A conservative solution to checkerboarding:
Examining the discrete Laplacian kernel using mesh connectivity

J.A. Hopman, F.X. Trias and J.Rigola

Heat and Mass Transfer Technological Center (CTTC)

Technical University of Catalonia (UPC), Terrassa, Spain

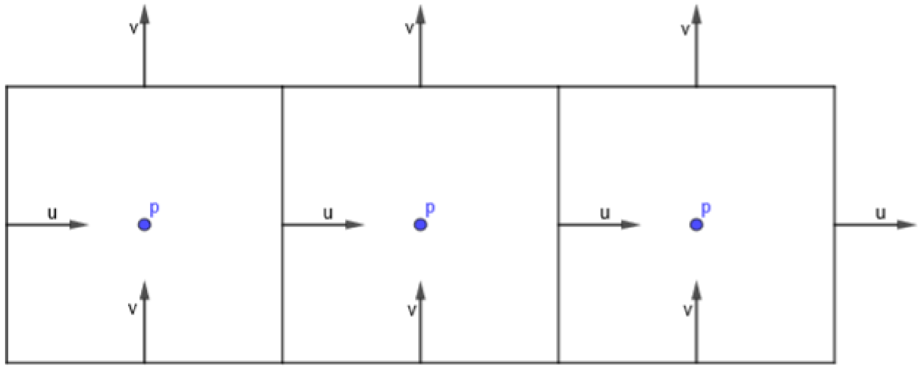
DLES13 2022 – Direct and Large Eddy Simulation
26-28 October 2022, Udine, Italy

Pressure-velocity coupling

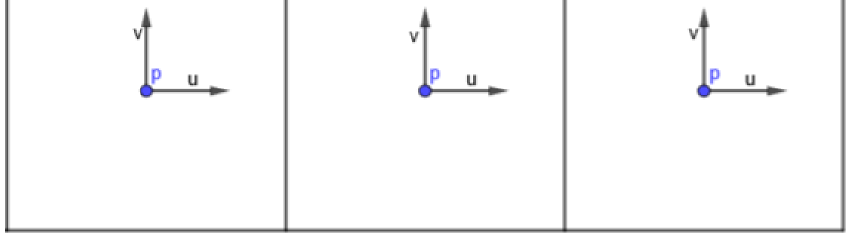
$$\mathbf{u}^{n+1} = \mathbf{u}^{p} - \nabla \mathbf{p}^{n+1}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

$$\nabla^2 \mathbf{p}^{n+1} = \nabla \cdot \mathbf{u}^{p}$$



staggered



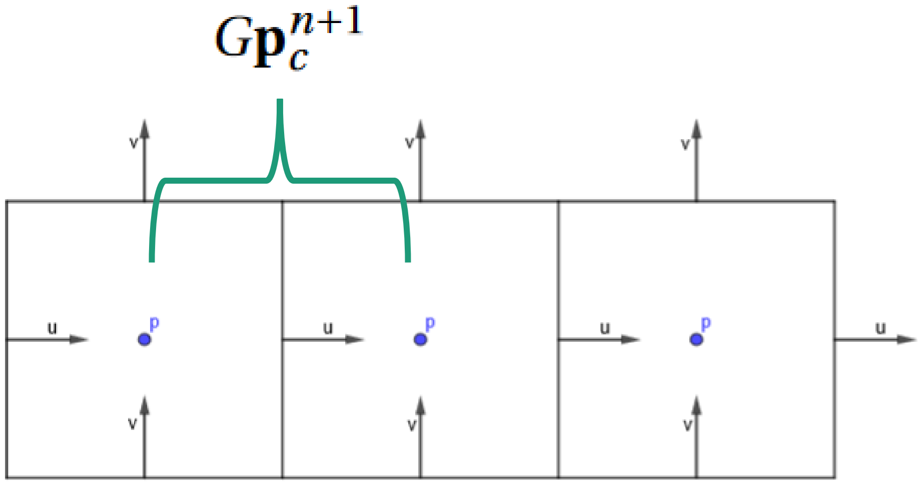
collocated

Pressure-velocity coupling

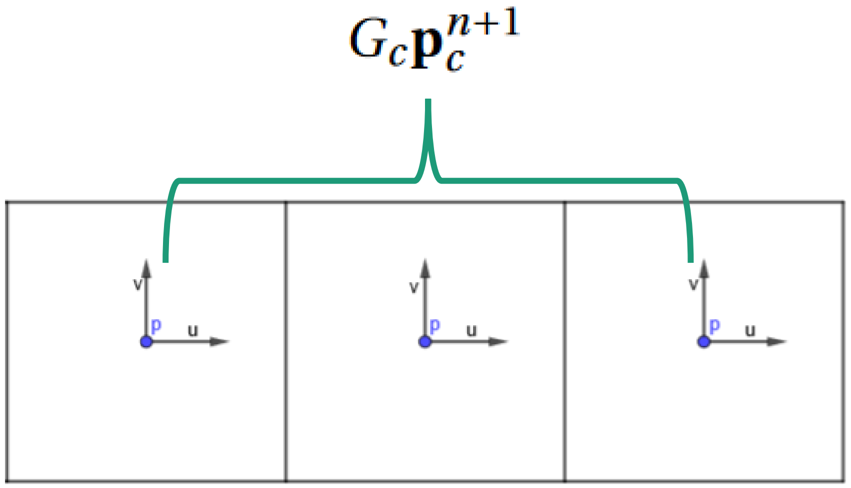
$$\mathbf{u}^{n+1} = \mathbf{u}^p - \nabla \mathbf{p}^{n+1}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

$$\nabla^2 \mathbf{p}^{n+1} = \nabla \cdot \mathbf{u}^p$$



staggered



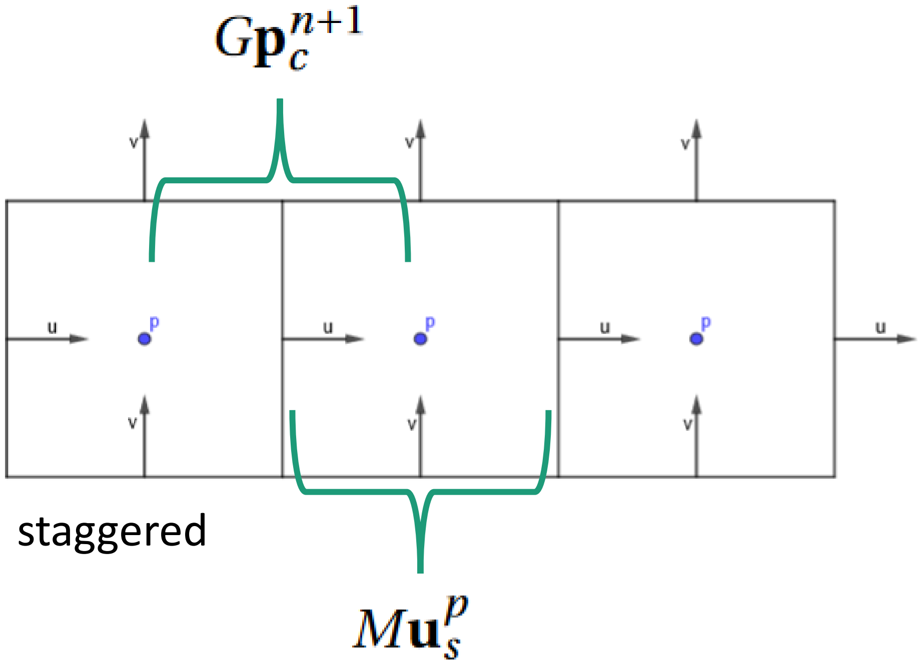
collocated

Pressure-velocity coupling

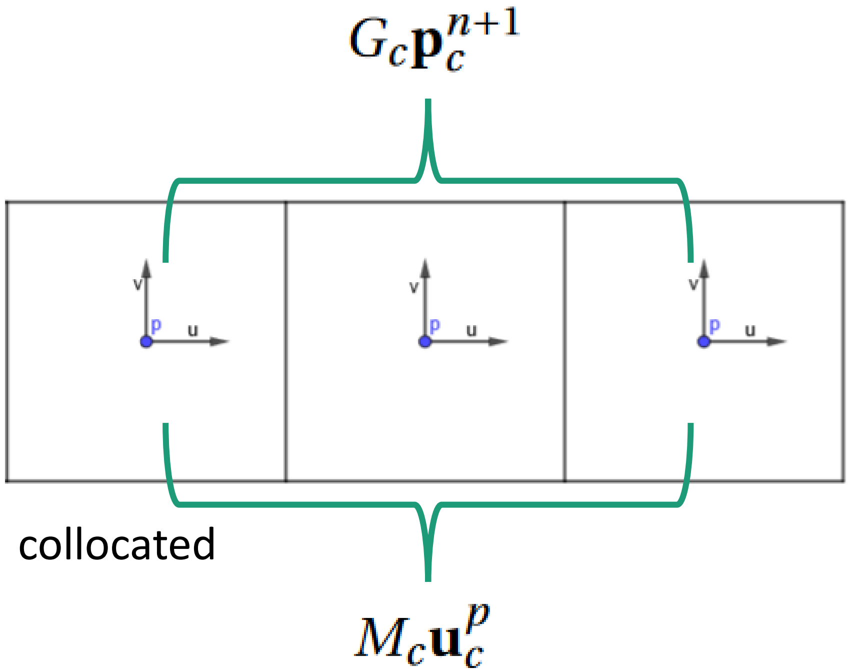
$$\mathbf{u}^{n+1} = \mathbf{u}^{p} - \nabla \mathbf{p}^{n+1}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

$$\nabla^2 \mathbf{p}^{n+1} = \nabla \cdot \mathbf{u}^{p}$$



staggered



collocated

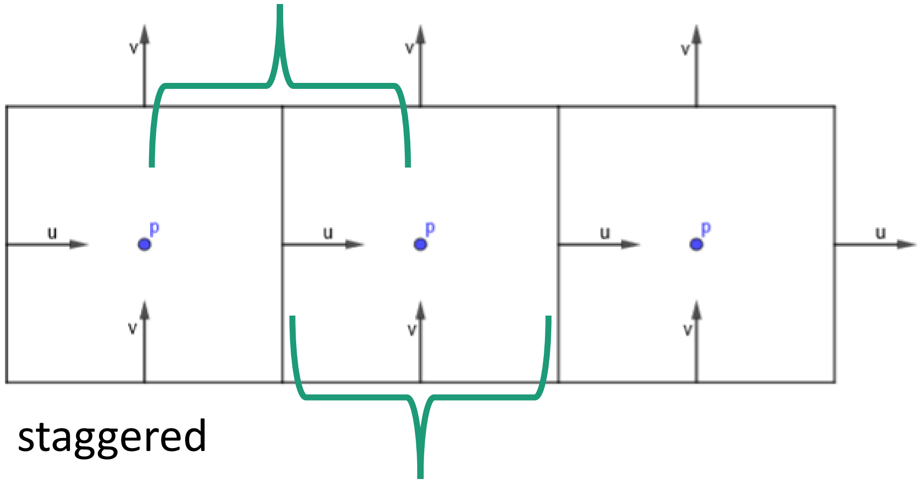
Pressure-velocity coupling

$$\mathbf{u}^{n+1} = \mathbf{u}^{p} - \nabla \mathbf{p}^{n+1}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

