Half-closed Discontinuous Galerkin discretisations[1]

Department of Mathematics, University of California, Berkeley
Mathematics Department, Lawrence Berkeley National Laboratory

WCCM-PANACM Vancouver 2024

July 24, 2024

BERKELEY LAB
Bringing Science Solutions to the World

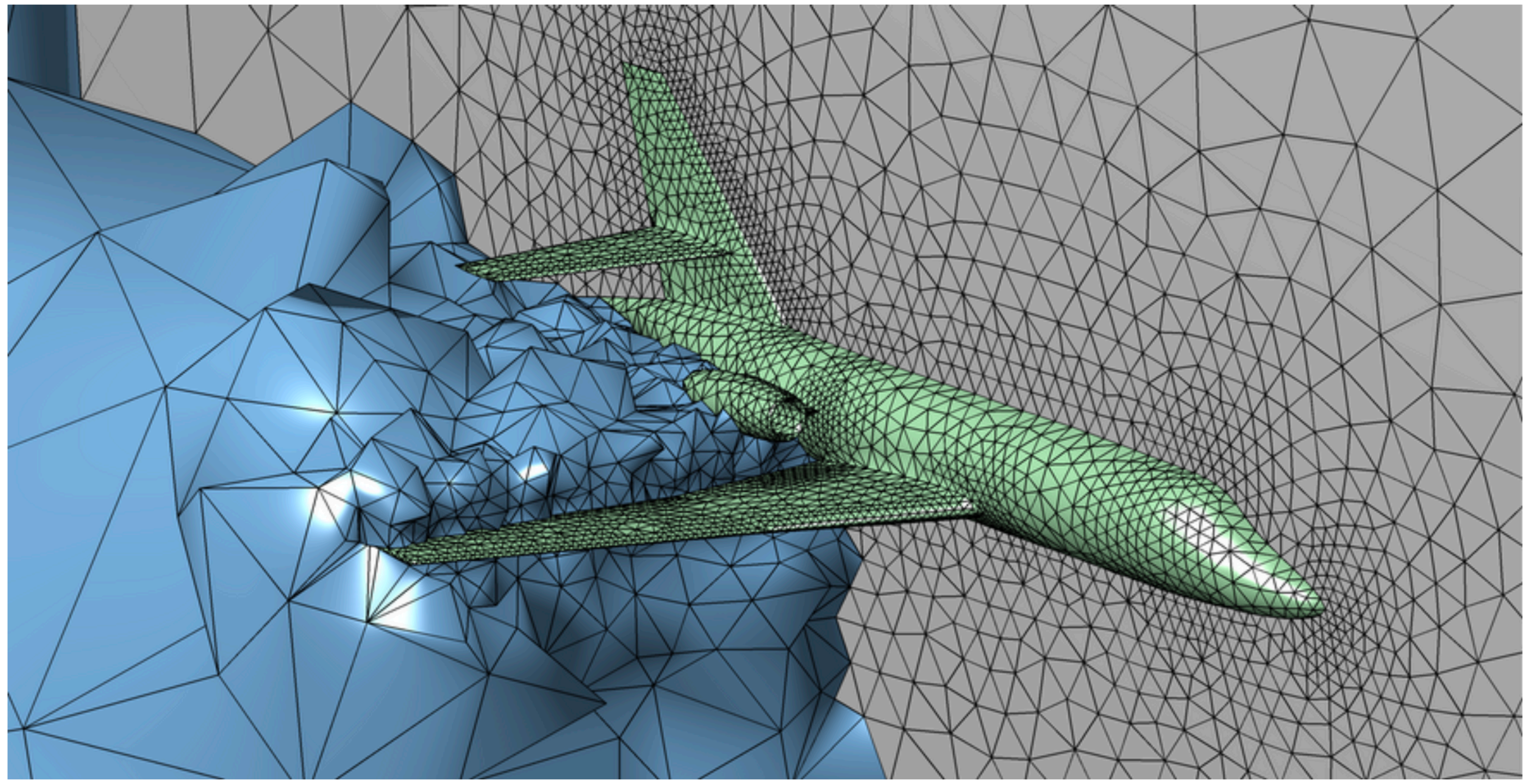
[1]Yulong Pan and Per-Olof Persson. Half-closed discontinuous galerkin discretisation. Submitted to Journal of Computational Physics. https://arxiv.org/abs/2405.12383

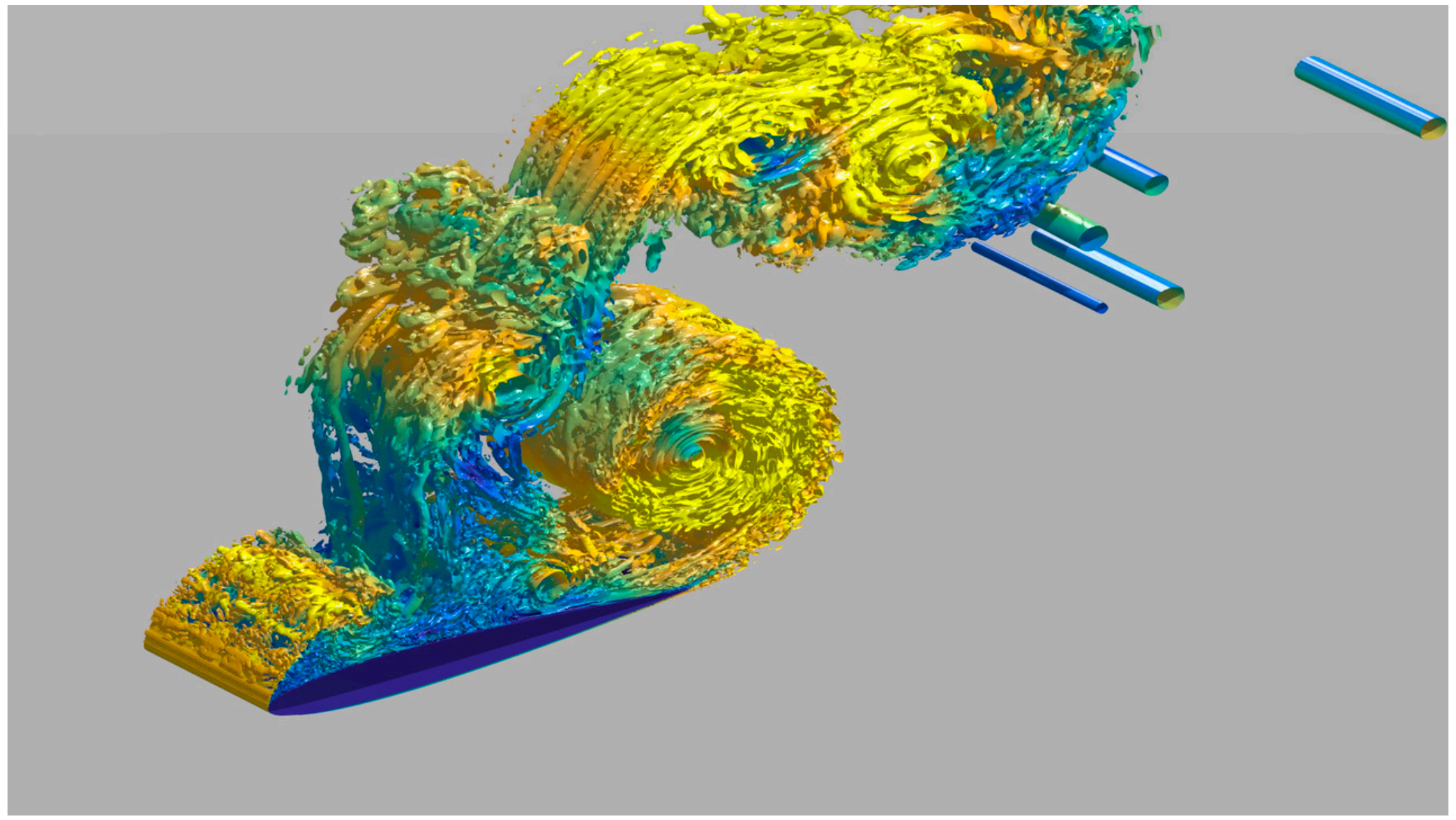
High-order methods for unstructured meshes

- Widely believed that high-order accurate methods will be required for challenging simulations of turbulent flows, wave propagation, multiscale phenomena, etc.

- In addition, fully unstructured meshes are necessary to handle complex geometries, with adapted resolution and full automation

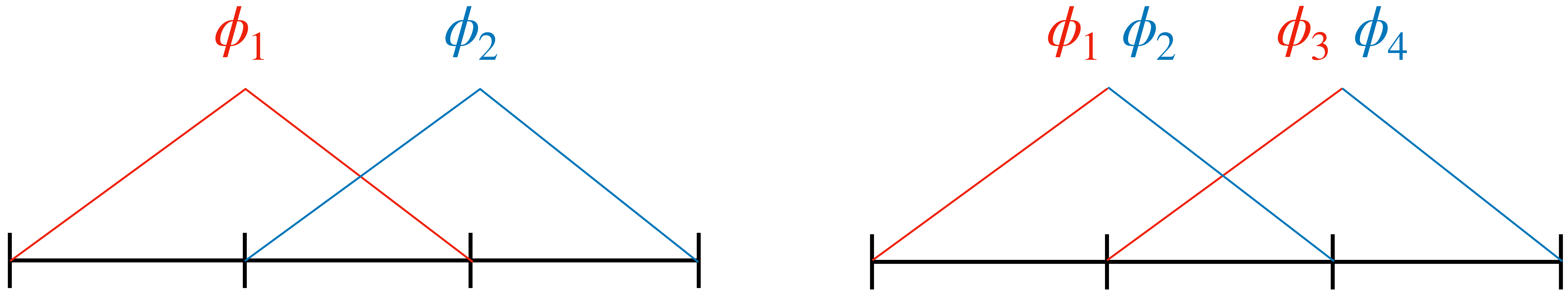
- Goal: Develop robust, efficient, and accurate high-order methods based on fully unstructured meshes





Discontinuous Galerkin methods

- One popular high-order method is the Discontinuous Galerkin (DG) method
- Extension of the Finite Element method (FEM) to allow for discontinuous solutions, with numerical fluxes from Finite Volumes



Discontinuous Galerkin methods

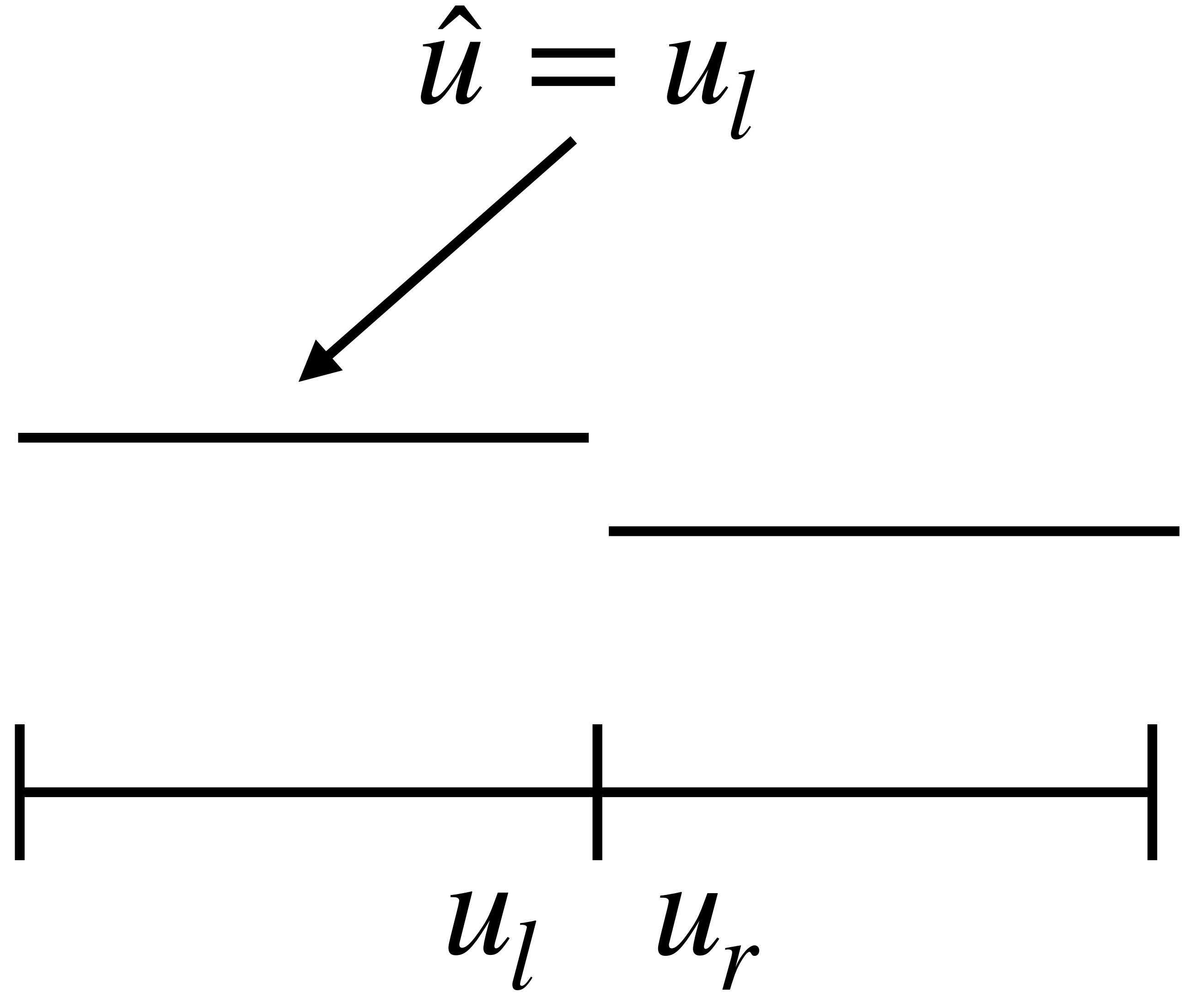
<u>Convection equation:</u>

$$u_t + \alpha \cdot \nabla u = 0$$

$$\sum_E \int_E v\ (u_t + \alpha \cdot \nabla u) = 0$$

$$\sum_E \partial_t \int_E vu\ dx + \int_{\partial E} v\hat{u}\alpha \cdot \mathbf{n} ds - \int_E \alpha \cdot \nabla vu\ dx = 0$$

$$\partial_t M\mathbf{u} + \alpha G\mathbf{u} = 0$$



Discontinuous Galerkin methods

Poisson's equation (LDG[1]):

$$-\nabla^2 u = f$$

$$q = \nabla u \qquad -\nabla \cdot q = f$$

$$Mq = Gu \quad -Dq = Mf$$

$$-D^d = (G^d)^T \qquad \text{(adjoint consistency)}$$

$$-\underbrace{\sum_d D^d M^{-1} G^d}_{L} u = f$$

[1]Bernardo Cockburn and Chi-Wang Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal. , 35(6):2440–2463, 1998

Discontinuous Galerkin methods

Pros:

- Stabilisation with Riemann solvers
- Easy to attain high-order accuracy by increasing polynomial order
- Block structured operators
- Easy to parallelise

Cons:

- Computational cost
    - Repeated boundary nodes
    - Interpolation for assembly
    - Flux calculations
- Decreased accuracy compared with FEM
    - Second order operators

Discontinuous Galerkin methods

Pros:

- Stabilisation with Riemann solvers
- Easy to attain high-order accuracy by increasing polynomial order
- Block structured operators
- Easy to parallelise

Cons:

- Computational cost
  - Repeated boundary nodes
  - Interpolation for assembly
  - Flux calculations
- Decreased accuracy compared with FEM
  - Second order operators

Is there a way to remedy (at least partially) some of these issues?

