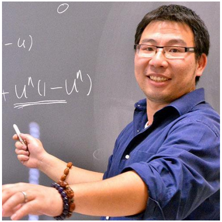
An Efficient and Effective FEM Solver for Diffusion Equation with Strong Anisotropy

D. Green, X. Hu, J. Lore, L. Mu, and M. Stowell

Department of Mathematics
University of Georgia, Athens

Dec. 13th, 2022, FEM@LLNL Seminar
Virtual Seminar

Collaborators

- David Green, Oak Ridge National Laboratory
- Xiaozhe Hu, Tufts University
- Jeremy Lore, Oak Ridge National Laboratory
- Mark L. Stowell, Lawrence Livermore National Laboratory

This work is partially supported by RF SciDAC-4 project from DOE.

Outline

Figure: Major Fusion Breakthrough @ LLNL

- https://www.llnl.gov/news/national-ignition-facility-achieves-fusion-ignition

Figure: Unlimited Energy and Amazing Machine: Video from ITER

- References (Iter Website)
- https://www.iter.org/

Motivation and Background

- Magnetically confined plasma applications for which anisotropy is generated by the magnetic field.
- The anisotropy ratio between $D_{\parallel}/D_{\perp}$ can range from 10^6 (boundary) to 10^{12} (core region).
- Computational Challenges:
  - **Numerical Pollution** in the perpendicular diffusion or transport, due to the fast parallel dynamics

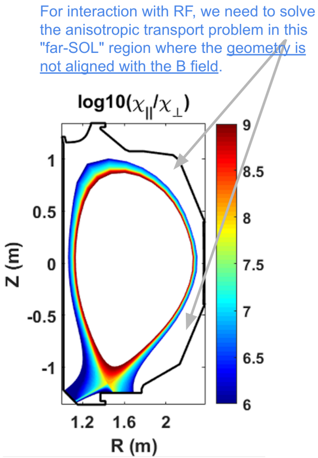


Figure: Demonstration of Anisotropy

Motivation and Background

- Magnetically confined plasma applications for which anisotropy is generated by the magnetic field.
- The anisotropy ratio between $D_{\|}/D_{\perp}$ can range from 10^6 (boundary) to 10^{12} (core region).
- Computational Challenges:
  - **Numerical Pollution** in the perpendicular diffusion or transport, due to the fast parallel dynamics
  - Traditional approach is a field aligned mesh (e.g., SOLPS / UEDGE) which avoids the pollution issue

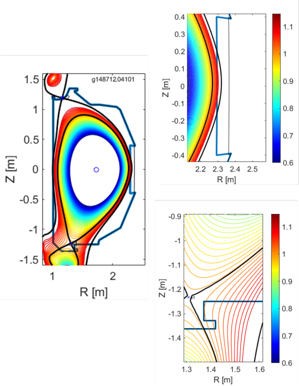


Figure: Demonstration of Anisotropy

Motivation and Background

- Magnetically confined plasma applications for which anisotropy is generated by the magnetic field.
- The anisotropy ratio between $D_{\|}/D_{\perp}$ can range from 10^{6} (boundary) to 10^{12} (core region).
- Computational Challenges:
  - **Numerical Pollution** in the perpendicular diffusion or transport, due to the fast parallel dynamics
  - Traditional approach is a field aligned mesh (e.g., SOLPS / UEDGE) which avoids the pollution issue
  - **For our far-SOL use case where we need non-field aligned coordinates to handle the high geometric fidelity in the boundary**

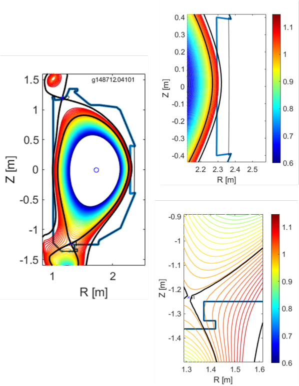
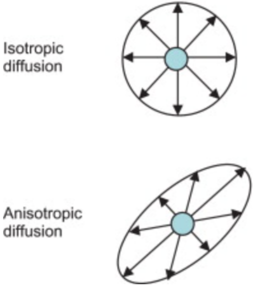
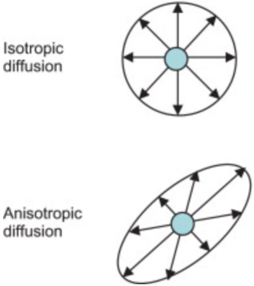


Figure: Demonstration of Anisotropy

Literature Review: Spatial Discretization

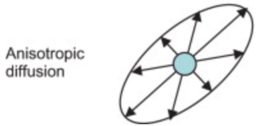


Literature Review: Spatial Discretization



- Aligned Mesh
  - FDM with flux algined coordinates, [Dudson 2009, van Es 2014]
  - Finite Volume Method [Crouseilles 2015]
  - FEM with anisotropy adaptive mesh [Li 2010]
- Non-aligned Mesh
  - FDM with modified interpolation schemes [Soler2020]
  - FVM with high-order scheme [Holleman 2013]
  - **FEM with high-order scheme** [Gunter 2007, Held 2016, Giorgiani 2020]
  - Spectral element [Meier 2010]

Literature Review: Spatial Discretization



- Aligned Mesh
  - FDM with flux algined coordinates, [Dudson 2009, van Es 2014]
  - Finite Volume Method [Crouseilles 2015]
  - FEM with anisotropy adaptive mesh [Li 2010]
- Non-aligned Mesh
  - FDM with modified interpolation schemes [Soler2020]
  - FVM with high-order scheme [Holleman 2013]
  - **FEM with high-order scheme** [Gunter 2007, Held 2016, Giorgiani 2020]
  - Spectral element [Meier 2010]
- Fast Solver
  - Schwarz [Antonietti 2007], Multilevel [Dobrev 2006], Multigrid [Brenner 2005] Methods
  - Subspace Correction [Xu 1992] and **Auxiliary Space Preconditioning** [S. Nepomnyaschikh 1992, Xu 1996]

Goal of this project

We will address the issues in the Diffusion Equations with **Strong Anisotropy** ($D_{\|}/D_{\perp} \geq$1E6):

- Accuracy on the **Non-aligned Mesh**
- Efficiency of the **Linear Solver**

Goal of this project

We will address the issues in the Diffusion Equations with **Strong Anisotropy** ($D_{||}/D_{\perp} \geq$1E6):

- Accuracy on the **Non-aligned Mesh**
    - By High-order Scheme
- Efficiency of the **Linear Solver**

Goal of this project

We will address the issues in the Diffusion Equations with **Strong Anisotropy** ($D_{\parallel}/D_{\perp} \geq$1E6):

- Accuracy on the **Non-aligned Mesh**
  - By High-order Scheme
- Efficiency of the **Linear Solver**
  - By Auxiliary Space Pre-conditioner

Problem Setting

We consider the following steady state anisotropic diffusion equation:

$$-\nabla \cdot (\mathbb{D}\nabla u) = f, \text{ in } \Omega, \quad (1)$$

$$u = 0, \text{ on } \partial\Omega, \quad (2)$$

where the diffusion coefficient tensor is given by

$$\mathbb{D} = \begin{pmatrix} b_1 & -b_2 \\ b_2 & b_1 \end{pmatrix} \begin{pmatrix} D_{\parallel} & 0 \\ 0 & D_{\perp} \end{pmatrix} \begin{pmatrix} b_1 & b_2 \\ -b_2 & b_1 \end{pmatrix}.$$

The direction of the anisotropy, or the magnetic field, is given by a unit vector $\mathbf{b} = (b_1, b_2)^\top$. Here $D_{\parallel}$ and $D_{\perp}$ represent the parallel and the perpendicular diffusion coefficient. For example, $D_{\perp} = 1$.

Finite Element Space

Let finite element spaces be

$$V_{\text{CG}} = \{v \in H_0^1(\Omega) |\ v|_T \in P_k(T), \forall T \in \mathcal{T}_h\}, \quad (3)$$

$$V_{\text{DG}} = \{v \in L^2(\Omega) |\ v|_T \in P_k(T), \forall T \in \mathcal{T}_h\}. \quad (4)$$

where $P_k(T)$ $(k \geq 1)$ denotes the polynomials with degree $\leq k$.

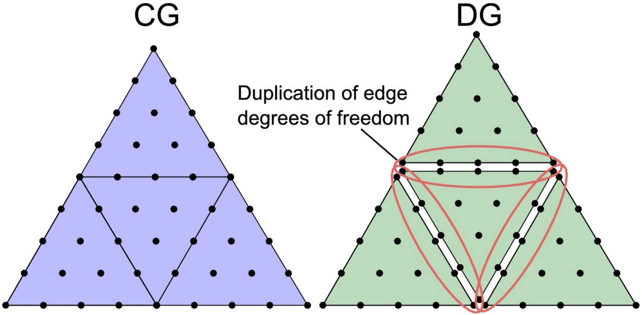


Figure: DG has more DoFs compared to CG on the same mesh.

Continuous Galerkin FEMs

The H^1-FEM is to find the numerical solution $u_h \in V_{\mathsf{CG}}$, such that

$$A_{\mathsf{CG}}(u_h, v) = \sum_{T \in \mathcal{T}_h} \int_T f v dT, \ \forall v \in V_{\mathsf{CG}}, \tag{5}$$

where the bilinear form

$$A_{\mathsf{CG}}(w, v) = \sum_{T \in \mathcal{T}_h} \int_T \mathbb{D} \nabla w \cdot \nabla v dT. \tag{6}$$

Continuous Galerkin FEMs

The H^1-FEM is to find the numerical solution $u_h \in V_{\mathsf{CG}}$, such that

$$A_{\mathsf{CG}}(u_h, v) = \sum_{T \in \mathcal{T}_h} \int_T f v dT, \ \forall v \in V_{\mathsf{CG}}, \tag{5}$$

where the bilinear form

$$A_{\mathsf{CG}}(w, v) = \sum_{T \in \mathcal{T}_h} \int_T \mathbb{D} \nabla w \cdot \nabla v dT. \tag{6}$$

Remark

- **Fewest** Degrees of Freedom (DoFs).
- **No** conservation preserving property
- The CG scheme will be **only** used in constructing the preconditioner.

Discontinuous Galerkin FEMs

The interior penalty discontinuous Galerkin (IPDG) numerical algorithm is to find $u_h \in V_{\mathsf{DG}}$ such that

$$A_{\mathsf{DG}}(u_h, v) = \sum_{T\in\mathcal{T}_h} \int_T f v dT, \ \forall v \in V_{\mathsf{DG}}, \text{ where} \tag{7}$$

$$\begin{aligned} A_{\mathsf{DG}}(u_h, v) = & \sum_{T\in\mathcal{T}_h} \int_T \mathbb{D}\nabla u_h \cdot \nabla v dT - \sum_{e\in\mathcal{E}_h} \int_e \{\!\{\mathbb{D}\nabla u_h \cdot \mathbf{n}\}\!\} [\![v]\!] ds \\ - & \ \beta \sum_{e\in\mathcal{E}_h} \int_e \{\!\{\mathbb{D}\nabla v \cdot \mathbf{n}\}\!\} [\![u_h]\!] ds + \sum_{e\in\mathcal{E}_h} \int_e \alpha [\![u_h]\!] [\![v]\!] ds. \end{aligned}$$

Remark:

- $\beta = 1$ - symmetric IPDG scheme; $\beta = -1$ - non-symmetric IPDG scheme
- For $\beta = 1$, α has to be chosen to ensure stability.
- **Features: Flexibility and Conservation**

Choice of Penalty Parameter

We shall only focus on the symmetric IPDG scheme

- Penalty parameter α can be chosen as

$$\alpha_1 = \frac{4k(k+1)D_{\parallel}}{h}, \ \alpha_2 = \frac{4k(k+1)}{h}\mathbf{n}\cdot(\mathbb{D}\mathbf{n}). \tag{8}$$

- Error Estimate

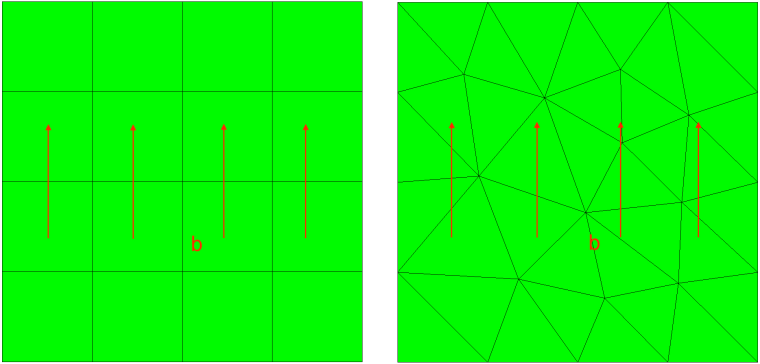
$$\|u-u_h\| \quad \approx \quad \frac{\lambda_{\max}}{\lambda_{\min}} h^{k+1}\|u\|_{k+1} \tag{9}$$

Remark:

- For our case, $\lambda_{\text{min}} = 1 = D_{\perp}$ with varying values in $\lambda_{\text{max}} = D_{\parallel}$.
- The big ratio of anisotropy $\dfrac{\lambda_{\max}}{\lambda_{\min}}$ may destroy the approximation.
- High order scheme with larger k will HELP!
- We may have **bad** condition number.