$$\nabla^2 \mathbf{p}^{n+1} = \nabla \cdot \mathbf{u}^{p}$$

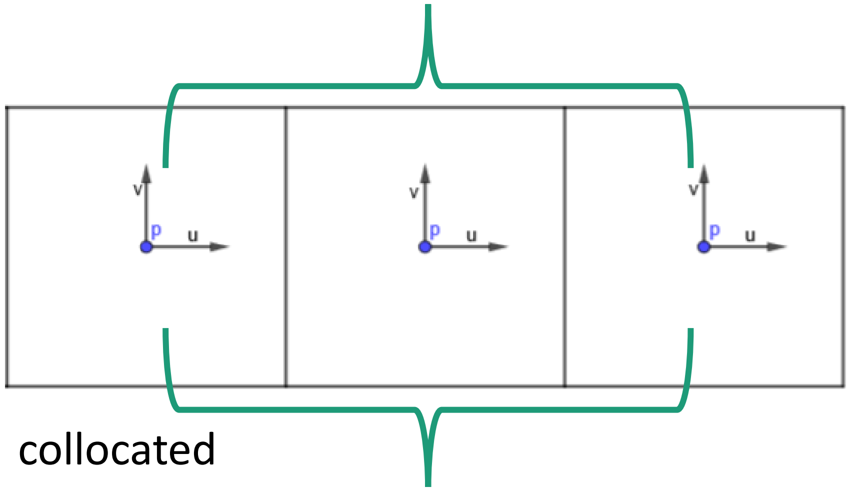
$G\mathbf{p}_c^{n+1}$



$M\mathbf{u}_s^{p}$

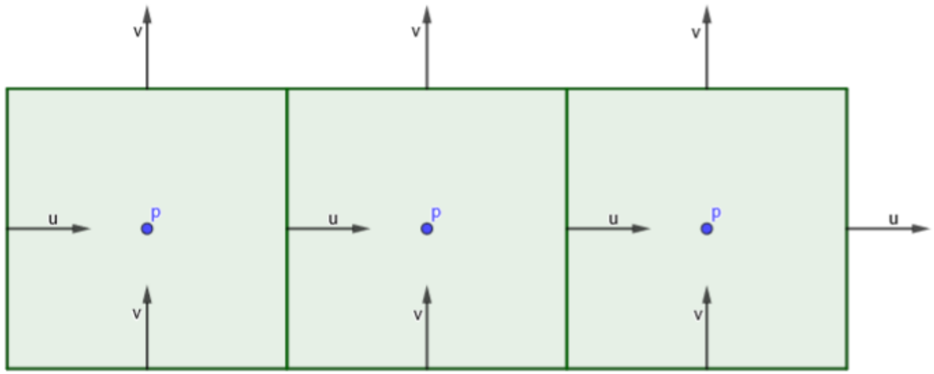
	Stag	Col
Grad	G	G_c
Div	M	M_c

$G_c\mathbf{p}_c^{n+1}$

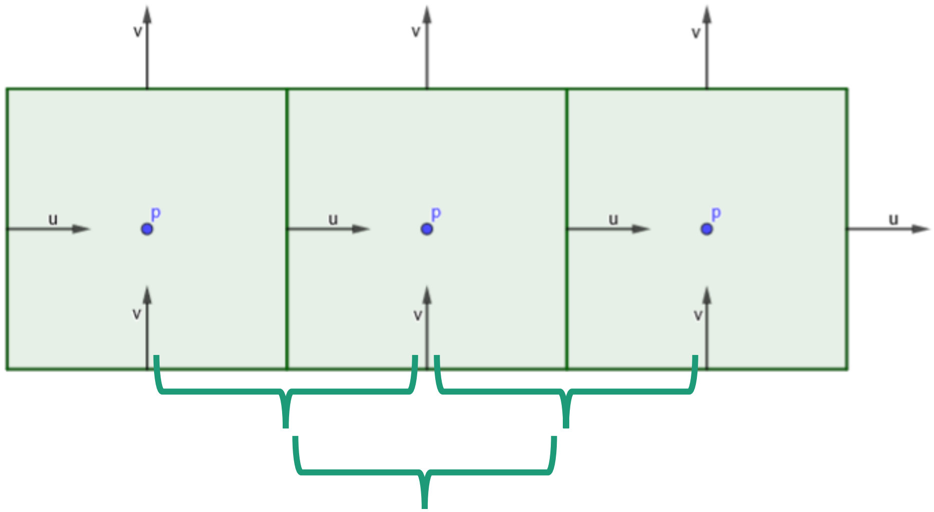


$M_c\mathbf{u}_c^{p}$

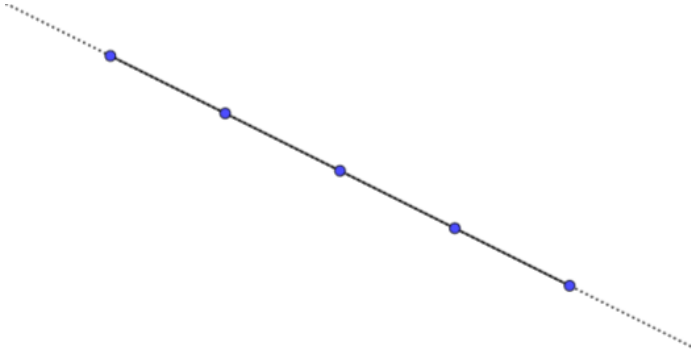
Compact vs wide stencil Laplacian



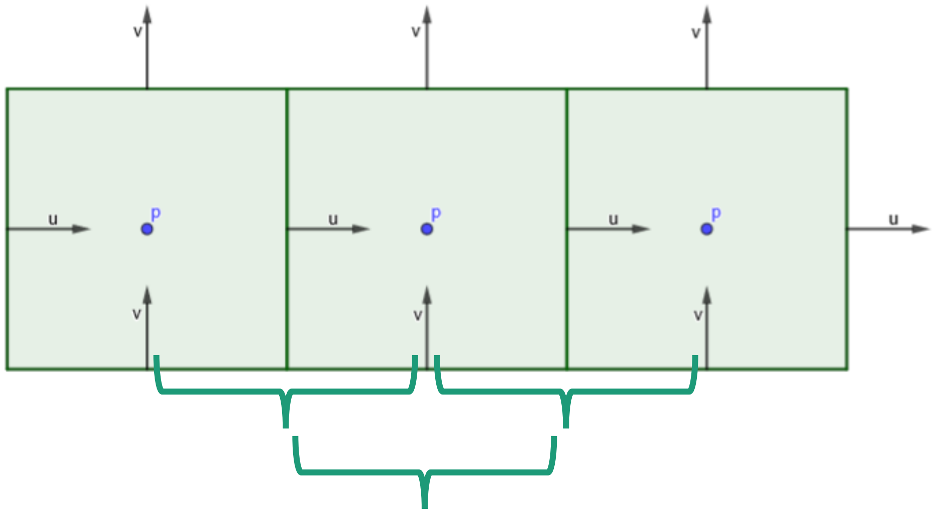
Compact vs wide stencil Laplacian



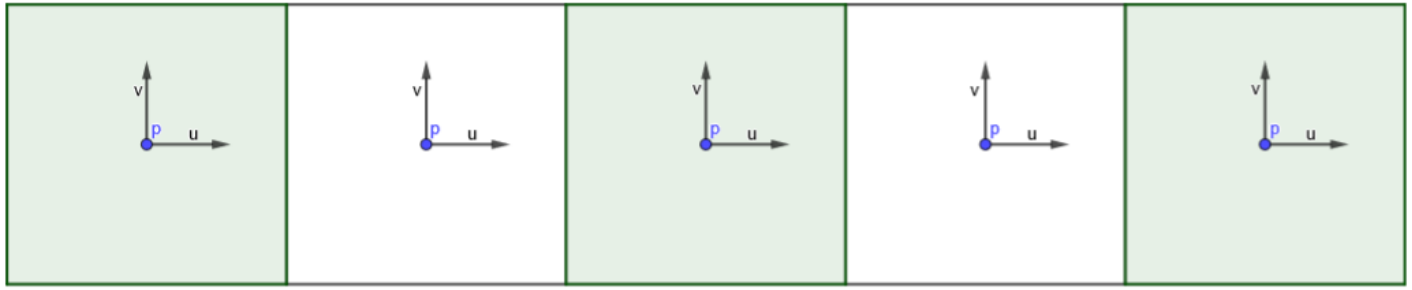
$L = MG$ compact stencil

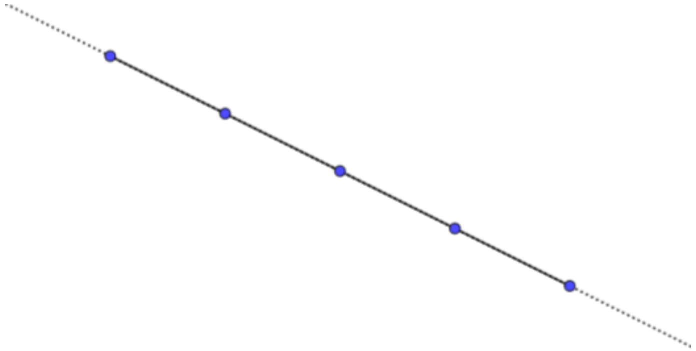


Compact vs wide stencil Laplacian

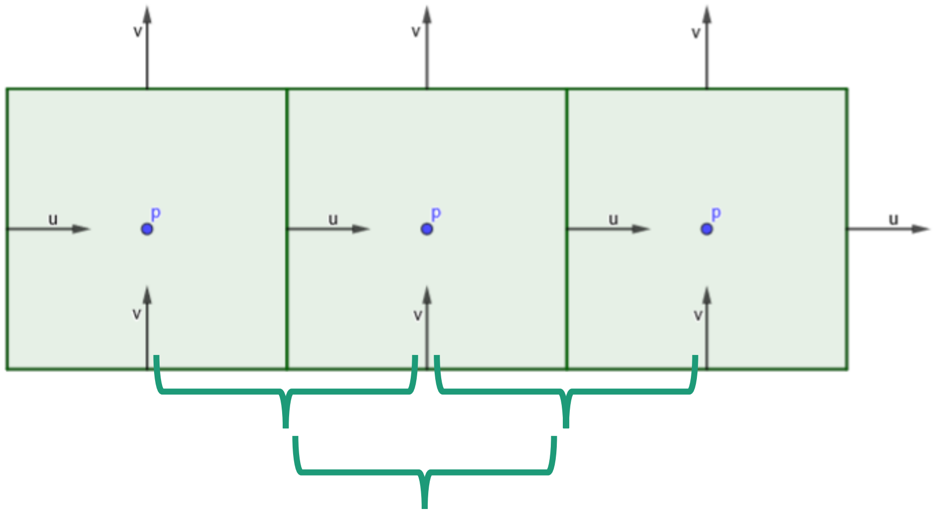


$L = MG$ compact stencil

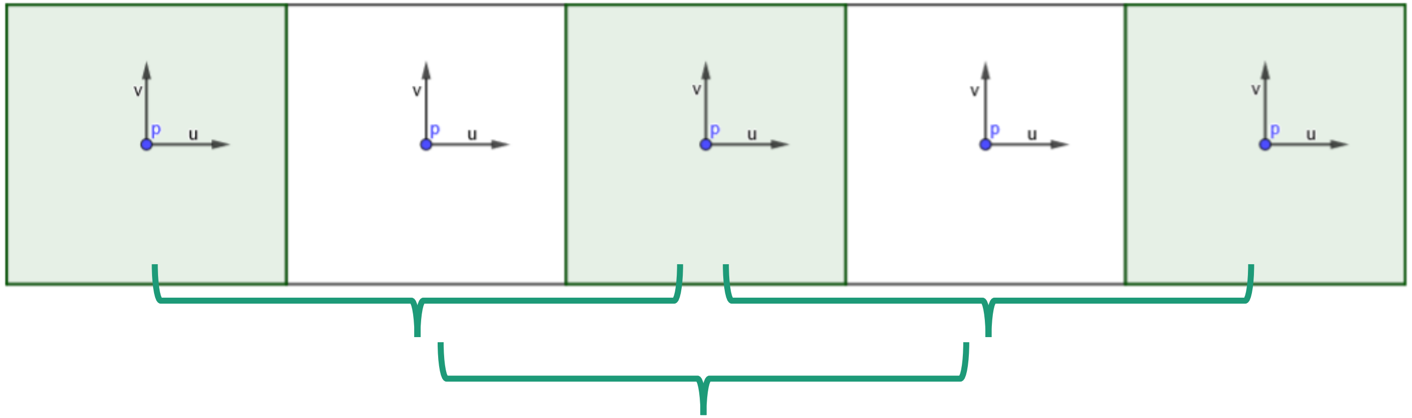




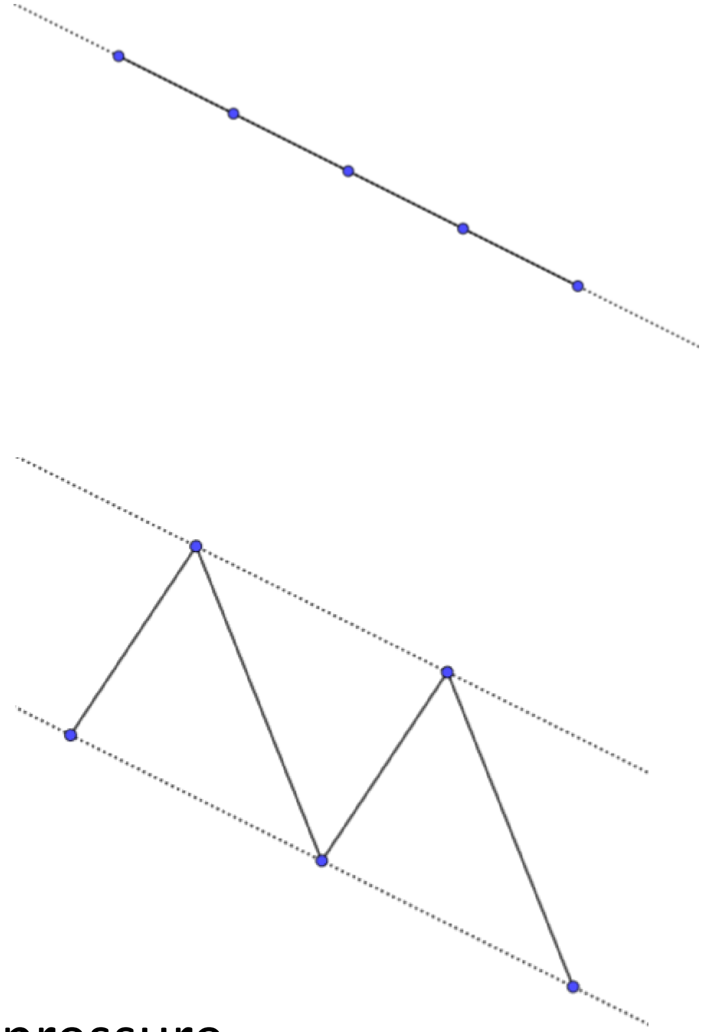
Compact vs wide stencil Laplacian



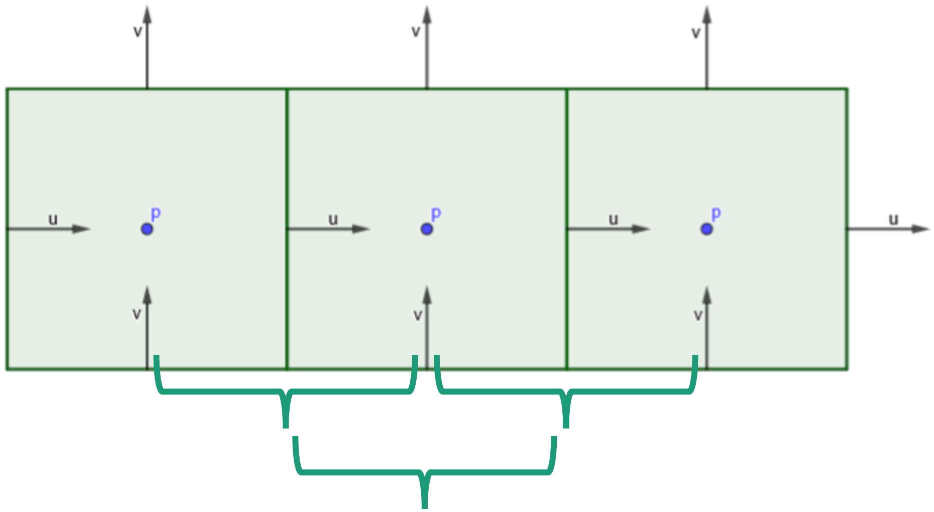
$L = MG$ compact stencil



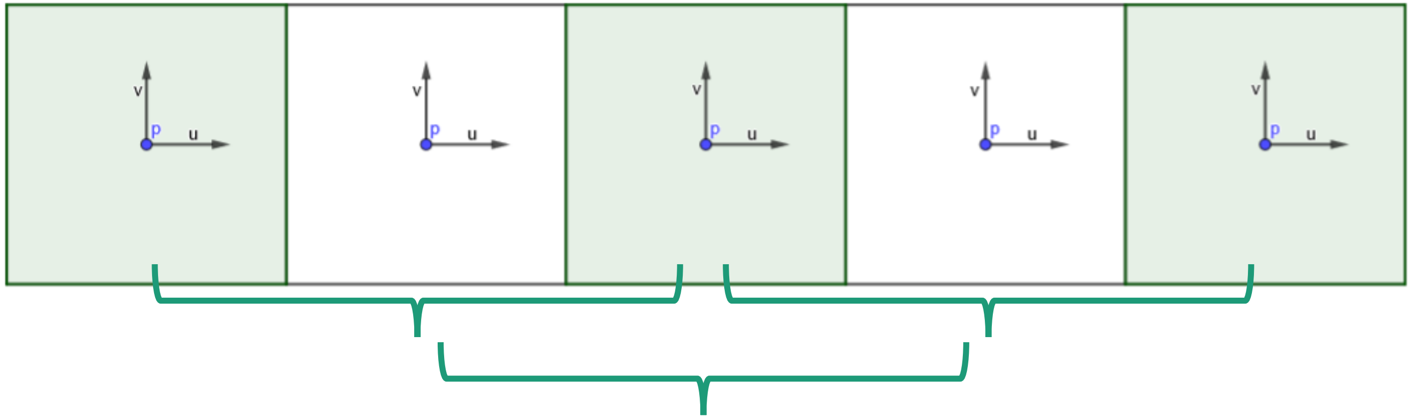
$L_c = M_c G_c$ wide stencil → decoupling of pressure



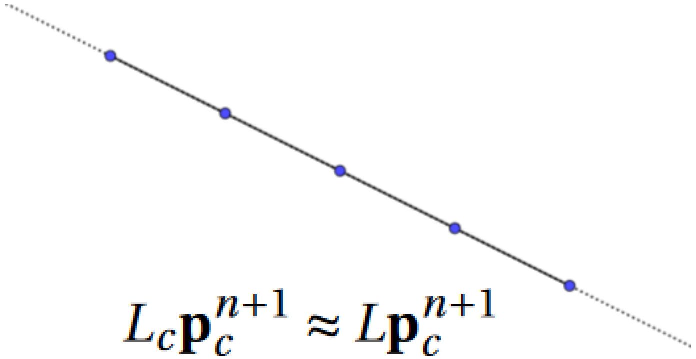
Compact vs wide stencil Laplacian



$L = MG$ compact stencil

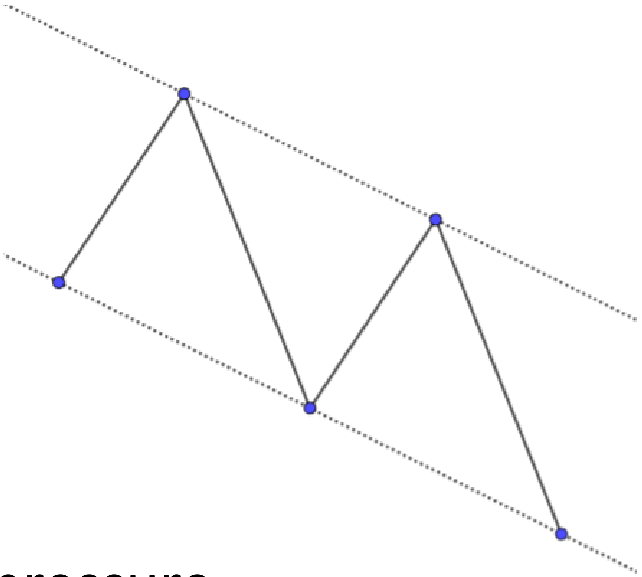


$L_c = M_c G_c$ wide stencil → decoupling of pressure



$L_c \mathbf{p}_c^{n+1} \approx L \mathbf{p}_c^{n+1}$

→ pressure error



Motivation

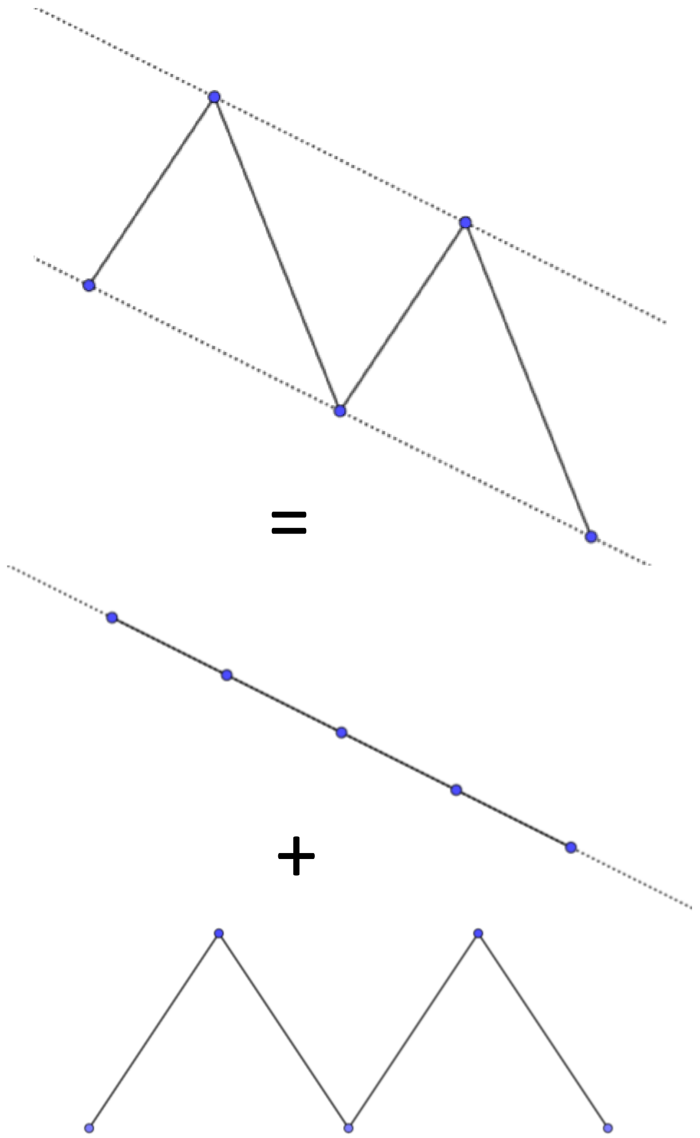
Can we remove spurious modes without pressure error?

Develop proper filtering for Cartesian meshes

Gain insight in filtering for unstructured meshes

Spurious modes – Kernel of Laplacian

$$\mathbf{p}_c = \mathbf{p}_c^+ + \mathbf{p}_c^-$$
$$\mathbf{p}_c^- \in \mathrm{Ker}(L_c)$$
$$L_c\mathbf{p}_c = L_c\mathbf{p}_c^+ + L_c\mathbf{p}_c^- = L_c\mathbf{p}_c^+$$

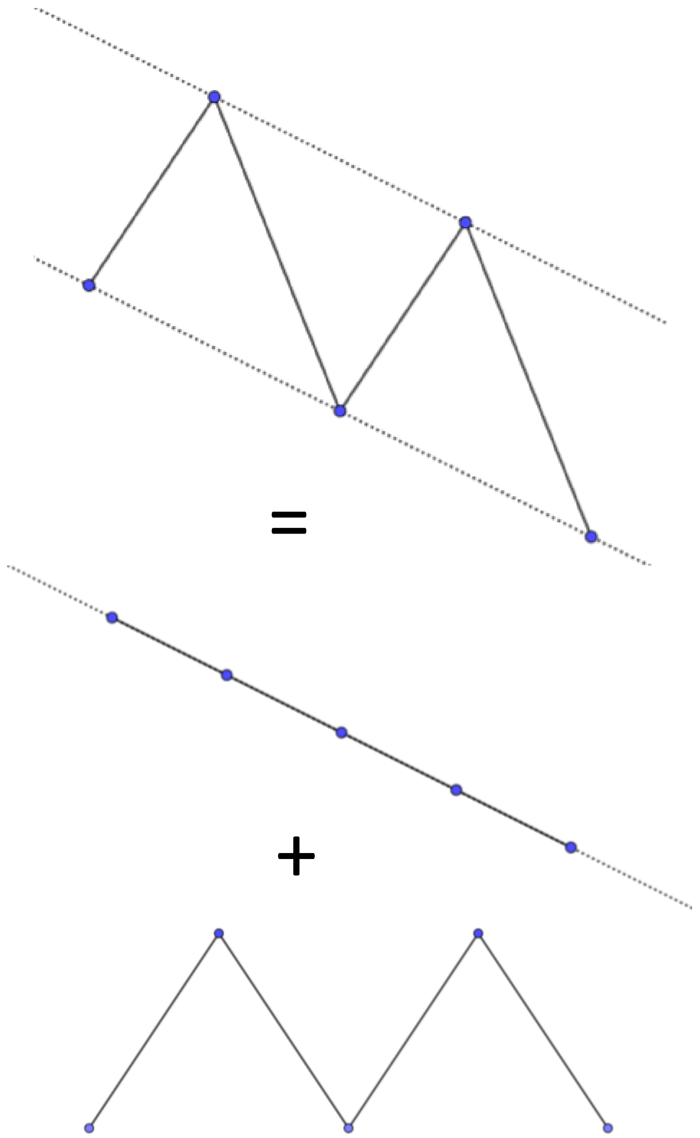


Spurious modes – Kernel of Laplacian

$$\mathbf{p}_c = \mathbf{p}_c^+ + \mathbf{p}_c^-$$
$$\mathbf{p}_c^- \in \mathrm{Ker}(L_c)$$
$$L_c\mathbf{p}_c = L_c\mathbf{p}_c^+ + L_c\mathbf{p}_c^- = L_c\mathbf{p}_c^+$$

