Variable Coefficients:

- 1D diffusion example: *For* $p(x) > 0$,

$$-\frac{d}{dx}p(x)\frac{du}{dx} \;=\; f(x) \;+\; \text{BCs} \begin{cases} u(0)=0 \\ u'(1)=0 \end{cases} \tag{1}$$

- Variational formulation: *Find* $u \in X_0^N$ *such that for all* $v \in X_0^N$,

$$a(v,u) \;:=\; (v_x, p\,u_x) \;=\; a(v,\tilde{u}). \tag{2}$$

- Deriving the bilinear form:

$$-\int_{-1}^{1} v\,\frac{d}{dx}p(x)\frac{du}{dx}\,dx \;=\; \int_{-1}^{1} v\,f(x)\,dx \tag{3}$$

$$\int_{-1}^{1} \frac{dv}{dx}p(x)\frac{du}{dx}\,dx \;=\; \int_{-1}^{1} v\,f(x)\,dx \qquad \text{(boundary term vanishes because of BCs)} \tag{4}$$

$$(D\underline{v})^T\,PBD\,\underline{u} \;=\; \underline{v}^T B\underline{f} \tag{5}$$

$$\underline{v}^T D^T PBD\,\underline{u} \;=\; \underline{v}^T B\underline{f} \tag{6}$$

$$A\underline{u} \;=\; B\underline{f} \tag{7}$$

$$A \;:=\; D^T PBD. \tag{8}$$

- With (*highly accurate*) GLL quadrature, it is sufficient to evaluate $p(x)$ at the GLL nodal points, x_k, $k=0,\ldots,N$
- The matrices P and B are thus both diagonal, *with positive entries*:

$$P_{kk} \;=\; p(x_k), \quad B_{kk} \;=\; \rho_k, \quad k=0,\ldots,N. \tag{9}$$

- 2D example: *For* $p(x,y) > 0$,

$$-\nabla \cdot [p(x,y)\nabla u] \;=\; f(x,y), \tag{10}$$

plus appropriate BCs.

- Variational form, *Find* $u \in X_0^N$ *such that for all* $v \in X_0^N$,

$$a(v,u) \;:=\; (\nabla v, p\,\nabla u) \;=\; a(v,\tilde{u}). \tag{11}$$

- Energy inner product:

$$\begin{aligned}
a(v,u) &= \int_\Omega p\,\nabla v \cdot \nabla u\, dV & (12)\\
&= \int_\Omega p \left(\frac{\partial v}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial u}{\partial y} \right) dV & (13)\\
&= \int_\Omega \left(\frac{\partial v}{\partial x}\, p\, \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\, p\, \frac{\partial u}{\partial y} \right) dV & (14)\\
&= \int_\Omega \nabla v \cdot (p\,\nabla u)\; dV \;=\; (\nabla v, p\nabla u) & (15)\\
&\approx (\nabla v, p\nabla u)_N. & (16)
\end{aligned}$$

- In 1D, (16) would be exact for $p \in \mathbb{P}_1$.
- In 2D, this is not the case because of the integration in the undifferentiated direction.
- Convergence is still exponential if p is smooth.

- Derive bilinear form by starting with the *strong form*,

$$-\int\int v\,\nabla\cdot[p\nabla u]\;dx\,dy \;=\; \int\int v\,f(x,y)\,dx\,dy \tag{17}$$

$$-\int\int v\,\left[\frac{\partial}{\partial x}p\frac{\partial u}{\partial x} \;+\; \frac{\partial}{\partial y}p\frac{\partial u}{\partial y}\right]\;dx\,dy \;=\; \int\int v\,f(x,y)\,dx\,dy \tag{18}$$

- First term on left, integrated by parts

$$\mathcal{I}_x \;=\; \int_0^{L_y}\left[\int_0^{L_x}\frac{\partial v}{\partial x}p\frac{\partial u}{\partial x}\,dx \;-\; v\,p\;\frac{\partial u}{\partial x}\bigg|_{x=0}^{x=1}\right]\,dy \tag{19}$$

- Let Γ_i denote the ith face of Ω,
  - i=1: left face
  - i=2: right face
  - i=3: lower face
  - i=4: upper face
- Let $\hat{\mathbf{n}}_i$ be the associated outward-pointing unit-normal on each face:
  - i=1: $\hat{\mathbf{n}}_1 = (\hat{\mathbf{n}}_{x,1}, \hat{\mathbf{n}}_{y,1}) = (-1, 0)$
  - i=2: $\hat{\mathbf{n}}_2 = (\hat{\mathbf{n}}_{x,2}, \hat{\mathbf{n}}_{y,2}) = (1, 0)$
  - i=3: $\hat{\mathbf{n}}_3 = (\hat{\mathbf{n}}_{x,3}, \hat{\mathbf{n}}_{y,3}) = (0, -1)$
  - i=4: $\hat{\mathbf{n}}_4 = (\hat{\mathbf{n}}_{x,4}, \hat{\mathbf{n}}_{y,4}) = (0, 1)$
- We can rewrite $\mathcal{I}_x$ as:

$$\mathcal{I}_x \;=\; \int_0^{L_y}\int_0^{L_x}\frac{\partial v}{\partial x}p\frac{\partial u}{\partial x}\,dx\,dy \;-\; \sum_{i=1}^{2}\int_{\Gamma_i} v\,p\,\frac{\partial u}{\partial x}\,\hat{\mathbf{n}}_{x,i}\,dy. \tag{20}$$