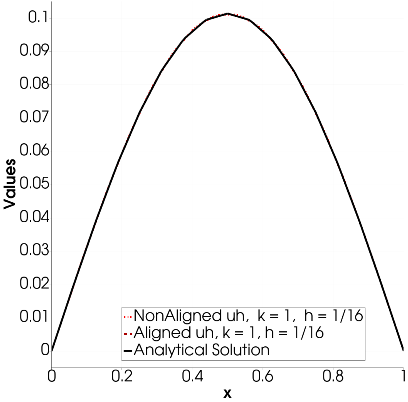
Accuracy Test 1

Set $f = \sin \pi x$ and Dirichelet BC at $x = 1$. Exact solution $u = \frac{1}{\pi^2} \sin \pi x$

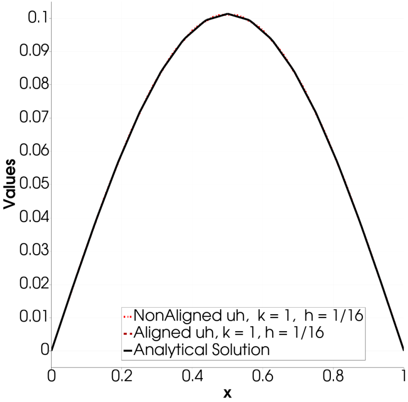


(a) Aligned Mesh (b) Non-aligned Mesh

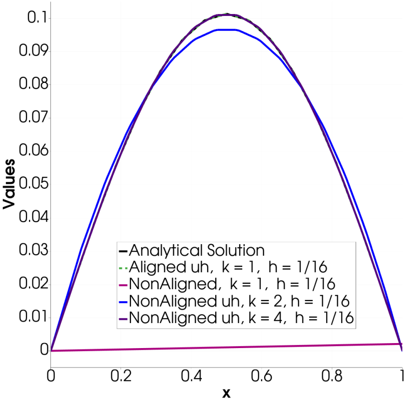
Results on (Non-)Aligned with $h = 1/16$



Results on (Non-)Aligned with $h = 1/16$



(a) $D_{||} = 1.0$

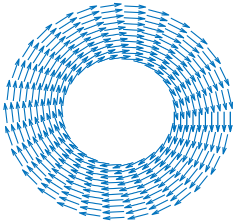


(b) $D_{||} = 10^9$

Accuracy Test 2: Annulus Test

Let Ω be an annulus with R = 1, r = 0.5 and

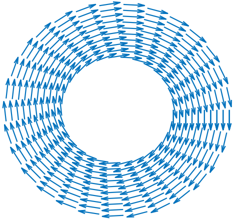
$$
\begin{aligned}
u &= \sqrt{\frac{3}{4r}}\sin(2\pi r - \pi),\\
b_1 &= \frac{y}{r},\ b_2 = -\frac{x}{r},\ \mathbf{b} = (\mathbf{b_1}, \mathbf{b_2})^\top\\
f &= \sqrt{\frac{3}{4r^5}}(4\pi^2 r^2 - \frac{1}{4})\sin(2\pi r - \pi).
\end{aligned}
$$



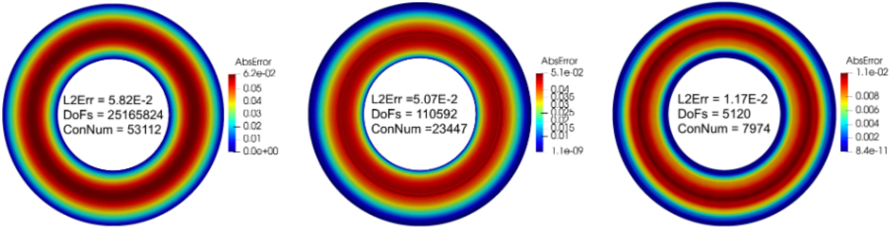
Accuracy Test 2: Annulus Test

Let Ω be an annulus with R = 1, r = 0.5 and

$$
\begin{aligned}
u &= \sqrt{\frac{3}{4r}}\sin(2\pi r - \pi), \\
b_1 &= \frac{y}{r},\ b_2 = -\frac{x}{r},\ \mathbf{b} = (\mathbf{b_1}, \mathbf{b_2})^\top \\
f &= \sqrt{\frac{3}{4r^5}}(4\pi^2 r^2 - \frac{1}{4})\sin(2\pi r - \pi).
\end{aligned}
$$



Error Plot $|u - u_h|$ for $D_{||}$ = 1E6 on:
(a) $N_r = 1024, k = 1$; (b) $N_r = 48, k = 2$; (c) $N_r = 8, k = 3$.

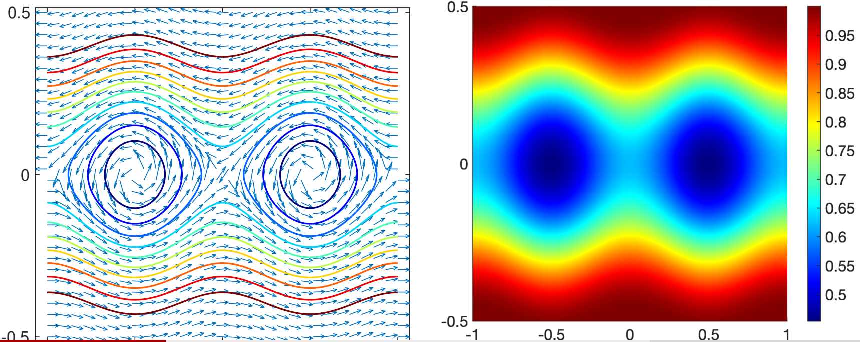


Accuracy Test 3: Two Magnetic Islands

Let $\Omega = [-1,1] \times [-0.5, 0.5]$ and the exact solution be

$$u = \cos\left(\frac{1}{10}\cos(2\pi(x - 3/2)) + \cos(\pi y)\right). \quad (10)$$

$$\mathbf{b} = \frac{\mathbf{B}}{|\mathbf{B}|}, \ \mathbf{B} = \begin{pmatrix} -\pi \sin(\pi y) \\ \frac{2\pi}{10}\sin(2\pi(x - 3/2)) \end{pmatrix}. \quad (11)$$



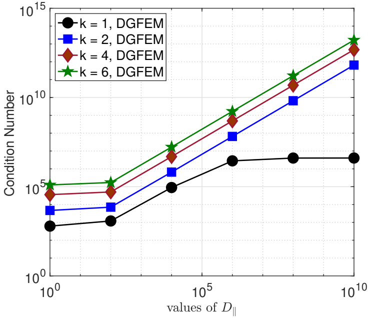
$1/h$	$D_\parallel = 1E1$ $\|u-u_h\|$	order	$D_\parallel = 1E2$ $\|u-u_h\|$	order	$D_\parallel = 1E4$ $\|u-u_h\|$	order	$D_\parallel = 1E6$ $\|u-u_h\|$	order	$D_\parallel = 1E8$ $\|u-u_h\|$	order
					$k=1$					
8	2.44E-02		5.54E-02		7.12E-02		7.14E-02		7.14E-02	
16	8.31E-03	1.56	3.26E-02	0.77	5.66E-02	0.33	5.70E-02	0.33	5.70E-02	0.33
32	2.39E-03	1.80	1.40E-02	1.22	4.73E-02	0.26	4.85E-02	0.23	4.85E-02	0.23
64	6.33E-04	1.92	4.50E-03	1.64	3.90E-02	0.28	4.27E-02	0.18	4.28E-02	0.18
128	1.62E-04	1.97	1.24E-03	1.86	2.84E-02	0.46	3.86E-02	0.15	3.87E-02	0.14
256	4.10E-05	1.99	3.23E-04	1.94	1.51E-02	0.91	3.54E-02	0.12	3.59E-02	0.11
					$k=2$					
4	5.68E-03		1.20E-02		1.73E-02		1.74E-02		1.74E-02	
8	6.01E-04	3.24	1.36E-03	3.14	7.80E-03	1.15	8.26E-03	1.08	8.26E-03	1.07
16	7.16E-05	3.07	1.22E-04	3.49	3.22E-03	1.28	4.91E-03	0.75	4.94E-03	0.74
32	8.84E-06	3.02	1.11E-05	3.45	4.93E-04	2.71	3.35E-03	0.55	3.56E-03	0.47
64	1.10E-06	3.00	1.19E-06	3.23	3.67E-05	3.75	1.56E-03	1.11	2.85E-03	0.32
128	1.38E-07	3.00	1.40E-07	3.08	2.42E-06	3.92	1.95E-04	3.00	2.23E-03	0.35
					$k=3$					
4	7.04E-04		1.15E-03		6.65E-03		7.34E-03		7.35E-03	
8	4.48E-05	3.98	6.43E-05	4.16	5.26E-04	3.66	9.12E-04	3.01	9.41E-04	2.96
16	2.78E-06	4.01	3.16E-06	4.35	5.17E-05	3.35	1.05E-03	-0.21	1.33E-03	-0.50
32	1.74E-07	4.00	1.80E-07	4.13	1.36E-06	5.25	8.98E-05	3.55	8.84E-04	0.59
64	1.09E-08	4.00	1.10E-08	4.03	3.79E-08	5.16	2.32E-06	5.27	1.32E-04	2.74
128	6.83E-10	4.00	6.83E-10	4.01	1.25E-09	4.92	6.45E-08	5.17	3.37E-06	5.29
					$k=4$					
4	3.64E-05		5.19E-05		9.96E-04		3.08E-03		3.14E-03	
8	1.03E-06	5.15	1.50E-06	5.11	1.70E-05	5.88	9.21E-05	5.06	9.92E-05	4.99
16	3.04E-08	5.08	3.76E-08	5.32	1.34E-07	6.99	1.31E-06	6.14	8.19E-06	3.60
32	9.16E-10	5.05	1.02E-09	5.20	2.45E-09	5.77	9.02E-09	7.18	5.38E-07	3.93
64	2.81E-11	5.03	2.96E-11	5.11	5.56E-11	5.46	4.43E-10	4.35	6.52E-08	3.04
					$k=5$					
2	2.62E-04		3.71E-04		4.12E-03		5.22E-03		5.23E-03	
4	3.12E-06	6.39	4.81E-06	6.27	9.10E-05	5.50	3.43E-04	3.93	3.53E-04	3.89
8	7.58E-08	5.36	8.81E-08	5.77	5.14E-07	7.47	1.59E-05	4.43	9.97E-05	1.82
16	1.09E-09	6.12	1.14E-09	6.27	5.62E-09	6.52	9.92E-08	7.32	2.01E-06	5.63
32	1.66E-11	6.03	1.68E-11	6.09	3.92E-11	7.16	4.08E-10	7.93	1.42E-08	7.15
					$k=6$					
1	1.50E-03		3.18E-03		2.83E-02		5.24E-02		5.29E-02	
2	1.70E-05	6.46	3.72E-05	6.42	7.55E-04	5.23	1.35E-03	5.28	1.36E-03	5.28
4	3.18E-07	5.74	5.40E-07	6.11	5.49E-06	7.10	1.10E-04	3.61	1.73E-04	2.97
8	1.91E-09	7.38	2.67E-09	7.66	2.84E-08	7.60	5.60E-07	7.62	9.63E-06	4.17
16	1.37E-11	7.12	1.63E-11	7.36	7.90E-11	8.49	2.53E-09	7.79	1.57E-08	9.26

We observe:

- High-order Scheme works well for resolving the non-alignment of mesh and anisotropy

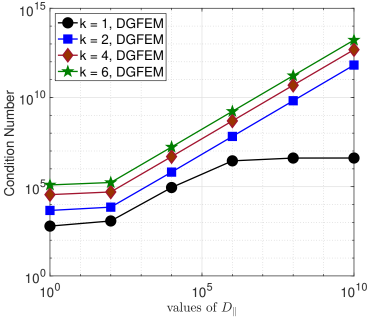
We observe:

- High-order Scheme works well for resolving the non-alignment of mesh and anisotropy
- However, with increasing anisotropy, the condition number in corresponding linear system is also increasing



We observe:

- High-order Scheme works well for resolving the non-alignment of mesh and anisotropy
- However, with increasing anisotropy, the condition number in corresponding linear system is also increasing



- We need an **effective and efficient fast solver**

Illustration of Auxiliary Space Pre-conditioner (ASP)