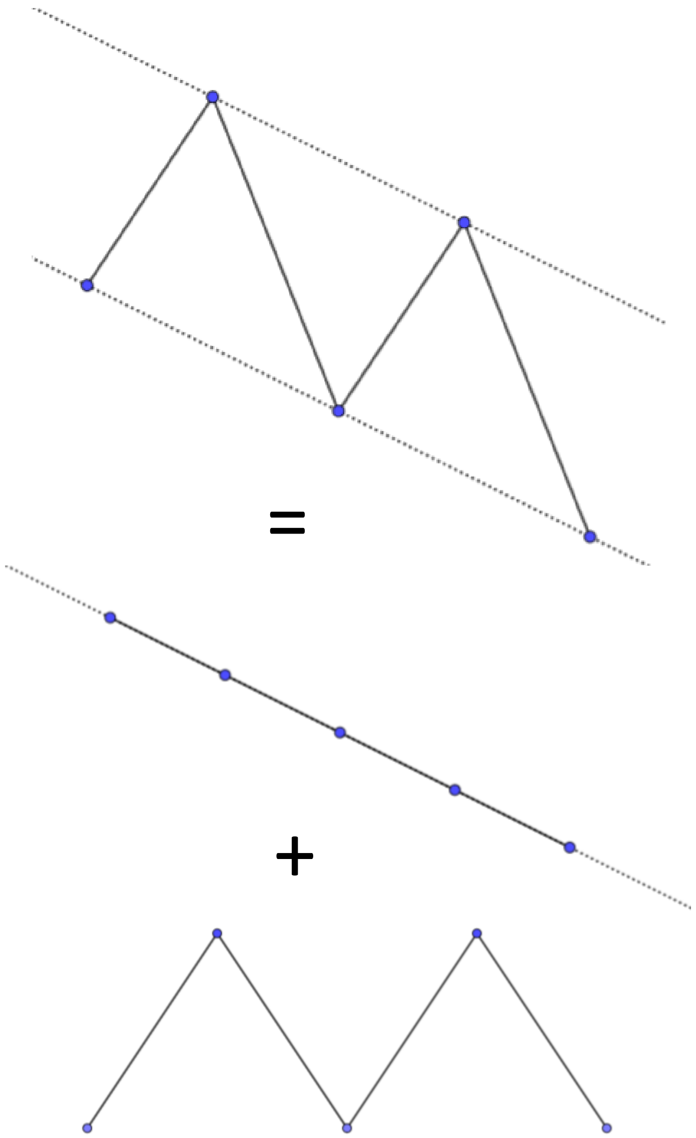
Nullity(L_c) > 1→ spurious modes
(Nullity(L) = 1: constant mode → reference pressure)

If we know Ker(L_c) we can just filter $\mathbf{p}^-$



Spurious modes – Kernel of Laplacian

$$\mathbf{p}_c = \mathbf{p}_c^+ + \mathbf{p}_c^-$$
$$\mathbf{p}_c^- \in \mathrm{Ker}(L_c)$$
$$L_c\mathbf{p}_c = L_c\mathbf{p}_c^+ + L_c\mathbf{p}_c^- = L_c\mathbf{p}_c^+$$



Nullity(L_c) > 1→ spurious modes
(Nullity(L) = 1: constant mode → reference pressure)

If we know Ker(L_c) we can just filter $\mathbf{p}^-$

Calculate using Singular Value Decomposition?

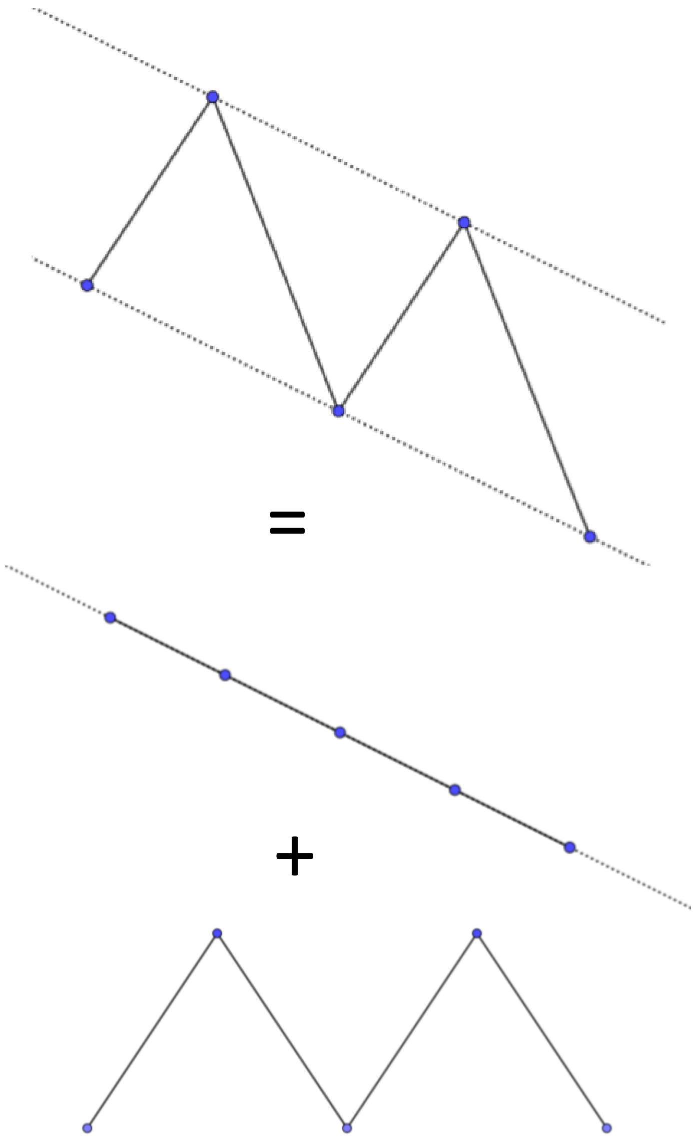
Spurious modes – Kernel of Laplacian

$$\mathbf{p}_c = \mathbf{p}_c^+ + \mathbf{p}_c^-$$
$$\mathbf{p}_c^- \in \mathrm{Ker}(L_c)$$
$$L_c\mathbf{p}_c = L_c\mathbf{p}_c^+ + L_c\mathbf{p}_c^- = L_c\mathbf{p}_c^+$$

Nullity(L_c) > 1→ spurious modes
(Nullity(L) = 1: constant mode → reference pressure)

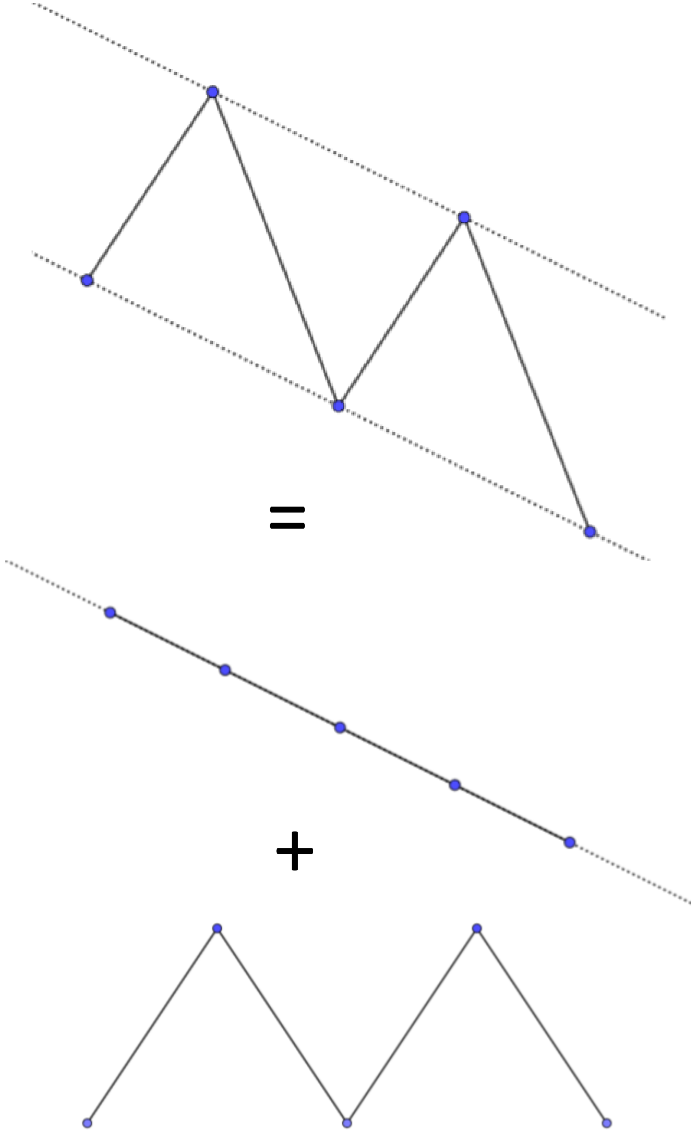
If we know Ker(L_c) we can just filter $\mathbf{p}^-$

Calculate using Singular Value Decomposition?
Cost cales with O(n^3)



Spurious modes – Kernel of Laplacian

$$\mathbf{p}_c = \mathbf{p}_c^+ + \mathbf{p}_c^-$$
$$\mathbf{p}_c^- \in \mathrm{Ker}(L_c)$$
$$L_c\mathbf{p}_c = L_c\mathbf{p}_c^+ + L_c\mathbf{p}_c^- = L_c\mathbf{p}_c^+$$



Nullity(L_c) > 1→ spurious modes
(Nullity(L) = 1: constant mode → reference pressure)

If we know Ker(L_c) we can just filter $\mathbf{p}^-$

Calculate using Singular Value Decomposition?
Cost cales with O(n^3)

L_c depends on mesh and discretisation;
can't we deduce Ker(L_c) from mesh and discretisation?

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\Gamma_{cs} \qquad G_c = \Gamma_{sc} G$$

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\underline{\Gamma_{cs}} \qquad G_c = \underline{\Gamma_{sc}}G$$

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\underline{\Gamma_{cs}} \qquad G_c = \underline{\Gamma_{sc}} G$$

$$-M^T = \Omega_s G$$
$$-M_c^T = \Omega_c G_c$$

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M \Gamma_{cs} \qquad G_c = \Gamma_{sc} G$$

$$-M^T = \Omega_s G$$
$$-M_c^T = \Omega_c G_c$$

$$\Gamma_{sc} = \Omega_c^{-1} \Gamma_{cs}^T \Omega_s$$

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\underline{\Gamma_{cs}} \qquad G_c = \underline{\Gamma_{sc}} G$$

$$-M^T = \Omega_s G$$
$$-M_c^T = \Omega_c G_c$$

$$\Gamma_{sc} = \Omega_c^{-1}\Gamma_{cs}^T\Omega_s$$

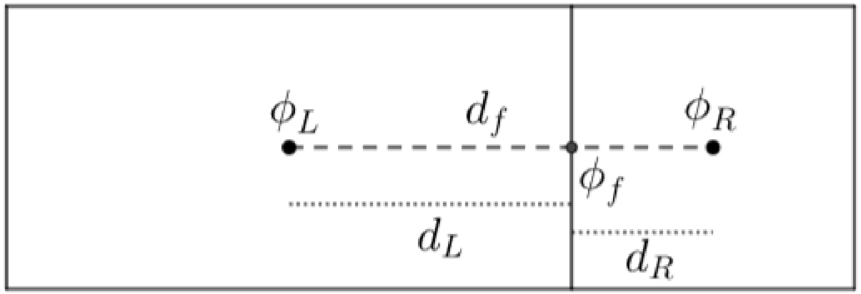
<u>Interpolators related, 1 d.o.f.</u>

Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\underline{\Gamma_{cs}} \qquad G_c = \underline{\Gamma_{sc}}G$$

$$-M^T = \Omega_s G$$
$$-M_c^T = \Omega_c G_c$$



$$\Gamma_{sc} = \Omega_c^{-1}\Gamma_{cs}^T\Omega_s$$

<u>Interpolators related, 1 d.o.f.</u>

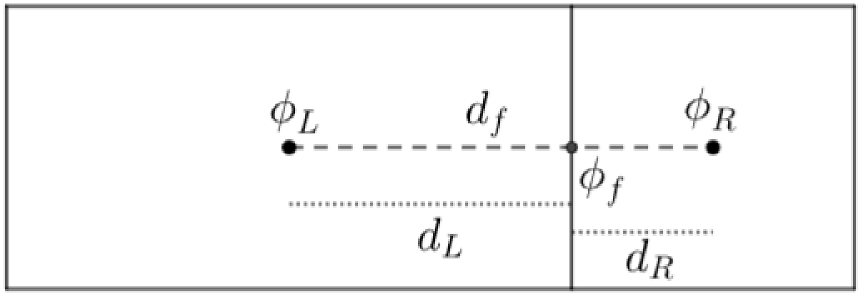
Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\Gamma_{cs} \qquad G_c = \Gamma_{sc} G$$

$$-M^T = \Omega_s G$$
$$-M_c^T = \Omega_c G_c$$

$$\Gamma_{sc} = \Omega_c^{-1} \Gamma_{cs}^T \Omega_s$$



Midpoint: $\Pi_{cs}^M \to \phi_f = \dfrac{\phi_L + \phi_R}{2}$

Linear: $\Pi_{cs}^L \to \phi_f = \dfrac{d_R \phi_L + d_L \phi_R}{d_f}$

Volumetric: $\Pi_{cs}^V \to \phi_f = \dfrac{d_L \phi_L + d_R \phi_R}{d_f}$

<u>Interpolators related, 1 d.o.f.</u>

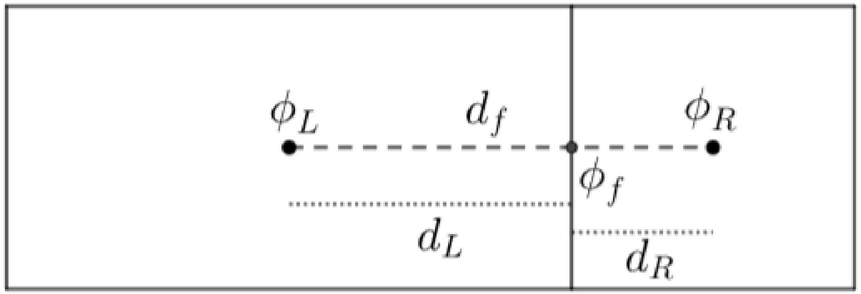
Laplacian matrix – interpolators

$$L_c = M_c G_c$$

$$M_c = M\Gamma_{cs} \qquad G_c = \Gamma_{sc} G$$

$$-M^T = \Omega_s G$$
$$-M_c^T = \Omega_c G_c$$

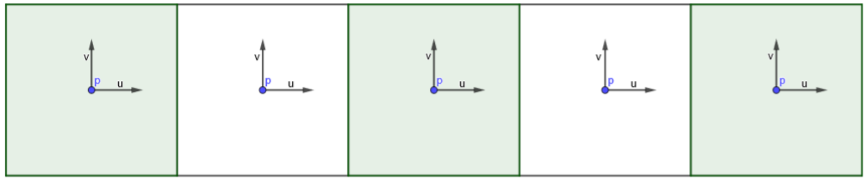
$$\Gamma_{sc} = \Omega_c^{-1} \Gamma_{cs}^T \Omega_s$$



Midpoint: $\Pi_{cs}^M \rightarrow \phi_f = \dfrac{\phi_L + \phi_R}{2}$

Linear: $\Pi_{cs}^L \rightarrow \phi_f = \dfrac{d_R\phi_L + d_L\phi_R}{d_f}$

Volumetric: $\Pi_{cs}^V \rightarrow \phi_f = \dfrac{d_L\phi_L + d_R\phi_R}{d_f}$



<u>Interpolators related, 1 d.o.f.</u>

Laplacian matrix – interpolators

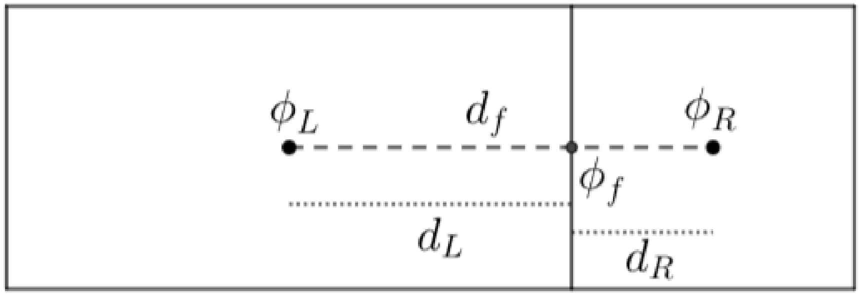
$$L_c = M_c G_c$$

$$M_c = M\underline{\Gamma_{cs}} \qquad G_c = \underline{\Gamma_{sc}} G$$

$$-M^T = \Omega_s G$$

$$-M_c^T = \Omega_c G_c$$

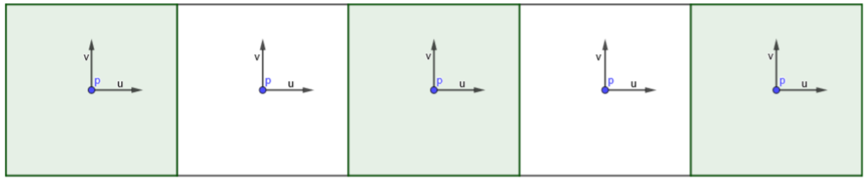
$$\Gamma_{sc} = \Omega_c^{-1} \Gamma_{cs}^T \Omega_s$$



Midpoint: $\Pi_{cs}^M \rightarrow \phi_f = \dfrac{\phi_L + \phi_R}{2}$

Linear: $\Pi_{cs}^L \rightarrow \phi_f = \dfrac{d_R \phi_L + d_L \phi_R}{d_f}$

Volumetric: $\Pi_{cs}^V \rightarrow \phi_f = \dfrac{d_L \phi_L + d_R \phi_R}{d_f}$



Interpolators related, 1 d.o.f.