- Similarly, $\mathcal{I}_y$ is

$$\mathcal{I}_y \;=\; \int_0^{L_y}\int_0^{L_x} \frac{\partial v}{\partial y} p \frac{\partial u}{\partial y}\, dx\, dy \;-\; \sum_{i=3}^{4} \int_{\Gamma_i} v\, p\, \frac{\partial u}{\partial y}\, \hat{\mathbf{n}}_{y,i}\, dy. \tag{21}$$

$$\tag{22}$$

- Combining $\mathcal{I}_x$ and $\mathcal{I}_y$,

$$\begin{aligned} \mathcal{I} &= \mathcal{I}_x + \mathcal{I}_y & (23)\\ &= \int_0^{L_y}\int_0^{L_x} \frac{\partial v}{\partial x} p \frac{\partial u}{\partial x} \;+\; \frac{\partial v}{\partial y} p \frac{\partial u}{\partial y}\, dx\, dy \;-\; \sum_{i=1}^{4} \int_{\Gamma_i} v\, p\, \left(\frac{\partial u}{\partial x}\, \hat{\mathbf{n}}_{x,i} \;+\; v\, p\, \frac{\partial u}{\partial y}\, \hat{\mathbf{n}}_{y,i} \right) dS. & (24)\\ &= \int_\Omega \nabla v \cdot (p \nabla u)\; dV \;-\; \int_{\partial\Omega} v\, p\, \nabla u \cdot \hat{\mathbf{n}}\, dS. & (25) \end{aligned}$$

- If $\displaystyle\int_{\partial\Omega} v\, p\, \nabla u \cdot \hat{\mathbf{n}}\Big|_{\partial\Omega} \;=\; 0$, then u satisfies

$$\int_{\Omega} \nabla v \cdot (p \nabla u)\; dV \;\;=\;\; \int_{\Omega} v\, f\, dV \quad \text{for all } v \in X_0^N. \tag{26}$$

$$\tag{27}$$

- **Note:** For any $u \in X^N$ we have the *gradient*

$$\nabla u \;\;=\;\; \begin{pmatrix} \frac{\partial u}{\partial x} \\ \\ \frac{\partial u}{\partial y} \end{pmatrix} \;=\; \begin{pmatrix} D_x \\ \\ D_y \end{pmatrix} \underline{u} \;=\; \begin{pmatrix} \hat{I} \otimes \frac{2}{L_x} \hat{D} \\ \\ \frac{2}{L_y} \hat{D} \otimes \hat{I} \end{pmatrix} \underline{u}. \tag{28}$$

$$\tag{29}$$

- Example:

$$x = \frac{L_x}{2}(z+1), \;\; y = \frac{L_y}{2}(z+1) \tag{30}$$

$$[X, Y] = ndgrid(x, y) \tag{31}$$

$$U = \sin \pi X \tag{32}$$

$$U_x = D_x U, \;\; U_y = D_y U \tag{33}$$

$$D_x = \hat{I} \otimes \frac{2}{L_x} \hat{D}, \;\; D_y = \frac{2}{L_y} \hat{D} \otimes \hat{I} \tag{34}$$

- Plot U_x, U_y.

- Returning to our weighted residual form, insertion of the bases and using quadrature yields

$$\mathcal{I} = \int_0^{L_y} \int_0^{L_x} \frac{\partial v}{\partial x} p \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} p \frac{\partial u}{\partial y} \, dx \, dy \tag{35}$$

$$= (D_x \underline{v})^T \, P \, \bar{B} \, D_x \underline{u} + (D_y \underline{v})^T \, P \, \bar{B} \, D_y \underline{u} \tag{36}$$

$$= \underline{v}^T \, D_x^T \, P \, \bar{B} \, D_x \underline{u} + \underline{v}^T \, D_y^T \, P \, \bar{B} \, D_y \underline{u}. \tag{37}$$

- Note that if the range of indices on $\underline{v}$ and $\underline{u}$ includes *all nodes*, including the boundaries, then we would have the form of our *Neumann operator*,

$$\bar{A} = D_x^T \, P \, \bar{B} \, D_x + D_y^T \, P \, \bar{B} \, D_y. \tag{38}$$

- To construct the A-matrix that governs the interior degrees of freedom, apply the restriction operator R to $\bar{A}$:

$$A = R \, \bar{A} \, R^T. \tag{39}$$

- In the case of a rectangular domain with *uniform boundary conditions along a given edge*, R restricts index sets in both the x and y directions and can be written in tensor-product form:

$$R = R_y \otimes R_x. \tag{40}$$

  As always, the R_x matrix is on the right, because it is acting on the fastest changing index in the $\underline{u}^T = [u_{00} \, u_{10} \, \ldots \, u_{NN}]^T$ lexicographical ordering of the unknowns.

