UPPSALA UNIVERSITET

# Learning preconditioners for interior point methods

**Aleix Nieto**

**Daiyuan Xu**

**Jonathan Franklin**

January 12, 2024

# CONTENTS

UPPSALA UNIVERSITET

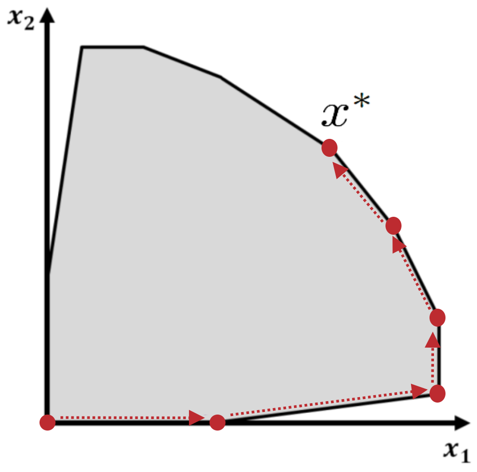
# Introduction

UPPSALA UNIVERSITET

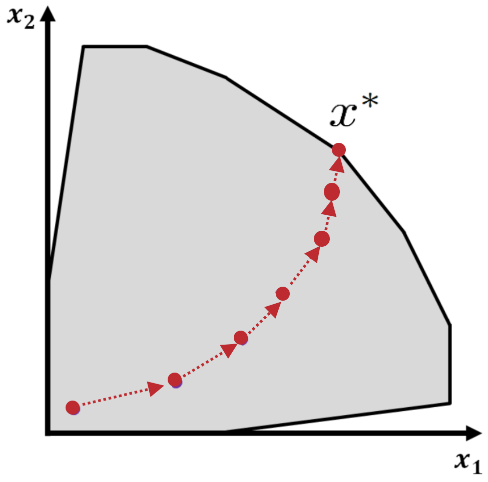
**Linear optimization** problems are ubiquitous in research and engineering

$$\min_x \; c^T x \;\; \text{s.t.} \;\; Ax = b, \; x \geq 0$$

Simplex method




Interior point method

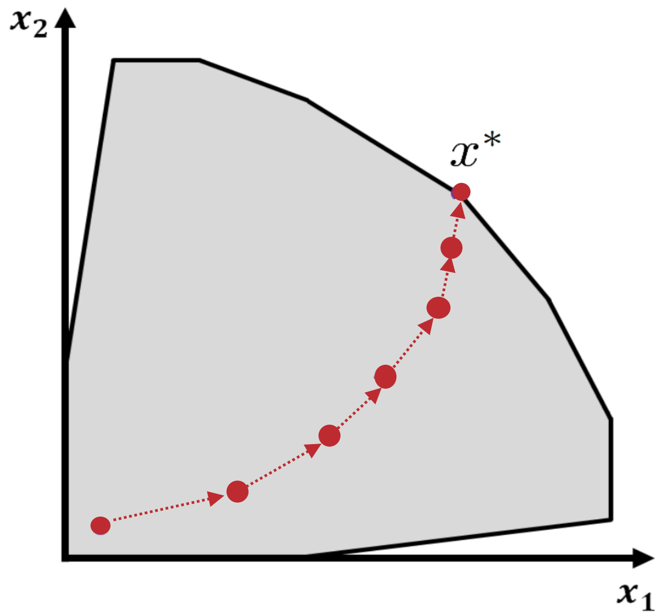


# Introduction



- IPMs initiate from a **feasible point** and determine a **search direction by solving a linear equation system** derived from the Karush-Kuhn-Tucker (KKT) optimality conditions

- The **computational bottleneck** in IPMs lies in solving this linear equation system. While iterative methods are common for large-scale problems, their efficiency often hinges on the availability of a **good preconditioner**



# General linear programming model



A general presentation of LP(linear programming) model is given as follows:

- A maximum/minimum linear objective function:

$$\min_{x} \; c^T x = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

- Several linear constraints(≤, ≥ or =):

$$\begin{aligned} a_{i,1}x_1 + a_{i,2}x_2 + \cdots + a_{i,n}x_n &= b_i \\ a_{j,1}x_1 + a_{j,2}x_2 + \cdots + a_{j,n}x_n &\leq b_j \\ a_{k,1}x_1 + a_{k,2}x_2 + \cdots + a_{k,n}x_n &\geq b_k \end{aligned}$$