Midpoint results in checkerboard

Relation between Ker(L_c) and mesh – Midpoint

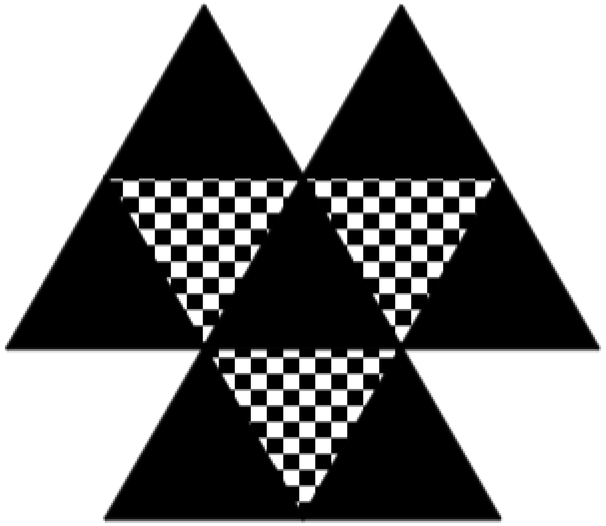
$$[L_c]_{i,k} = \sum_j \frac{1}{4\,[\Omega_c]_j} \left(A_{i,j}\mathbf{n}_{i,j}\right) \cdot \left(A_{j,k}\mathbf{n}_{j,k}\right)$$

Group together the second neighbours

Relation between Ker(L_c) and mesh – Midpoint

$$[L_C]_{i,k} = \sum_j \frac{1}{4\,[\Omega_C]_j} \left(A_{i,j}\mathbf{n}_{i,j}\right) \cdot \left(A_{j,k}\mathbf{n}_{j,k}\right)$$

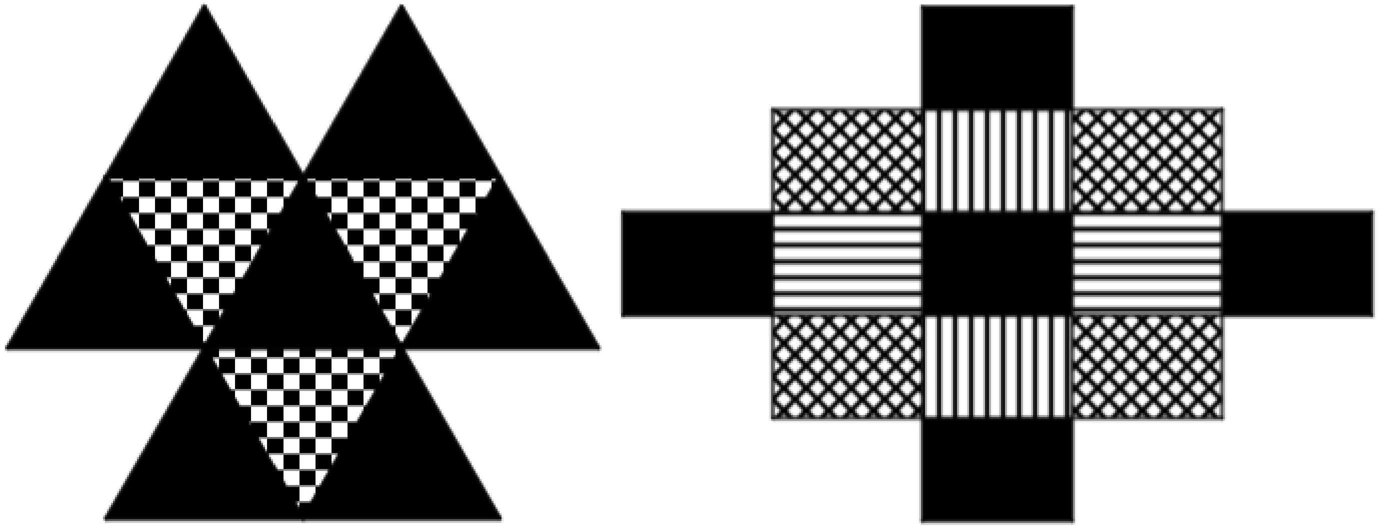
Group together the second neighbours



Relation between Ker(L_c) and mesh – Midpoint

$$[L_c]_{i,k} = \sum_j \frac{1}{4\,[\Omega_c]_j} \left(A_{i,j}\mathbf{n}_{i,j}\right) \cdot \left(A_{j,k}\mathbf{n}_{j,k}\right)$$

Group together the second neighbours

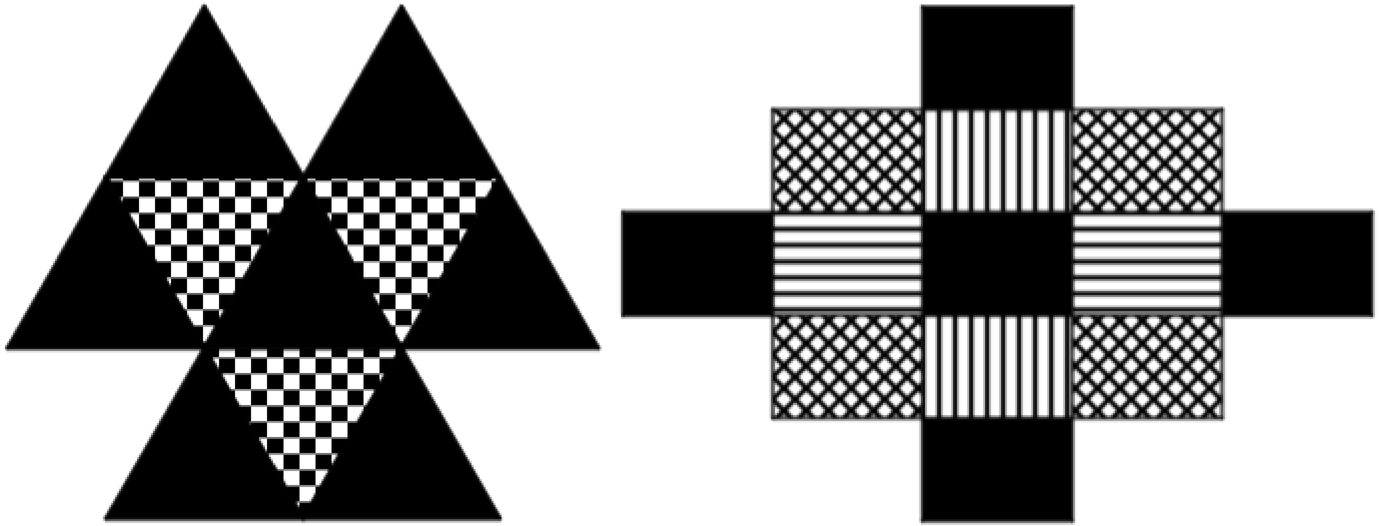


Relation between Ker(L_c) and mesh – Midpoint

$$[L_c]_{i,k} = \sum_j \frac{1}{4\,[\Omega_c]_j} \underbrace{(A_{i,j}\mathbf{n}_{i,j}) \cdot (A_{j,k}\mathbf{n}_{j,k})}$$

Group together the non-orthogonal second neighbours

= 0 for orthogonal faces → Cartesian meshes

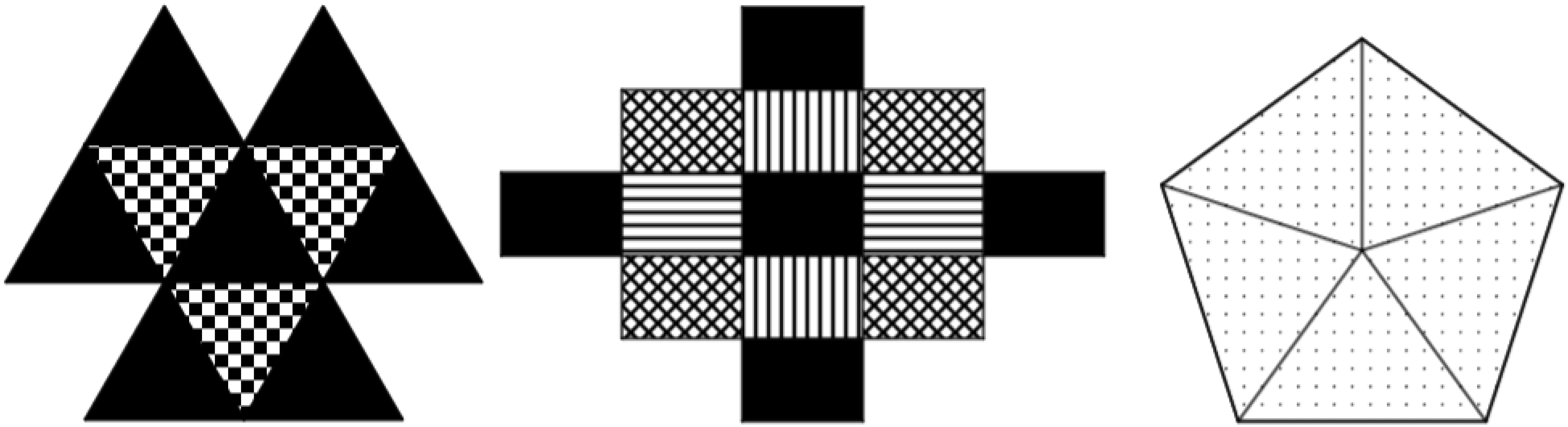


Relation between Ker(L_c) and mesh – Midpoint

$$[L_c]_{i,k} = \sum_j \frac{1}{4\,[\Omega_c]_j} \underbrace{(A_{i,j}\mathbf{n}_{i,j}) \cdot (A_{j,k}\mathbf{n}_{j,k})}$$

Group together the non-orthogonal second neighbours

= 0 for orthogonal faces → Cartesian meshes

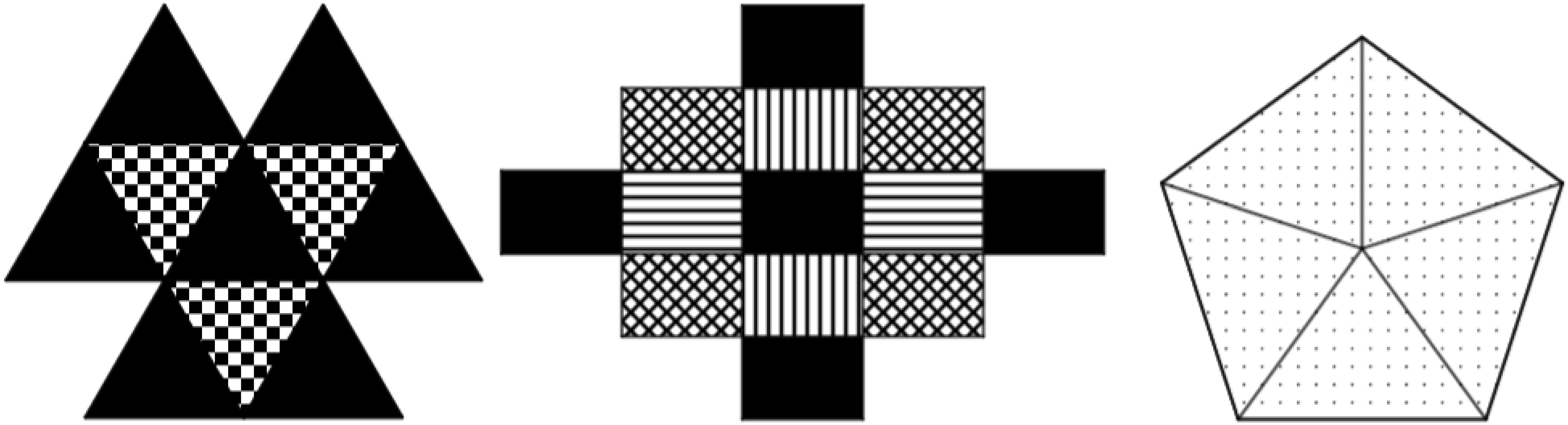


Relation between Ker(L_c) and mesh – Midpoint

$$[L_c]_{i,k} = \sum_j \frac{1}{4\,[\Omega_c]_j} \underbrace{(A_{i,j}\mathbf{n}_{i,j}) \cdot (A_{j,k}\mathbf{n}_{j,k})}$$

Group together the non-orthogonal second neighbours

= 0 for orthogonal faces → Cartesian meshes



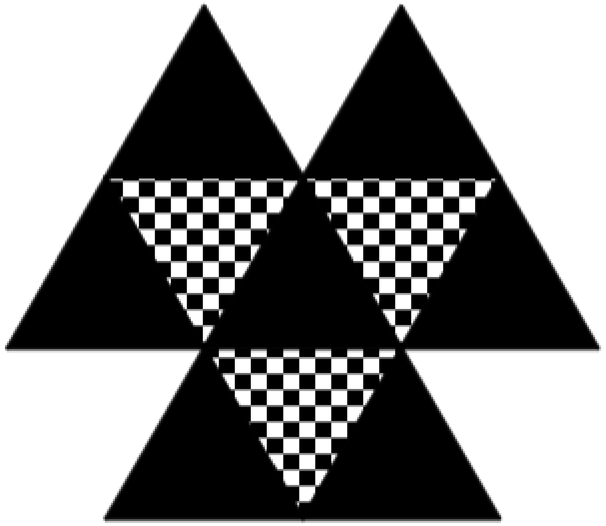
Vectors that span the kernel can be derived from the mesh.
Nullity = number of disconnected cell groups

Relation between Ker(L_c) and mesh – Midpoint

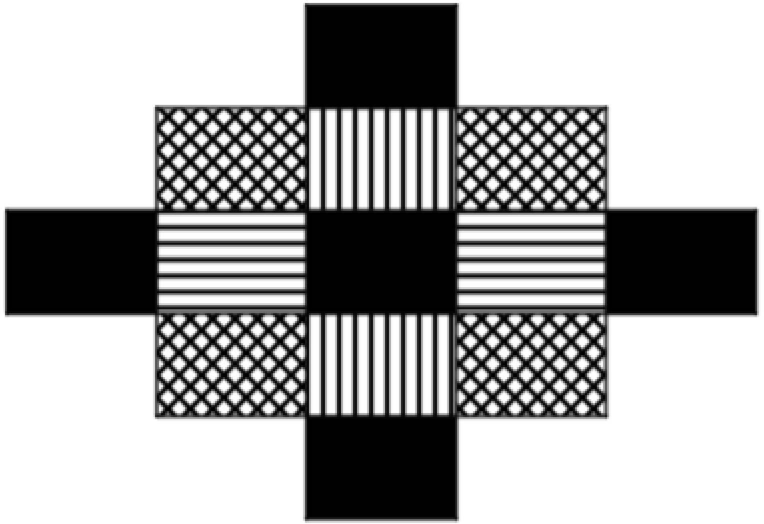
$$[L_c]_{i,k} = \sum_j \frac{1}{4\,[\Omega_c]_j} \underbrace{(A_{i,j}\mathbf{n}_{i,j}) \cdot (A_{j,k}\mathbf{n}_{j,k})}$$

Group together the non-orthogonal second neighbours

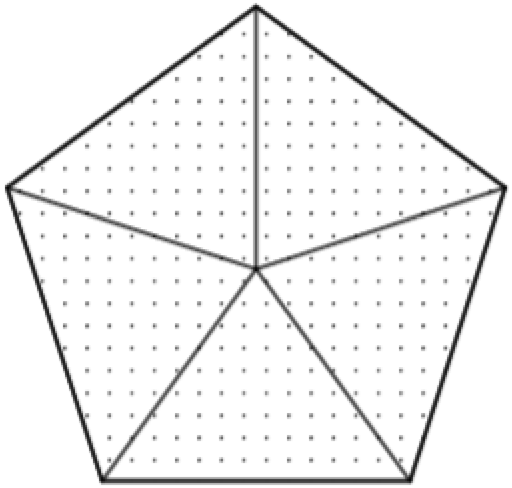
= 0 for orthogonal faces → Cartesian meshes



Nullity = 2



Nullity = 2^{Dim} = 4

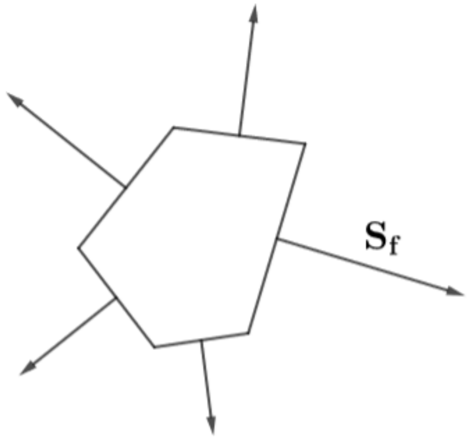


Nullity = 1

Vectors that span the kernel can be derived from the mesh.
Nullity = number of disconnected cell groups

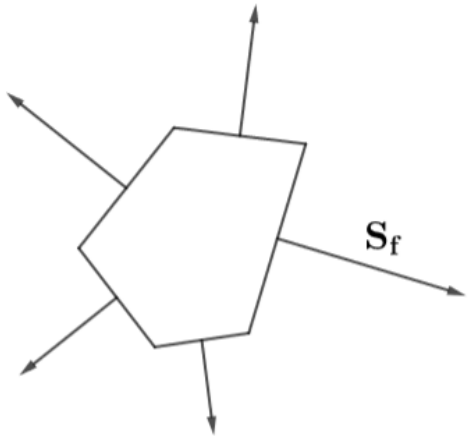
Rewriting the gradient operator

Gauss Gradient: $$G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$$



Rewriting the gradient operator

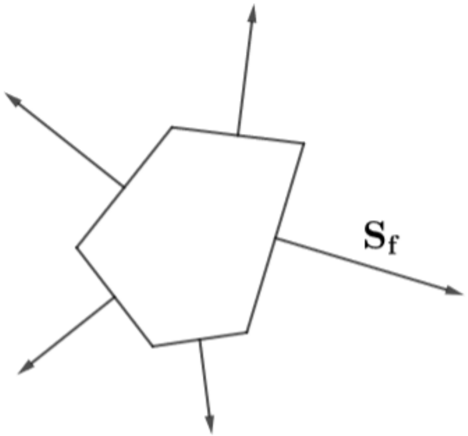
Gauss Gradient: $$G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$$



$$\sum \phi_i \mathbf{S}_f = \mathbf{0}$$

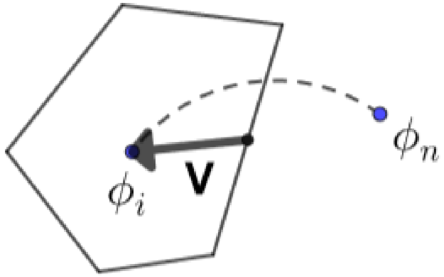
Rewriting the gradient operator

Gauss Gradient: $$G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$$



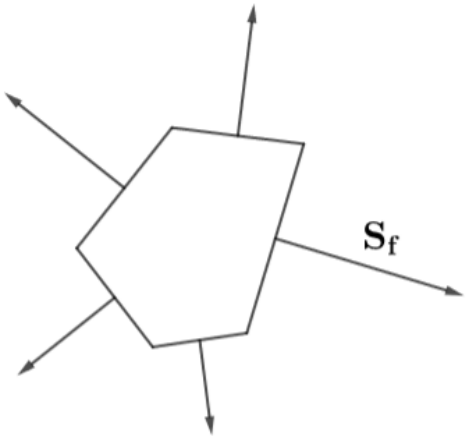
$$\sum \phi_i \mathbf{S}_f = \mathbf{0}$$

$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} {\Gamma_{cs}^{V}}^{T} \Omega_s G\phi$$



Rewriting the gradient operator

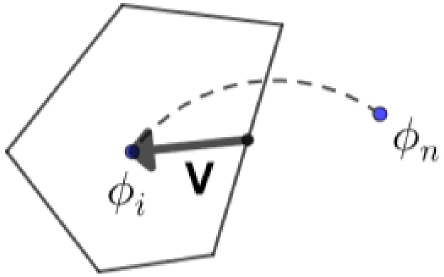
Gauss Gradient: $G_G \phi_f = \frac{1}{V} \sum \phi_f \mathbf{S}_f$