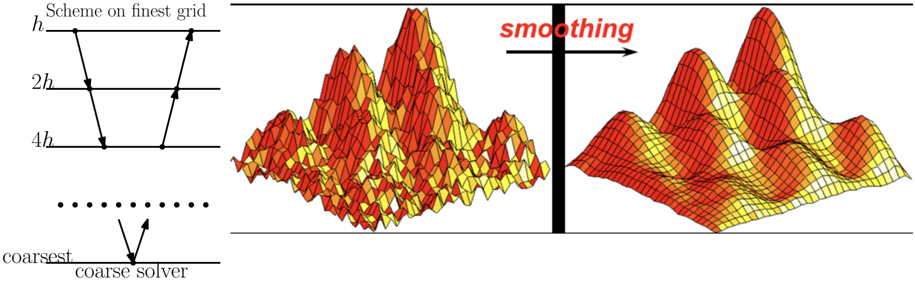
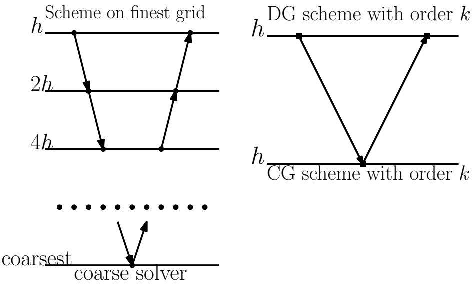


Figure: (a). Multigrid V-cycle;

Illustration of Auxiliary Space Pre-conditioner (ASP)



(a) (b)

Figure: (a). Multigrid V-cycle; (b). One-level Auxiliary Space;

- Proposed in [S. Nepomnyaschikh 1992]
- b). Use an auxiliary space as “coarse” space, which is easier to solve

Illustration of Auxiliary Space Pre-conditioner (ASP)

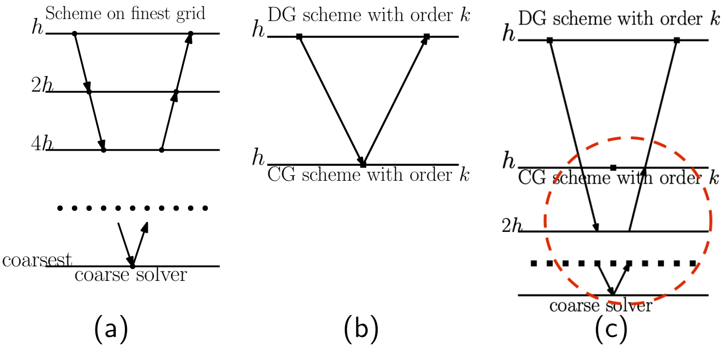


Figure: (a). Multigrid V-cycle; (b). One-level Auxiliary Space; (c). Multi-grid Auxiliary Space.

- Proposed in [S. Nepomnyaschikh 1992]
- b). Use an auxiliary space as "coarse" space, which is easier to solve
- c). Replace the "coarse" solver by existing solvers/preconditioners

Auxiliary Space Pre-conditioner - CG Scheme

Why CG Scheme as the ASP?

- Less DoFs
- Well developed CG - fast solver

Auxiliary Space Pre-conditioner (Efficiency)

Given $\boldsymbol{f} \in V_{\mathsf{DG}}'$, find $\boldsymbol{u} \in V_{\mathsf{DG}}$ such that

$$\boldsymbol{A}_{\mathsf{DG}}\boldsymbol{u} = \boldsymbol{f},$$

Auxiliary Space Pre-conditioner (Efficiency)

Given $\boldsymbol{f} \in V'_{\mathsf{DG}}$, find $\boldsymbol{u} \in V_{\mathsf{DG}}$ such that

$$\boldsymbol{A}_{\mathsf{DG}}\boldsymbol{u} = \boldsymbol{f},$$

- As in [Antonietti 2017], choose the product of auxiliary spaces: $\bar{V}_{\mathsf{DG}} = V_{\mathsf{DG}} \times V_{\mathsf{CG}}$

Auxiliary Space Pre-conditioner (Efficiency)

Given $\boldsymbol{f} \in V'_{\mathsf{DG}}$, find $\boldsymbol{u} \in V_{\mathsf{DG}}$ such that

$$\boldsymbol{A}_{\mathsf{DG}}\boldsymbol{u} = \boldsymbol{f},$$

- As in [Antonietti 2017], choose the product of auxiliary spaces: $\bar{V}_{\mathsf{DG}} = V_{\mathsf{DG}} \times V_{\mathsf{CG}}$
- Since $V_{\mathsf{CG}} \subset V_{\mathsf{DG}}$, we can use subset decomposition $V_{\mathsf{DG}} = V_{\mathsf{DG}} + V_{\mathsf{CG}}$.

Auxiliary Space Pre-conditioner (Efficiency)

Given $\boldsymbol{f} \in V'_{\mathsf{DG}}$, find $\boldsymbol{u} \in V_{\mathsf{DG}}$ such that

$$\boldsymbol{A}_{\mathsf{DG}}\boldsymbol{u} = \boldsymbol{f},$$

- As in [Antonietti 2017], choose the product of auxiliary spaces: $\bar{V}_{\mathsf{DG}} = V_{\mathsf{DG}} \times V_{\mathsf{CG}}$
- Since $V_{\mathsf{CG}} \subset V_{\mathsf{DG}}$, we can use subset decomposition $V_{\mathsf{DG}} = V_{\mathsf{DG}} + V_{\mathsf{CG}}$.

Introduce the auxiliary space preconditioner, $\boldsymbol{B}_{\mathsf{DG}} : V'_{\mathsf{DG}} \mapsto V_{\mathsf{DG}}$

$$\boldsymbol{B}_{\mathsf{DG}} = \underbrace{\boldsymbol{S}_{\mathsf{DG}}}_{\text{smoother: Gauss-Seidel}} + \underbrace{\boldsymbol{\Pi}}_{\text{inclusion operator}} \boldsymbol{A}_{\mathsf{CG}}^{-1} \underbrace{\boldsymbol{\Pi}^{\top}}_{L^2 \text{ projection}}, \tag{12}$$

Here $\boldsymbol{\Pi} : V_{\mathsf{CG}} \to V_{\mathsf{DG}}$ and $\boldsymbol{\Pi}^{\top} : V'_{\mathsf{DG}} \to V'_{\mathsf{CG}}$.

Auxiliary Space Pre-conditioner (Efficiency)

Given $\boldsymbol{f} \in V_{\mathsf{DG}}'$, find $\boldsymbol{u} \in V_{\mathsf{DG}}$ such that

$$\boldsymbol{A}_{\mathsf{DG}}\boldsymbol{u} = \boldsymbol{f},$$

- As in [Antonietti 2017], choose the product of auxiliary spaces: $\bar{V}_{\mathsf{DG}} = V_{\mathsf{DG}} \times V_{\mathsf{CG}}$
- Since $V_{\mathsf{CG}} \subset V_{\mathsf{DG}}$, we can use subset decomposition $V_{\mathsf{DG}} = V_{\mathsf{DG}} + V_{\mathsf{CG}}$.

Introduce the auxiliary space preconditioner, $\boldsymbol{B}_{\mathsf{DG}} : V_{\mathsf{DG}}' \mapsto V_{\mathsf{DG}}$

$$\boldsymbol{B}_{\mathsf{DG}} = \underbrace{\boldsymbol{S}_{\mathsf{DG}}}_{\text{smoother: Gauss-Seidel}} + \underbrace{\boldsymbol{\Pi}}_{\text{inclusion operator}} \boldsymbol{A}_{\mathsf{CG}}^{-1} \underbrace{\boldsymbol{\Pi}^{\top}}_{L^2 \text{ projection}}, \tag{12}$$

Here $\boldsymbol{\Pi} : V_{\mathsf{CG}} \to V_{\mathsf{DG}}$ and $\boldsymbol{\Pi}^{\top} : V_{\mathsf{DG}}' \to V_{\mathsf{CG}}'$.

Remark:

Here we need to invert $\boldsymbol{A}_{\mathsf{CG}}$ exactly. We shall first show the effectiveness of the proposed pre-conditioner.

Efficiency Test 1 - Solve H^1-problem exactly

- Set $\boldsymbol{B}_{\mathsf{DG}} = \boldsymbol{S}_{\mathsf{DG}} + \boldsymbol{\Pi} \boldsymbol{A}_{\mathsf{CG}}^{-1} \boldsymbol{\Pi}^\top$
- Solve pre-conditioned system by GMRES with tol $=$ 1E-6
- $\mathbf{b} = [0, 1]^\top$ on unstructured triangular mesh

Efficiency Test 1 - Solve H^1-problem exactly

- Set $\boldsymbol{B}_{\mathsf{DG}} = \boldsymbol{S}_{\mathsf{DG}} + \boldsymbol{\Pi}\boldsymbol{A}_{\mathsf{CG}}^{-1}\boldsymbol{\Pi}^\top$
- Solve pre-conditioned system by GMRES with tol $=$ 1E-6
- $\mathbf{b} = [0,1]^\top$ on unstructured triangular mesh

$D_\parallel$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
1	11	11	11	11	11	11	11	11
1E+2	12	10	10	10	10	10	10	11
1E+4	20	8	8	8	8	8	8	9
1E+6	20	14	5	5	5	5	6	7
1E+8	20	4	5	5	5	6	6	5

Table: Iterations for $\boldsymbol{B}_{\mathsf{DG}}$ (solve the H^1 problem exactly) when $h = 1/10$ on non-aligned mesh.

Efficiency Test 1 - Solve H^1-problem exactly

- Set $\boldsymbol{B}_{\text{DG}} = \boldsymbol{S}_{\text{DG}} + \boldsymbol{\Pi} \boldsymbol{A}_{\text{CG}}^{-1} \boldsymbol{\Pi}^\top$
- Solve pre-conditioned system by GMRES with tol $=$ 1E-6
- $\mathbf{b} = [0, 1]^\top$ on unstructured triangular mesh

$D_\parallel$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
1	11	11	11	11	11	11	11	11
1E+2	12	10	10	10	10	10	10	11
1E+4	20	8	8	8	8	8	8	9
1E+6	20	14	5	5	5	5	6	7
1E+8	20	4	5	5	5	6	6	5

Table: Iterations for $\boldsymbol{B}_{\text{DG}}$ (solve the H^1 problem exactly) when $h = 1/10$ on non-aligned mesh.

Remark:

- Almost constant iteration number⟶**Effective and Robust**

Efficiency Test 1 - Solve H^1-problem exactly

- Set $\boldsymbol{B}_{\text{DG}} = \boldsymbol{S}_{\text{DG}} + \boldsymbol{\Pi} \boldsymbol{A}_{\text{CG}}^{-1} \boldsymbol{\Pi}^\top$
- Solve pre-conditioned system by GMRES with tol $=$ 1E-6
- $\mathbf{b} = [0, 1]^\top$ on unstructured triangular mesh

$D_\parallel$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
1	11	11	11	11	11	11	11	11
1E+2	12	10	10	10	10	10	10	11
1E+4	20	8	8	8	8	8	8	9
1E+6	20	14	5	5	5	5	6	7
1E+8	20	4	5	5	5	6	6	5

Table: Iterations for $\boldsymbol{B}_{\text{DG}}$ (solve the H^1 problem exactly) when $h = 1/10$ on non-aligned mesh.

Remark:

- Almost constant iteration number $\longrightarrow$ **Effective and Robust**
- However, $\boldsymbol{A}_{\text{CG}}^{-1}$ challenging with large size, high polynomial degree, strong anisotropy.

Efficiency Test 1 - Solve H^1-problem exactly

- Set $\boldsymbol{B}_{\mathsf{DG}} = \boldsymbol{S}_{\mathsf{DG}} + \boldsymbol{\Pi}\boldsymbol{A}_{\mathsf{CG}}^{-1}\boldsymbol{\Pi}^\top$
- Solve pre-conditioned system by GMRES with tol $=$ 1E-6
- $\mathbf{b} = [0,1]^\top$ on unstructured triangular mesh

$D_\parallel$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
1	11	11	11	11	11	11	11	11
1E+2	12	10	10	10	10	10	10	11
1E+4	20	8	8	8	8	8	8	9
1E+6	20	14	5	5	5	5	6	7
1E+8	20	4	5	5	5	6	6	5

Table: Iterations for $\boldsymbol{B}_{\mathsf{DG}}$ (solve the H^1 problem exactly) when $h = 1/10$ on non-aligned mesh.