# *Network flow (maximum flow)*

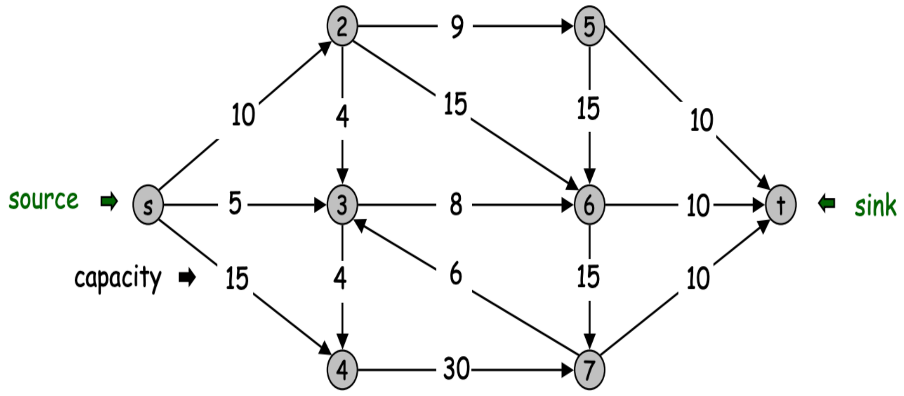


**Abstract of a network in a maximum flow problem**

- Directed Graph with vertex set (**V**) and edge set (**E**)
- Capacity on edges
- Several Sources(***s***)
- Several Sinks(***t***)

**Goal: Find the maximum flow from *s* to *t***

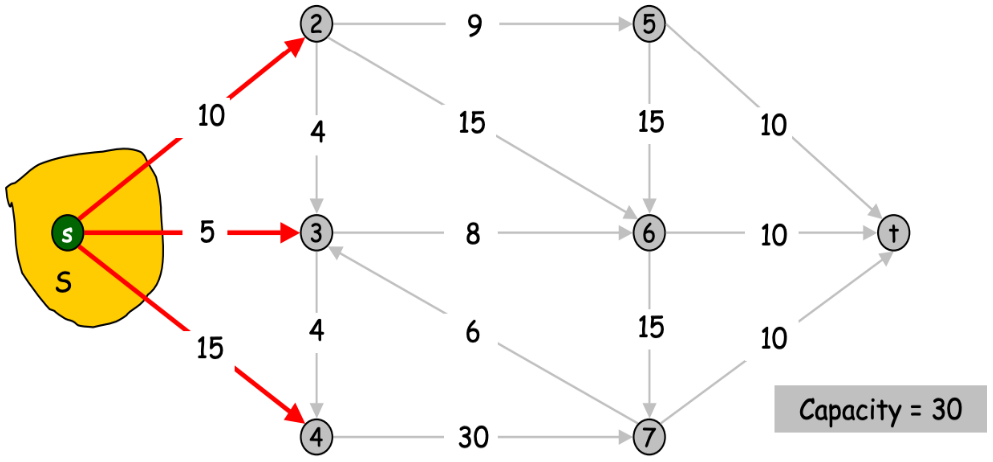
- Equalize inflow and outflow at every intermediate vertex
- Maximize flow sent from ***s*** to ***t***



# *LP for the dual problem (minimum cut)*



- A **cut** is a node partition **(S, T)** such that ***s*** is in **S** and ***t*** is in **T**
- *capacity**(S, T)*** = sum of weights of edges leaving **S**
- **Goal: Find a cut with the minimum capacity**