$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

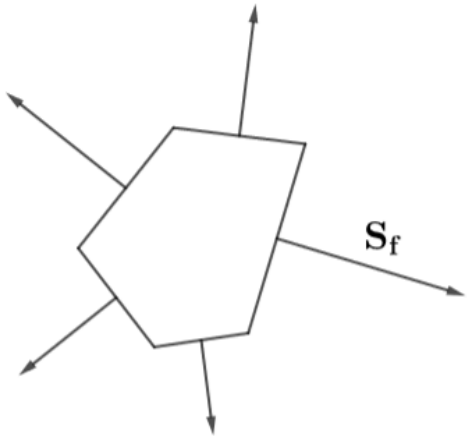
$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} \Gamma_{cs}^{V\ T} \Omega_s G\phi$$

$$= \frac{1}{V} \sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} & \\ \frac{-1}{d_f} w_{fi} & \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



Rewriting the gradient operator

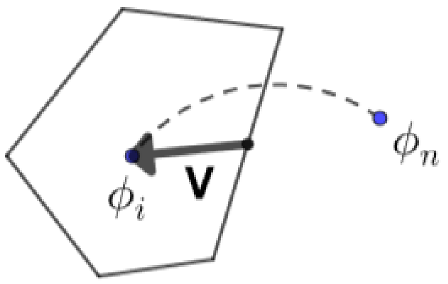
Gauss Gradient: $G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$



$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

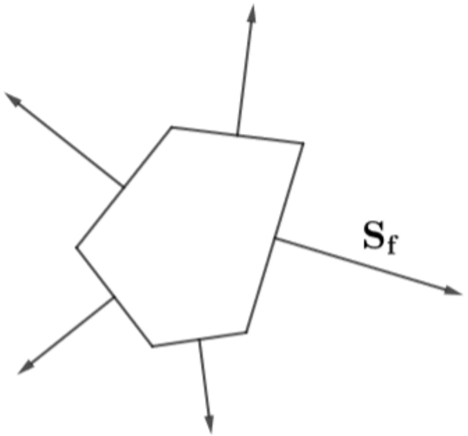
$$\Gamma_{sc}^{V} G\phi = \underline{\Omega}^{-1} \Gamma_{cs}^{V\ T} \Omega_s G\phi$$

$$= \underline{\frac{1}{V}} \sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} \\ \frac{-1}{d_f} w_{fi} \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



Rewriting the gradient operator

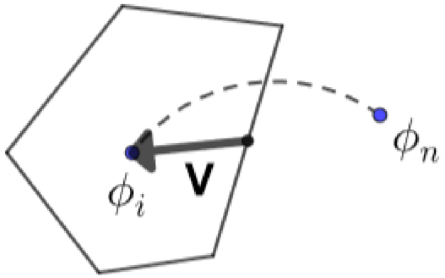
Gauss Gradient: $G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$



$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

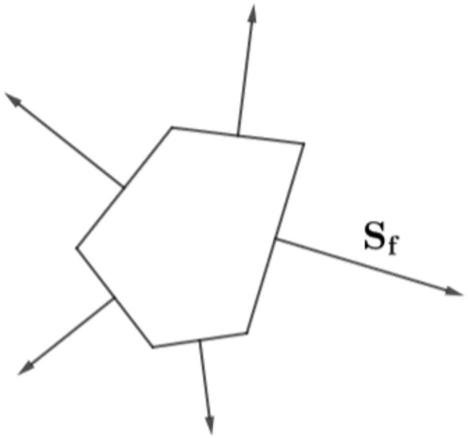
$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} {\Gamma_{cs}^{V}}^{T} \Omega_s G\phi$$

$$= \frac{1}{V}\sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} & \\ \frac{-1}{d_f} w_{fi} & \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



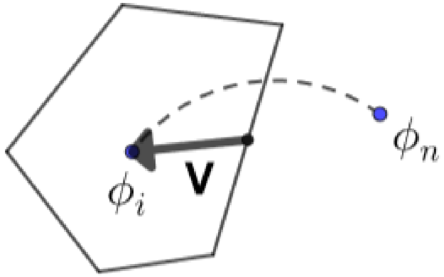
Rewriting the gradient operator

Gauss Gradient: $$G_G \phi_f = \frac{1}{V} \sum \phi_f \mathbf{S}_f$$



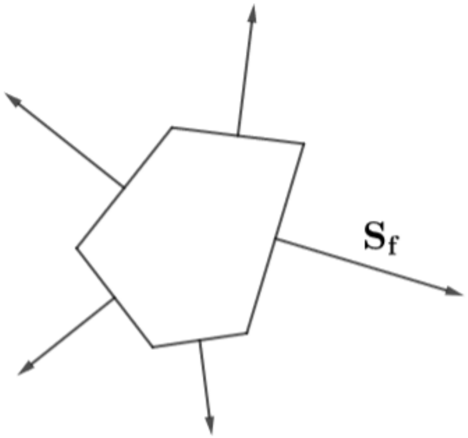
$$\sum \phi_i \mathbf{S}_f = \mathbf{0}$$

$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} \Gamma_{cs}^{V\ T} \Omega_s G\phi$$
$$= \frac{1}{V} \sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} \\ \frac{-1}{d_f} w_{fi} \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



Rewriting the gradient operator

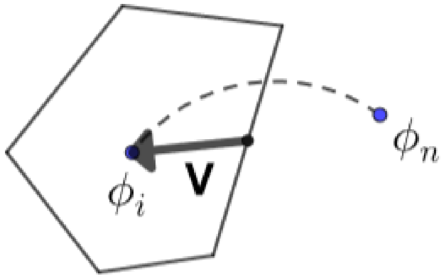
Gauss Gradient: $G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$



$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

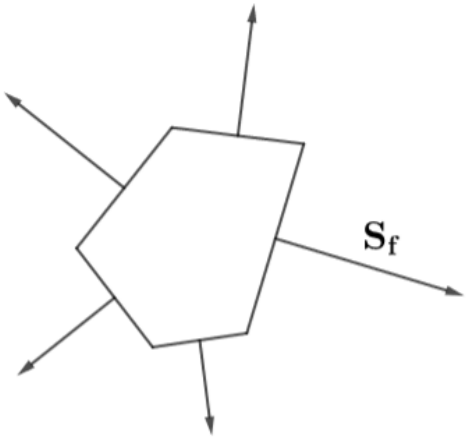
$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} {\Gamma_{cs}^{V}}^{T} \Omega_s G\phi$$

$$= \frac{1}{V}\sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} \\ \frac{-1}{d_f} w_{fi} \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



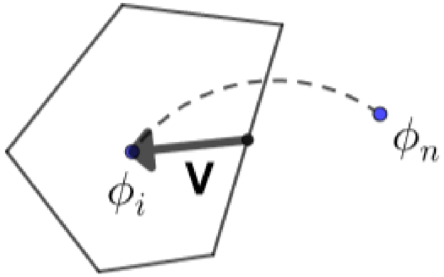
Rewriting the gradient operator

Gauss Gradient: $G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$



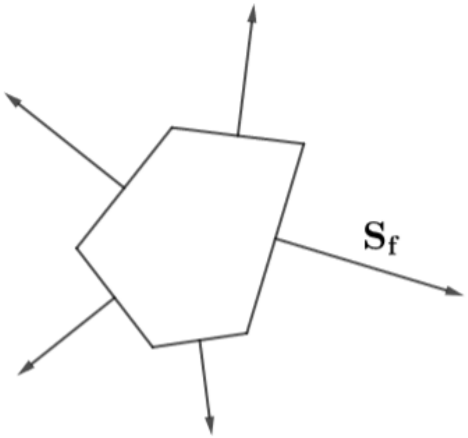
$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} {\Gamma_{cs}^{V}}^{T} \Omega_s G\phi$$
$$= \frac{1}{V}\sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} \\ \frac{-1}{d_f}(1 - w_{fn}) \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



Rewriting the gradient operator

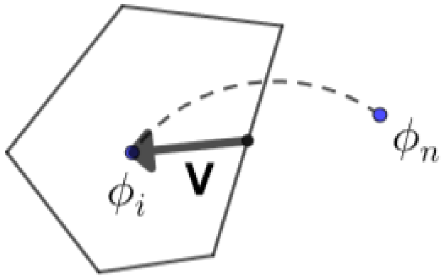
Gauss Gradient: $G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$



$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

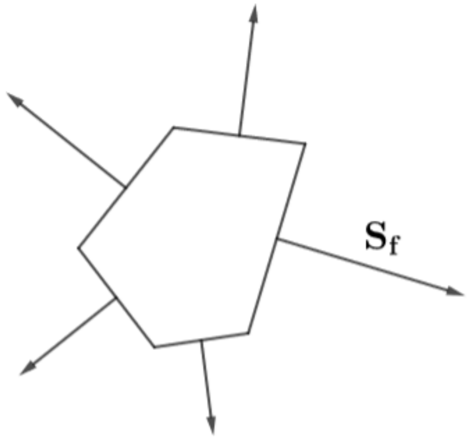
$$\Gamma_{sc}^{V} G\phi = \Omega^{-1}{\Gamma_{cs}^{V}}^{T}\Omega_s G\phi$$

$$= \frac{1}{V}\sum \mathbf{S}_f d_f \begin{bmatrix} \frac{1}{d_f} w_{fi} \\ \frac{-1}{d_f}(\cancel{1} - w_{fn}) \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



Rewriting the gradient operator

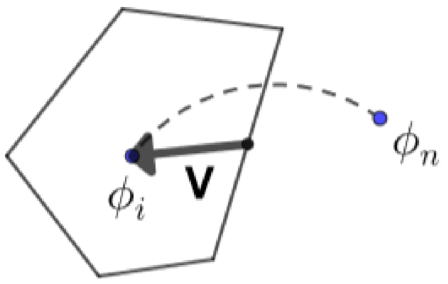
Gauss Gradient: $G_G\phi_f = \frac{1}{V}\sum \phi_f \mathbf{S}_f$



$\sum \phi_i \mathbf{S}_f = \mathbf{0}$

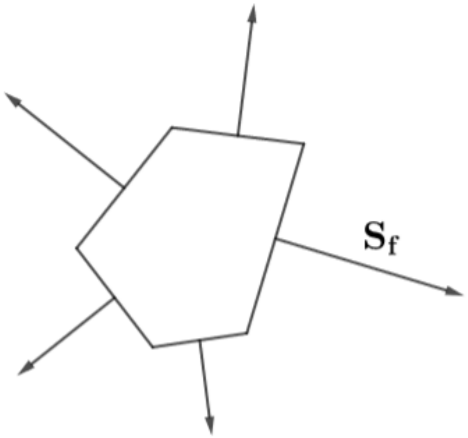
$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} {\Gamma_{cs}^{V}}^{T} \Omega_s G\phi$$

$$= \frac{1}{V}\sum \mathbf{S}_f \begin{bmatrix} w_{fi} \\ w_{fn} \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$



Rewriting the gradient operator

Gauss Gradient: $$G_G \phi_f = \frac{1}{V} \sum \phi_f \mathbf{S}_f$$

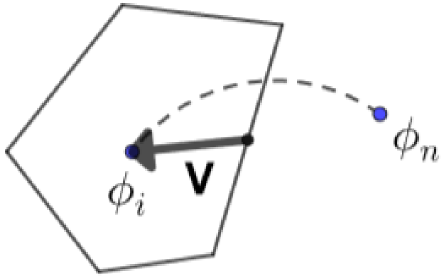


$$\sum \phi_i \mathbf{S}_f = \mathbf{0}$$

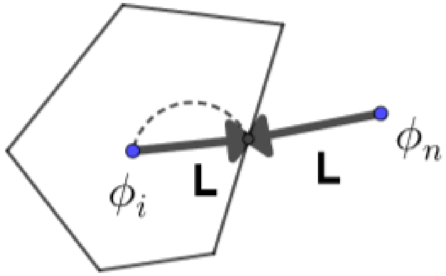
$$\Gamma_{sc}^{V} G\phi = \Omega^{-1} {\Gamma_{cs}^{V}}^{T} \Omega_s G\phi$$

$$= \frac{1}{V} \sum \mathbf{S}_f \begin{bmatrix} w_{fi} \\ w_{fn} \end{bmatrix} \cdot \begin{bmatrix} \phi_n \\ \phi_i \end{bmatrix}$$

$$= G_G \Pi_{cs}^{L} \phi$$

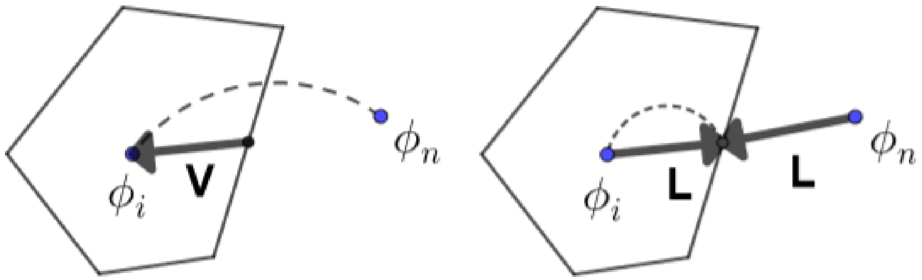


=



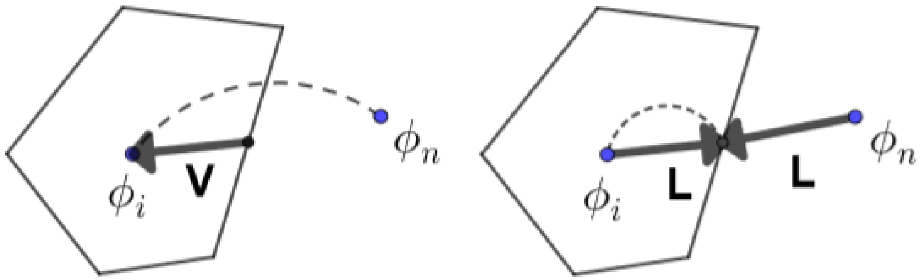
Relations between gradient operators

	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^M G$	$G_G \Pi_{cs}^M$
Linear	$\Gamma_{sc}^L G$	$G_G \Pi_{cs}^L$
Volumetric	$\Gamma_{sc}^V G$	$G_G \Pi_{cs}^V$



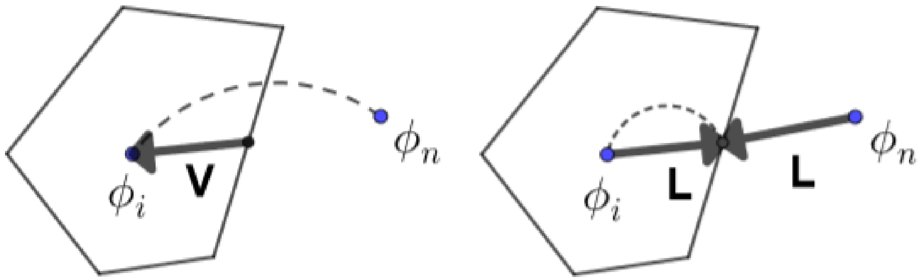
Relations between gradient operators

	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^M G$ =	$G_G \Pi_{cs}^M$
Linear	$\Gamma_{sc}^L G$	$G_G \Pi_{cs}^L$
Volumetric	$\Gamma_{sc}^V G$	$G_G \Pi_{cs}^V$



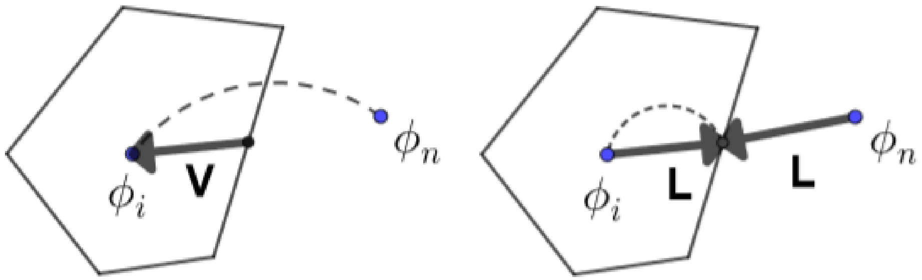
Relations between gradient operators

	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^{M} G$ =	$G_G \Pi_{cs}^{M}$
Linear	$\Gamma_{sc}^{L} G$	$G_G \Pi_{cs}^{L}$
Volumetric	$\Gamma_{sc}^{V} G$	$G_G \Pi_{cs}^{V}$



Relations between gradient operators

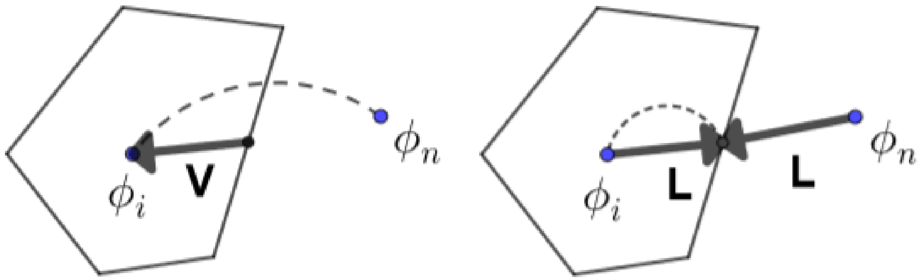
	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^{M} G$ =	$G_G \Pi_{cs}^{M}$
Linear	$\Gamma_{sc}^{L} G$	$G_G \Pi_{cs}^{L}$
Volumetric	$\Gamma_{sc}^{V} G$	$G_G \Pi_{cs}^{V}$



Relations between gradient operators

	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^{M} G$ =	$G_G \Pi_{cs}^{M}$
Linear	$\Gamma_{sc}^{L} G$	$G_G \Pi_{cs}^{L}$
Volumetric	$\Gamma_{sc}^{V} G$	$G_G \Pi_{cs}^{V}$

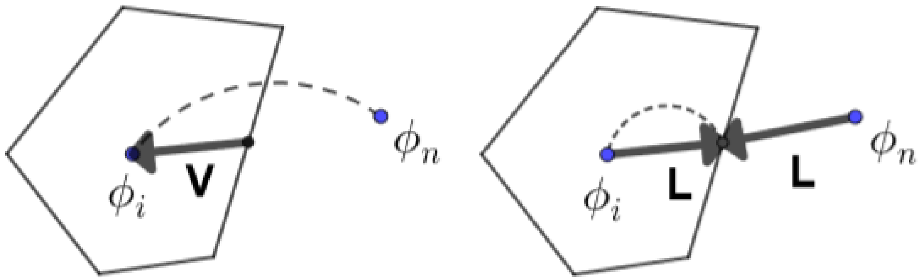
- Less options for L_c



Relations between gradient operators

	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^M G$ =	$G_G \Pi_{cs}^M$
Linear	$\Gamma_{sc}^L G$	$G_G \Pi_{cs}^L$
Volumetric	$\Gamma_{sc}^V G$	$G_G \Pi_{cs}^V$

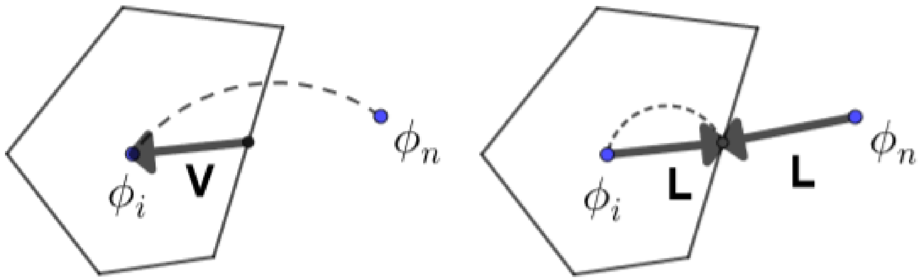
- Less options for L_c
- G_G is easier to implement