Remark:

• Almost constant iteration number⟶**Effective and Robust**

• However, $\boldsymbol{A}_{\mathsf{CG}}^{-1}$ challenging with large size, high polynomial degree, strong anisotropy. $\rightarrow$ **use preconditioner $B_{\mathsf{CG}} \approx A_{\mathsf{CG}}^{-1}$**

Auxiliary Space Pre-conditioner (Efficiency)

$$\boldsymbol{B}_{\mathsf{DG}} = \boldsymbol{S}_{\mathsf{DG}} + \boldsymbol{\Pi} \boldsymbol{A}_{\mathsf{CG}}^{-1} \boldsymbol{\Pi}^{\top}, \text{ will be replaced by}$$

Auxiliary Space Pre-conditioner (Efficiency)

$$
\begin{aligned}
\boldsymbol{B}_{\mathsf{DG}} &= \boldsymbol{S}_{\mathsf{DG}} + \boldsymbol{\Pi} \boldsymbol{A}_{\mathsf{CG}}^{-1} \boldsymbol{\Pi}^{\top}, \text{ will be replaced by} \\
\boldsymbol{B}_{\mathsf{DG}}^{\mathsf{inexact}} &= \boldsymbol{S}_{\mathsf{DG}} + \boldsymbol{\Pi} \boldsymbol{B}_{\mathsf{CG}} \boldsymbol{\Pi}^{\top}, \text{ here, } \boldsymbol{B}_{\mathsf{CG}} = \boldsymbol{S}_{\mathsf{CG}} + \boldsymbol{B}_{\mathsf{MG}}
\end{aligned}
$$

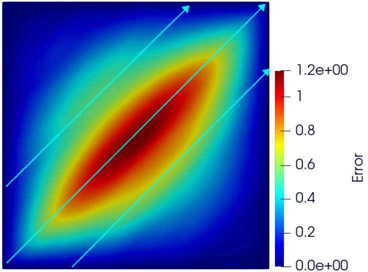
Auxiliary Space Pre-conditioner (Efficiency)

$$\begin{aligned} B_{\mathsf{DG}} &= S_{\mathsf{DG}} + \Pi A_{\mathsf{CG}}^{-1} \Pi^{\top}, \text{ will be replaced by} \\ B_{\mathsf{DG}}^{\mathsf{inexact}} &= S_{\mathsf{DG}} + \Pi B_{\mathsf{CG}} \Pi^{\top}, \text{ here, } B_{\mathsf{CG}} = S_{\mathsf{CG}} + B_{\mathsf{MG}} \end{aligned}$$

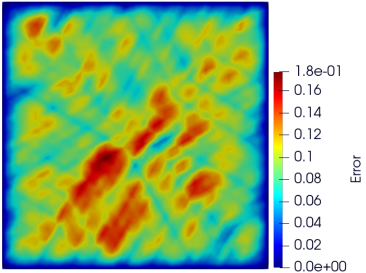
- Here B_{MG}: multi-grid solver for CG-FEM.
- S_{CG}: Schwarz-type block line smoother [Pavarino 1994, Antonietti 2017]

Error plot for $\mathbf{b} = [1,1]^\top/\sqrt{2}$ and exact $u = 0$

Let $\Omega = [-1,1]^2$ and start the iterative solver with a random initial guess.

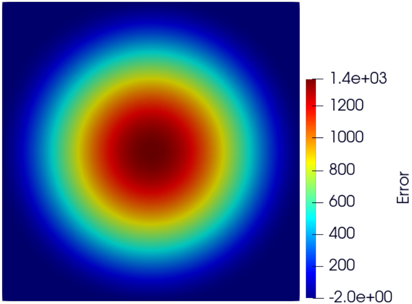


(a) before smoothing

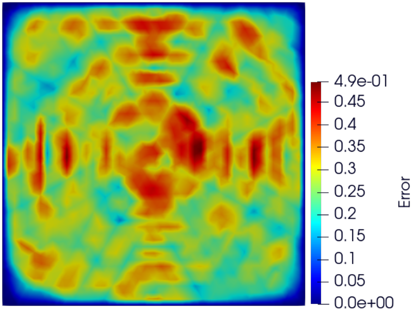


(b) with perpendicular line smoother

Error plot for circular b and exact $u = 0$



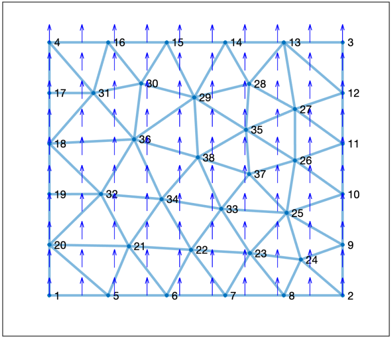
(a) before smoothing



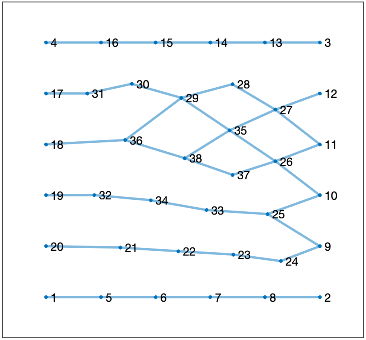
(b) with perpendicular line smoother

- high frequency pattern along the direction perpendicular to the anisotropy

Illustration of Line Smoother for $\mathbf{b} = [0, 1]^\top$

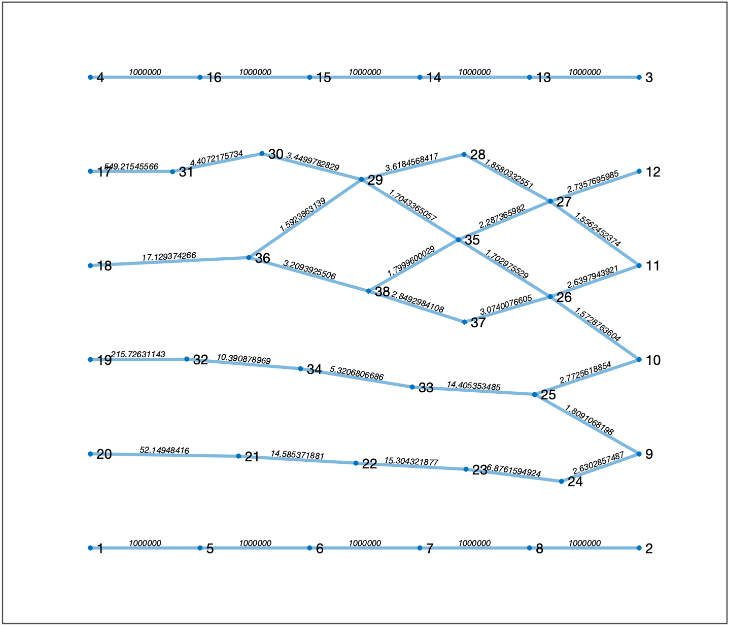


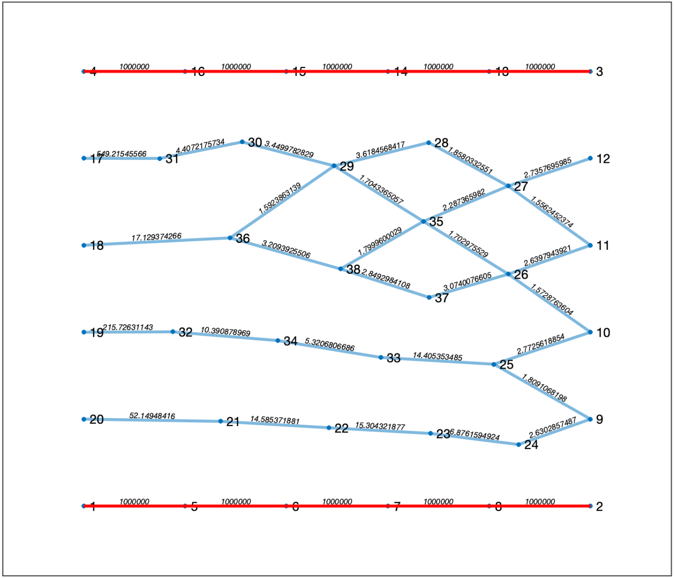
(a) mesh

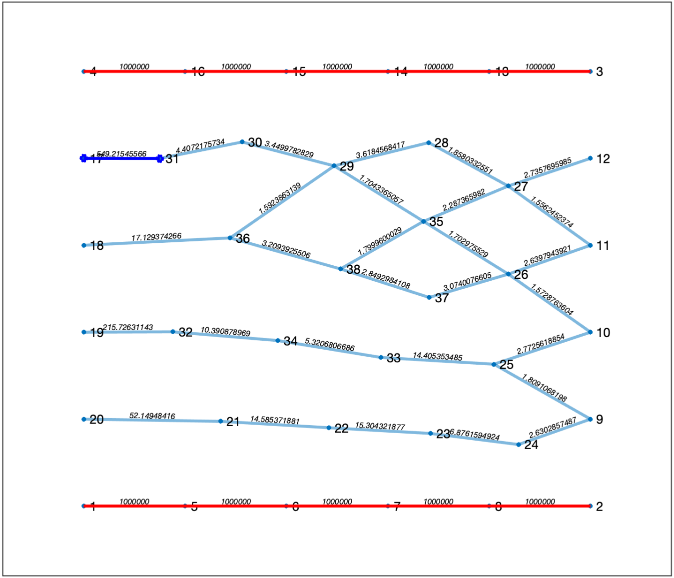


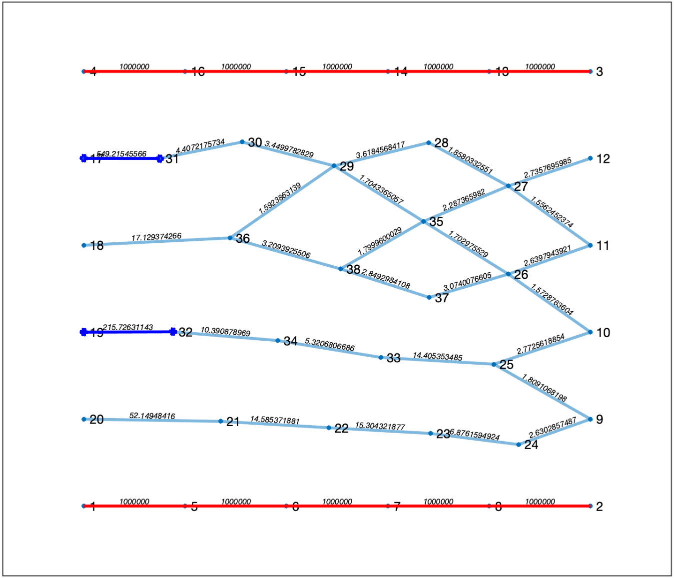
(b) dropping aligned edges

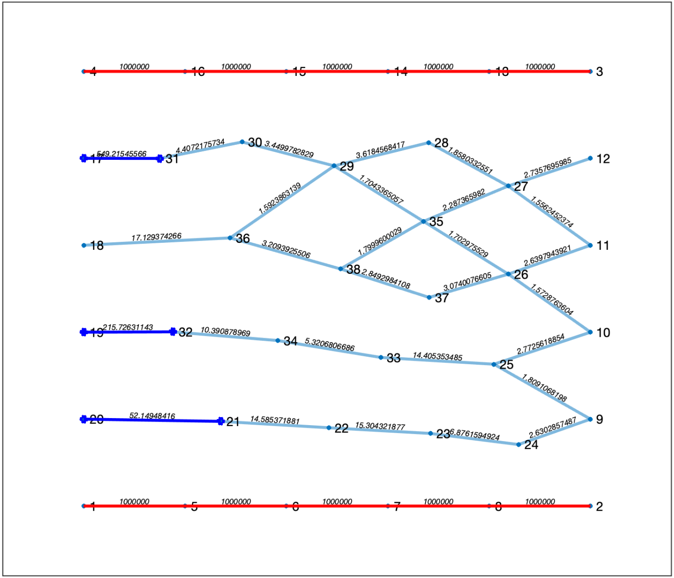
- Weights: $\omega_e \leftarrow \dfrac{1.0}{|\cos(\theta)| + 10^{-6}}$, $\cos(\theta) \leftarrow \dfrac{\boldsymbol{t}^\top \mathbf{b}_{\text{mid}}}{\|\boldsymbol{t}\| \|\mathbf{b}_{\text{mid}}\|}$
- For example, we choose threshold $\eta = 2.0$

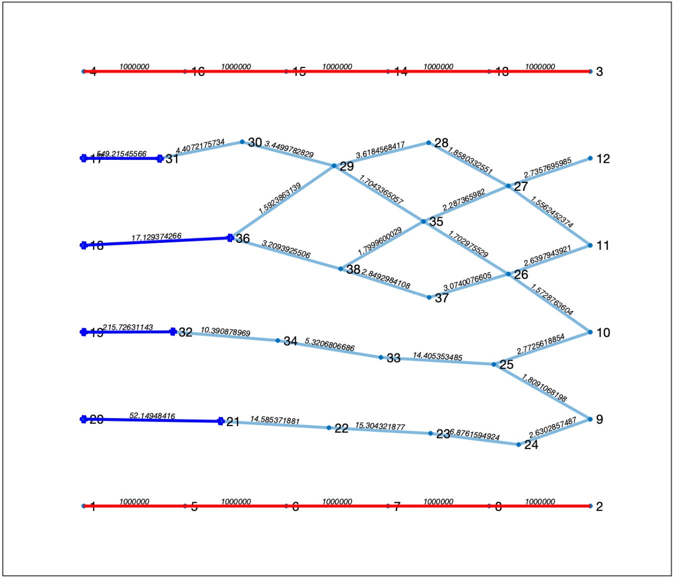

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
11
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
26
3.0740076605
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