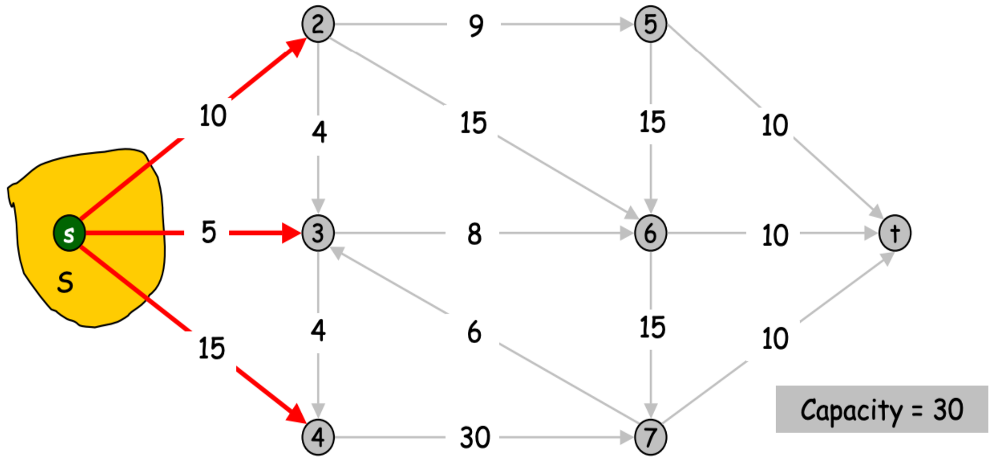

$$
\begin{aligned}
&\text{minimize} \quad \sum_{(u,v)\in E} c(u,v) y_{u,v}, \\
&\text{subject to} \\
&\qquad y_v + y_{s,v} \geq 1, && \forall v : (s,v) \in E. \\
&\qquad y_v - y_u + y_{u,v} \geq 0, && \forall (u,v) \in E, u \neq s, v \neq t. \\
&\qquad -y_u + y_{u,t} \geq 0, && \forall u : (u,t) \in E. \\
&\qquad y_{u,v} \geq 0, && \forall (u,v) \in E.
\end{aligned}
$$

# Standard form



$$\min_x \; c^T x \;\; \text{s.t.} \;\; Ax = b, \; x \geq 0$$

- For all greater equal constraints, minus a slack variable $r_{s,v}$
- For all unconstrained variables $y_u$, rewrite them as $a_u - b_u$




$$\begin{aligned}
\min \quad & \sum_{(u,v)\in E} c(u,v) y_{u,v}, \\
\text{subject to} \quad & \\
& a_v - b_v + y_{s,v} - r_{s,v} = 1, && \forall v : (s,v) \in E. \\
& a_v - b_v - a_u + b_u + y_{u,v} - r_{s,v} = 0, && \forall (u,v) \in E, u \neq s, v \neq t. \\
& -a_u + b_u + y_{u,t} - r_{u,t} = 0, && \forall u : (u,t) \in E. \\
& y_{u,v}, r_{u,v} \geq 0, && \forall (u,v) \in E. \\
& a_u, b_u \geq 0, && \forall u \in V, u \neq s,t.
\end{aligned}$$

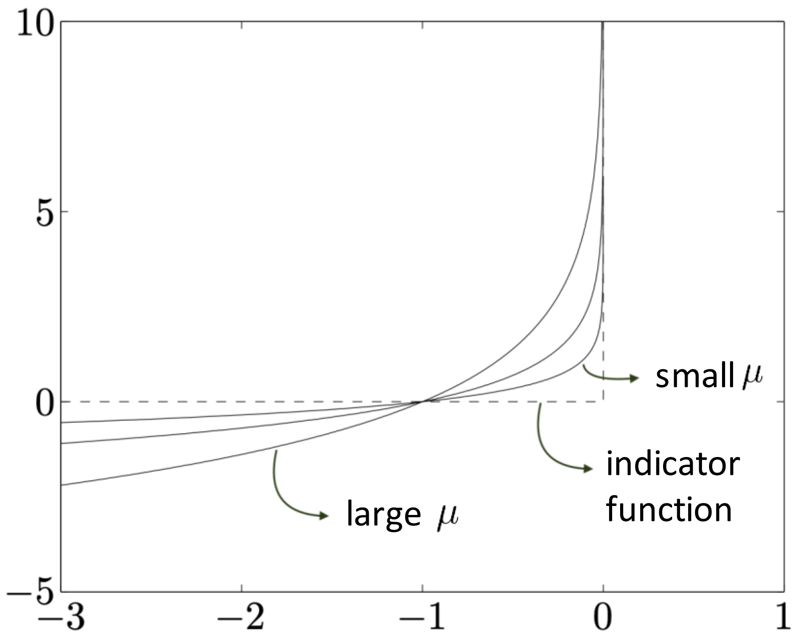
# *Interior point method (IPM) - log barrier function*