Relations between gradient operators

	$G_c = \Gamma_{sc}G$	$G_c = G_G\Pi_{cs}$
Midpoint	$\Gamma_{sc}^{M}G$	$G_G\Pi_{cs}^{M}$
Linear	$\Gamma_{sc}^{L}G$	$G_G\Pi_{cs}^{L}$
Volumetric	$\Gamma_{sc}^{V}G$	$G_G\Pi_{cs}^{V}$

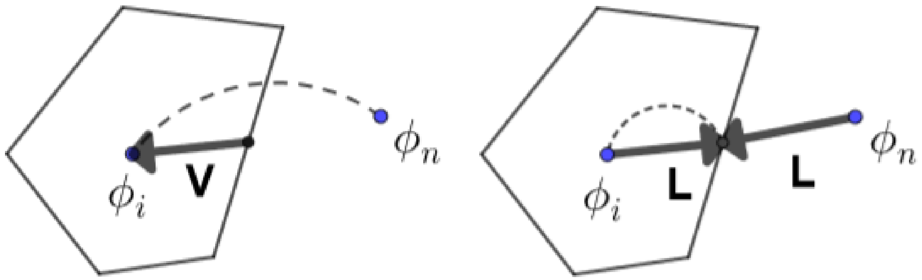
- Less options for L_c
- G_G is easier to implement
- Many solvers use $M\Gamma_{cs}^{L}G_G\Pi_{cs}^{L}$ which is non-symmetric



Relations between gradient operators

	$G_c = \Gamma_{sc} G$	$G_c = G_G \Pi_{cs}$
Midpoint	$\Gamma_{sc}^M G$ =	$G_G \Pi_{cs}^M$
Linear	$\Gamma_{sc}^L G$	$G_G \Pi_{cs}^L$
Volumetric	$\Gamma_{sc}^V G$	$G_G \Pi_{cs}^V$

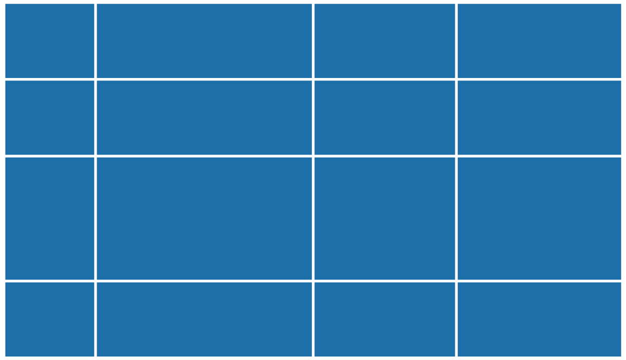
- Less options for L_c
- G_G is easier to implement
- Many solvers use $M\Gamma_{cs}^L G_G \Pi_{cs}^L$ which is non-symmetric
- Useful in deriving kernel vectors



Predicting kernel vectors

Predicting kernel vectors

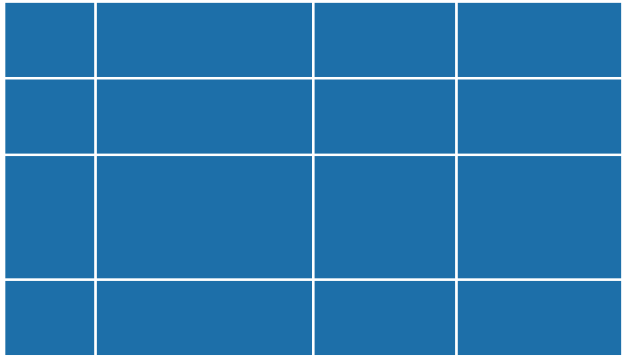
$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$

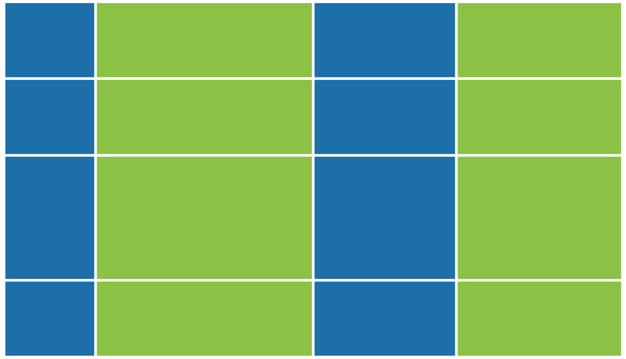


Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$

$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i$$



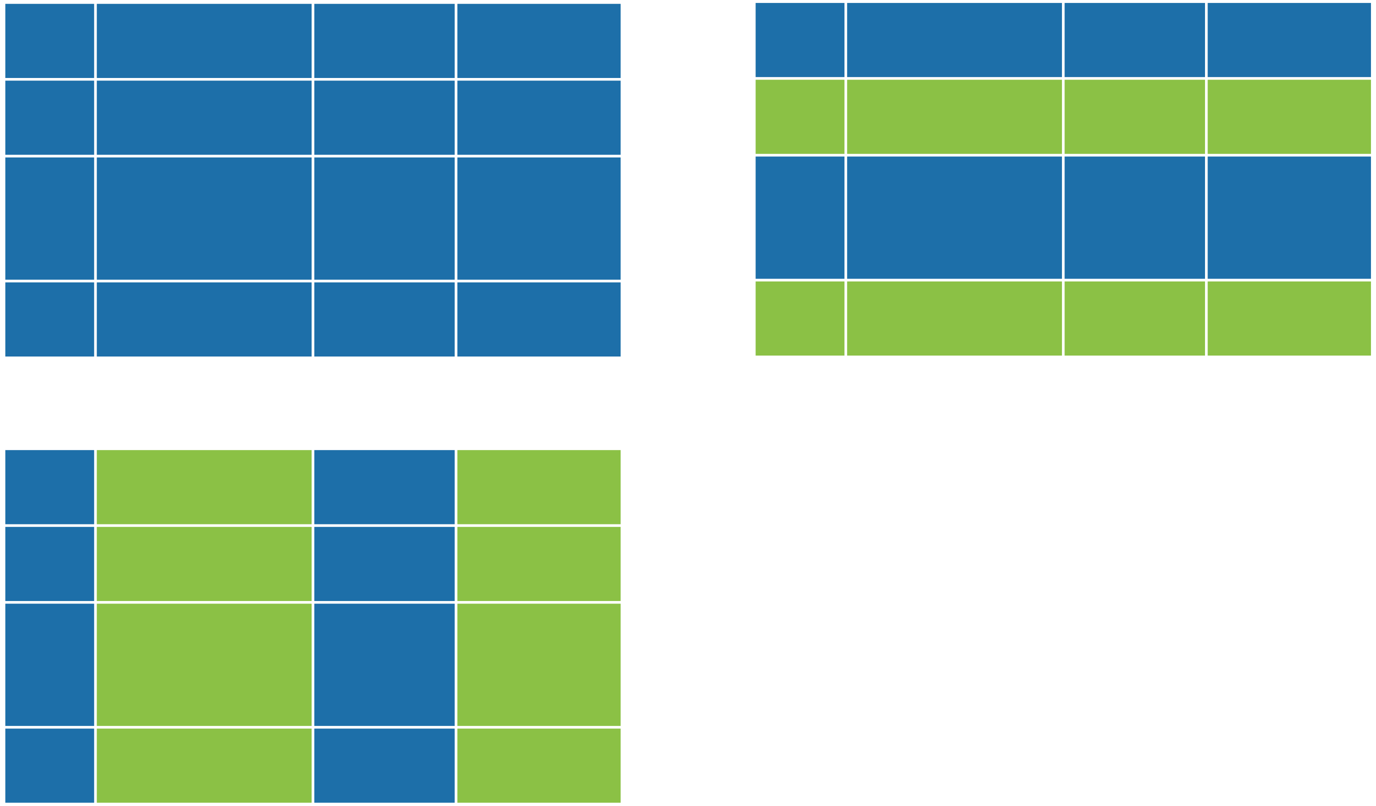


Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$

$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i$$

$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^j$$



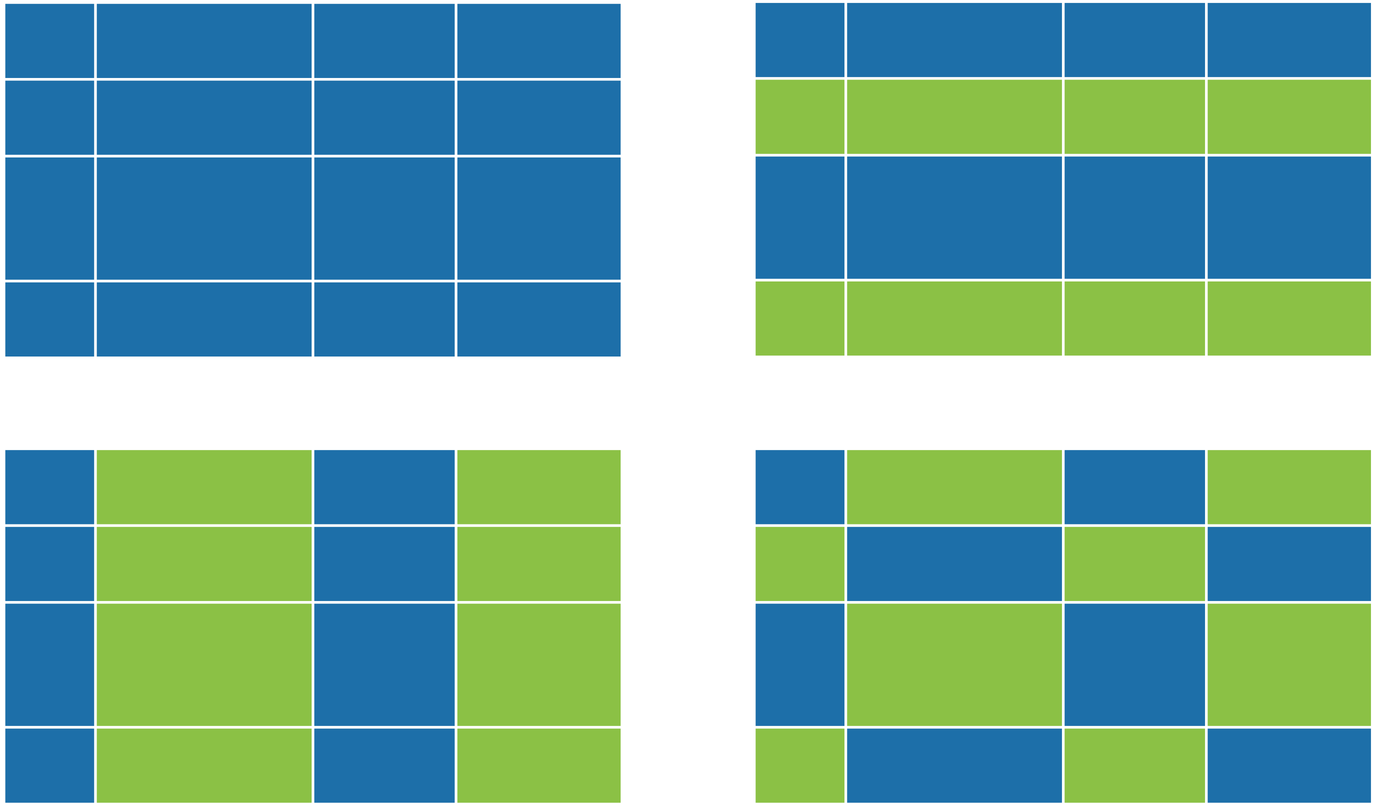
Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$

$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i$$

$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^j$$

$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j}$$



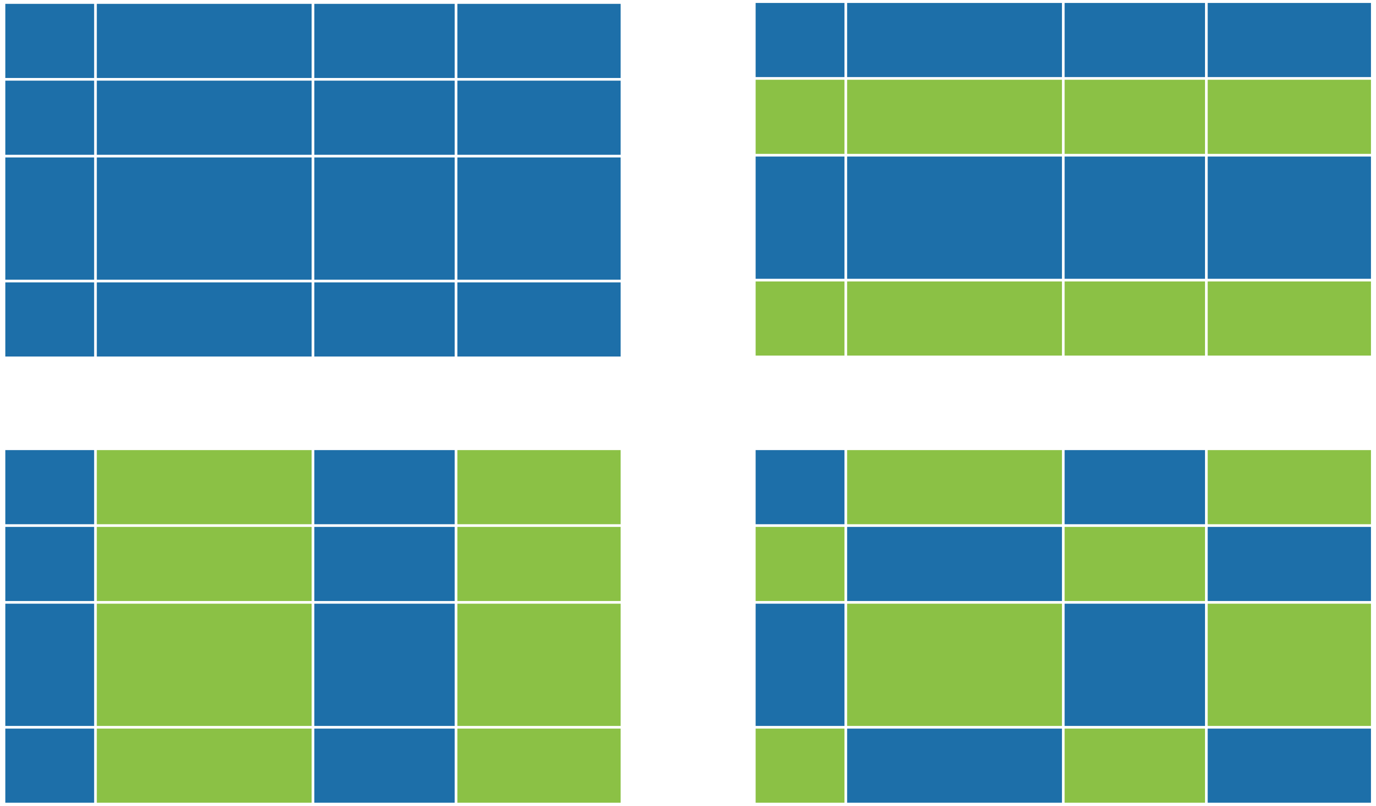
Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$

$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^{i}$$

$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^{j}$$

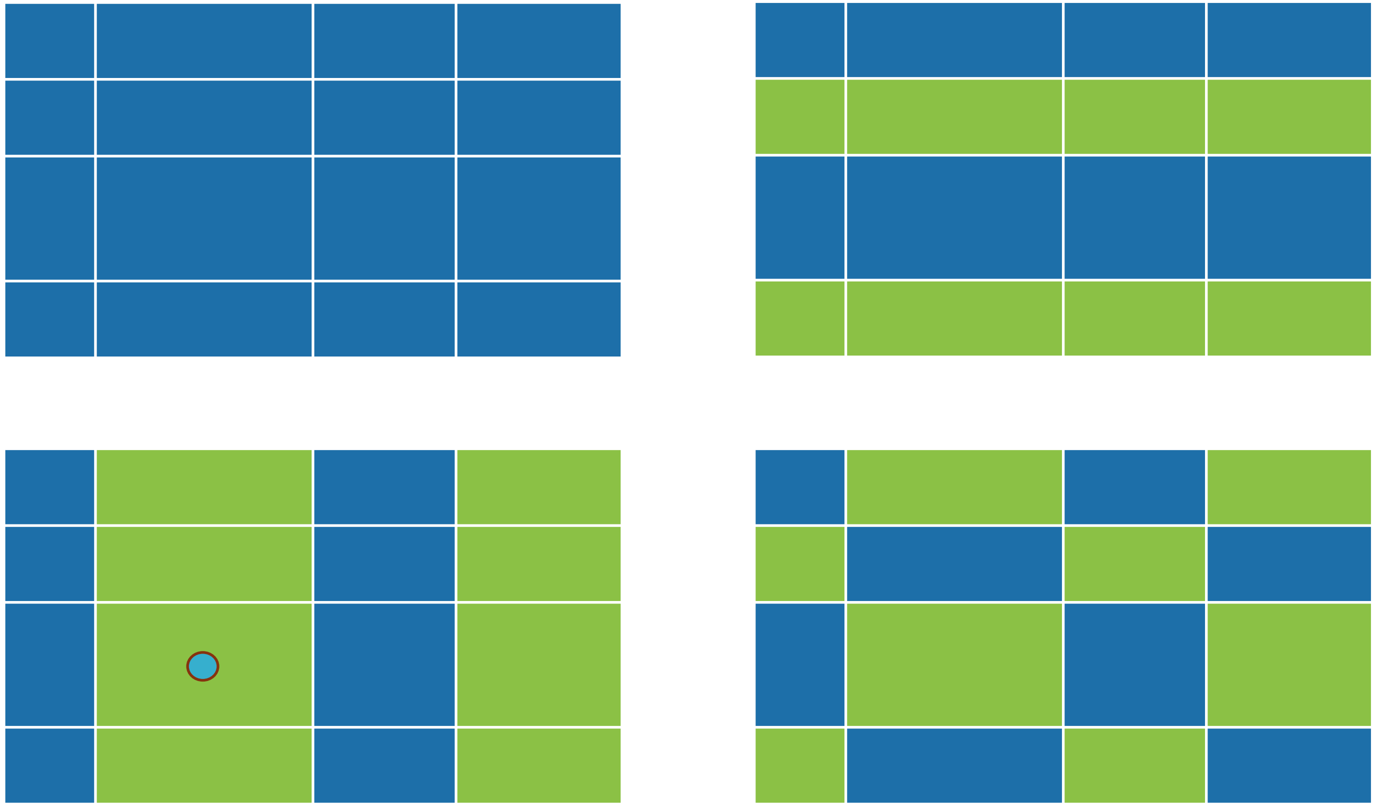
$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j}$$



$$G_c = G_G \Pi_{cs}^{L}: \quad \mathbf{p}_c^{-(1)} = \Delta x_0 \quad -\Delta x_1 \quad \Delta x_2 \quad -\Delta x_3$$