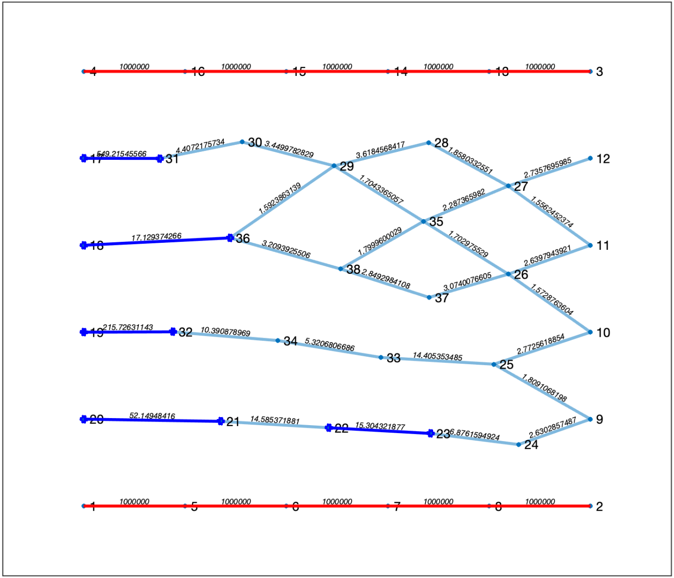

4
1000000
16
15
14
13
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
3.2093925506
35
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
10
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
1.8091068198
9
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
1
5
6
7
8
2

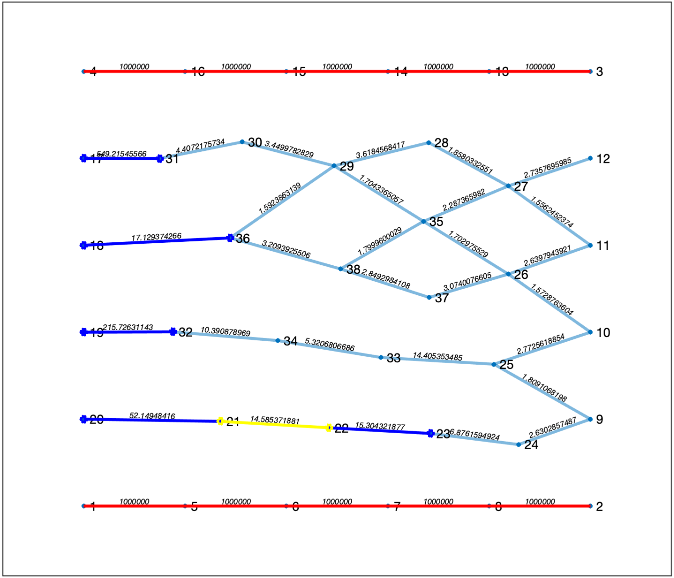

4 1000000 16 1000000 15 1000000 14 1000000 13 1000000 3
17 549.21545566 31 4.4072175734 30 3.4499782829 29 3.6184568417 28 1.8580332551 27 2.7357695985 12
1.5923863139
1.7043365057
2.287365982
35
1.5562452374
18 17.129374266 36 3.2093925506 38 2.8492984108 37 3.0740076605 26 2.6397943921 11
1.7999600029
1.702975529
1.5728763604
19 215.72631143 32 10.390878969 34 5.3206806686 33 14.405353485 25 2.7725618854 10
1.8091068198
20 52.14948416 21 14.585371881 22 15.304321877 23 6.8761594924 24 2.6302857487 9
1 1000000 5 1000000 6 1000000 7 1000000 8 1000000 2

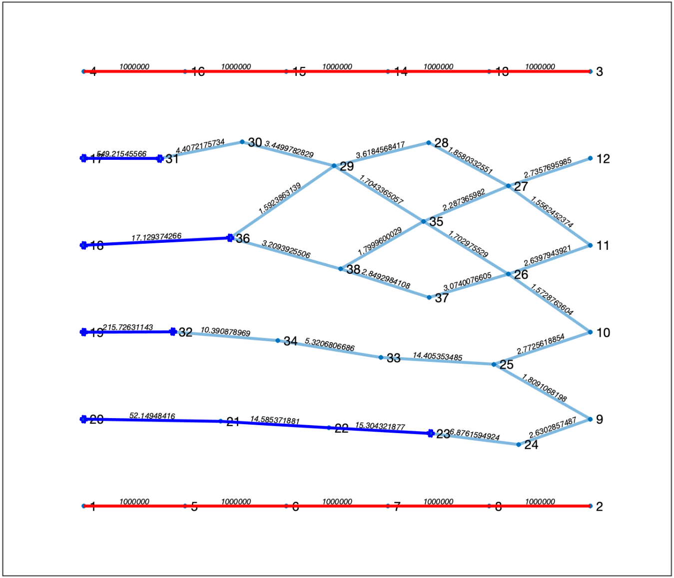

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
35
18
17.129374266
36
3.2093925506
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2


4
16
15
14
13
3
1000000
12
11
10
9
2
28
29
30
31
17
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
27
2.7357695985
1.5923863139
1.7043365057
2.287365982
1.5562452374
35
36
18
17.129374266
3.2093925506
1.7999600029
1.702975529
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
1
5
6
7
8

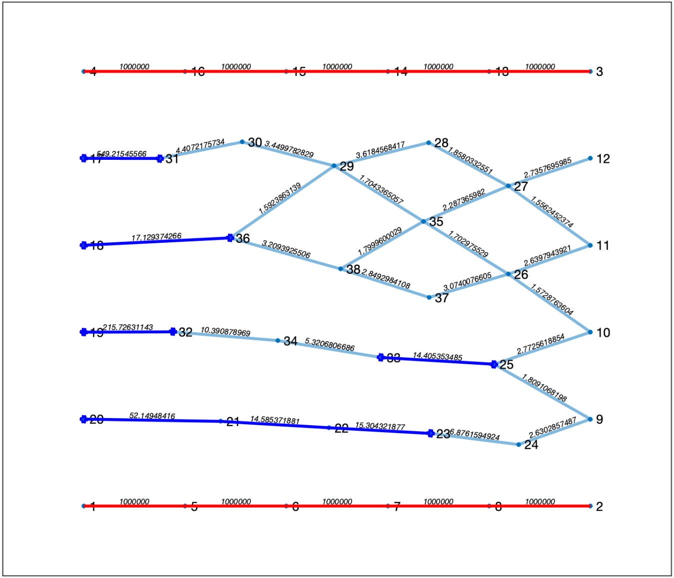

4
1000000
16
1000000
15
1000000
14
1000000
1000000
3
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
17.129374266
36
35
3.2093925506
1.7999600029
1.702975529
11
38
2.8492984108
2.6397943921
26
3.0740076605
37
1.5728763604
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
2.6302857487
9
24
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

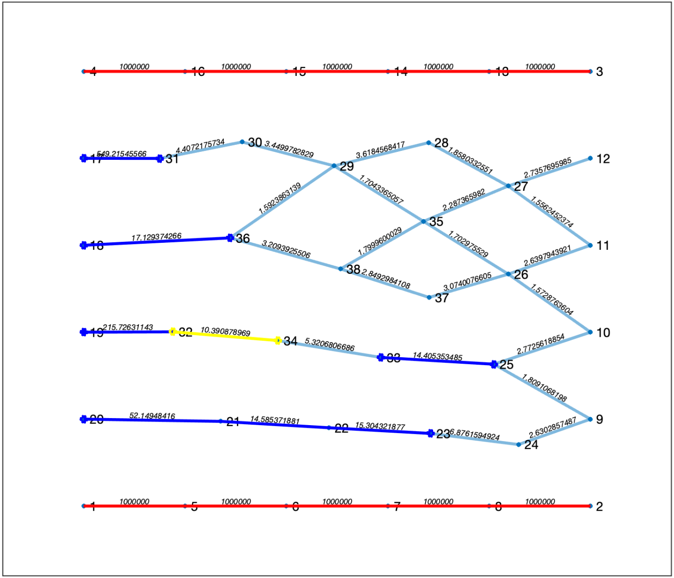

4
16
15
14
13
3
1000000
12
11
10
9
2
1
5
6
7
8
17
31
30
29
28
27
18
36
35
38
37
26
19
32
34
33
25
20
21
22
23
24
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
1.5923863139
1.7043365057
2.287365982
1.5562452374
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
2.8492984108
3.0740076605
1.5728763604
215.72631143
10.390878969
5.3206806686
14.405353485
2.7725618854
1.8091068198
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487

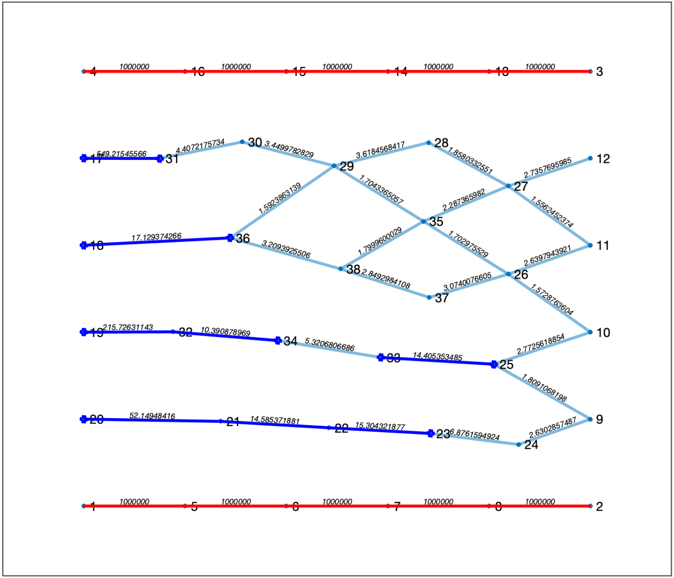

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
11
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
26
3.0740076605
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

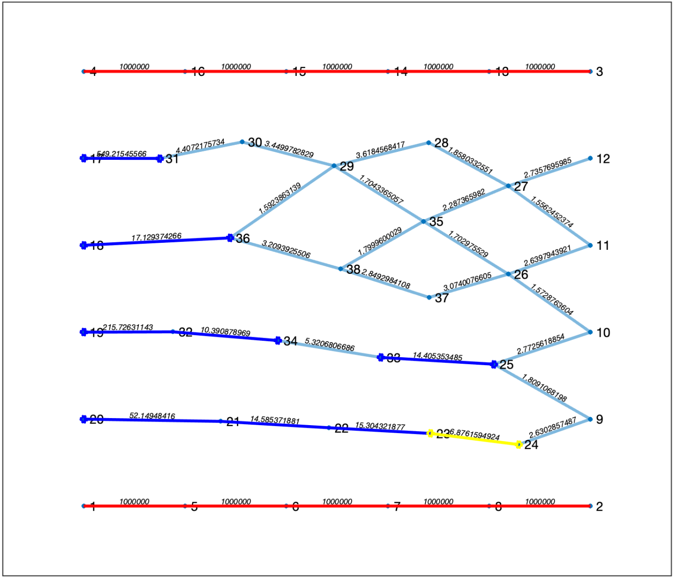


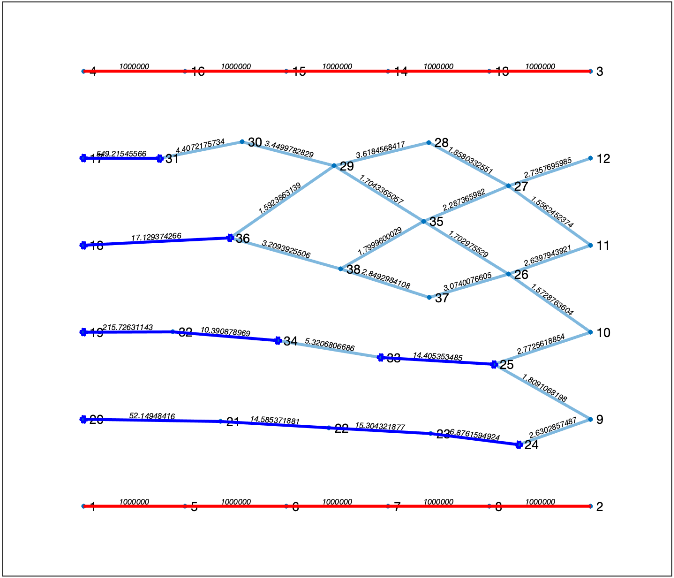
4
16
15
14
13
3
1000000
1000000
1000000
1000000
1000000
17
31
30
28
29
12
27
35
11
18
36
38
26
37
10
19
32
34
33
25
9
20
21
22
23
24
1
5
6
7
8
2
1000000
1000000
1000000
1000000
1000000
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
1.5923863139
1.7043365057
2.287365982
1.5562452374
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
2.8492984108
3.0740076605
1.5728763604
215.72631143
10.390878969
5.3206806686
14.405353485
2.7725618854
1.8091068198
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487