UPPSALA UNIVERSITET

$$\left.\begin{aligned} &\min_x \; c^T x \;\; \text{s.t.} \;\; Ax = b,\; x \geq 0 \\ &\mathcal{S} = \{x \in \mathbb{R}^n | Ax = b, x > 0\} \end{aligned}\right\} \quad \min_{x \in \mathcal{S}} \; c^T x - \mu \sum_{i=1}^{n} I_{x \geq 0}(x)$$

We approximate the indicator function using the log barrier function

$$\min_{x \in \mathcal{S}} \; c^T x - \mu \sum_{i=1}^{n} \log(x_i)$$

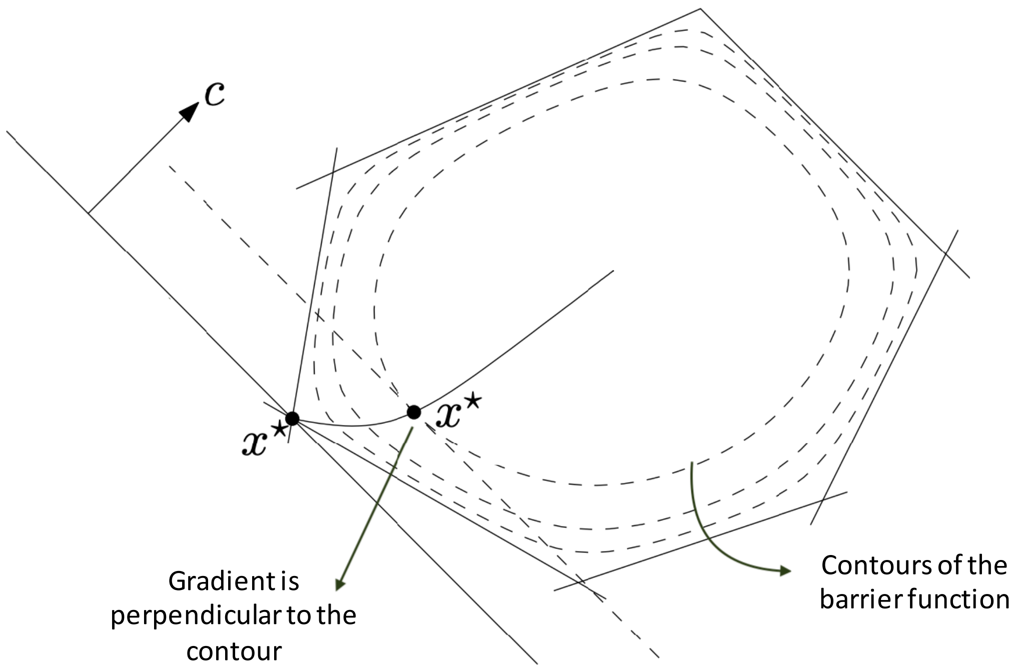


# Interior point method (IPM) - central path

UPPSALA UNIVERSITET

$$x(\mu) = \arg\min_{x \in \mathcal{S}} \; c^T x - \mu \sum_{i=1}^{n} \log(x_i)$$

- The primal central path is the curve, parameterized by $\mu$, described by $x(\mu)$
- When $\mu_k \longrightarrow 0$, the trajectory followed by points $x(\mu)$ constitute the **central path**). All limit points of $\{x(\mu_k)\}$ are solutions of the original LP problem
- When $\mu_k \longrightarrow \infty$ recover the original LP



# *Interior point method (IPM) - KKT conditions*

UPPSALA UNIVERSITET

**Lagrangian function:**

$$\mathcal{L}(x, \lambda, s) = c^T x + \lambda^T (b - Ax) - s^T x = (c - A^T \lambda - s)^T x + \lambda^T b$$

**KKT optimality conditions:**

$$\begin{aligned} Ax - b &= 0 && \text{(primal constraint)} \\ A^T \lambda + s &= c && \text{(dual constraint)} \\ XSe &= \mu e && \text{(complementarity constraint)} \\ (x, s) &> 0 && \text{(primal constraint and dual constraints)} \end{aligned}$$

The primal-dual central path is the curve described by *(x(μ), λ*(μ), *s(μ)*) where *(x(μ), λ(μ), s(μ))* solves the KKT conditions

