimized Hyperparameters
Regularization strength (Lambda): 6.6861e-10
First layer size: 72
Second layer size: 29

Hyperparameter Search Range
Regularization strength (Lambda): 3.1864e-11-0.31864
First layer size: 1-300
Second layer size: 1-300

(a) Neural network training results

Training Results

RMSE (Validation)	4601.9
R-Squared (Validation)	1.00
MSE (Validation)	2.1178e+07
MAE (Validation)	1608.8
Prediction speed	~1400000 obs/sec
Training time	60.374 sec
Model size (Compact)	~14 kB

Model Hyperparameters

Preset: Optimizable Tree
Surrogate decision splits: Off

Optimized Hyperparameters
Minimum leaf size: 10

Hyperparameter Search Range
Minimum leaf size: 1-156918

(b) Tree training results

Cost model: optimizing the parameters

- We have a trained black-box cost model
- For a given problem, now we can obtain the best parameters to use for our solver using a global optimization method
- Matlab's optimization toolbox contains: **simulated annealing**, pattern search, surrogate, swarm, Bayesian optimization methods.
- We use simulated annealing, with at most 400 iterations such that, given the cost model f, and input parameters *outer_tol*, *mesh_size* and *viscosity*, find the the remaining input parameters that minimizes the cost
- Search domain: all dropping tolerances and thresholds are between 10^{-1} and 10^{-9}, window size is between 0 and 20, all innner stopping tolerances are between 10^{-1} and the outer tolerance

Results: two models (Outer tol. $= 10^{-5}$)

- **Neural network based model**

Outer fGMRES(10) iterations.

Mesh/Re	16×16	32×32	64×64
20	7	7	20
10	7	7	21
2	7	6	20

$\text{nnz}(\tilde{Z})+\text{nnz}(\tilde{Q})+\text{nnz}(\tilde{W})$

Mesh/Re	16×16	32×32	64×64
20	73832	1544161	24376889
10	73910	1547189	23743927
2	42879	1100810	23837370

Other number of iterations: Average inner fGMRES(10) $3 - 7.8$, MRS $1 - 2.8$ (larger for higher Reynold's number), LSQR $19.1 - 55.4$ (larger for finer mesh)

- **Tree based model**

Outer fGMRES(10) iterations.

Mesh/Re	16×16	32×32	64×64
20	14	260	9
10	15	325	9
2	15	> 1000	9

$\text{nnz}(\tilde{Z})+\text{nnz}(\tilde{Q})+\text{nnz}(\tilde{W})$

Mesh/Re	16×16	32×32	64×64
20	11362	62311	15785138
10	11361	62314	15778891
2	11362	62314	15764788

Other number of iterations: Average inner fGMRES(10) $21.9 - 198.6$ (higher for finer mesh), MRS $1 - 10.4$ (higher for larger Reynold's number and finer mesh), LSQR $5.1 - 39.2$ (larger for finer mesh)

Other saddle point problems

#	Matrix Name	n	nnz
1	reorientation_3	2,513	32,166
2	dynamicSoaringProblem_2	1,591	15,588
3	tumorAntiAngiogenesis_3	410	3,952
4	tumorAntiAngiogenesis_4	455	4,593
5	tumorAntiAngiogenesis_8	490	4,776

- These are optimal control problems, obtained from SuiteSparse Matrix Collection (Davis and Hu 2011) by Vehicle Dynamics & Optimization Lab, UF Anil Rao and Begum Senses, University of Florida (http://vdol.mae.ufl.edu)
- We just use the Neural Network based performance prediction model obtained earlier
- For the mesh size, we use $\sqrt{n}$
- For the Reynold's number, we use 10
- For comparison, we use GMRES(10) with Matlab's ILUTP(τ) preconditioner (after RCM reordering)
- Stopping the iterations when the preconditioned (ℓ_2-norm) relative residual $\leq 10^{-5}$

Summary of results

Number of fGMRES/GMRES iterations

#	Prop. Met.	ILUTP(τ) 10^{-4}	10^{-3}	10^{-2}	10^{-1}
1	2	†	†	†	†
2	10	5	10	†	†
3	3	$4^\star$	$6^\star$	†	†
4	3	$4^\star$	$5^\star$	†	†
5	1	$3^\star$	$7^\star$	†	†

Final true relative residuals ($\ell_2 - norm$)

#	Prop. Met.	ILUTP(τ) 10^{-4}	10^{-3}	10^{-2}	10^{-1}
1	2.09×10^{-6}	†	†	†	†
2	1.55×10^{-5}	4.07×10^{-8}	7.54×10^{-7}	†	†
3	4.50×10^{-6}	$\mathbf{2.24 \times 10^{-1}}$	$\mathbf{1.41 \times 10^{-1}}$	†	†
4	3.06×10^{-6}	$\mathbf{1.45 \times 10^{-1}}$	$\mathbf{1.23 \times 10^{-1}}$	†	†
5	5.28×10^{-6}	$\mathbf{3.31 \times 10^{-1}}$	$\mathbf{6.48 \times 10^{-1}}$	†	†

- $\star$: true relative residual is much larger than the stopping criterion
- †: a zero pivot is encountered during the incomplete LU fact.