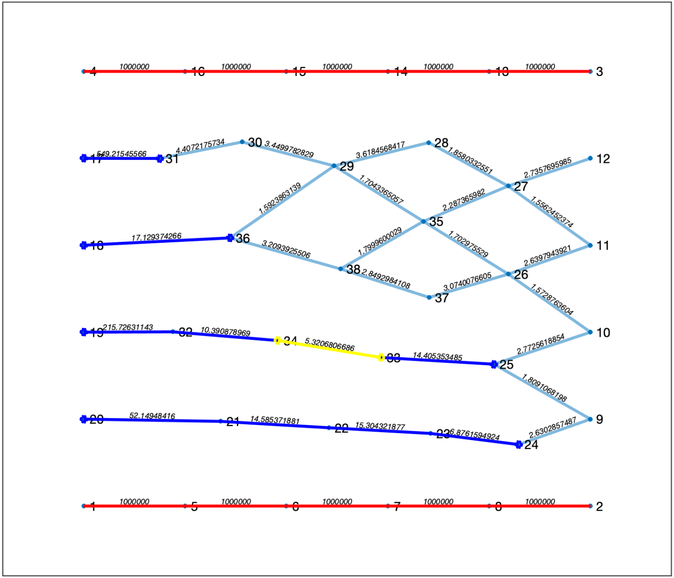

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
3.2093925506
1.7999600029
1.702975529
11
38
2.8492984108
2.6397943921
26
3.0740076605
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

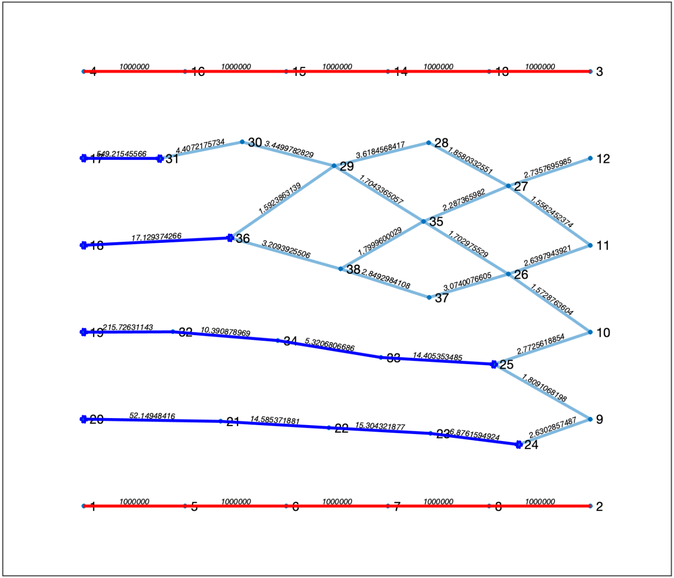

4
16
15
14
13
3
1000000
17
31
30
28
29
12
27
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
1.5923863139
1.7043365057
2.287365982
1.5562452374
35
18
36
11
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
26
37
3.0740076605
1.5728763604
19
32
34
10
215.72631143
10.390878969
5.3206806686
33
25
14.405353485
2.7725618854
1.8091068198
20
21
22
23
24
9
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487
1
5
6
7
8
2

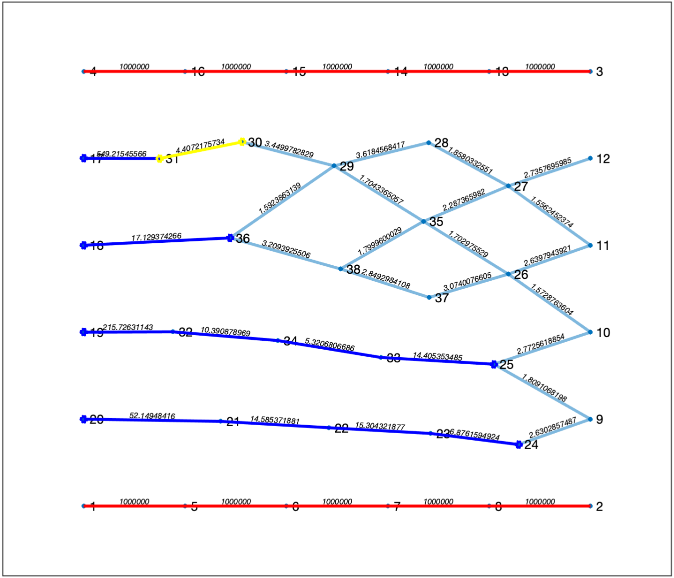

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
11
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
26
3.0740076605
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
2.6302857487
9
24
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

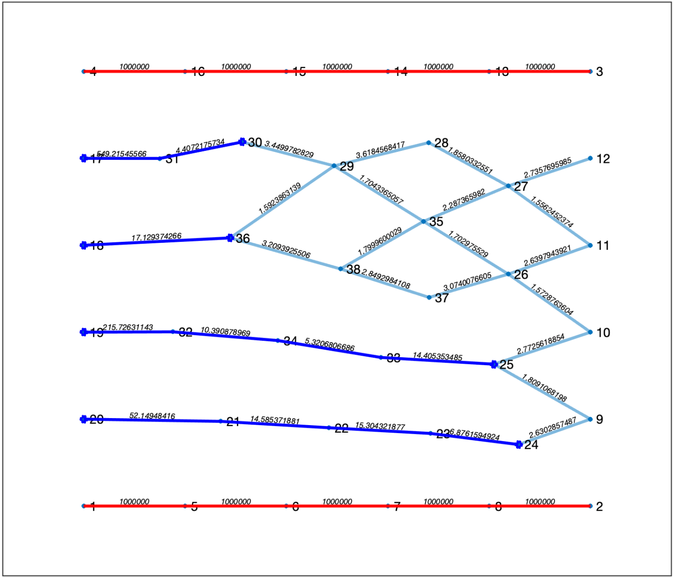

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
3.2093925506
35
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

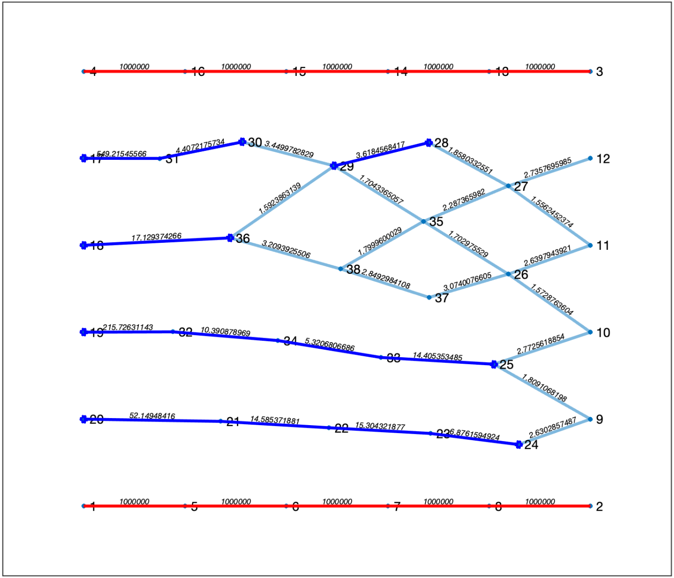

4
16
15
14
13
3
1000000
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
3.2093925506
35
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
9
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
1
5
6
7
8
2


4
1000000
16
1000000
15
1000000
14
1000000
1000000
3
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
35
1.5562452374
17.129374266
36
3.2093925506
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
215.72631143
32
10.390878969
5.3206806686
14.405353485
25
2.7725618854
10
1.8091068198
52.14948416
21
14.585371881
22
15.304321877
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

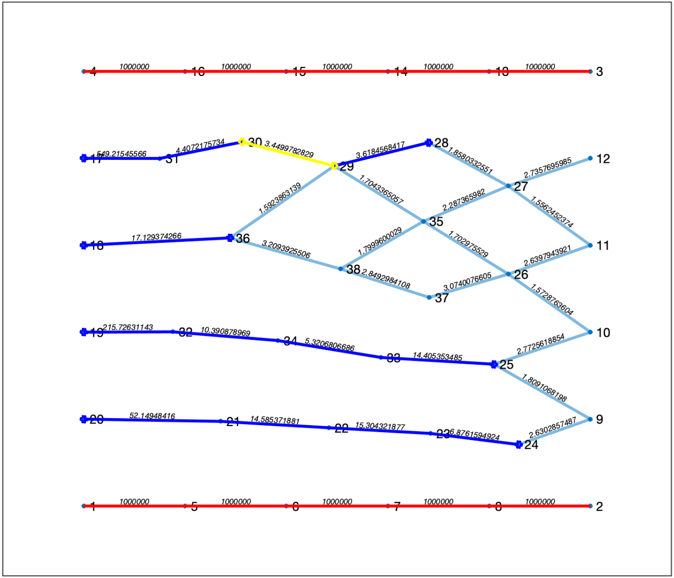

4
16
15
14
13
3
1000000
1000000
1000000
1000000
1000000
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
3.2093925506
35
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
5
6
7
8
2
1000000
1000000
1000000
1000000
1000000


1000000
2.7357695985
1.5562452374
2.6397943921
1.5728763604
2.7725618854
1.8091068198
2.6302857487

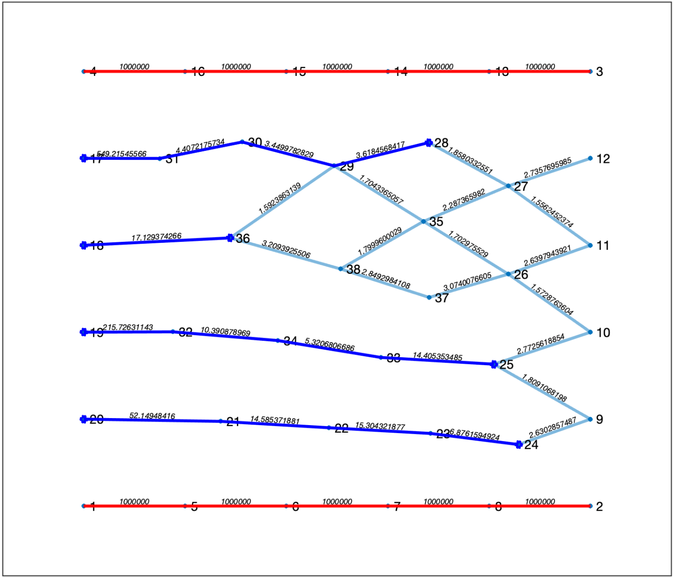

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
11
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
26
3.0740076605
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2

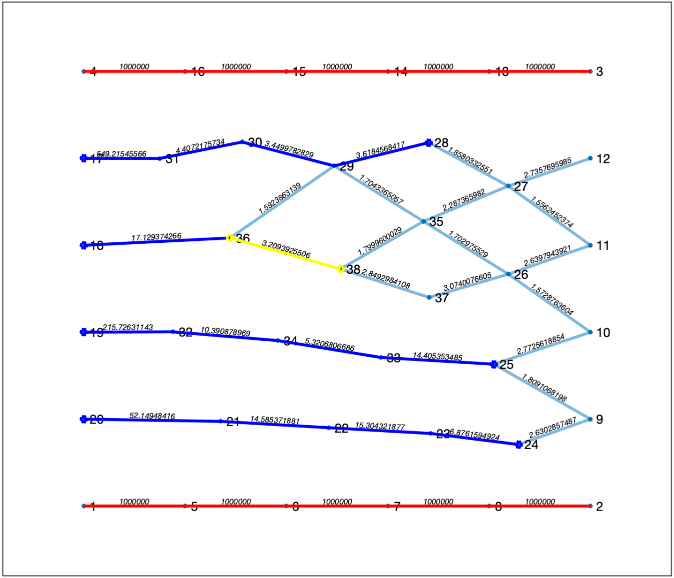


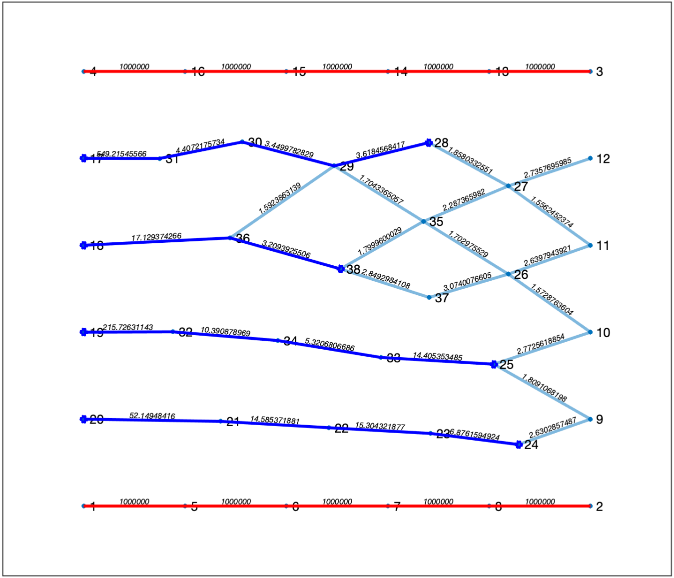
4 1000000 16 1000000 15 1000000 14 1000000 13 1000000 3
17 549.21545566 31 4.4072175734 30 3.4499782829 29 3.6184568417 28 1.8580332551 27 2.7357695985 12
1.5923863139 1.7043365057 2.287365982 1.5562452374
18 17.129374266 36 3.2093925506 38 1.7999600029 35 1.702975529 26 2.6397943921 11
38 2.8492984108 37 3.0740076605 1.5728763604
19 215.72631143 32 10.390878969 34 5.3206806686 33 14.405353485 25 2.7725618854 10
1.8091068198
20 52.14948416 21 14.585371881 22 15.304321877 23 6.8761594924 24 2.6302857487 9
1 1000000 5 1000000 6 1000000 7 1000000 8 1000000 2