# Interior point method (IPM) - primal–dual algorithm

UPPSALA UNIVERSITET

**Algorithm 1** Primal-dual method

1: **Input:** : System of linear equations $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$
2: **Output:** Solution to the linear equation system $x_\star$
3: Initialize starting guess $(x^{(0)}, \lambda^{(0)}, s^{(0)})$, with $(x^{(0)}, s^{(0)}) > 0$ a strictly feasible point
4: $\eta^{(0)} \leftarrow (x^{(0)})^T s^{(0)}$; $\sigma \in (0,1)$
5: **for** $k = 0, 1, \ldots,$ until $\eta^{(k+1)} \leq \delta$ and $\delta \leq \left(\|r_{\text{primal}}\|_2^2 + \|r_{\text{dual}}\|_2^2\right)^{\frac{1}{2}}$ **do**
6: $\quad \mu \leftarrow \sigma\eta^{(k)}/n$
7: $\longrightarrow$

Solve $\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \nabla x \\ \nabla \lambda \\ \nabla s \end{bmatrix} = -r(x,\lambda,s) = -\begin{bmatrix} Ax - b \\ A^T\lambda + s - c \\ -XSe + \mu e \end{bmatrix}$

with $(x, \lambda, s) = (x^{(k)}, \lambda^{(k)}, s^{(k)})$

8: $\quad$ Update $y^{(k+1)} = y^{(k)} + \alpha_k \nabla y$.
9: $\quad \eta^{(k+1)} \leftarrow (x^{(k+1)})^T s^{(k+1)}$; $k \leftarrow k+1$
10: **end for**

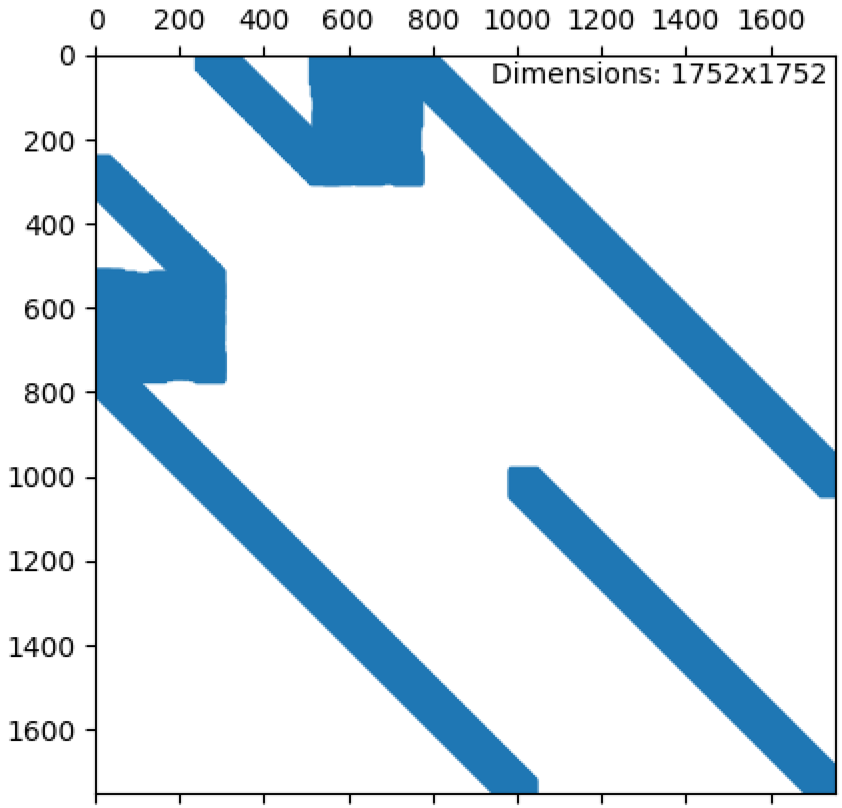
We want to apply **preconditioners** to this linear system

# *Solvers*



- Matrix properties (non spd, non diagonally dominant)
- Direct solver with LU decomposition
- Non-feasible solvers: Conjugate Gradient
- Iterative solvers tried: GMRES, BICGSTAB, TFQMR
- Difference in solve times

***Matrix sparsity pattern***



# Preconditioning



**Purpose:** Improve the spectral properties of the equation system $\boldsymbol{MAx} = \boldsymbol{Mb}$ instead (for a suitable matrix $\boldsymbol{M}$) and thus obtaining faster convergence