Landscape of high-order methods

- There have been numerous developments of other high-order methods to address these issues, to name a few:
  - DG-Spectral Element Method (DGSEM)[1]
  - Spectral Differences[2]/Flux Reconstruction[3]
  - Line-based Discontinuous Galerkin[4]
- Challenges however remain with each method
- More work still required

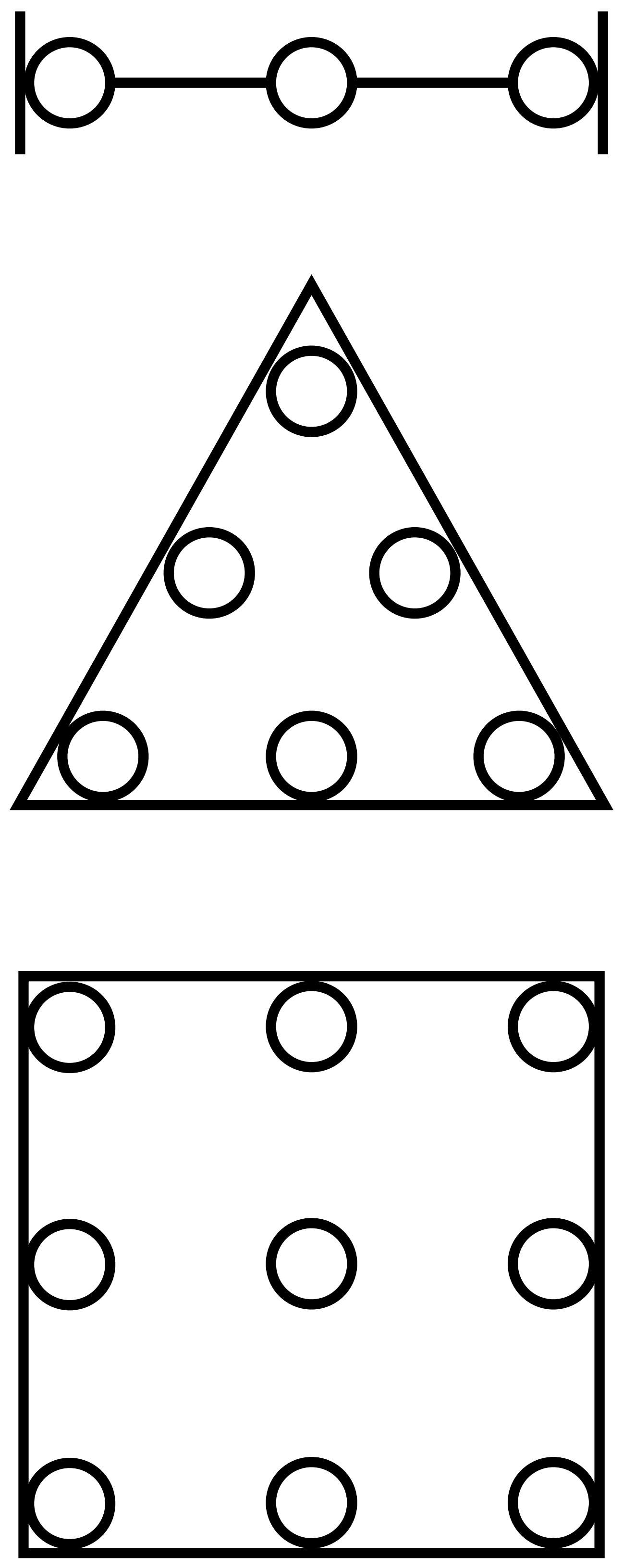
[1]David A Kopriva and John H Kolias. A conservative staggered-grid chebyshev multidomain method for compressible flows. Journal of Computational Physics, 125(1):244–261, 1996.
[2]Yen Liu, Marcel Vinokur, and Zhi Jian Wang. Spectral difference method for unstructured grids I: Basic formulation. Journal of Computational Physics, 216(2):780–801, 2006.
[3]Hung T Huynh. A flux reconstruction approach to high-order schemes including discontinuous galerkin methods. In 18th AIAA Computational Fluid Dynamics Conference, page 4079, 2007.
[4]Per-Olof Persson. A sparse and high-order accurate line-based discontinuous galerkin method for un-structured meshes. Journal of Computational Physics, 233:414–429, 2013.

Half-closed DG

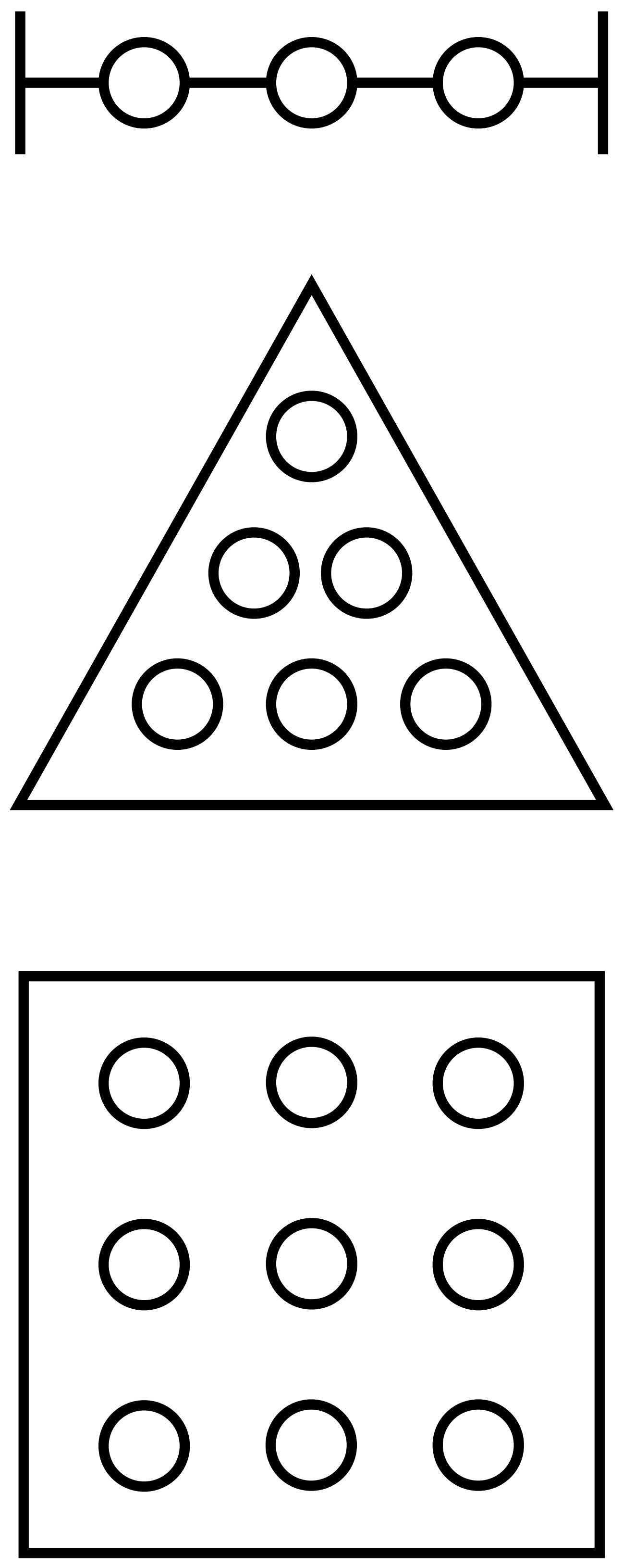
Half-closed nodes

- Commonly in nodal DG, nodes are placed either on all element boundaries or none of them

Closed, e.g. Gauss-Lobatto



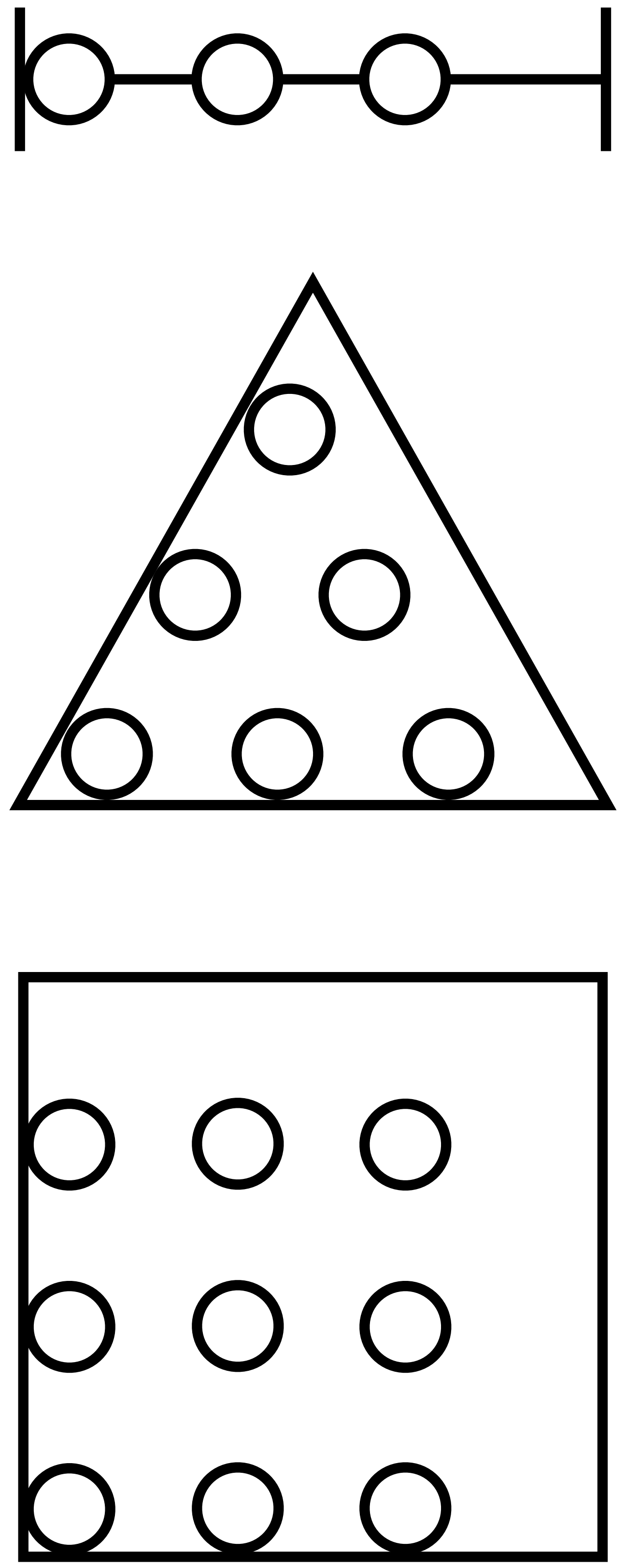
Open, e.g. Gauss-Legendre



Half-closed nodes

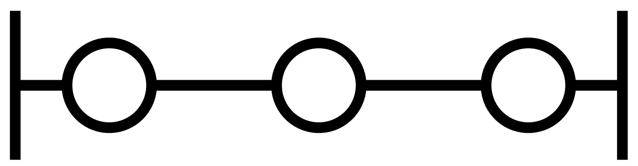
- Half-closed nodes are placed on a subset of element boundaries, on block elements (quads/hexes) they are placed on exactly half of them

Half-closed, e.g. Gauss-Radau

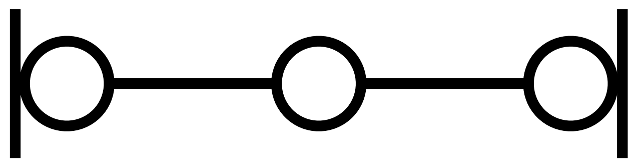


Quadrature precision

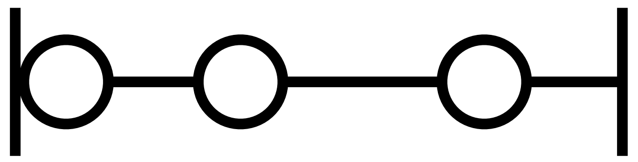
- Relaxing the constraint of requiring nodes on all boundaries allows for increased freedom in node placement
- Consider 1D for now for simplicity, but easy to generalise (e.g. outer products)
- This extra degree of freedom enables higher quadrature precision, with n points
  - Open: Gauss-Legendre attains 2n-1



  - Closed: Gauss-Lobatto attains 2n-3



  - Half-closed: Gauss-Radau attains 2n-2



Nodal integration

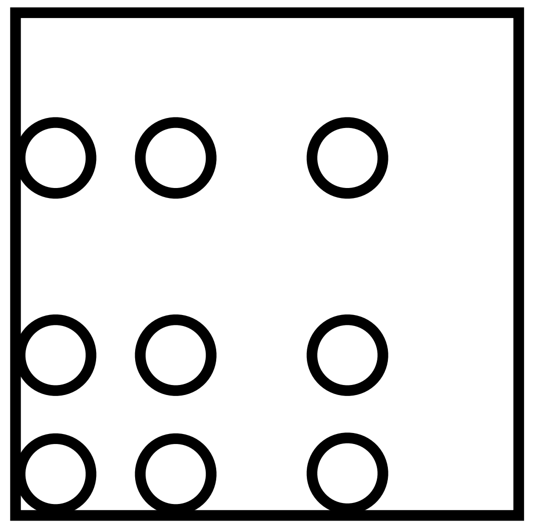
- Half-closed: n Gauss-Radau points → quadrature precision = 2n-2
- For example the mass matrix, degree p element, p+1 nodes

$$
\begin{aligned}
M_{ij} &= \int_E \phi_i \phi_j dx && \text{(integrand degree = 2p)} \\
&= \sum_k w_k \phi_i(x_k) \phi_j(x_k) && \text{(exact)} \\
&= w_i \delta_{ij}
\end{aligned}
$$

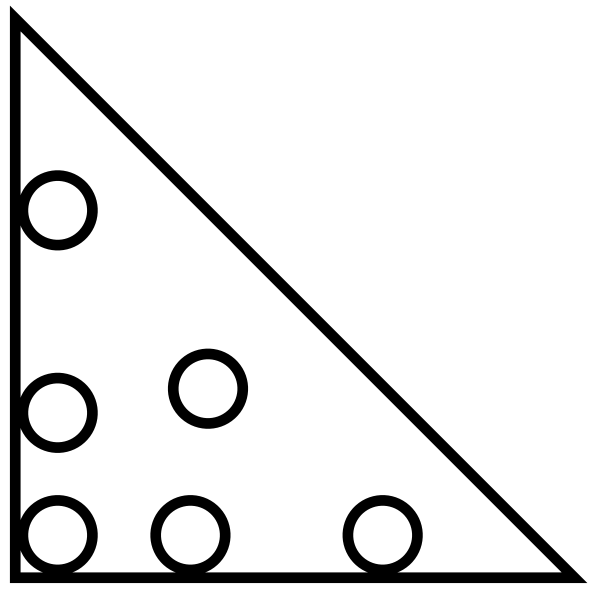
- (Exact) diagonal mass matrix without any interpolation required for assembly
- Extra degree of precision over Gauss-Lobatto quadrature

Half-closed nodes

- For block elements, half-closed nodes we focus on are Gauss-Radau nodes
    - For higher dimensions, outer product of 1D nodes
    - Boundary nodes placed on half of element boundaries



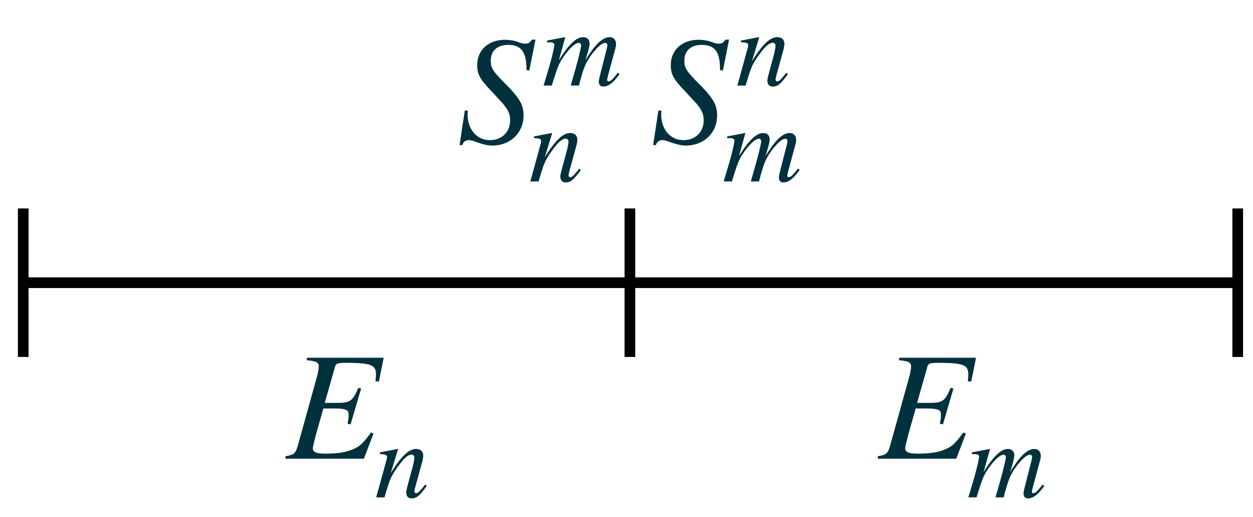
- For simplex elements more complicated, will not focus on for this talk (see publication)
    - For degree p simplex, $\binom{p+d}{d}$ nodes, insufficient to integrate degree 2p polynomial
    - More complex construction required
    - Boundary nodes placed on d boundaries