Predicting kernel vectors

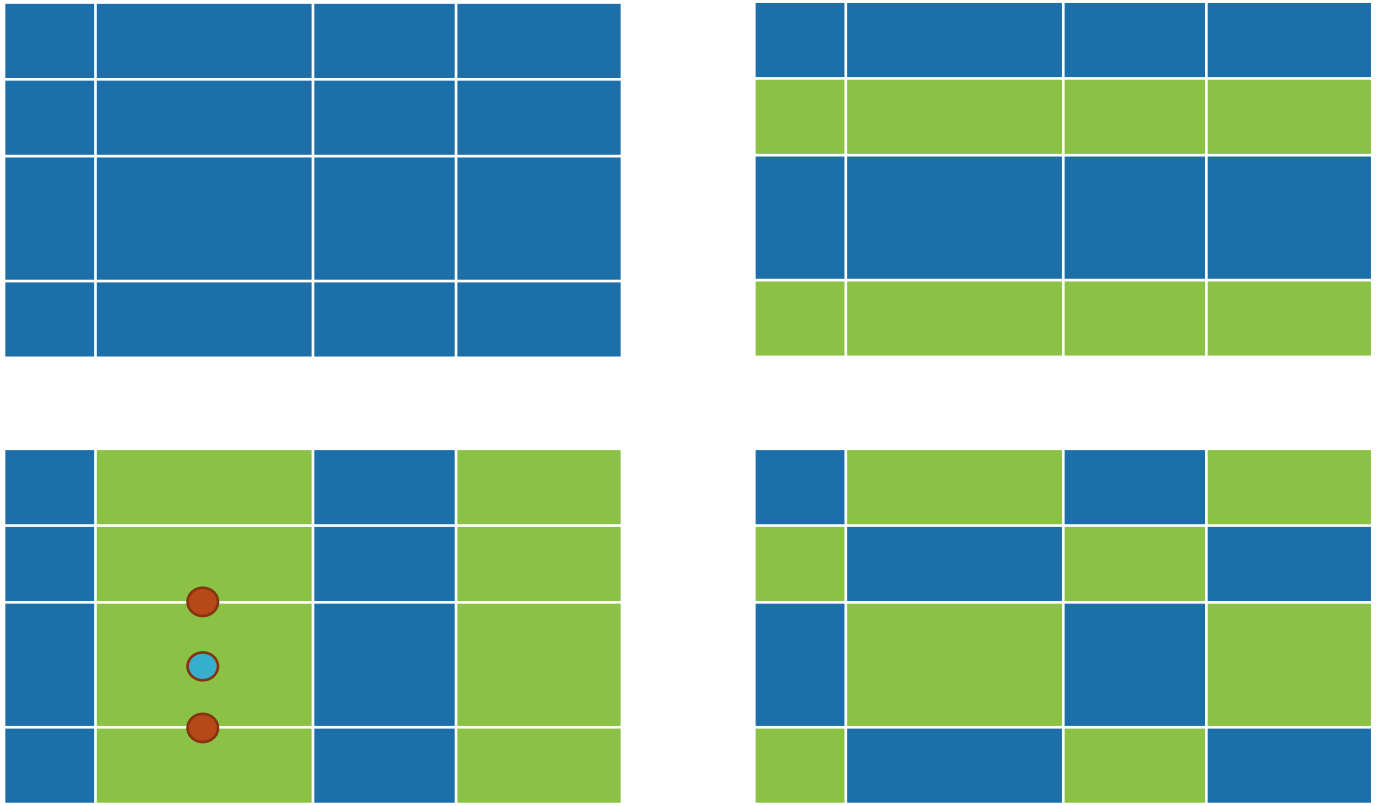
$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$
$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i$$
$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^j$$
$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j}$$



$$G_c = G_G \Pi_{cs}^L : \quad \mathbf{p}_c^{-(1)} = \Delta x_0 \quad -\Delta x_1 \quad \Delta x_2 \quad -\Delta x_3$$

Predicting kernel vectors

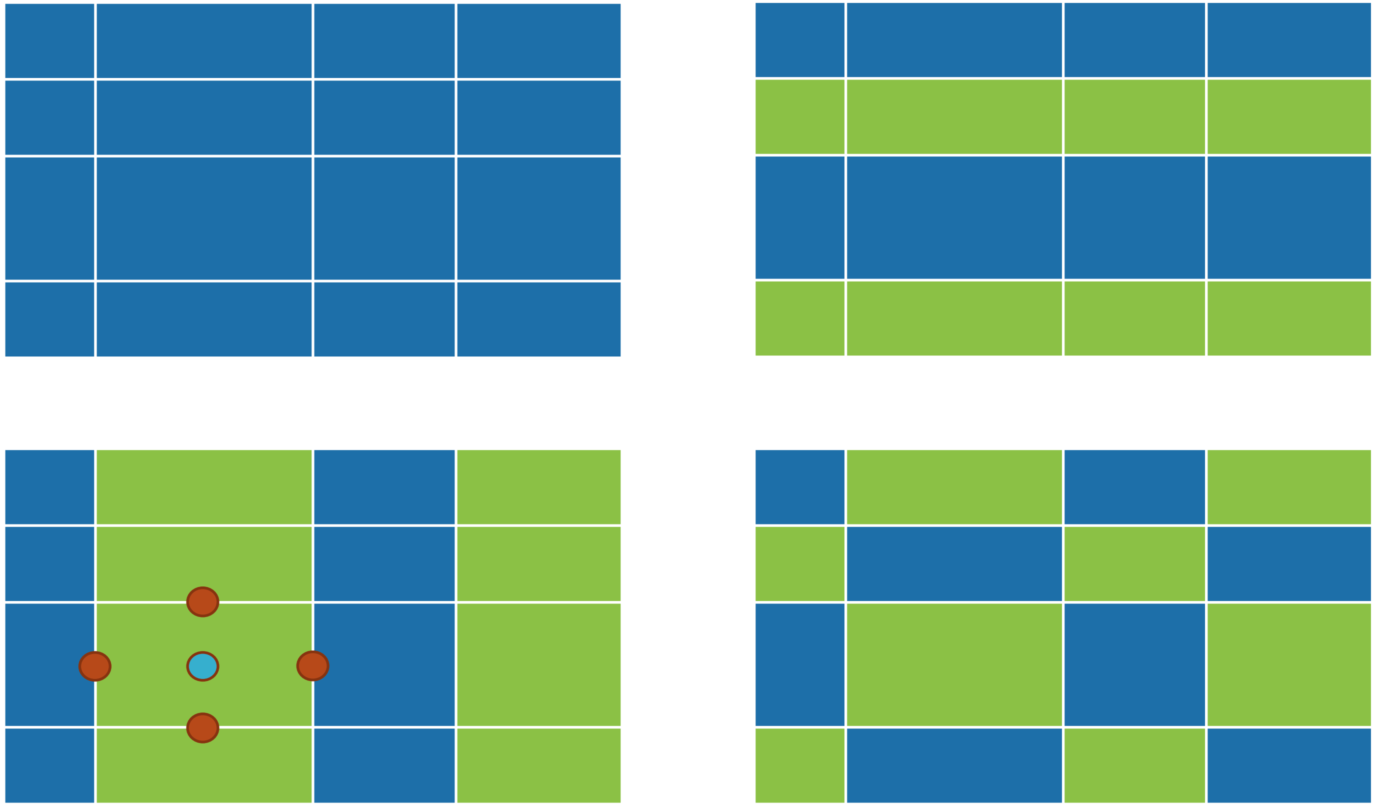
$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$
$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i$$
$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^j$$
$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j}$$



$$G_c = G_G \Pi_{cs}^L : \quad \mathbf{p}_c^{-(1)} = \Delta x_0 \quad -\Delta x_1 \quad \Delta x_2 \quad -\Delta x_3$$

Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$
$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^{i}$$
$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^{j}$$
$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j}$$



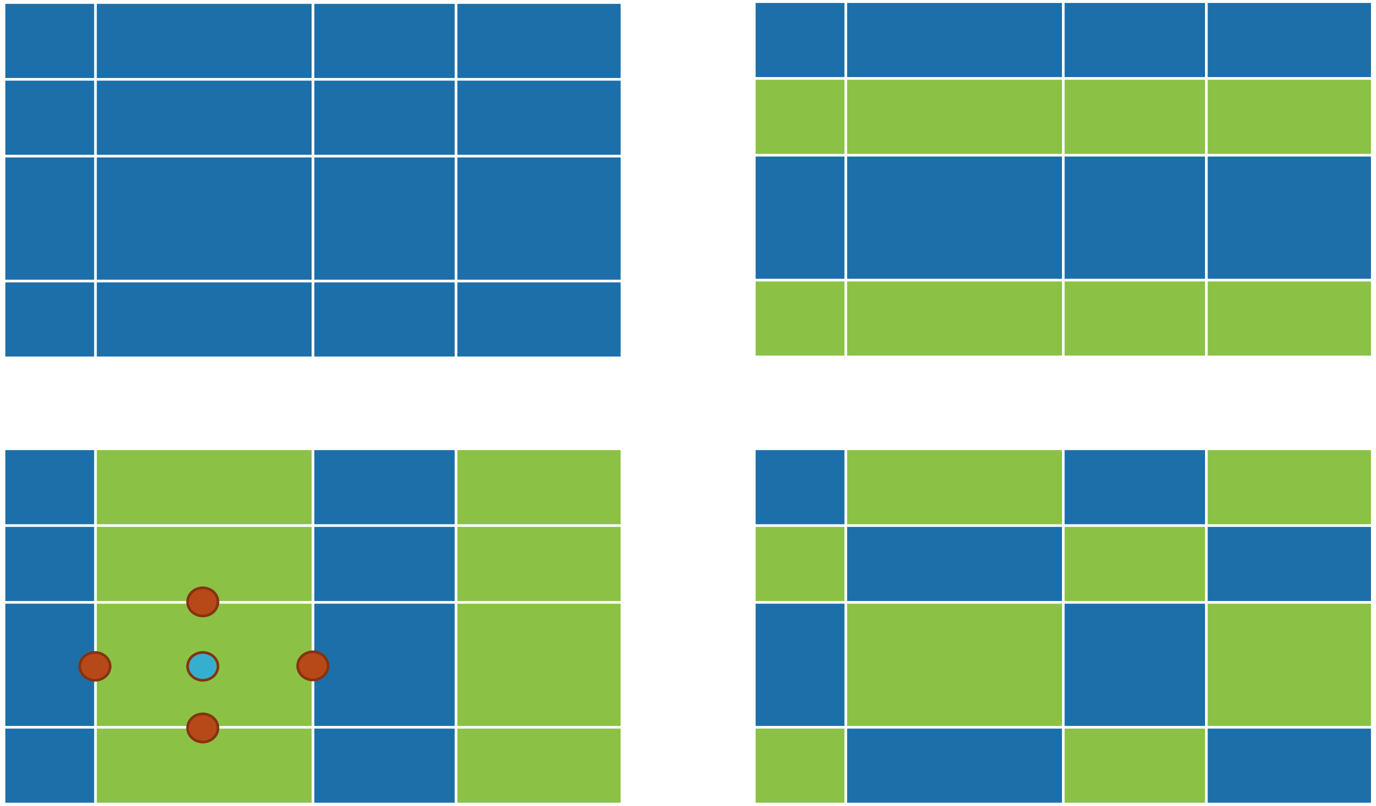
$$G_c = G_G \Pi_{cs}^{L}: \quad \mathbf{p}_c^{-(1)} = \Delta x_0 \quad -\Delta x_1 \quad \Delta x_2 \quad -\Delta x_3$$

$$\Pi_{cs}^{L} \rightarrow \phi_w = \frac{\Delta x_0 \Delta x_1 - \Delta x_1 \Delta x_0}{\Delta x_0 + \Delta x_1} = 0$$

Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$
$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i (\Delta x_i)^\alpha$$
$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^j (\Delta y_j)^\alpha$$
$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j} (\Delta x_i \Delta y_j)^\alpha$$

$$\alpha = \begin{cases} 1, & \text{if linear} \\ 0, & \text{if midpoint} \\ -1, & \text{if volumetric} \end{cases}$$



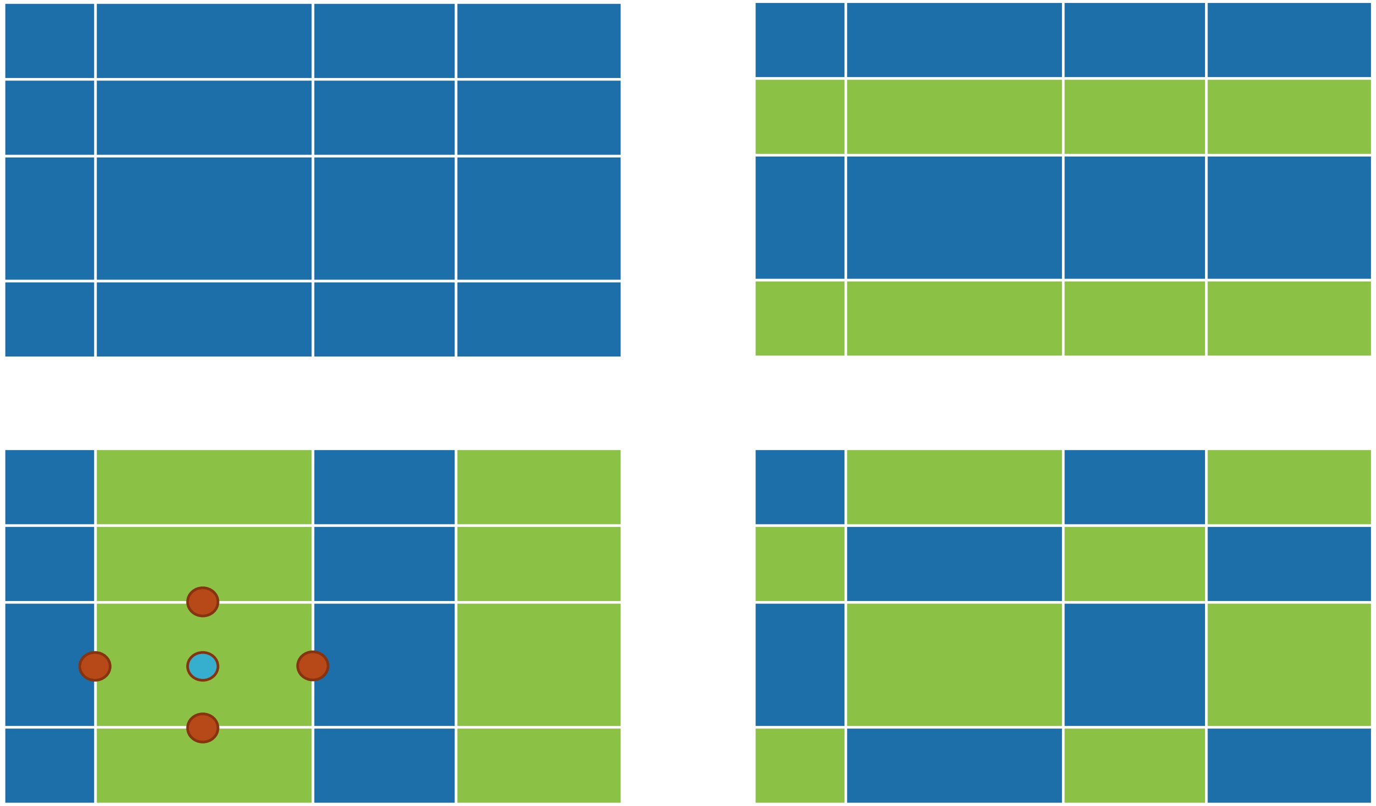
$$G_c = G_G \Pi_{cs}^L : \quad \mathbf{p}_c^{-(1)} = \Delta x_0 \quad -\Delta x_1 \quad \Delta x_2 \quad -\Delta x_3$$

$$\Pi_{cs}^L \rightarrow \phi_w = \frac{\Delta x_0 \Delta x_1 - \Delta x_1 \Delta x_0}{\Delta x_0 + \Delta x_1} = 0$$

Predicting kernel vectors

$$[\mathbf{p}_c^{-(0)}]_{i,j} = 1$$
$$[\mathbf{p}_c^{-(1)}]_{i,j} = (-1)^i (\Delta x_i)^\alpha$$
$$[\mathbf{p}_c^{-(2)}]_{i,j} = (-1)^j (\Delta y_j)^\alpha$$
$$[\mathbf{p}_c^{-(3)}]_{i,j} = (-1)^{i+j} (\Delta x_i \Delta y_j)^\alpha$$

$$\alpha = \begin{cases} 1, & \text{if linear} \\ 0, & \text{if midpoint} \\ -1, & \text{if volumetric} \end{cases}$$



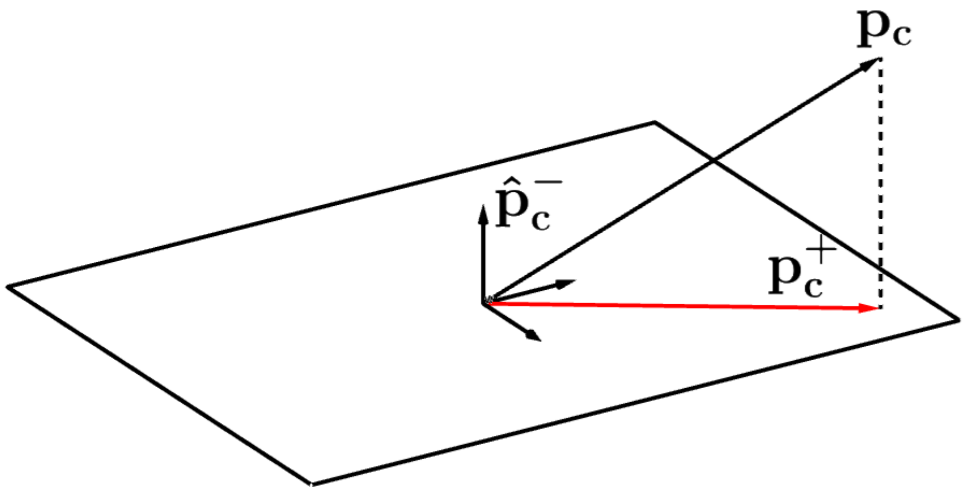
$$G_c = G_G \Pi_{cs}^L : \quad \mathbf{p}_c^{-(1)} = \Delta x_0 \quad -\Delta x_1 \quad \Delta x_2 \quad -\Delta x_3$$

$$\Pi_{cs}^L \to \phi_w = \frac{\Delta x_0 \Delta x_1 - \Delta x_1 \Delta x_0}{\Delta x_0 + \Delta x_1} = 0$$

Although not necessarily orthogonal, they are linearly independent, spanning the nullspace of L_c