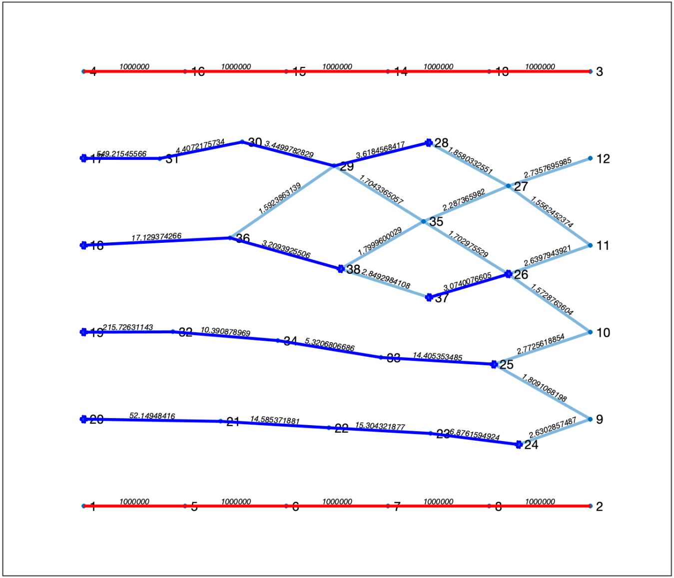


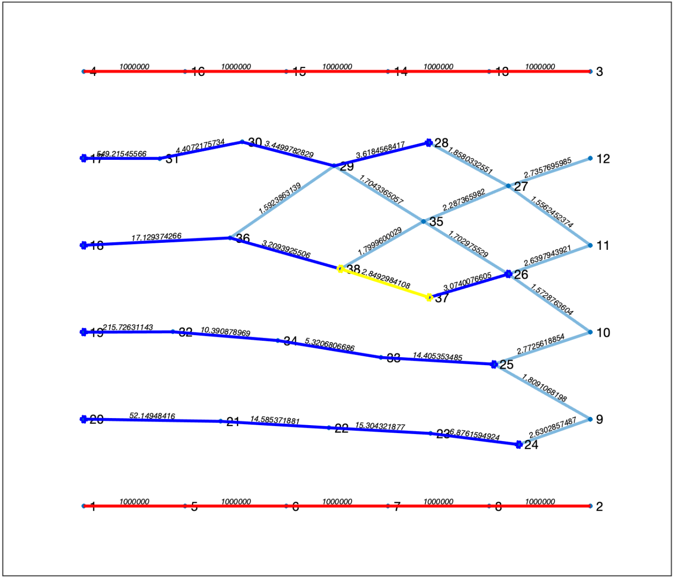
4
16
15
14
13
3
1000000
17
31
30
29
28
12
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
27
1.5923863139
1.7043365057
2.287365982
35
1.5562452374
18
36
11
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
26
37
3.0740076605
1.5728763604
19
32
34
33
25
10
215.72631143
10.390878969
5.3206806686
14.405353485
2.7725618854
1.8091068198
20
21
22
23
24
9
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487
1
5
6
7
8
2

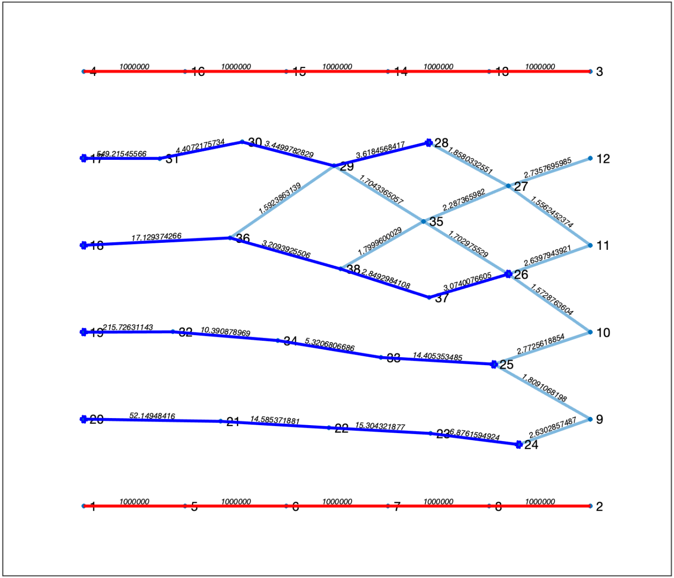

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
35
18
17.129374266
36
3.2093925506
1.7999600029
1.702975529
11
2.6397943921
38
2.8492984108
26
3.0740076605
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2


4
16
15
14
13
3
1000000
17
31
30
29
28
12
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
27
1.7043365057
1.5923863139
2.287365982
1.5562452374
35
18
36
11
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
26
2.8492984108
3.0740076605
37
1.5728763604
19
32
34
33
25
10
215.72631143
10.390878969
5.3206806686
14.405353485
2.7725618854
1.8091068198
20
21
22
23
24
9
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487
1
5
6
7
8
2

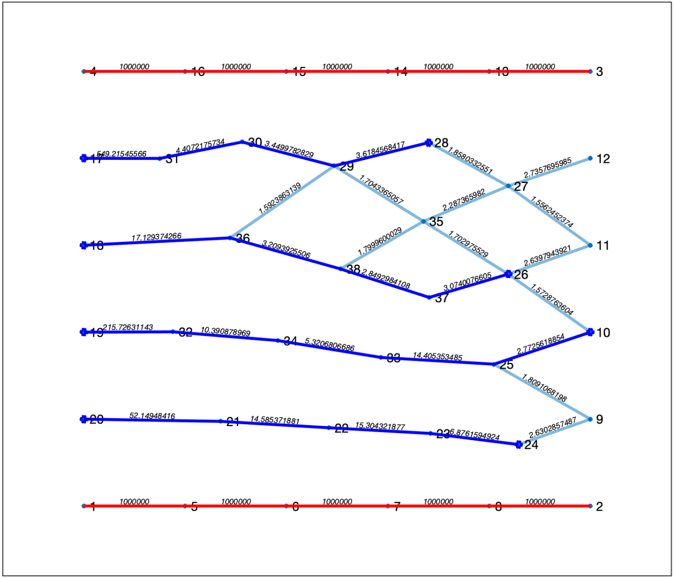

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5562452374
11
1.5923863139
1.7043365057
35
2.287365982
18
17.129374266
36
3.2093925506
38
1.7999600029
2.8492984108
37
3.0740076605
26
2.6397943921
1.702975529
1.5728763604
10
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
1.8091068198
9
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2


4
16
15
14
13
3
1000000
1000000
1000000
1000000
1000000
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
35
1.5562452374
11
18
17.129374266
36
3.2093925506
38
1.7999600029
1.702975529
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
10
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
1.8091068198
9
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
1
5
6
7
8
2
1000000
1000000
1000000
1000000
1000000

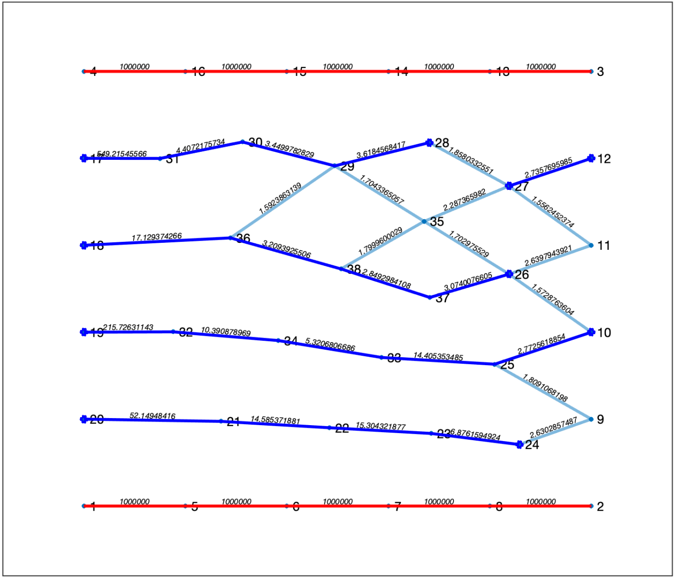

4
16
15
14
13
3
1000000
17
31
30
29
28
12
27
35
11
18
36
38
37
26
10
19
32
34
33
25
9
20
21
22
23
24
1
5
6
7
8
2
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
1.5923863139
1.7043365057
2.287365982
1.5562452374
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
2.8492984108
3.0740076605
1.5728763604
215.72631143
10.390878969
5.3206806686
14.405353485
2.7725618854
1.8091068198
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487

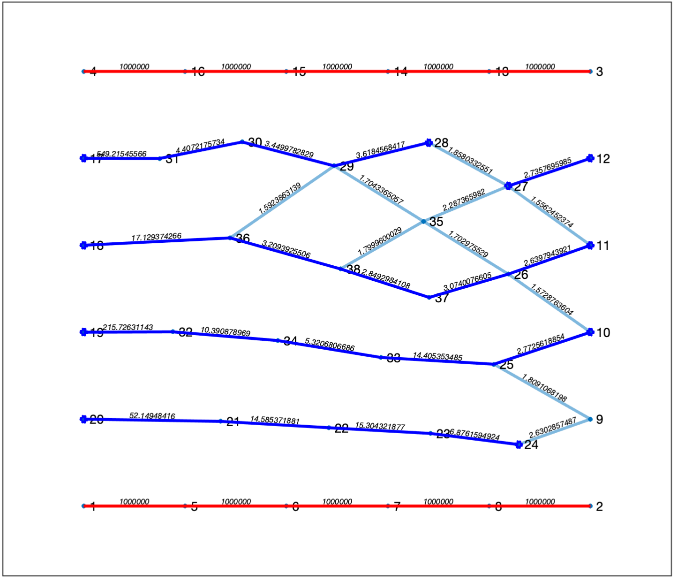


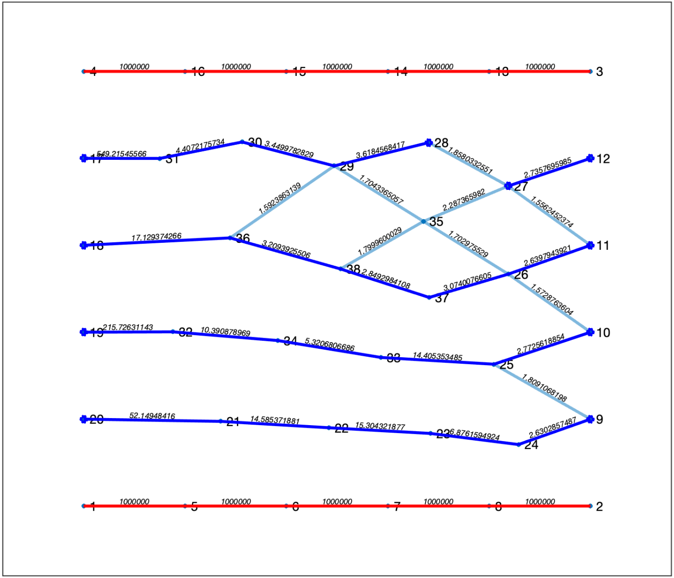
1000000
4
16
15
14
13
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
3.2093925506
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
5
6
7
8
2

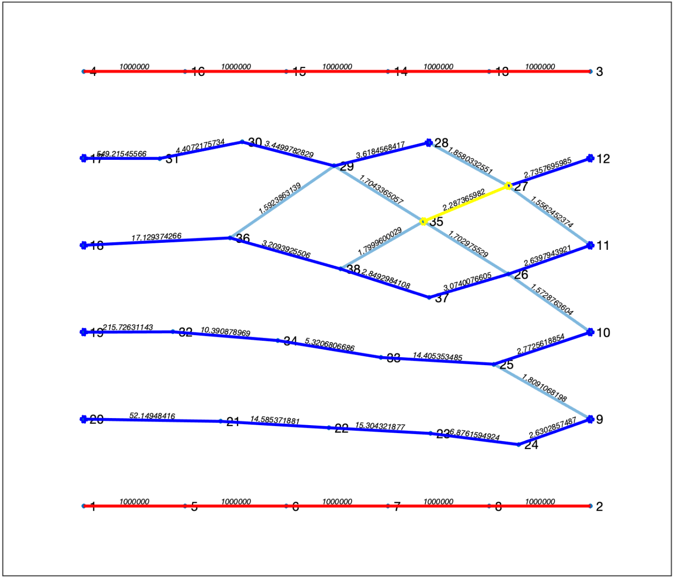


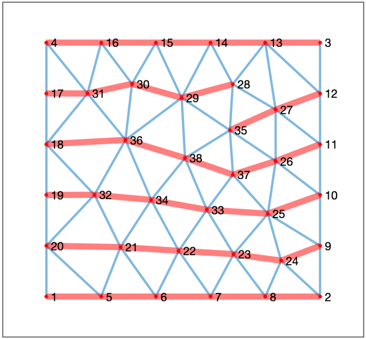
1000000
1000000
1000000
1000000
1000000
549.21545566
4.4072175734
3.4499782829
3.6184568417
1.8580332551
2.7357695985
1.5923863139
1.7043365057
2.287365982
1.5562452374
17.129374266
3.2093925506
1.7999600029
1.702975529
2.6397943921
2.8492984108
3.0740076605
1.5728763604
215.72631143
10.390878969
5.3206806686
14.405353485
2.7725618854
1.8091068198
52.14948416
14.585371881
15.304321877
6.8761594924
2.6302857487