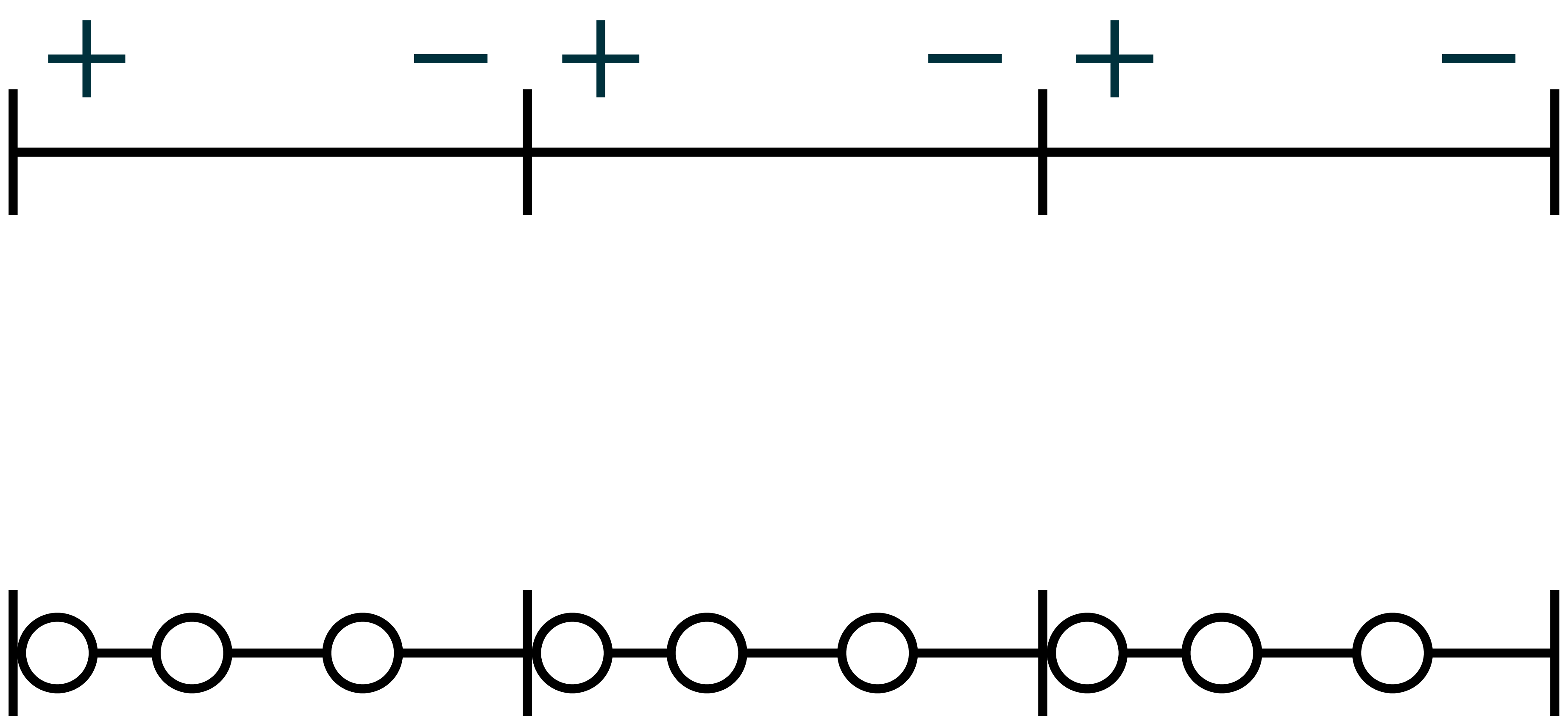
Half-closed node placement

- As nodes are not symmetric, need to determine how they are placed
- Simple procedure using switch functions from Local Discontinuous Galerkin method (LDG)

<u>Switch function</u>



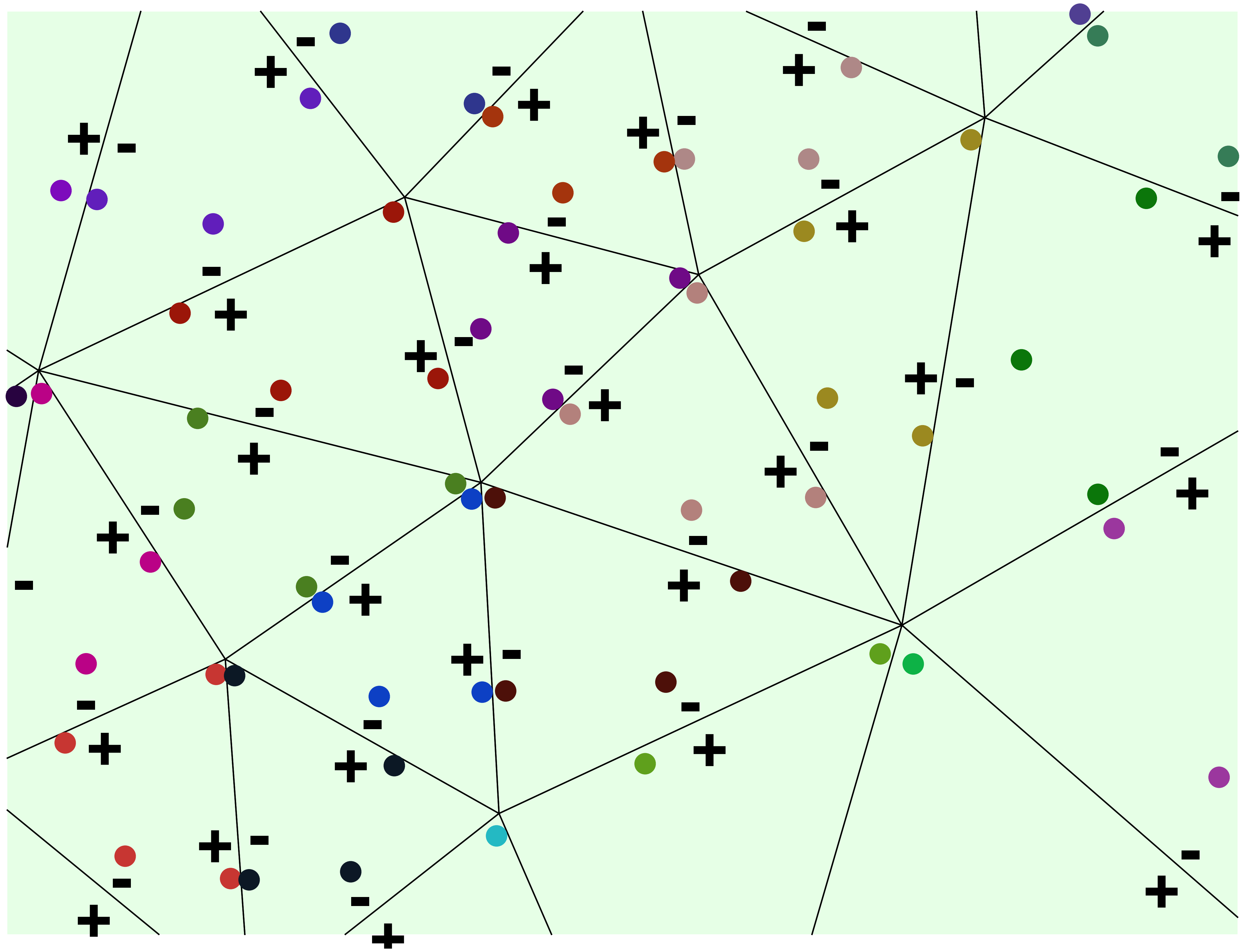
$$S_n^m = \{-1, 1\} \qquad S_n^m + S_m^n = 0$$

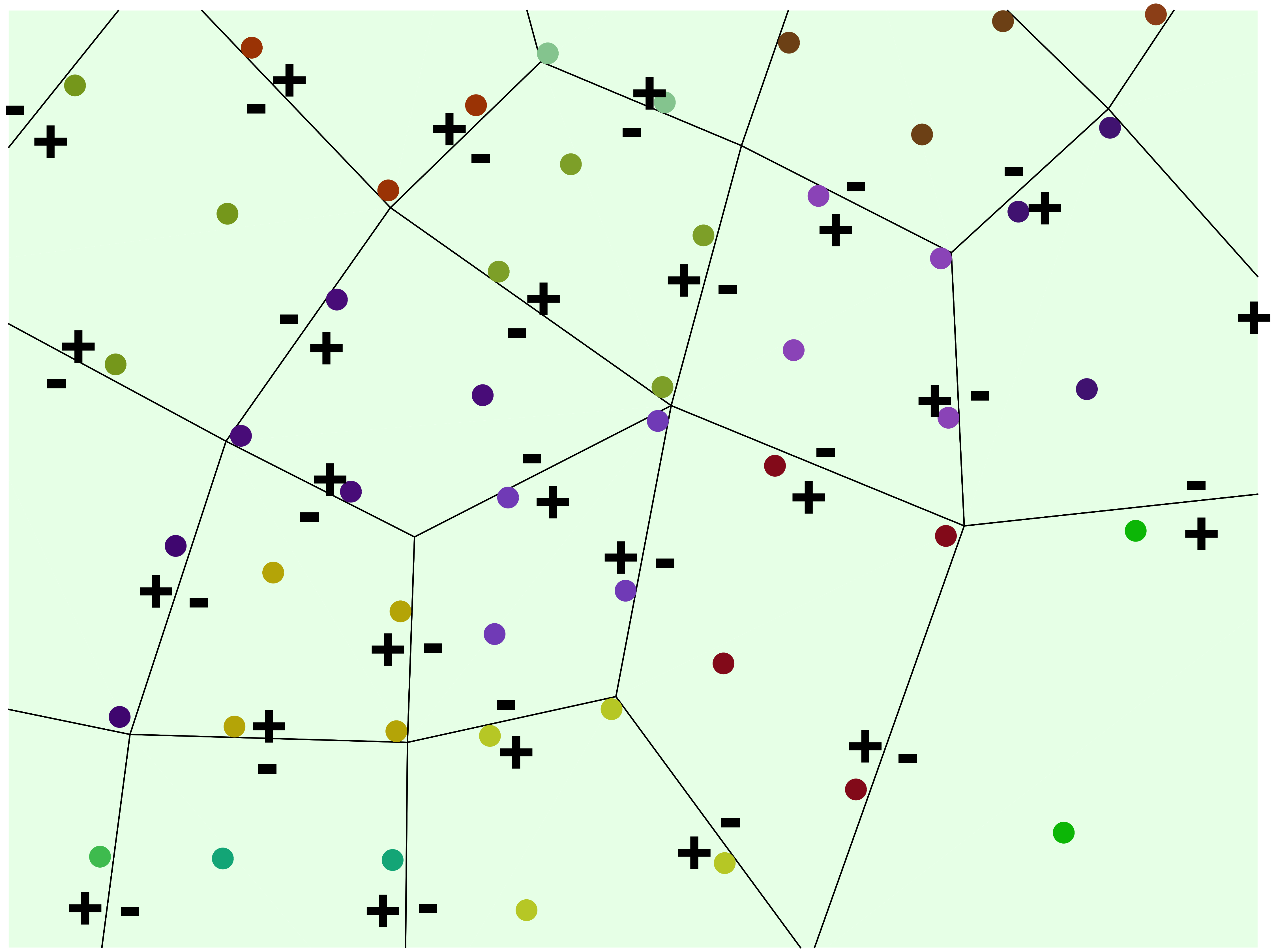


- Boundary nodes are placed on edges where the switch is equal to +1

Half-closed node placement

- Boundary nodes are placed on edges where the switch is equal to +1





Operator sparsity

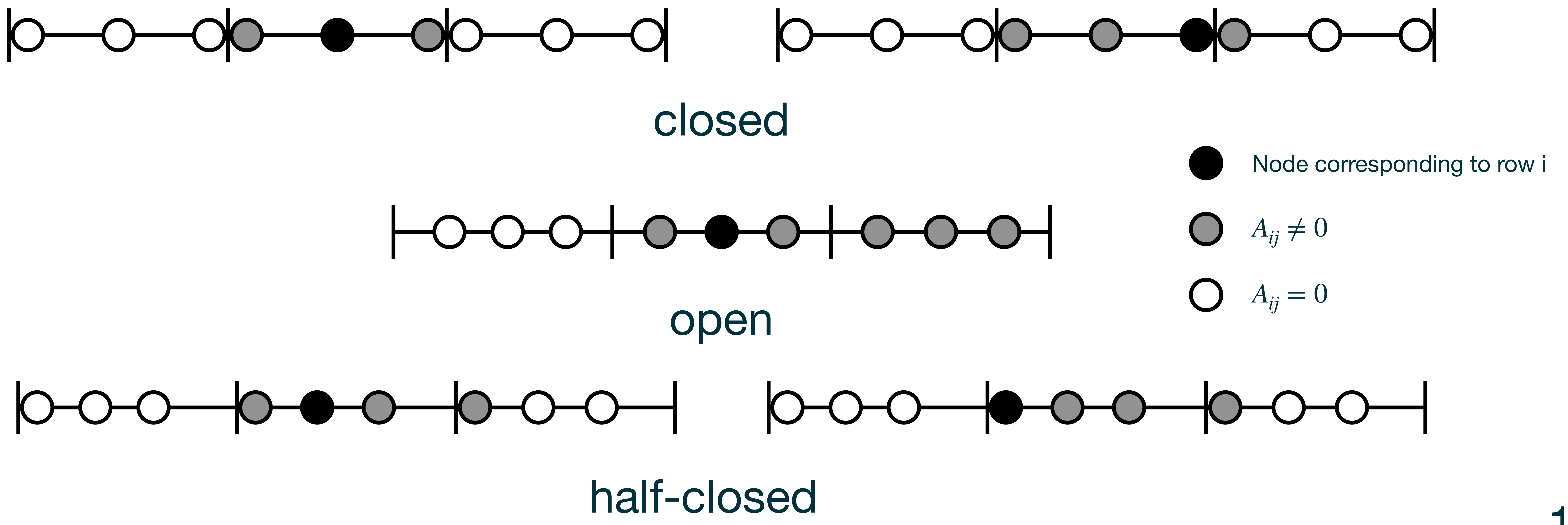
- How does choice of nodes affect DG operator sparsity?
  - Operators of interest:
    - Mass matrix - M - diagonal
    - Gradient operator - G
    - Divergence operator - D

      (1st order)
    - Laplace operator (using LDG) - $L = DM^{-1}G$ (2nd order)