- Preconditioners improve solver efficiency by **approximating the inverse** of matrix ***A*** with the matrix ***M***
- **Depends on matrix properties:** Symmetry, positive definiteness, diagonal-dominance, size, sparsity
- **Types of preconditioners:** Jacobi, ILU, ILUT, spectral shift
- Graph neural network **Neuralif** to predict a suitable preconditioner

# *Experiments*

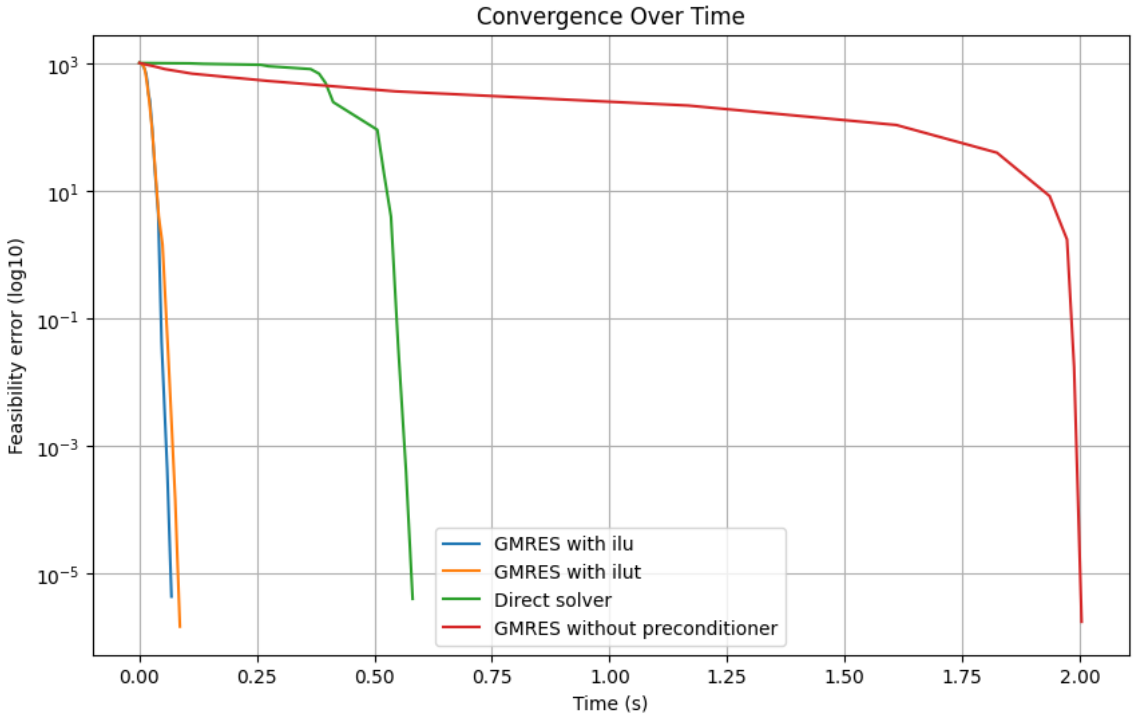


- Tested on different graph structures (varying nodes, edges, sources and sinks)
- Tested direct solver
- Tested the iterative solvers and the performance of each preconditioner
- Tried Neuralif on our matrices
- Other optimizations, hyperparameter tuning, matrix formats, etc.

# Results

UPPSALA UNIVERSITET

| Solver | Preconditioner | $(\|r_{\text{primal}}\|_2^2 + \|r_{\text{dual}}\|_2^2)^{\frac{1}{2}}$ | P-time ↓ | PD-time (iter.) ↓ | Failure rate ↓ |
|---|---|---|---|---|---|
| Direct | None | 2.45e−6 | - | 0.58 (14.53) | 0.0 |
| GMRES | None | 2.29e−6 | - | 14.67 (15.52) | 0.03 |
| GMRES | ILU | 2.61e−6 | 7.72e−6 | 0.25 (14.46) | 0.0 |
| GMRES | ILUT | 2.64e−6 | 6.06e−6 | 0.20 (14.57) | 0.2 |



# Conclusions



- Direct solver was faster than iterative without preconditioners
- Preconditioners improved the convergence time of iterative solvers (GMRES)
- ILU preconditioner reduced the failure rate of GMRES
- Some preconditioners are very unsuitable based on matrix properties
- **Numerical challenges** as iterative solvers approaches the solution

# *Future work*



- Transform matrices into spd form
- More testing needed for larger problems
- Optimizing parameters