4
1000000
16
15
14
13
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
18
17.129374266
36
35
11
3.2093925506
1.7999600029
1.702975529
2.6397943921
38
2.8492984108
3.0740076605
26
37
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
2.6302857487
24
9
1
5
6
7
8
2

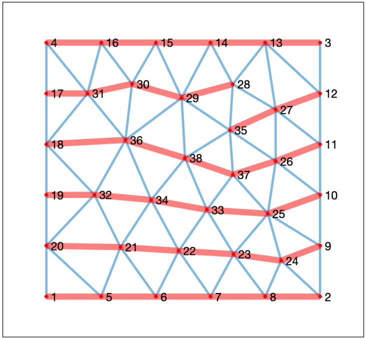

4
16
15
14
13
3
1000000
549.21545566
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
31
1.5923863139
1.7043365057
2.287365982
1.5562452374
35
11
17.129374266
36
3.2093925506
1.7999600029
1.702975529
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
52.14948416
21
14.585371881
22
15.304321877
6.8761594924
2.6302857487
9
24
1
5
6
7
8
2


4
1000000
16
15
14
3
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
1.5562452374
17.129374266
36
3.2093925506
35
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
5
6
7
8
2

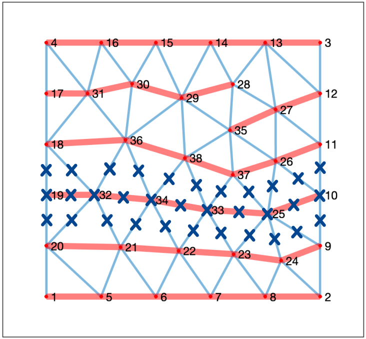

4
1000000
16
1000000
15
1000000
14
1000000
13
1000000
3
17
549.21545566
31
4.4072175734
30
3.4499782829
29
3.6184568417
28
1.8580332551
27
2.7357695985
12
1.5923863139
1.7043365057
2.287365982
35
1.5562452374
18
17.129374266
36
3.2093925506
1.7999600029
1.702975529
11
38
2.8492984108
37
3.0740076605
26
2.6397943921
1.5728763604
19
215.72631143
32
10.390878969
34
5.3206806686
33
14.405353485
25
2.7725618854
10
1.8091068198
20
52.14948416
21
14.585371881
22
15.304321877
23
6.8761594924
24
2.6302857487
9
1
1000000
5
1000000
6
1000000
7
1000000
8
1000000
2



4 16 15 14 13 3
17 31 30 29 28 12
27
35 11
18 36 38 26
37
19 32 34 33 25 10
20 21 22 23 24 9
1 5 6 7 8 2



(c) Path Cover



(b) DoFs of one block for P_2 basis

Fast Solver Test 1: $\mathbf{b} = [0, 1]^\top$

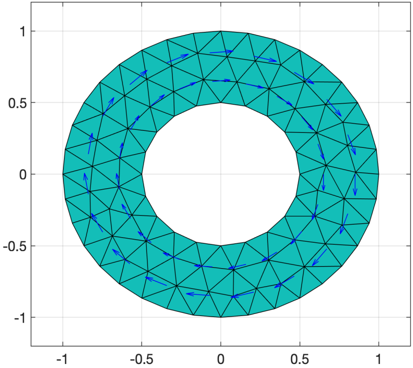
- $\mathbf{S}_{\mathsf{DG}}$: symmetric block Gauss-Seidel smoother using vertex patches
- $\mathbf{S}_{\mathsf{CG}}$: block line smoother
- $\mathbf{B}_{\mathsf{MG}}$: NG-Solve build-in MG method for high-order CG
- $\mathbf{A}_{\mathsf{DG}}^{-1} \approx \mathbf{S}_{\mathsf{DG}} + \Pi \mathbf{B}_{\mathsf{CG}} \Pi^\top$ and $\mathbf{B}_{\mathsf{CG}} = \mathbf{S}_{\mathsf{CG}} + \mathbf{B}_{\mathsf{MG}} \approx \mathbf{A}_{\mathsf{CG}}^{-1}$
- $\mathsf{tol}_{\mathsf{DG}} = 10^{-6}$ and $\mathsf{tol}_{\mathsf{CG}} = 10^{-4}$

Fast Solver Test 1: $\mathbf{b} = [0, 1]^\top$

- $\mathbf{S}_{\mathrm{DG}}$: symmetric block Gauss-Seidel smoother using vertex patches
- $\mathbf{S}_{\mathrm{CG}}$: block line smoother
- $\mathbf{B}_{\mathrm{MG}}$: NG-Solve build-in MG method for high-order CG
- $\mathbf{A}_{\mathrm{DG}}^{-1} \approx \mathbf{S}_{\mathrm{DG}} + \Pi \mathbf{B}_{\mathrm{CG}} \Pi^\top$ and $\mathbf{B}_{\mathrm{CG}} = \mathbf{S}_{\mathrm{CG}} + \mathbf{B}_{\mathrm{MG}} \approx \mathbf{A}_{\mathrm{CG}}^{-1}$
- $\mathrm{tol}_{\mathrm{DG}} = 10^{-6}$ and $\mathrm{tol}_{\mathrm{CG}} = 10^{-4}$

$D_{\parallel}$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
number of iterations for $B_{\mathrm{DG}}^{\mathrm{inexact}}$ when $h = 1/10$								
1	11	11	11	11	11	11	11	11
1E+2	12	11	10	11	11	11	11	11
1E+4	17	10	9	9	9	9	9	9
1E+6	15	13	10	9	8	7	7	7
1E+8	12	5	6	8	7	7	7	7

Fast Solver Test 2: Circular Test



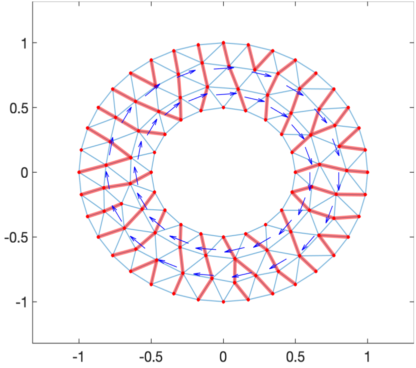


Figure: Illustration of computational Mesh level 1 (left) and line smoother (right).

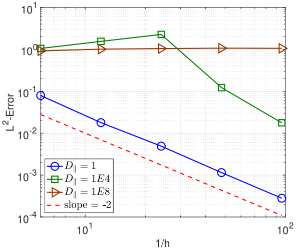
Number of Iterations for Exact Solver

$D_{\|}$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
	$h=1/6$							
1	11	10	10	10	10	10	9	9
1E+2	22	15	12	11	11	11	11	11
1E+4	50	52	33	20	13	11	10	10
1E+6	49	61	66	41	39	20	9	9
1E+8	49	66	70	52	39	13	11	11
	$h=1/12$							
1	12	12	12	12	12	12	12	12
1E+2	21	13	11	11	11	11	11	11
1E+4	86	61	27	16	12	10	10	9
1E+6	94	114	91	56	17	8	7	7
1E+8	94	122	125	41	13	11	9	9
	$h=1/24$							
1	13	12	12	12	12	12	12	12
1E+2	21	13	11	11	11	11	11	11
1E+4	111	58	22	14	12	10	10	9
1E+6	205	191	99	52	17	8	7	7
1E+8	210	249	178	41	13	11	9	9

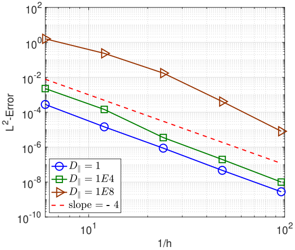
Number of Iterations for In-exact Solver

$D_{\parallel}$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
	$h=1/6$							
1	10	9	9	8	8	8	8	8
1E+2	20	13	10	10	10	10	10	10
1E+4	51	52	27	17	11	10	9	9
1E+6	54	65	64	37	24	17	8	16
1E+8	57	73	70	48	10	14	9	10
	$h=1/12$							
1	11	10	10	10	10	10	10	10
1E+2	18	12	10	10	10	10	10	10
1E+4	79	54	20	13	11	9	9	8
1E+6	98	116	84	32	19	16	8	9
1E+8	109	129	85	44	16	12	9	9
	$h=1/24$							
1	13	12	12	12	12	12	12	12
1E+2	21	13	11	11	11	11	11	11
1E+4	111	58	24	19	16	13	11	10
1E+6	205	191	122	87	19	16	8	9
1E+8	210	249	178	44	15	12	9	9

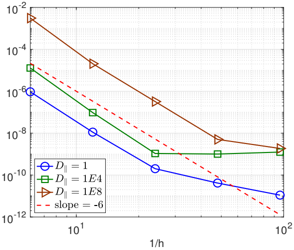
Accuracy Test for Annulus Test with Varying Values in $D_\parallel$



(a). $k = 1$



(b). $k = 3$



(c). $k = 5$

Fast Solver Test 3: WEST tokamak 2D with circular $\mathbf{b}$

A Gaussian source is diffused towards an identical sink

$$f_{sc} = D_{\|} \exp\left(-r_{sc}^2/0.05^2\right)$$
$$f_{sk} = -D_{\|} \exp\left(-r_{sk}^2/0.05^2\right),$$

where $r_{sc} = \sqrt{(x-1.5)^2 + y^2}$ and $r_{sk} = \sqrt{(x+1.5)^2 + y^2}$. The magnetic field direction is chosen as $\mathbf{b} = (b_1, b_2)^\top$ where

$$b_1 = \frac{y}{r}, \; b_2 = -\frac{x}{r}.$$

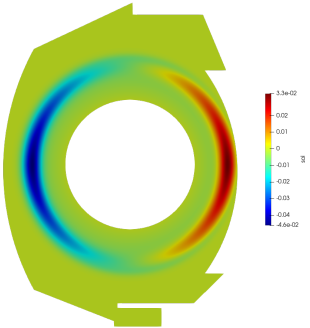
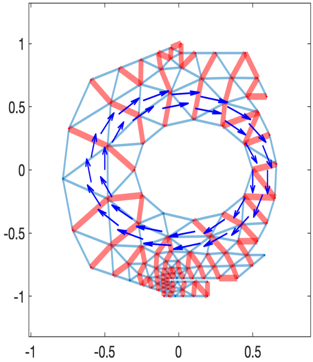
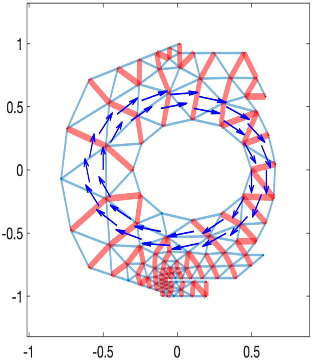


Figure: Plot of solution with $D_{\|}$ =1E4.

Illustration of line smoother



(a) η= 1.05



(b) η=1.2

Numerical Performance for Solvers

$D_{\parallel}$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
number of iterations for $\boldsymbol{B}_{\mathsf{DG}}$ when $h=1/10$								
1	8	8	8	8	8	8	7	7
1E+2	17	14	12	11	11	11	11	11
1E+4	21	20	18	15	15	14	13	13
1E+6	21	20	18	17	17	16	17	18
1E+8	21	20	18	17	17	16	15	18
number of iterations for $\boldsymbol{B}_{\mathsf{DG}}^{\mathsf{inexact}}$ when $h=1/10$								
1	8	8	8	8	8	8	7	7
1E+2	17	13	11	11	10	10	10	10
1E+4	27	26	20	18	15	13	12	12
1E+6	28	30	26	24	23	22	21	20
1E+8	28	30	29	28	26	26	25	25

Conclusions and Future Work

Conclusion

- High order scheme can resolve the numerical pollution on the non-aligned mesh
- Auxiliary Space Preconditioner (ASP) is efficient and effective in solving the linear system with large anisotropy

Future Work

- Numerical analysis
- Incorporate with M. Stowell and D. Copeland in MFEM implementation
- More applications with complicated magnetic fields

Thank you!

Reference

- D. Green, X. Hu, J. Lore, L. Mu, and M. Stowell, An Efficient High-order Numerical Solver for Diffusion Equations with Strong Anisotropy, CPC, 276, 2022.
- D. Green, X. Hu, J. Lore, L. Mu, and M. Stowell, An Efficient High-order Solver for Diffusion Equations with Strong Anisotropy on Non-anisotropy-aligned Meshes, Submitted.