Divergence operator

- For the divergence operator, comparing the three types of nodes

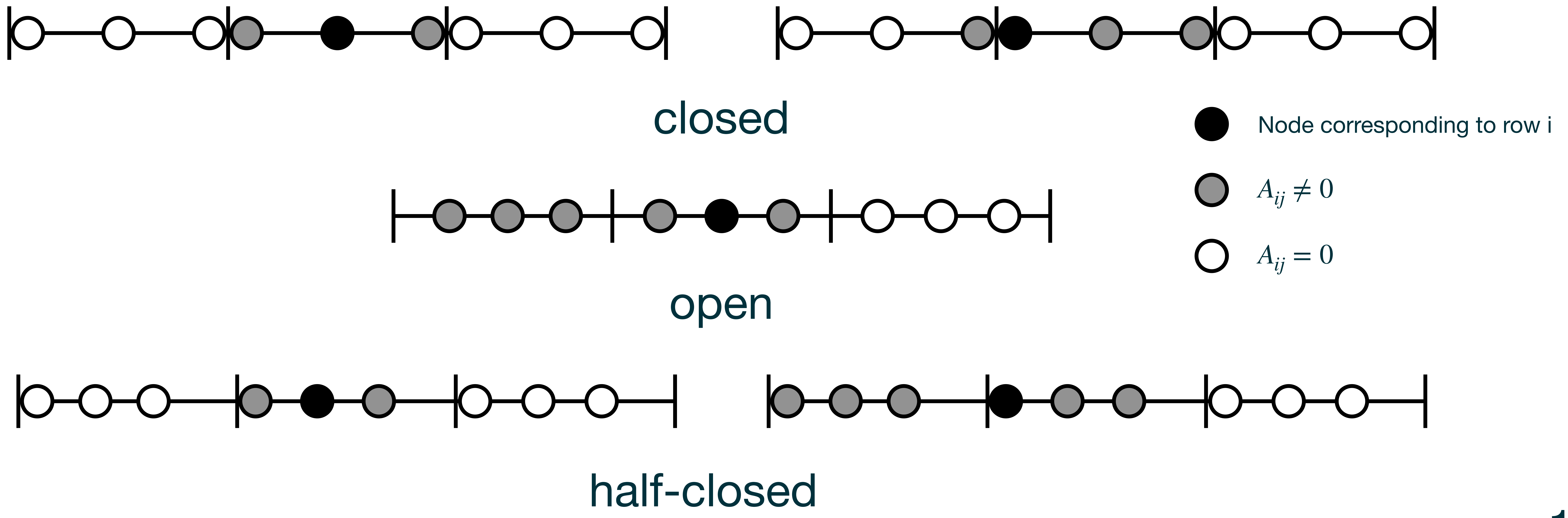
$$D_{ij} = \sum_E \int_E \frac{\partial \phi_i}{\partial x_d} \phi_j dx - \int_{\partial E} \phi_i \hat{\phi}_j n_d ds \qquad \hat{\phi}_j = \phi_r$$



Gradient operator

- For the gradient operator, comparing the three types of nodes

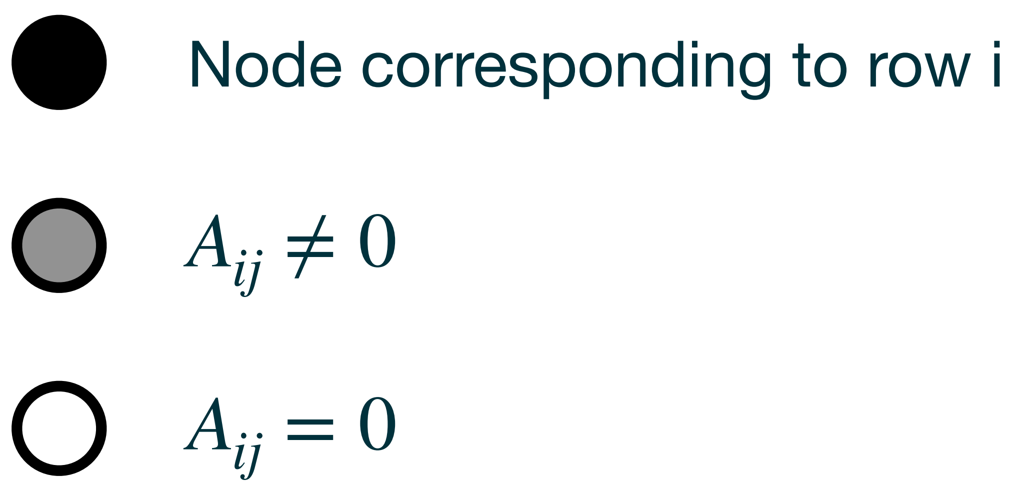
$$G_{ij} = \sum_E \int_E \frac{\partial \phi_i}{\partial x_d} \phi_j dx - \int_{\partial E} \phi_i \hat{\phi}_j n_d ds \qquad \hat{\phi}_j = \phi_l$$



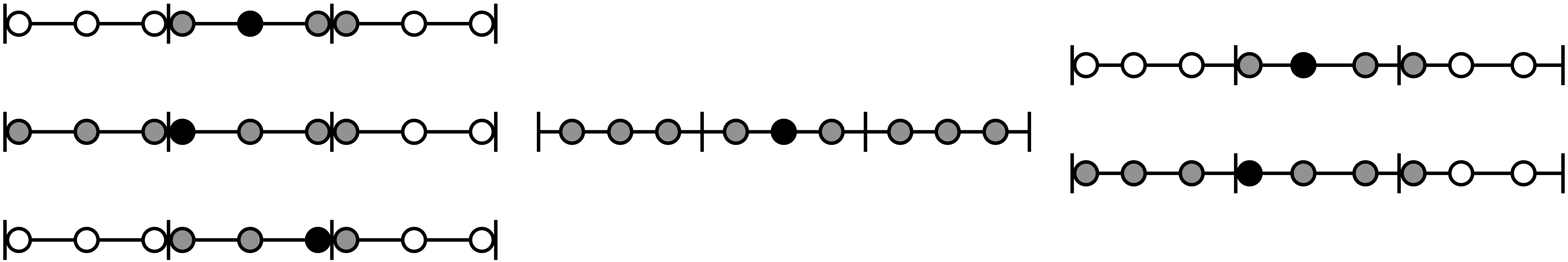
Laplace operator

- Overall for the Laplace operator, comparing the different types of nodes

$$L = DM^{-1}G$$



closed open half-closed

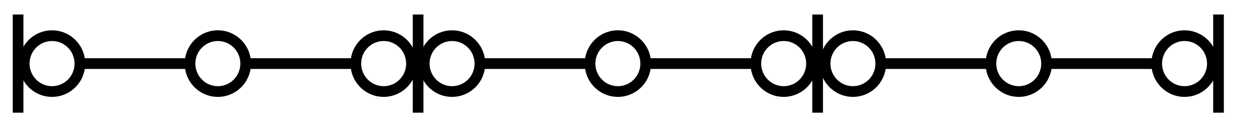


Operator sparsity

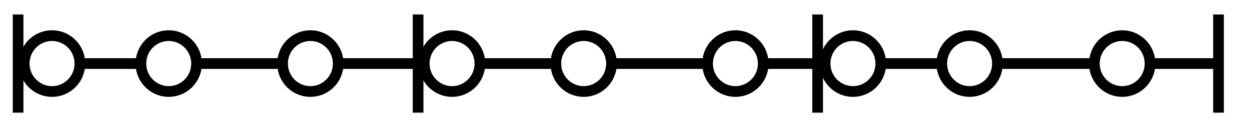
- Sparsity of gradient/divergence operators
    - closed > half-closed > open
- Sparsity of Laplace operator
    - closed = half-closed > open
- For all nodes
    - (sparsity pattern gradient/divergence) $\subset$ (sparsity pattern laplacian)