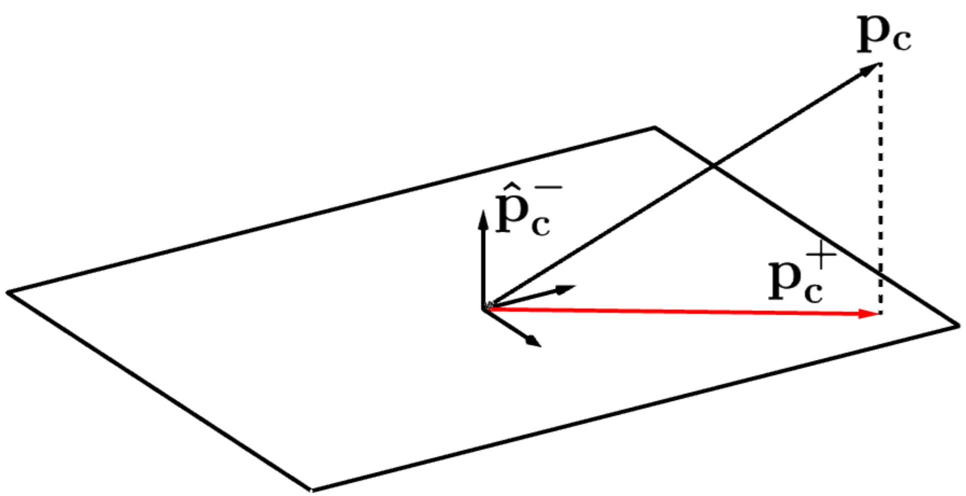
Filtering spurious modes

$$\mathbf{p}_c^+ = \mathbf{p}_c - \sum_i \left(\mathbf{p}_c \cdot \hat{\mathbf{p}}_c^{-(i)} \right) \hat{\mathbf{p}}_c^{-(i)}$$



Filtering spurious modes

$$\mathbf{p}_c^{+} = \mathbf{p}_c - \sum_i \left(\mathbf{p}_c \cdot \hat{\mathbf{p}}_c^{-(i)}\right) \hat{\mathbf{p}}_c^{-(i)}$$



4428 Shashank et al. / Journal of Computational Physics 229 (2010) 4425–4430

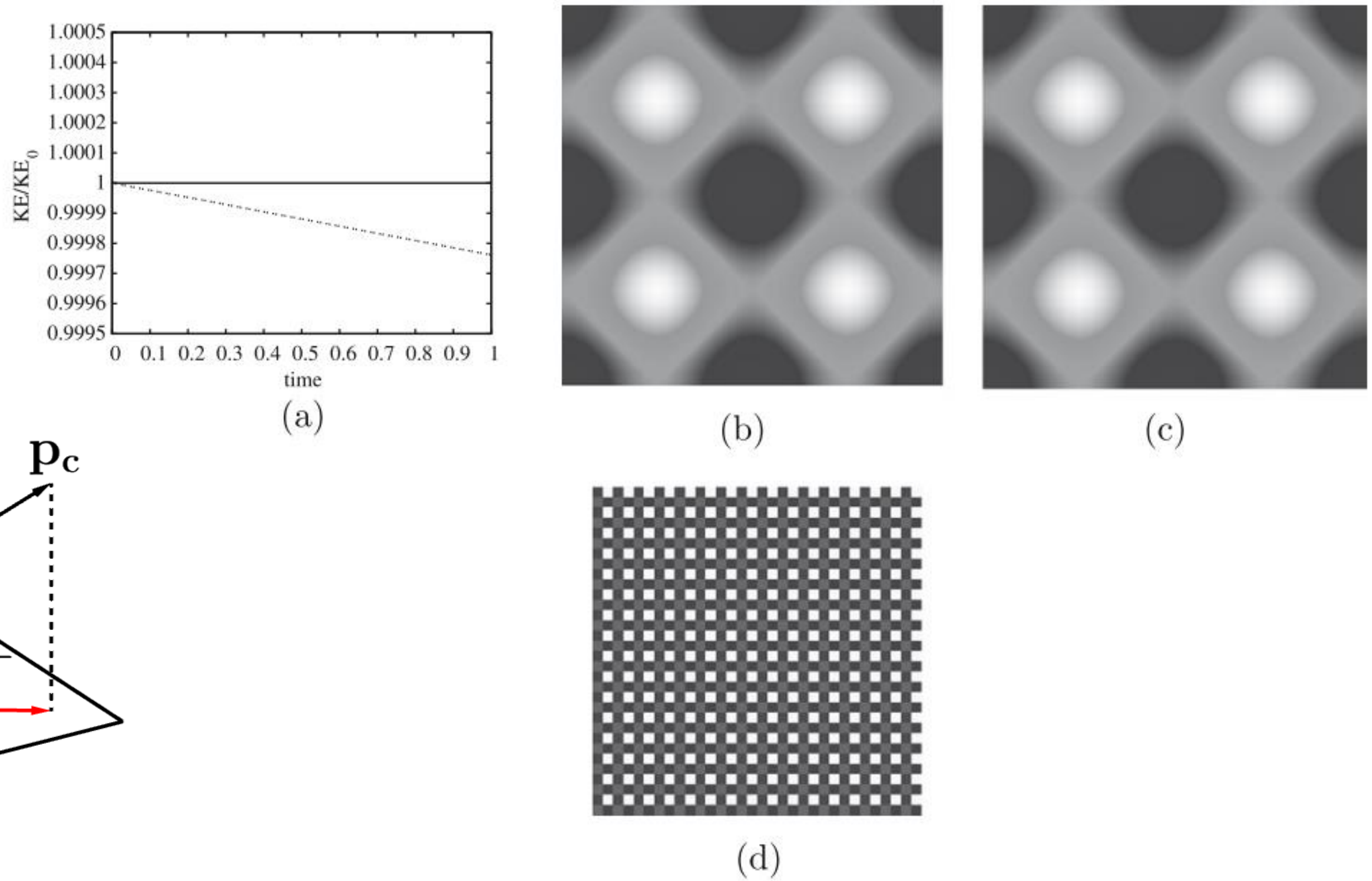


Fig. 1. Inviscid Taylor vortex. (a) Temporal evolution of kinetic energy using the present (solid) and Rhie–Chow (dotted) methods. (b–d) Pressure contours for the present (b), Rhie–Chow method (c) and without any correction (d).

Discussion

Discussion

How do spurious modes arise?

Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?

Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?

Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

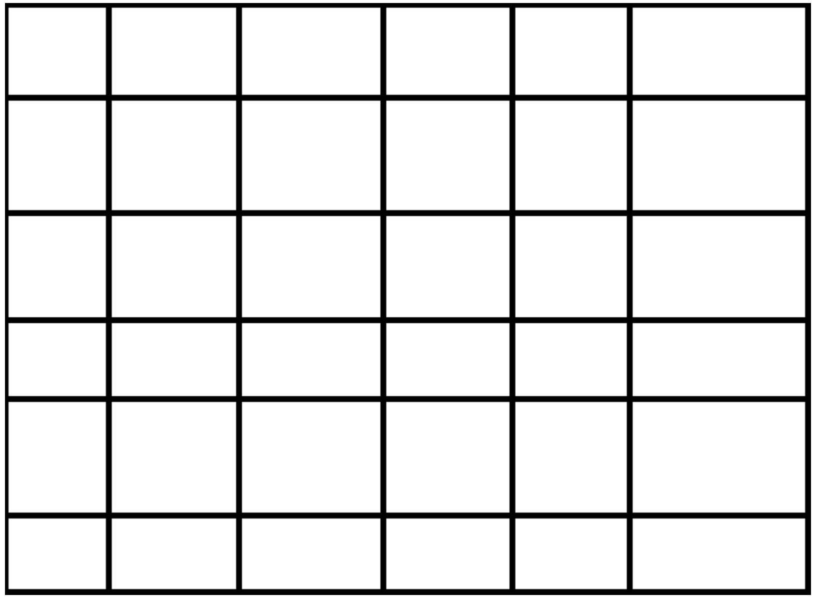
Simple mesh changes can reduce nullity to 1.

Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

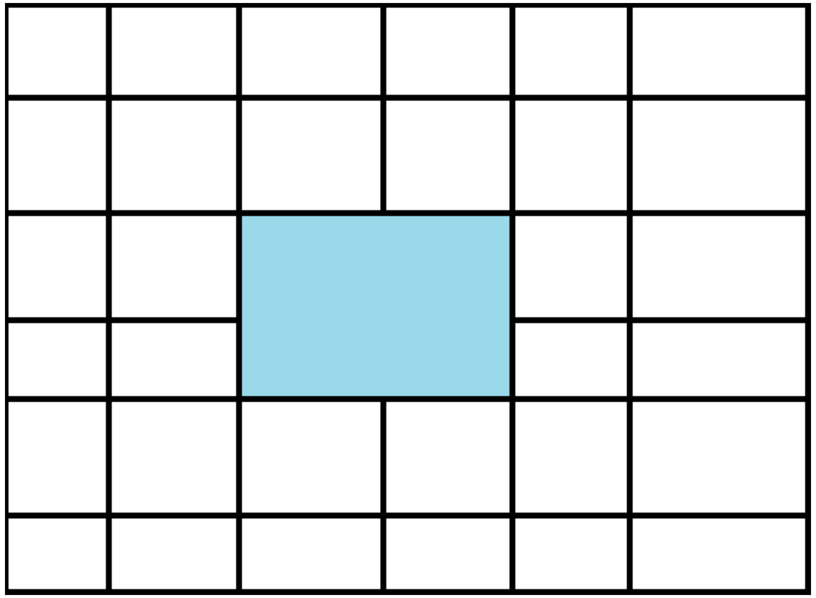


Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

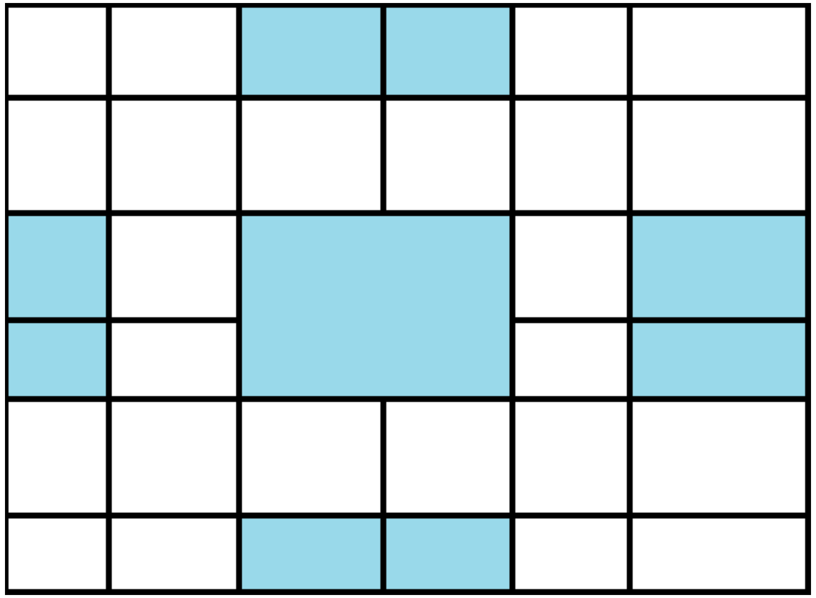


Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

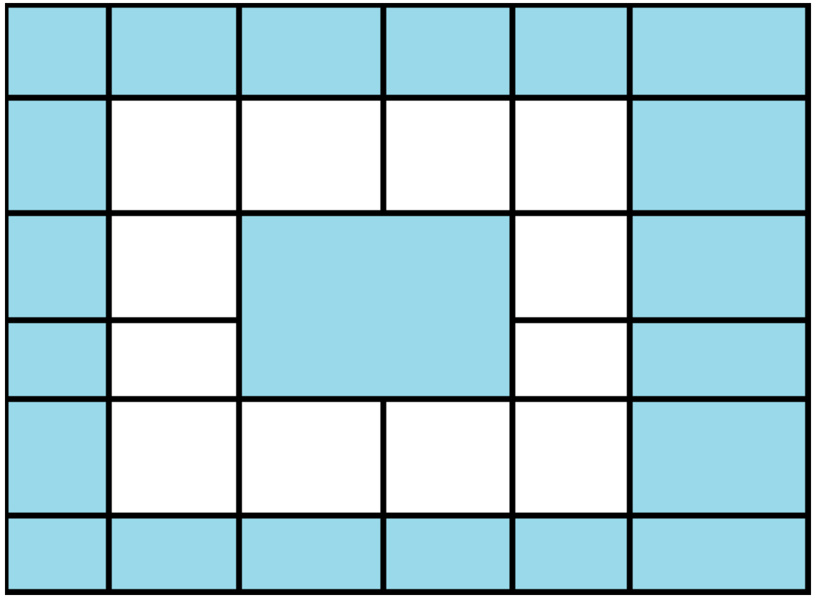


Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

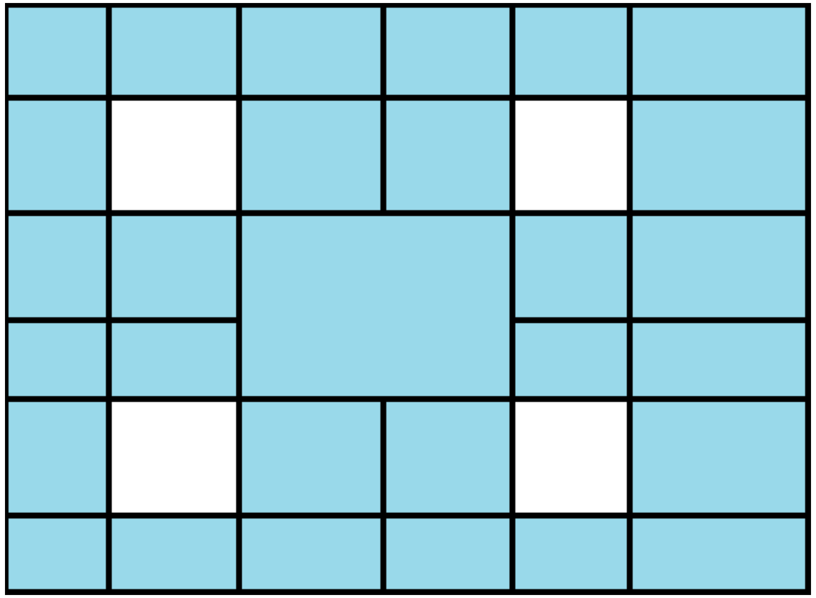


Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

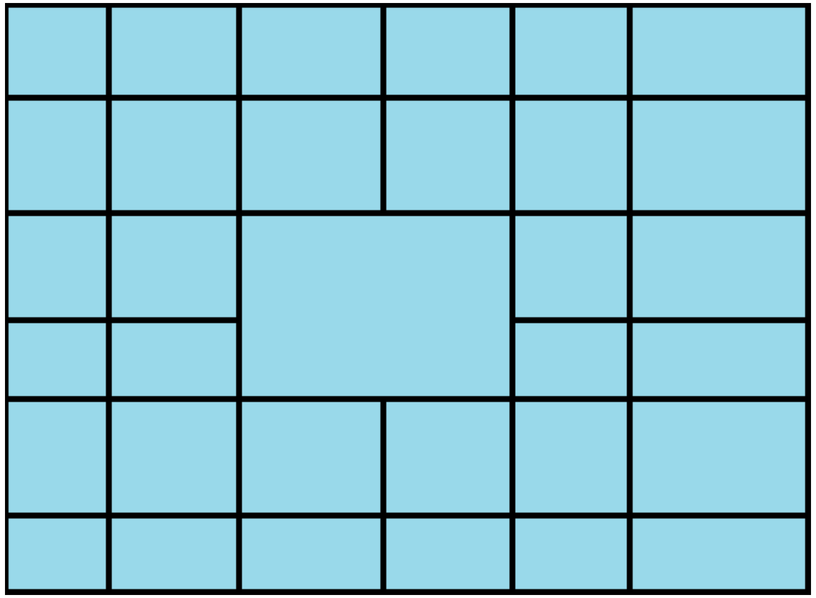


Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.



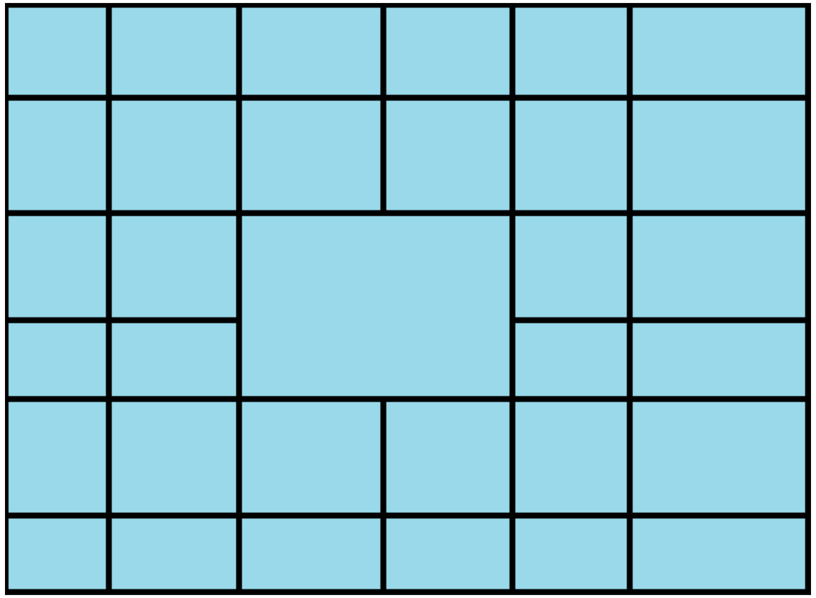
Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

- Will this eliminate checkerboarding?



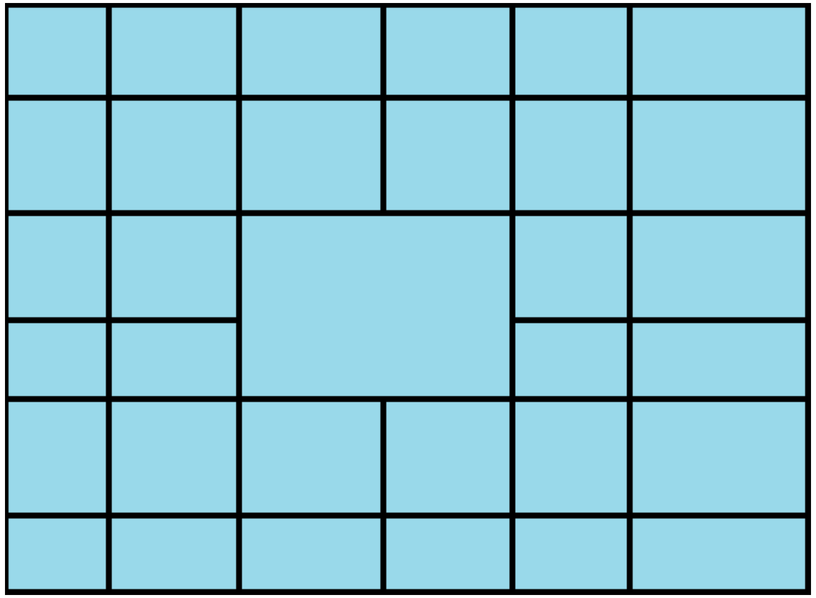
Discussion

How do spurious modes arise?

- Non-symmetry of Laplacian operator?
- Solver?
- Rounding errors?

Simple mesh changes can reduce nullity to 1.

- Will this eliminate checkerboarding?
- Are other (low EV) modes also problematic?



Thank you for attending!