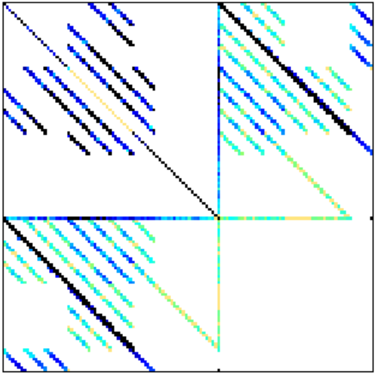
Summary of results

- We also report the total number of nonzeros for the proposed method (i.e. $nnz(\tilde{Z}) + nnz(\tilde{Q}) + nnz(\tilde{W})$) and ILUTP($\tau$) (i.e. $nnz(L) + nnz(U)$)

Number of nonzeros

#	Prop. Met.	ILUTP(τ) 10^{-4}	10^{-3}	10^{-2}	10^{-1}
1	3,961	†	†	†	†
2	29,718	265,734	144,674	†	†
3	584	$51,408^{\star}$	$34,241^{\star}$	†	†
4	629	$50,116^{\star}$	$36,094^{\star}$	†	†
5	665	$57,502^{\star}$	$39,290^{\star}$	†	†

A closer look: reorientation_3 (#1)



Spy plot (obtained from SuiteSparse matrix collection, colorized based on the magnitudes of nonzeros.)

A closer look: reorientation_3 (#1)

Parameters of the proposed solver obtained by simulated annealing using the neural network based cost model:

- $\tau_{SAROC} = 4.1 \times 10^{-3}$
- $\rho_{SAROC} = 9.5 \times 10^{-3}$
- $\tau_{SAMO} = 8.42 \times 10^{-4}$
- $w_{SAMO} = 16$
- $\tau_{FSAI} = 2.61 \times 10^{-5}$
- $\rho_{FSAI} = 3.75 \times 10^{-4}$
- inner fGMRES/LSQR stop. tol. 1.53×10^{-5}
- MRS stop. tol. 7.38×10^{-2}

Resulting in:

- 2 outer fGMRES(10) iterations
- 1 average MRS iterations
- 2 average inner fGMRES(10) iterations
- **763** average inner LSQR iterations

A closer look: reorientation_3 (#1) - improvements

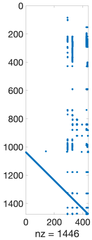
Let us do some fine tuning to improve the results:

1. Inner fGMRES/LSQR stop. tol. $1.53 \times 10^{-5} \rightarrow 10^{-3}$, the resulting iteration numbers are:
   - 2 outer fGMRES(10) iterations (no change)
   - 1 average MRS iterations (no change)
   - 1 average inner fGMRES(10) iterations (previously 2)
   - **212.5** average inner LSQR iterations (previously 763)

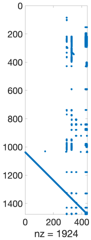
   with a final true relative residual of 9.6×10^{-6}
2. Inner fGMRES/LSQR stop. tol. $1.53 \times 10^{-5} \rightarrow 5 \times 10^{-3}$, the resulting iteration numbers are:
   - 3 outer fGMRES(10) iterations (previously 2)
   - 1 average MRS iterations (no change)
   - 1 average inner fGMRES(10) iterations (previously 2)
   - **70.17** average inner LSQR iterations (previously 763!)

   with a final true relative residual of 4.9×10^{-6}

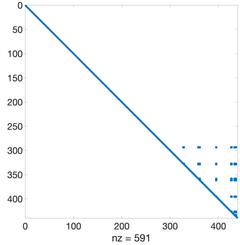
A closer look: reorientation_3 (#1) - spy plots



(a) $\tilde{Z}$: null-space basis.



(b) $\tilde{Q}$: M-orthogonalized null-space basis.



(c) $\tilde{W}$: FSAI preconditiner.

Conclusions and future work

- A robust iterative scheme based on the null-space method is proposed and within the proposed method:
  1. An approximate method for computing a sparse basis for the null-space of B^T
  2. An approximate method for computing sparse M-orthoganization based on the MGS procedure
  3. An implicit FSAI

  are proposed/used
- Furthermore, we also propose and train a neural network based performance prediction model for the proposed iterative scheme and train it on the 2D lid-driven cavity problem
- Use the trained model to predict the best parameters of the proposed scheme for solving lid-driven cavity problem as well as problems from other application domains
- As future work: the neural network model trained with other applications (such as in optimization where there are many saddle point problems that are solved) might be interesting

Thank you !