Numerical tests - linear advection

$$u_t + \alpha \cdot \nabla u = 0 \quad \Omega = [-1,1]$$



Gauss-Lobatto

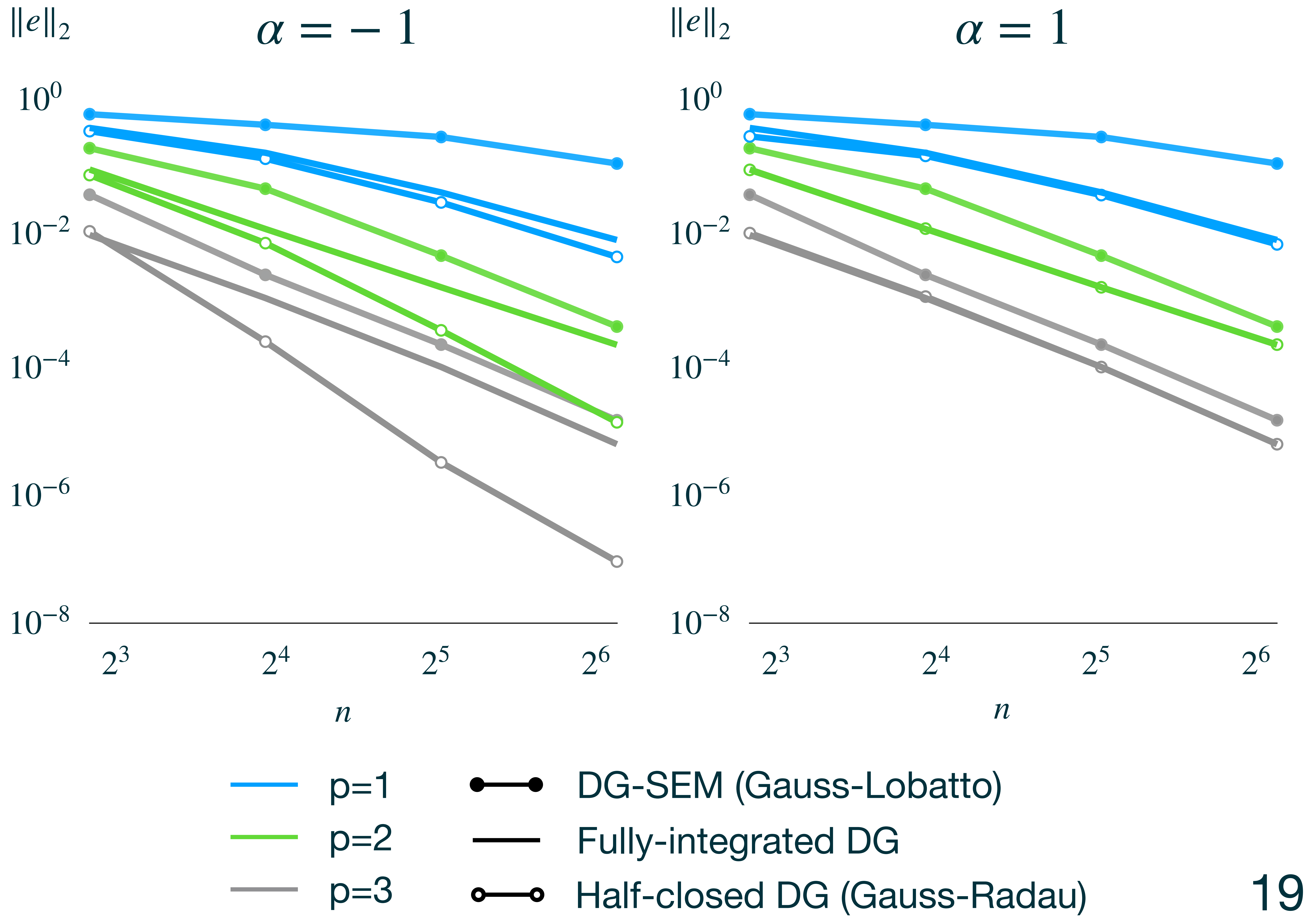


Gauss-Radau

Periodic, RK4

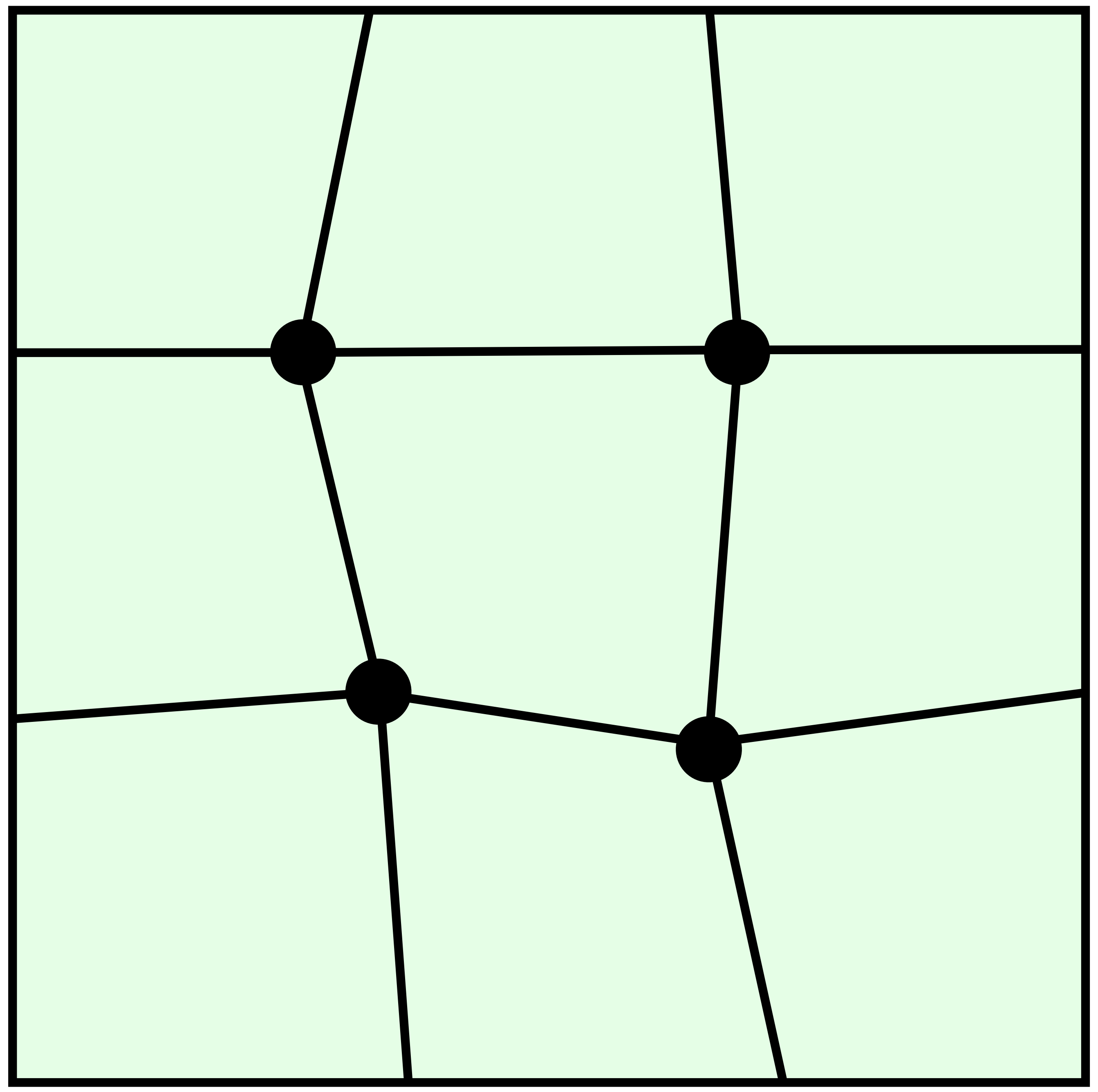
$$u_0(x) = e^{-20x^2}$$

$$T = 2.0 \qquad \Delta t = 10^{-3}$$

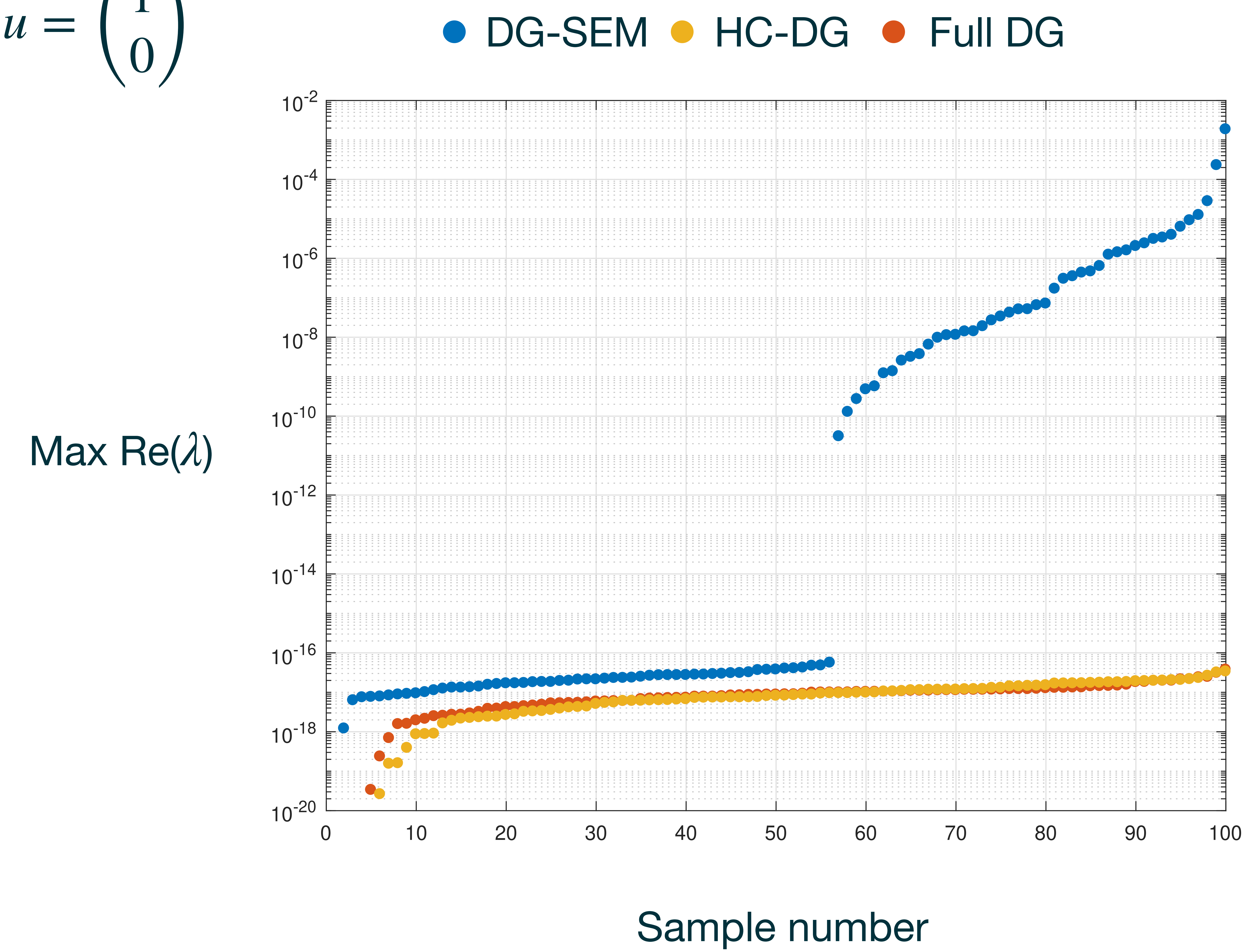


Linear stability

- Stability for linear convection, $u = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

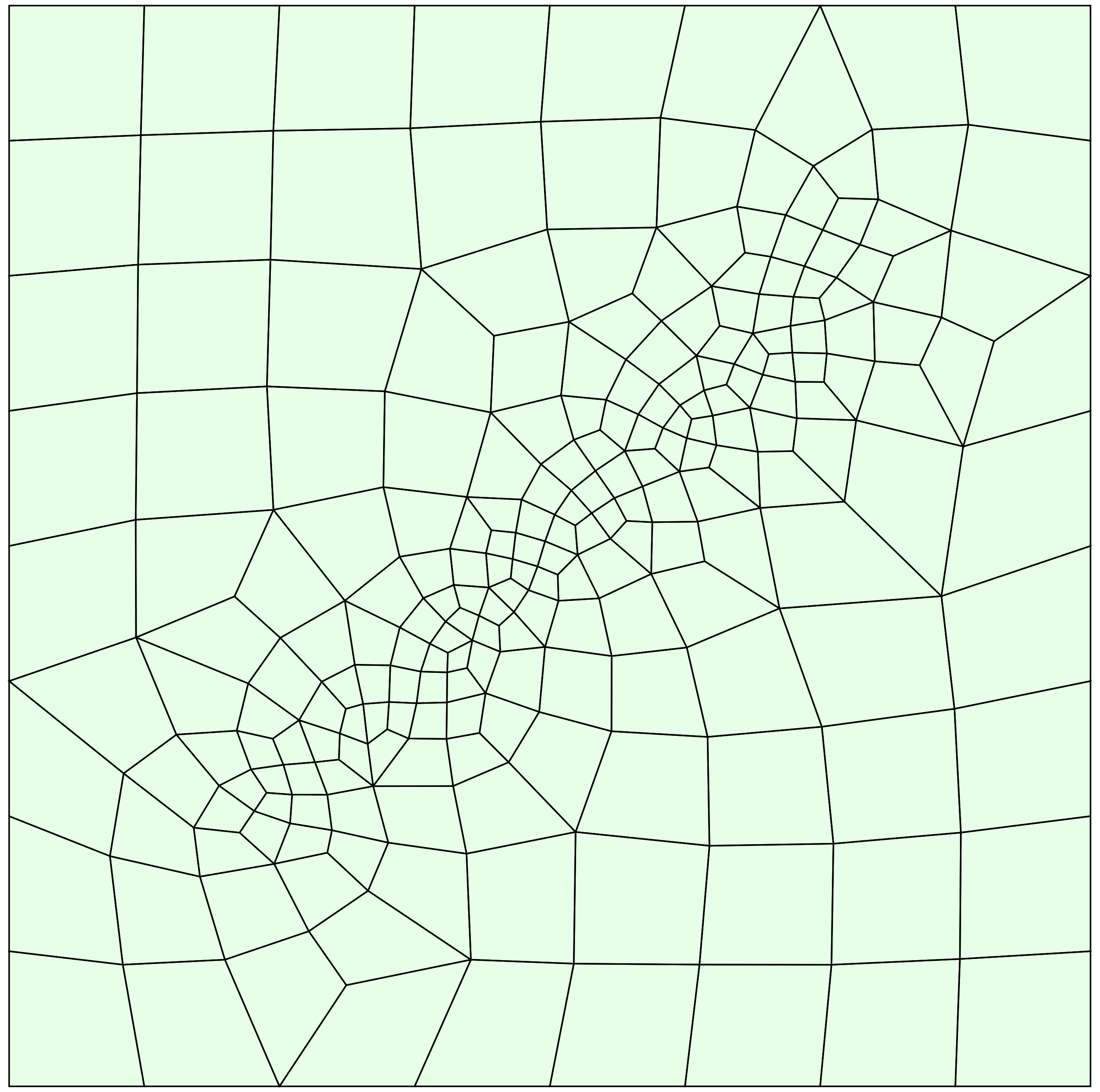


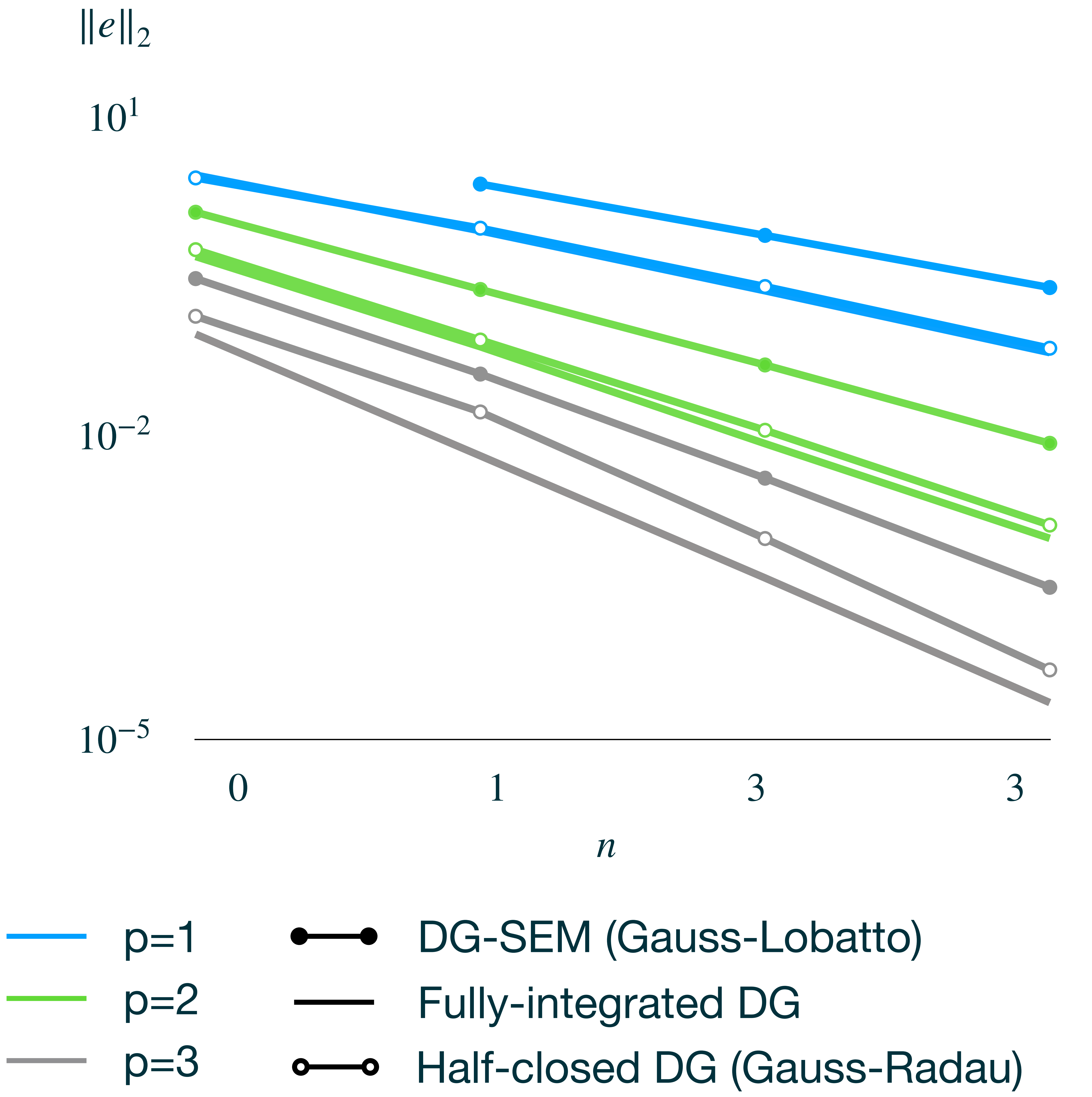
Randomise position of central points



Numerical tests - Euler vortex

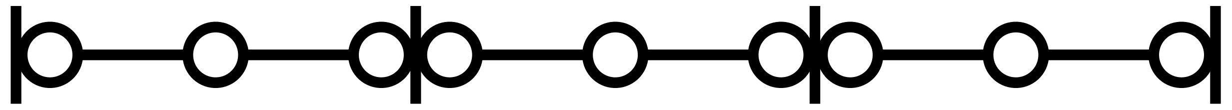
$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho u_x \\ \rho u_y \\ \rho E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u_x \\ \rho u_x^2 \\ \rho u_x u_y \\ u_x(\rho E + p)\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho u_y \\ \rho u_x u_y \\ \rho u_y^2 \\ u_y(\rho E + p)\end{pmatrix} = 0$$



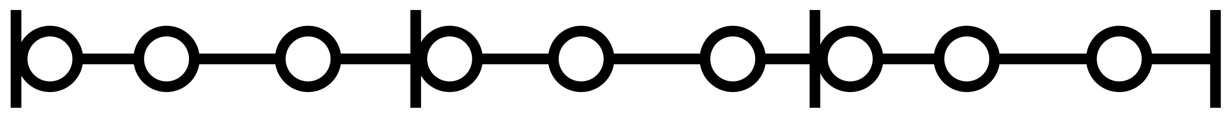


Numerical tests - Poisson's equation

$$-\nabla \cdot \nabla u = f \quad \Omega = [-1,1]$$

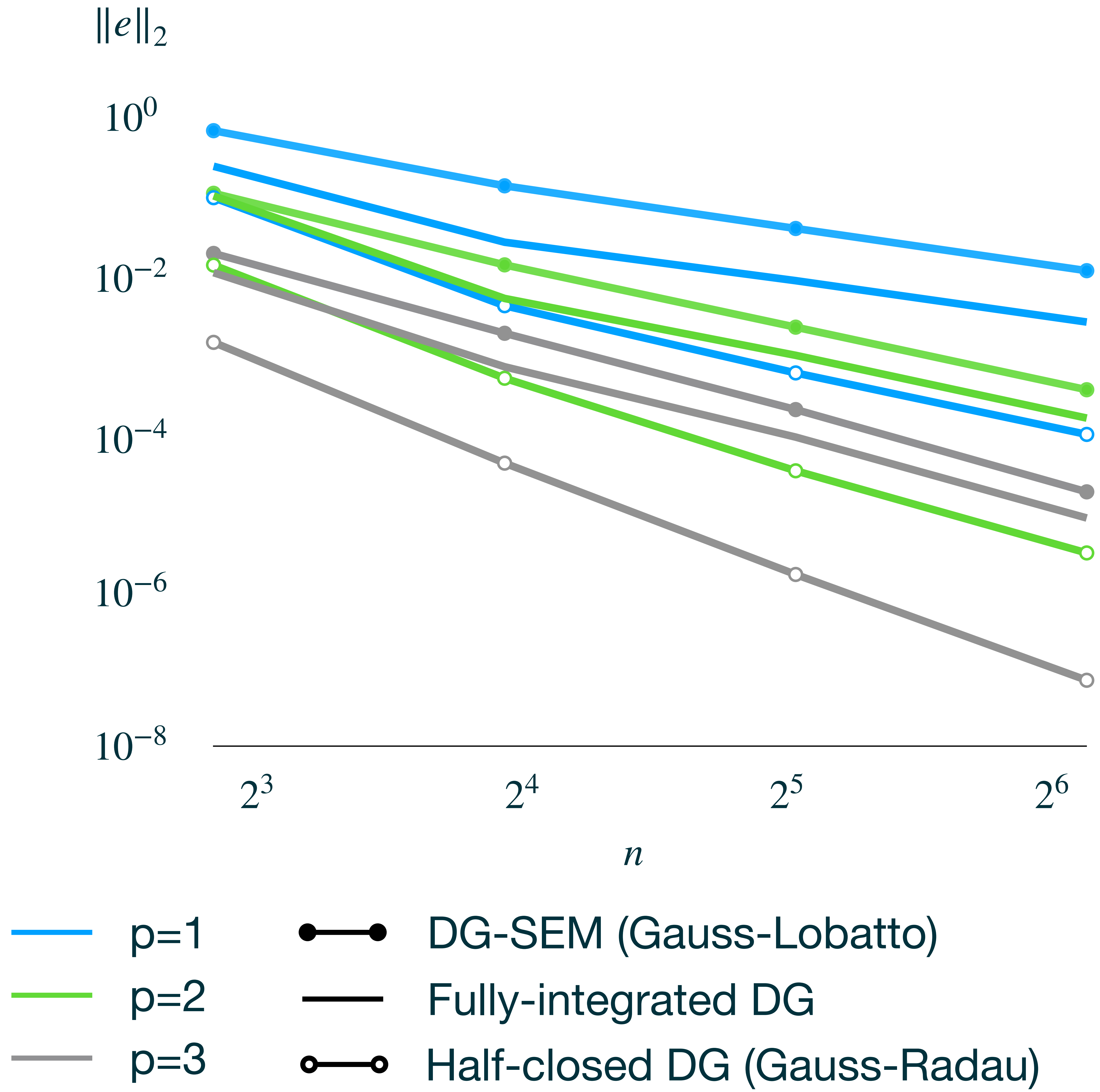


Gauss-Lobatto



Gauss-Radau

$$u(x) = e^{\sin(x)} - 1$$

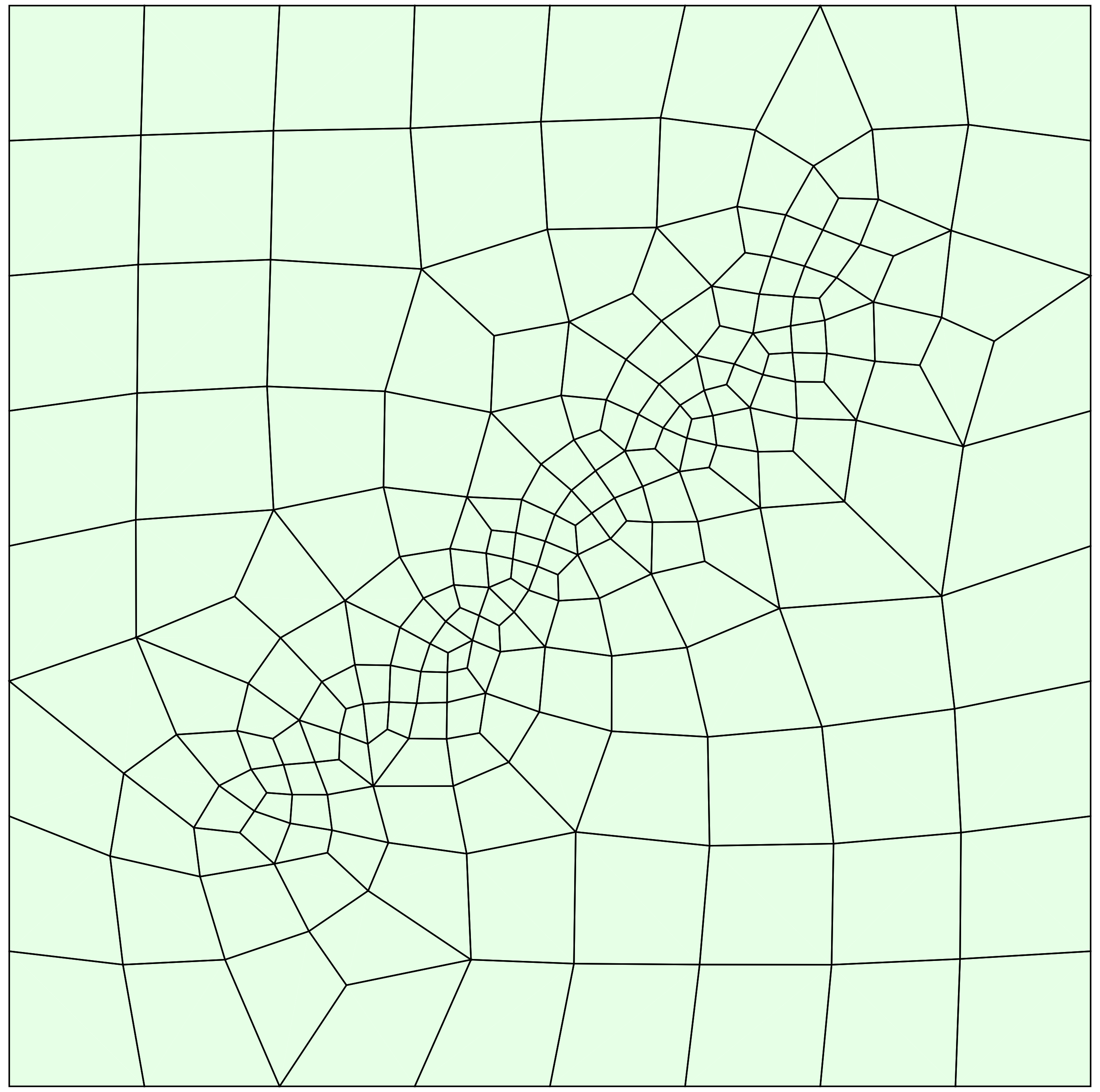


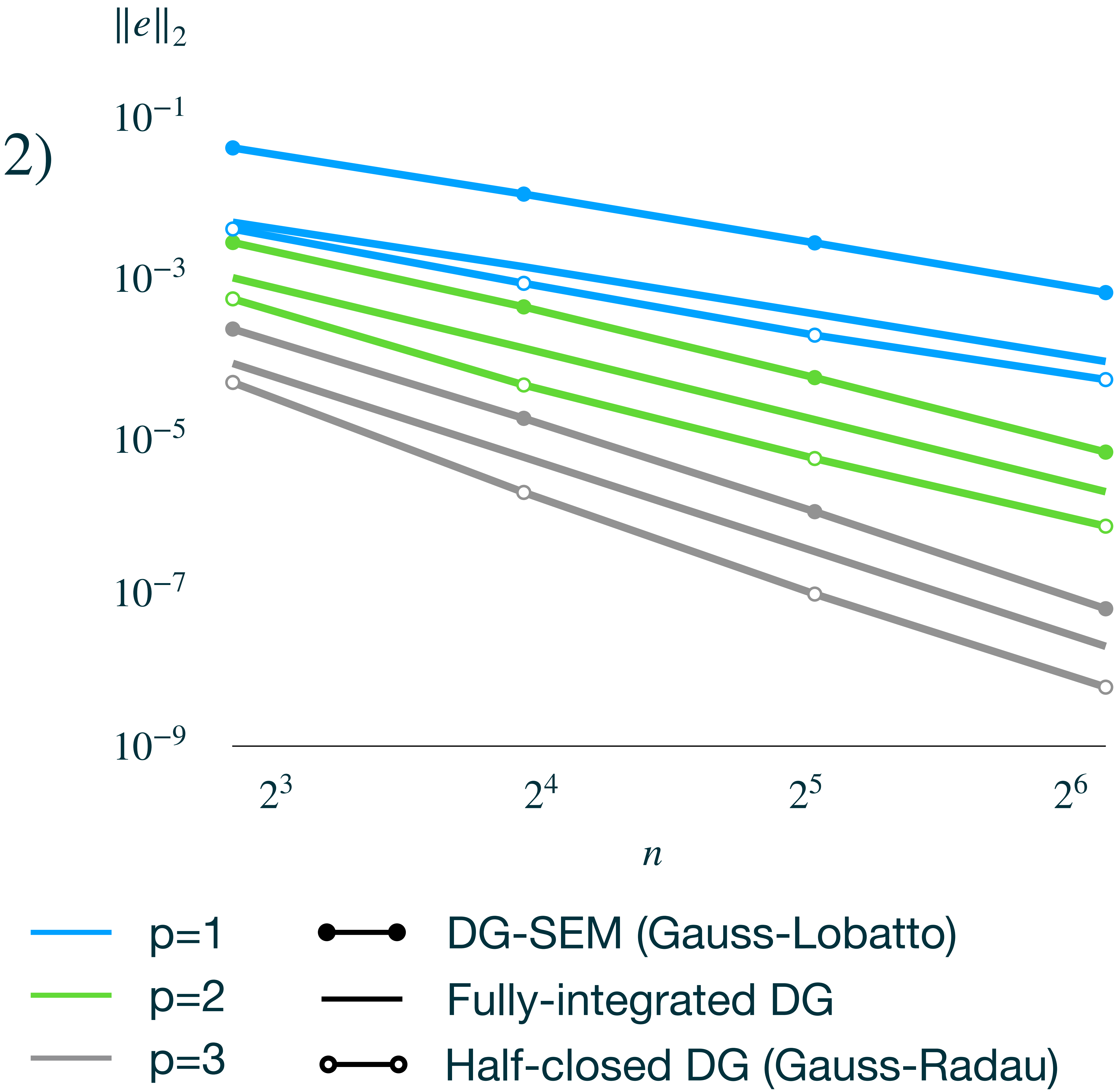
Numerical tests - Poisson's equation

$$-\nabla \cdot \nabla u = f \quad \Omega = [-1,1]^2$$

$$u(x, y) = G(x, y; 0.25, 0.2) + G(x, y; 0.75, 0.2)$$

$$G(x, y; a, r) = \exp\left(\frac{1}{r}(x-a)^2(y-a)^2\right)$$





Summary - properties

- Half-closed nodes place nodes only on half the boundaries of each element
    - Define switch function, then place nodes on all edges of element where switch value $S_n^m = +1$
    - Gauss-Radau quadrature extra integration order over Gauss-Lobatto quadrature
        - Nodal integration possible - exact diagonal mass matrix
- Sparsity of DG operators:
    - Grad/Div: closed > half-closed > open
    - Laplacian: closed = half-closed > open
    - (Sparsity Grad/Div) $\subset$ (Sparsity Laplacian)

Summary - numerical tests

- Half-closed DG with GR nodes attains similar accuracy to fully integrated DG for convection dominated systems
    - Improved accuracy + stability over nodally integrated DG on Gauss-Lobatto nodes
- Half-closed DG with GR nodes attains greater accuracy than standard DG for diffusion dominated systems
    - Gauss-Radau projections
- Half-closed DG with GR nodes recovers linear stability of fully integrated DG

Linear solvers

Linear solvers

- Recall similar sparsity patterns for DG operators with closed/half-closed nodes
- Construct solvers to take advantage of sparsity structure
    - **Techniques described from this point apply to both closed & half-closed**
- Two main techniques focused on:
    - Static condensation/Guyan reduction
    - Block methods (e.g. block Jacobi, block Gauss-Seidel, …)

Static condensation

- Commonly used with Finite Element methods

- Linear system $Ax = b$, split unknowns in independent/dependent degrees of freedom

$$\begin{pmatrix} A_{ii} & A_{id} \\ A_{di} & A_{dd} \end{pmatrix} \begin{pmatrix} x_i \\ x_d \end{pmatrix} = \begin{pmatrix} f_i \\ f_d \end{pmatrix}$$

- Eliminate dependent degrees of freedom via Schur complement

$$A_{ii}x_i + A_{id}x_d = f_i \qquad A_{di}x_i + A_{dd}x_d = f_d$$

$$x_d = A_{dd}^{-1}f_d - A_{dd}^{-1}A_{di}x_i$$

$$\rightarrow \left(A_{ii} - A_{id}A_{dd}^{-1}A_{di}\right)x_i = f_i - A_{id}A_{dd}^{-1}f_d$$

Static condensation

- Schur complement
  - Solve smaller system for independent degrees of freedom

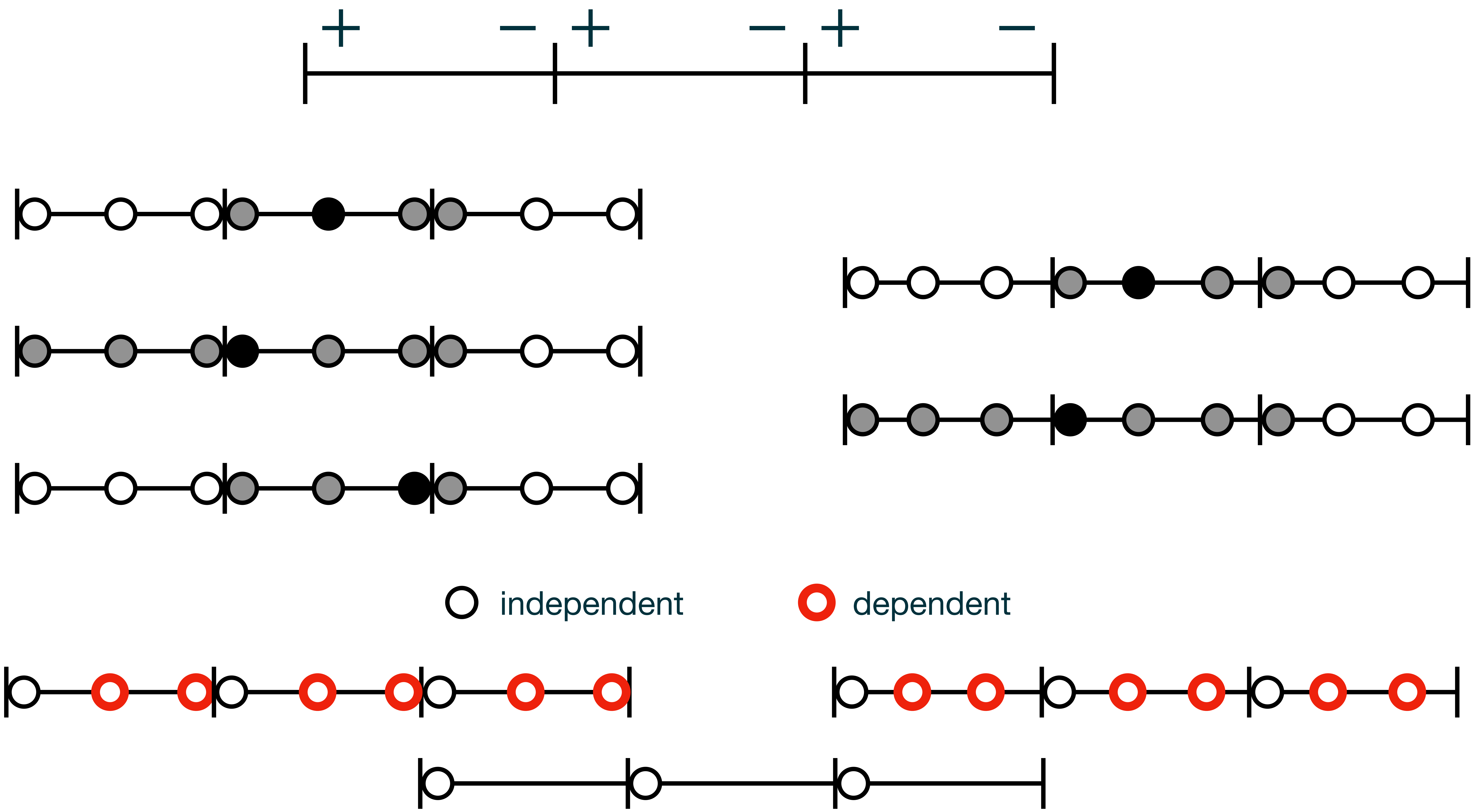
$$Sx_i = \left(A_{ii} - A_{id}A_{dd}^{-1}A_{di}\right)x_i = f_i - A_{id}A_{dd}^{-1}f_d$$

  - Solve for dependent degrees of freedom

$$x_d = A_{dd}^{-1}f_d - A_{dd}^{-1}A_{di}x_i$$

- If A_{dd}^{-1} sparse then S also sparse (in particular if A_{dd} block-diagonal)
- Pick independent and dependent nodes such that A_{dd} is block-diagonal

Static condensation



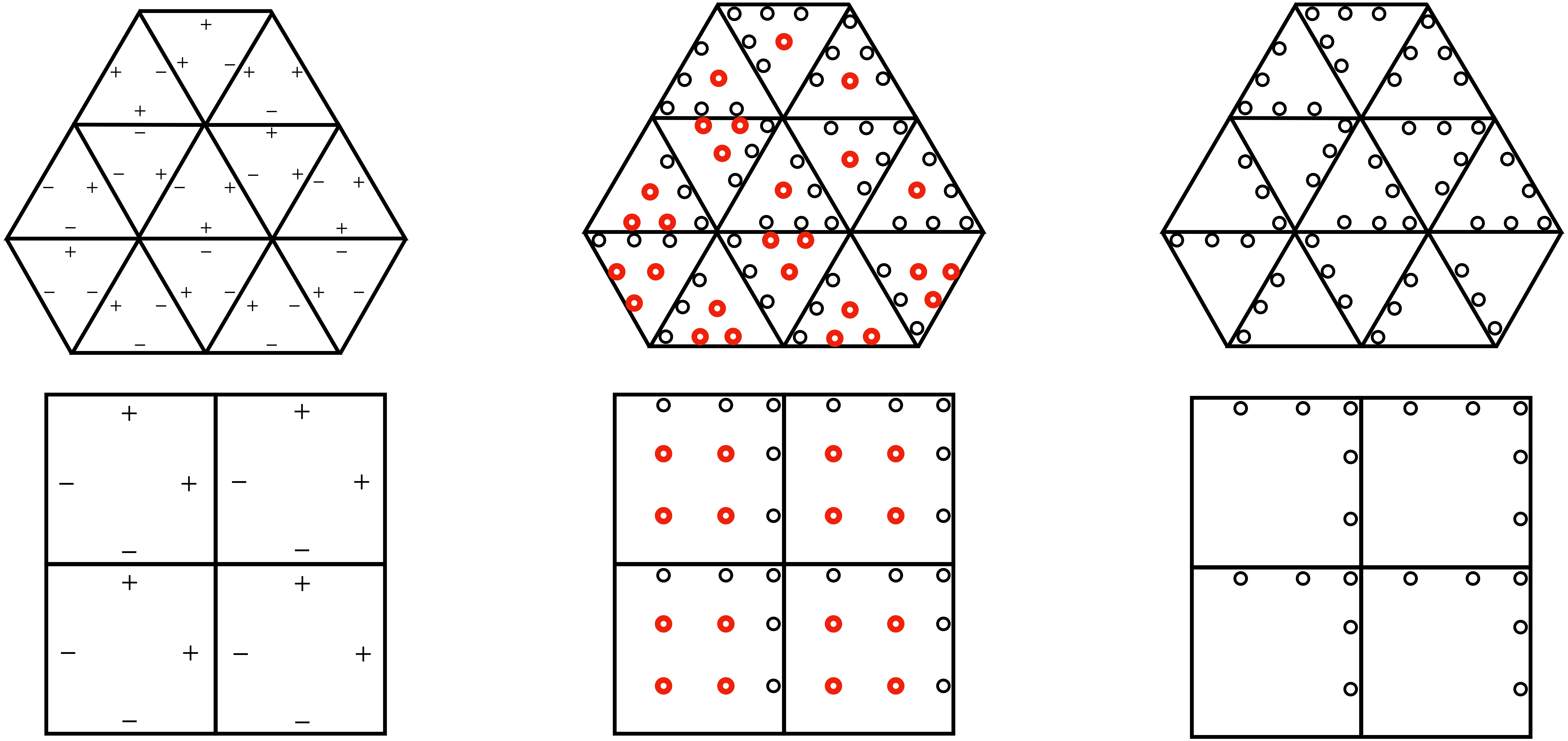
sparsity

partition

- Eliminate all nodes not on a + boundary

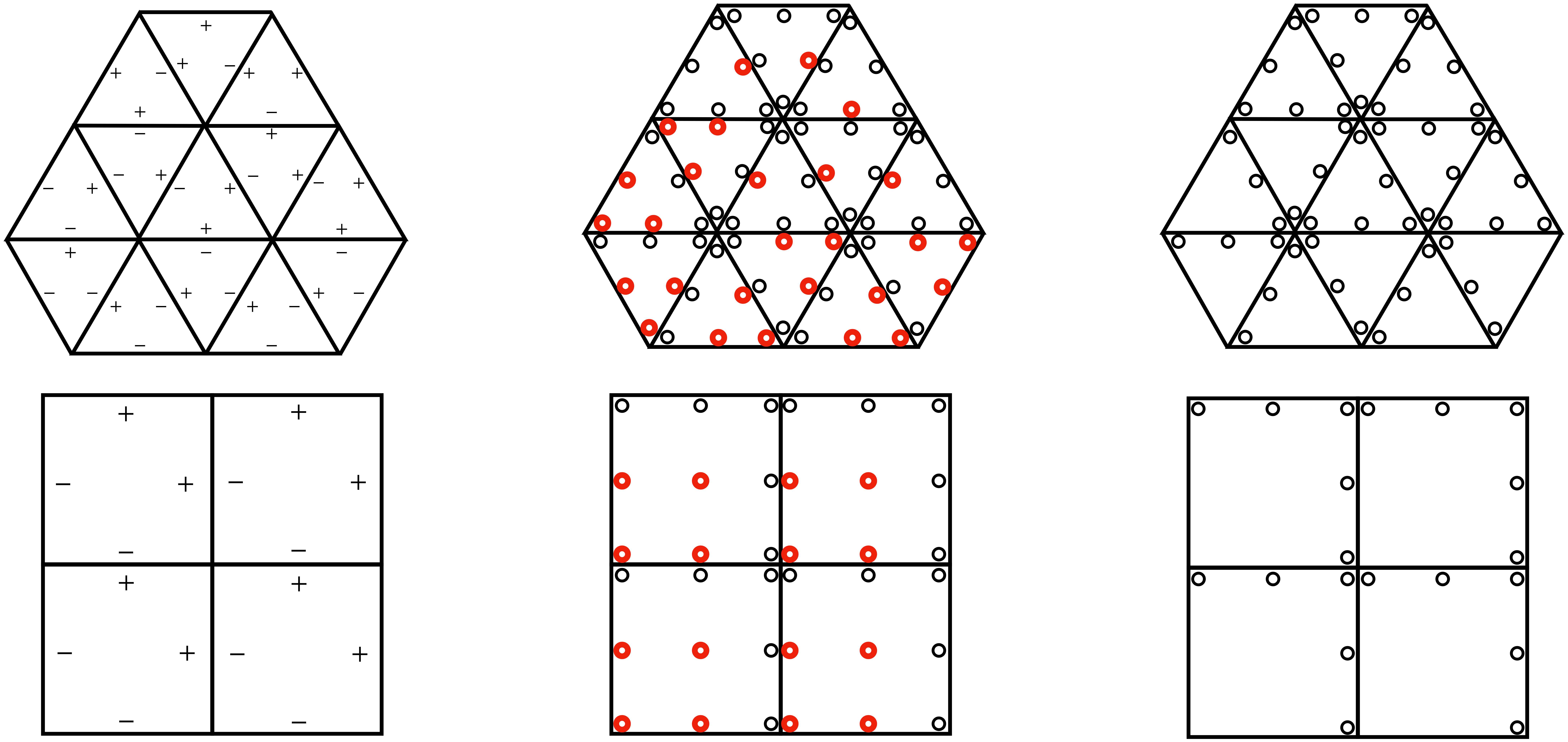
Static condensation (half-closed)

- Eliminate all nodes not on a + boundary

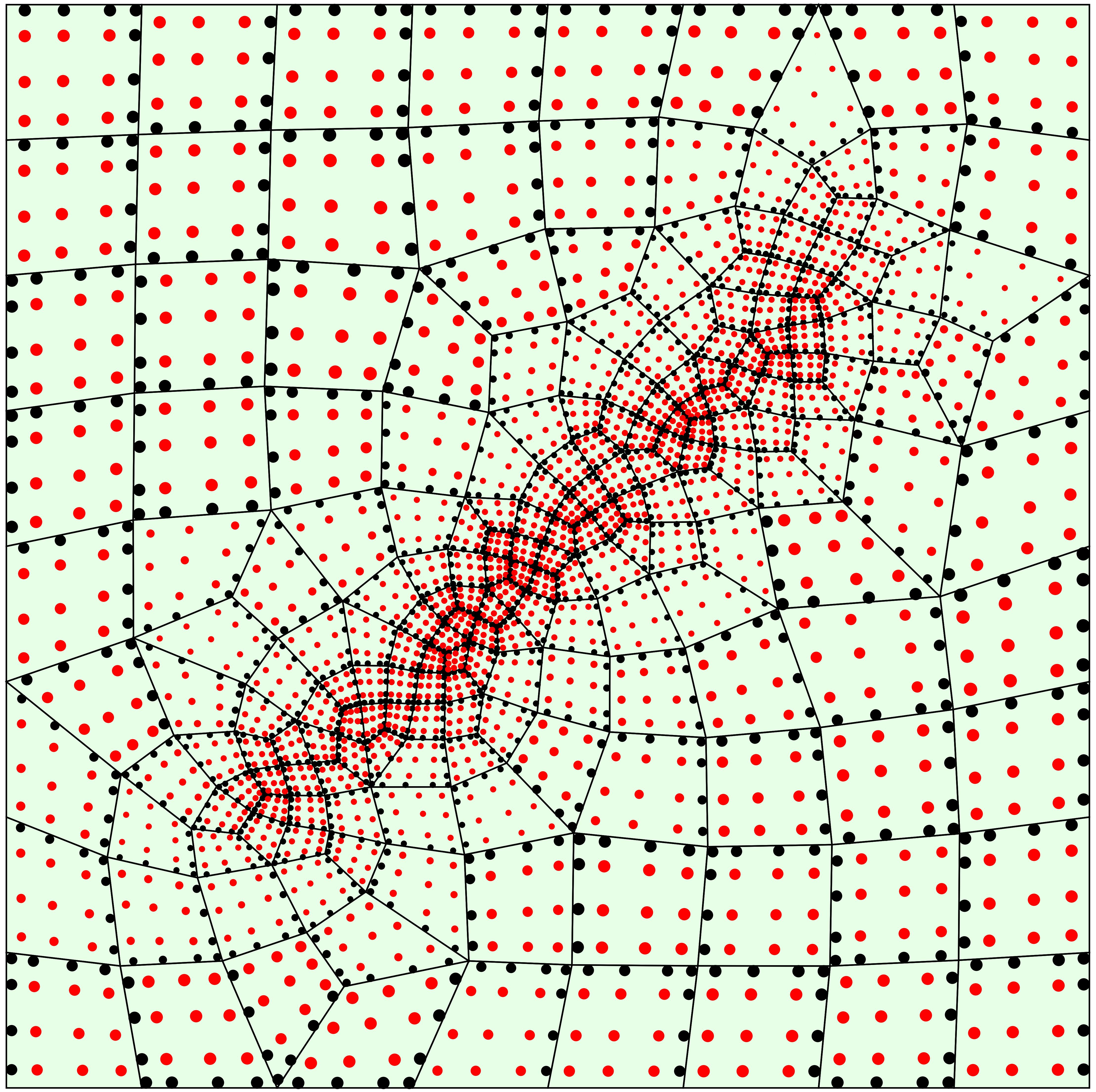


Static condensation (closed)

- Eliminate all nodes not on a + boundary



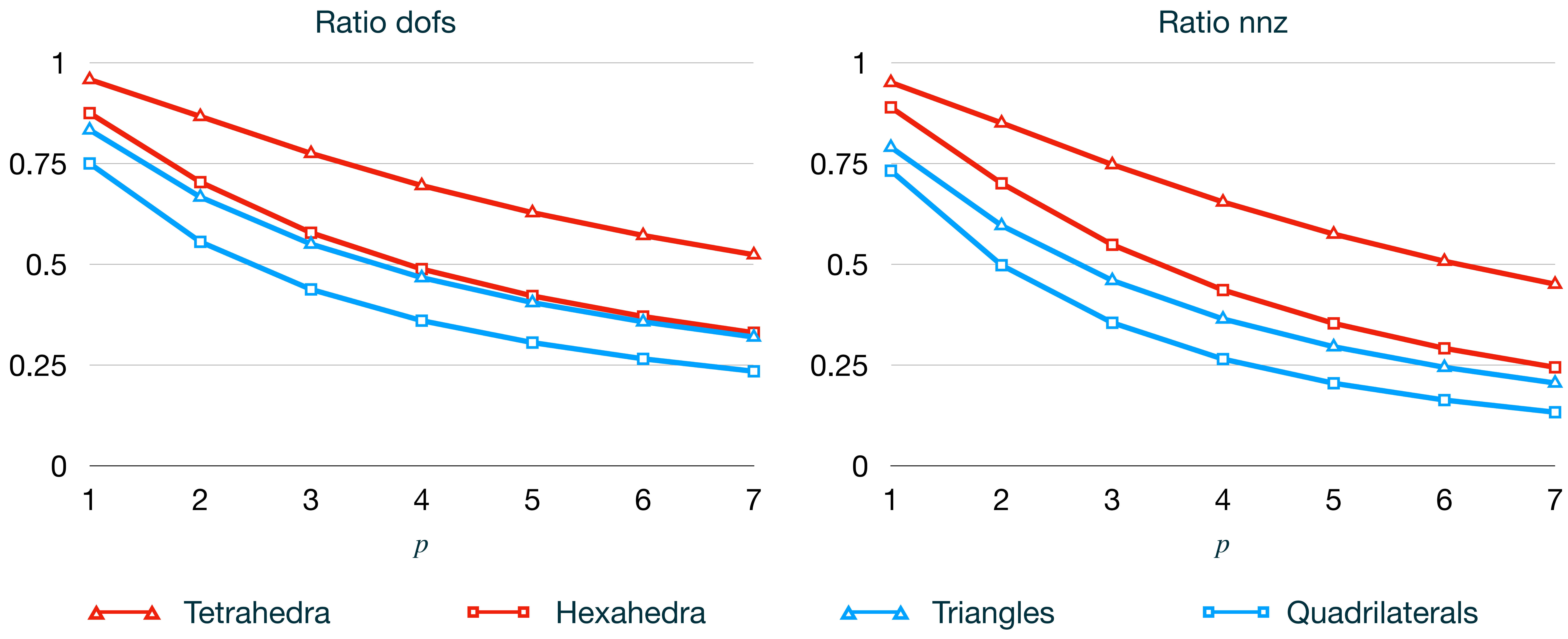
Static condensation (unstructured)



All nodes in red are marked as dependent and eliminated via static condensation

Static condensation

- Comparison of degrees of freedom and number of non-zeros in Laplace operator pre/post static condensation



Static condensation

- Eliminate according to switch function
- Any node not on a boundary where the switch value $S_n^m = +1$ is designated as dependent and eliminated
- Results in eliminated system similar to that of eliminated Finite Elements
    - Fewer inter-element connections
- Can be done for both closed and half-closed nodes
    - BUT not for open nodes

Preconditioners

- Splitting methods:

$$(A + P - P)x = b$$

$$Px^{n+1} = b - (A - P)x^n$$

$$x^{n+1} = P^{-1}b + (I - P^{-1}A)x^n$$

- Iteration matrix:

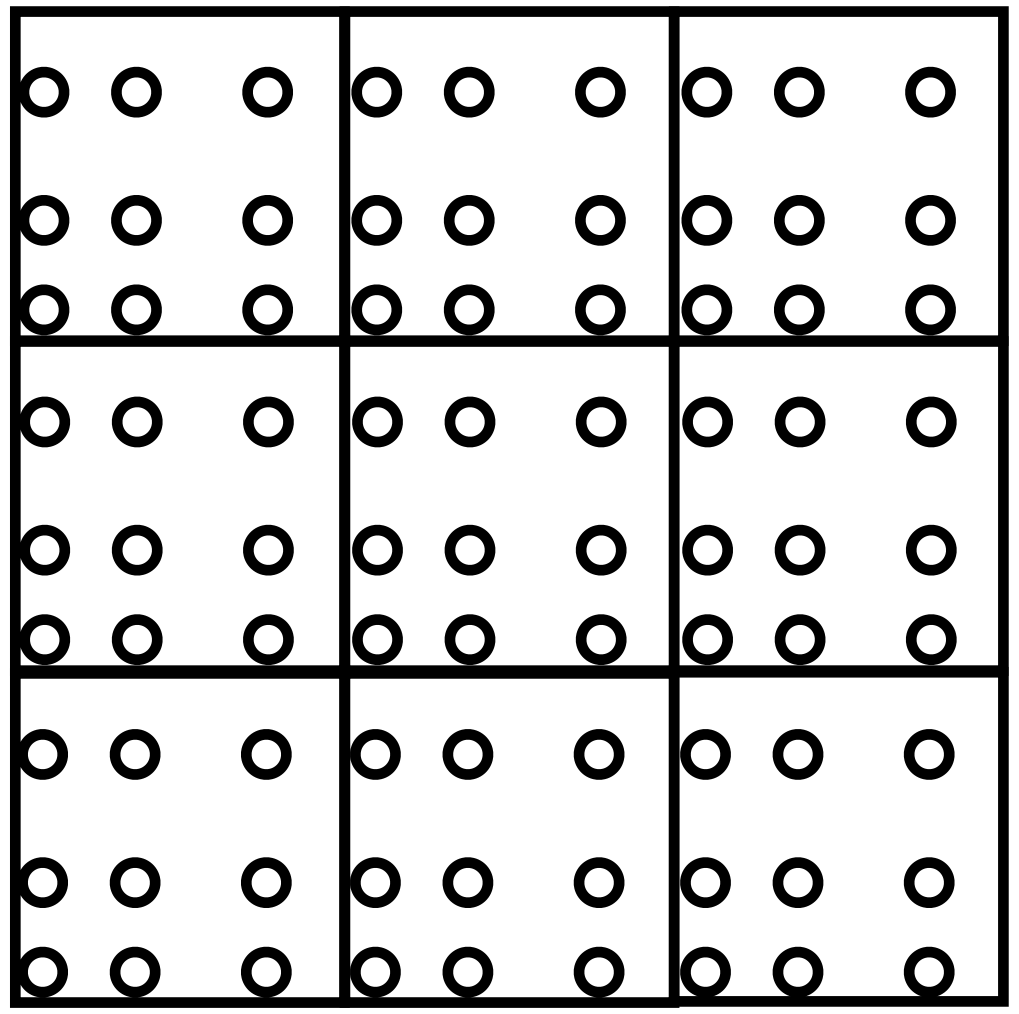
$$\lambda(I - P^{-1}A)$$

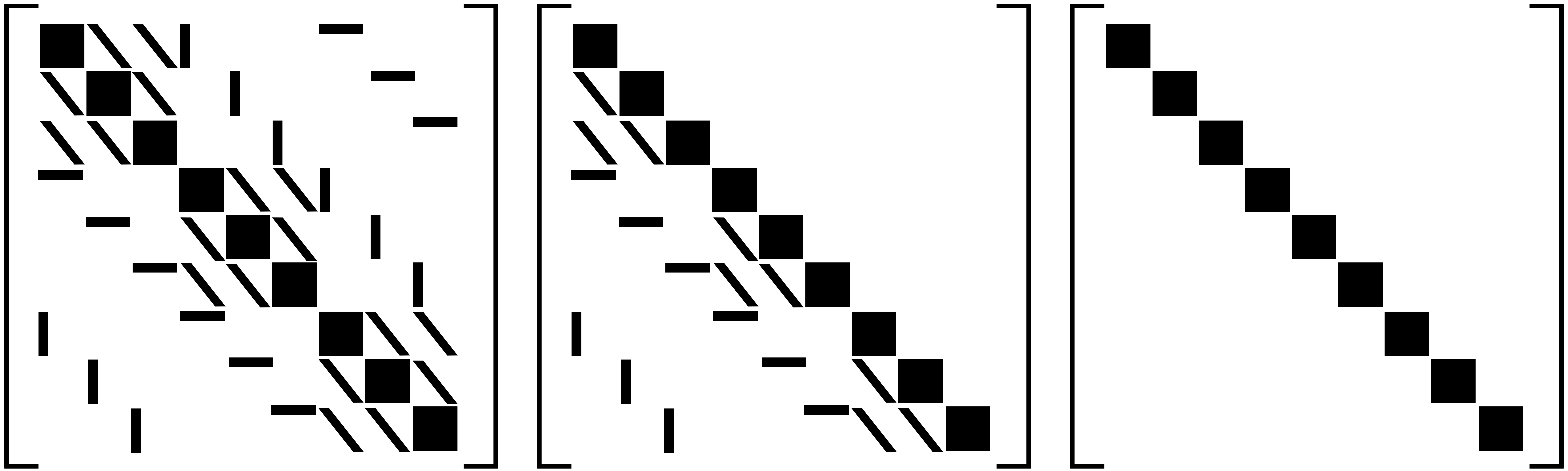
e.g. $P = D$ (block Jacobi), $P = L$ (block Gauss-Seidel)

Want eigenvalues to be as close to zero as possible

Block based solvers

- Commonly used with Discontinuous Galerkin methods
- Iteration matrix: $I - P^{-1}A$





A $\qquad\qquad$ L $\qquad\qquad$ D

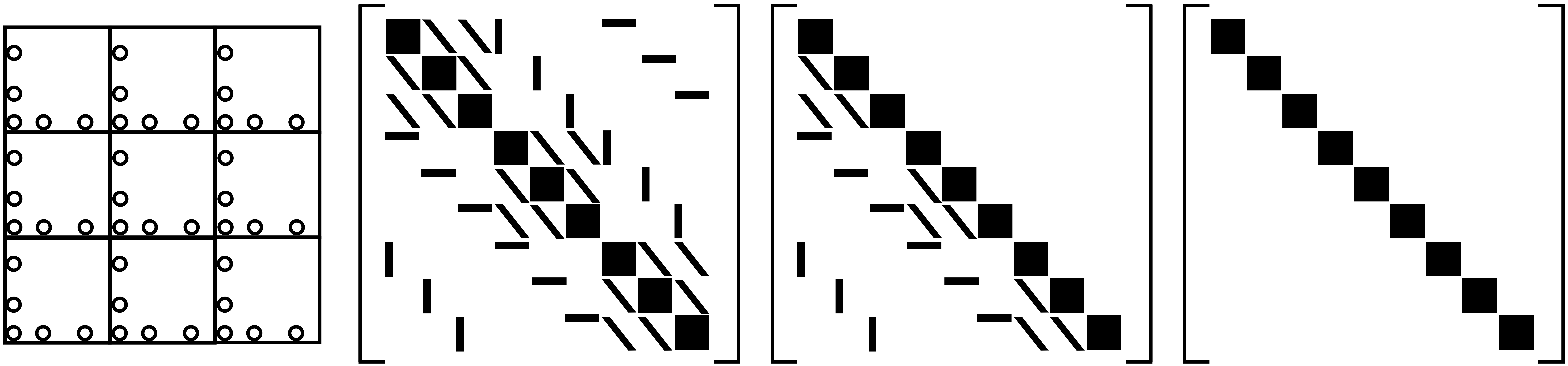
$P = L$, block Gauss-Seidel $\qquad$ $P = L$, block Jacobi

Static condensation + block based solvers

- Expect better performance when applying block based linear solver techniques on eliminated system vs on original DG system
  - Smaller block sizes → cheaper linear operations
  - Improved smoothing properties → fewer iterations
- Important linear operator properties are preserved under Schur complement
  - Symmetry
  - Positive/Negative (Semi)-definiteness
  - Diagonal dominance

Static condensation + block based solvers

- Same block structure post elimination, but with smaller block sizes $O(p^d) \to O(p^{d-1})$
- Iteration matrix: $I - P^{-1}A$



S L_S L_D

Cost of applying P^{-1} reduces from $O(p^{2d}) \to O(p^{2d-2})$

Static condensation + block based solvers

- Want spectrum $\lambda(I - P^{-1}A)$ close to 0:

original

$$\begin{pmatrix} A_{ii} & A_{id} \\ A_{di} & A_{dd} \end{pmatrix} \begin{pmatrix} x_i \\ x_d \end{pmatrix} = \begin{pmatrix} f_i \\ f_d \end{pmatrix}$$

Iterate on A

$$\begin{pmatrix} I - P_{ii}^{-1}A_{ii} & P_{ii}^{-1}A_{id} \\ P_{dd}^{-1}A_{di} & I - P_{dd}^{-1}A_{dd} \end{pmatrix} \begin{pmatrix} x_i^{n+1} \\ x_d^{n+1} \end{pmatrix} = \begin{pmatrix} x_i^n \\ x_d^n \end{pmatrix}$$

eliminated

$$\begin{pmatrix} A_{ii} & A_{id} \\ A_{di} & A_{dd} \end{pmatrix} \begin{pmatrix} x_i \\ x_d \end{pmatrix} = \begin{pmatrix} f_i \\ f_d \end{pmatrix}$$

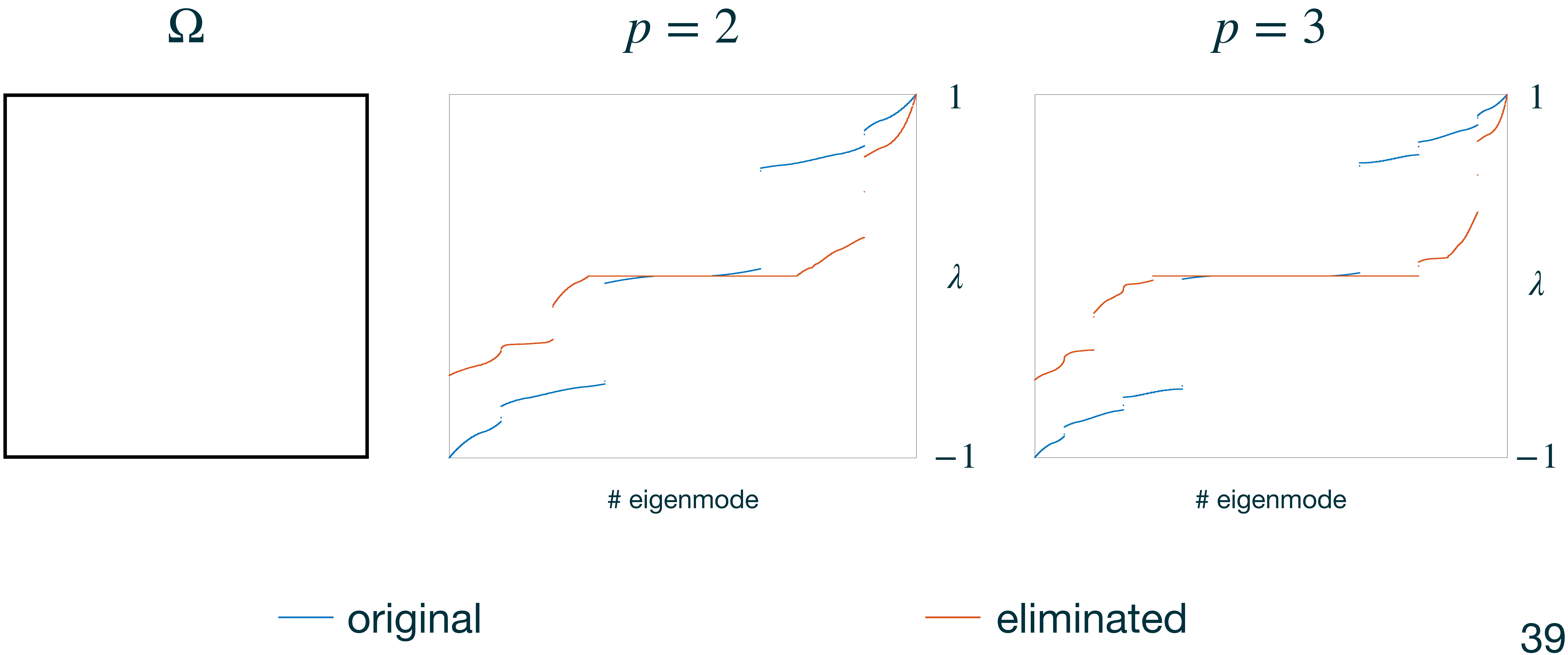
1. Eliminate $A \to S$
2. Iterate on S

$$\begin{pmatrix} I - P_S^{-1}S & 0 \\ C & 0 \end{pmatrix} \begin{pmatrix} x_i^{n+1} \\ x_d^{n+1} \end{pmatrix} = \begin{pmatrix} x_i^n \\ x_d^n \end{pmatrix}$$

$\to$ Improved smoothing properties on eliminated system

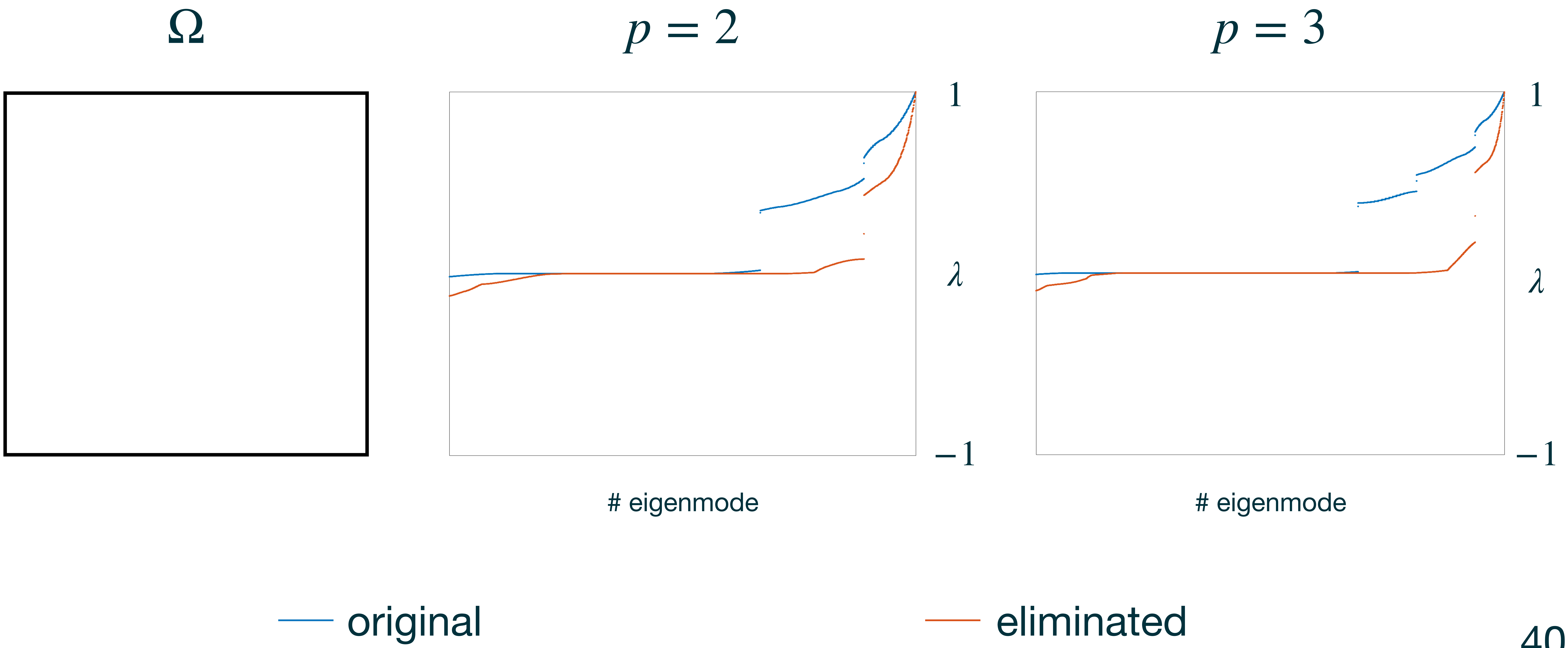
Static condensation + block based solvers

- Poisson's equation : $\lambda(I - P^{-1}A),\ P = D$



Static condensation + block based solvers

- Poisson's equation : $\lambda(I - P^{-1}A)$, $P = L$



Static condensation + block based solvers

- p-multigrid iterations for Poisson's equation
  - Number of iterations to reach 10^{-8} error
  - # iterations on eliminated $\approx 0.5 \times$ # iterations on original

	p=2	p=3	p=5	p=8
Eliminated	8	9	9	8
Original	13	16	16	15

Summary - linear solvers

- Take advantage of sparsity pattern of DG operators using closed/half-closed nodes
  - Static condensation (Commonly used in FEM)
    - Eliminate everywhere except on edges where switch function $S_n^m = +1$
  - Block techniques (Commonly used in DG)
    - Block Jacobi, Block Gauss-Seidel, Block-ILU, …
- Combine methods: cheaper operations + fewer iterations
  - For example: p=4 quads, ~4x fewer jacobian entries, ~2x fewer iterations

Conclusion

Conclusion

- Half-closed nodes for DG
  - Nodes placed only on subset of boundaries according to switch function
  - 2nd order operators using LDG
  - Cost:
    - Nodal integration (GR nodes) → efficient assembly
    - For convection-diffusion, same operator sparsity as with using closed nodes
  - Accuracy:
    - Similar accuracy with standard DG for convection-dominated problems
    - Improved accuracy over standard DG for diffusion dominated problems

Conclusion

- Linear solver techniques for DG using half-closed/closed nodes (but not open)
  - Static condensation, elimination according to LDG switch function
  - Block-based techniques
  - Ability to combine the two to construct more efficient solvers
- Future work:
  - More complex solvers techniques
  - Benchmarks

Thank you!

Yulong Pan and Per-Olof Persson. Half-closed discontinuous galerkin discretisation. Submitted to Journal of Computational Physics. https://arxiv.org/abs/2405.12383