Gradient-Based Form

- It is sometimes convenient to write $\bar{A}$ in a higher-level form.
- We can also write

$$\bar{A} \;=\; \begin{bmatrix} D_x \\ D_y \end{bmatrix}^T \begin{bmatrix} P\bar{B} & 0 \\ 0 & P\bar{B} \end{bmatrix} \begin{bmatrix} D_x \\ D_y \end{bmatrix} = \mathbf{D}^T \mathbf{P}\,\mathbf{B}\,\mathbf{D}. \tag{41}$$

- Here, we use the bold font to indicate that the matrices are working on or producing **vector fields**, i.e., inputs and/or outputs that have two components in 2D, three components in 3D, etc.
- Obviously, the $\mathbf{D}$ matrix is the discrete gradient operator introduced above.

Quick Summary

- Variable coefficients are central to PDEs.
- For Navier-Stokes, the advecting velocity, $\mathbf{u}(\mathbf{x})$ can be viewed as a variable coefficient.
- For deformed domains, we'll also find that the Jacobian and the metric terms associated with the map, $\mathbf{x}(\mathbf{r})$: $\hat{\Omega} \rightarrow \Omega$, lead to a variable coefficient formulation in $\hat{\Omega}$.
- Here, we have considered only $p(\mathbf{x}) > 0$, which is necessary to ensure that our energy inner-product remains positive definite.
- When discretized with a variational formulation, this problem leads to a new discrete inner product for the gradients with positive quadrature weights, $P_{kk}B_{kk}$, where B_{kk} are the diagonal entries of the standard 2D mass matrix, B.

- On a rectangle, these would be stored in an array on the full domain as

$$(PB)_{ij} \;\; = \;\; P_{ij} \, \frac{L_x L_y}{4} \, \rho_i \, \rho_j, \qquad i,j \in \{0,\ldots,N_x\} \times \{0,\ldots,N_y\}. \tag{42}$$

- Because P and B are diagonal matrices, one can effect the matrix-vector product, $\underline{w} = PB\,\underline{u}$ as a simple pointwise multiply for each point in the domain,

$$w_{ij} \;\; = \;\; (P_{ij} B_{ij}) \, u_{ij}. \tag{43}$$

  (No sum on the indices—just pointwise multiply.)

- In 3D we would have the analogous form,

$$w_{ijk} \;\; = \;\; (P_{ijk} B_{ijk}) \, u_{ijk}. \tag{44}$$

Advection-Diffusion in 2D/3D

- Here we are concerned with what is effectively the bulk of the Navier-Stokes equations, namely, *unsteady advection-diffusion*,

$$\frac{\partial u}{\partial t} \; + \; \mathbf{c} \cdot \nabla \, u \;\; = \;\; \nu \nabla^2 u \; + \; q(\mathbf{x}), \tag{45}$$

  which is subject to our usual set of BCs and ICs.

- Here, $\mathbf{c}(\mathbf{x})$ is the **advecting vector field**,

$$2D: \;\; \mathbf{c} \;\; := \;\; \begin{pmatrix} c_x \\ \\ c_y \end{pmatrix}, \qquad\qquad 3D: \;\; \mathbf{c} \;\; := \;\; \begin{pmatrix} c_x \\ \\ c_y \\ \\ c_z \end{pmatrix}. \tag{46}$$

- We will normally assume that $\mathbf{c}$ is *divergence-free*, which is characteristic of *incompressible flows.*

- For example, in 2D we'd have:

$$\nabla \cdot \mathbf{c} \;\; = \;\; \frac{\partial c_x}{\partial x} \; + \; \frac{\partial c_y}{\partial y} \; \equiv \; 0. \tag{47}$$

Bilinear form for Advection

- Consider, with $\nabla \cdot \mathbf{c} = 0$, the bilinear form,

$$c(v,u) \;:=\; \int_\Omega v\,\mathbf{c}\cdot\nabla u\,dV. \tag{48}$$

- Note that, for certain boundary conditions to be described, this form is *skew symmetric*:

$$c(v,u) \;:=\; -c(u,v). \tag{49}$$

- To see this, integrate (48) by parts,

$$c(v,u) \;=\; \int_\Omega v\,\mathbf{c}\cdot\nabla u\,dV \tag{50}$$

$$= \;-\int_\Omega u\,\nabla\cdot(\mathbf{c}\,v)\;dV \;+\; \underbrace{\int_{\partial\Omega} v\,u\,\mathbf{c}\cdot\hat{\mathbf{n}}\,dS}_{=:\;b} \tag{51}$$

$$= \;-\int_\Omega u\,[\,v\,\underbrace{\nabla\cdot\mathbf{c}}_{\equiv\;0} + \mathbf{c}\cdot\nabla v\,]\,dV \;+\; b \tag{52}$$

$$= \;-c(v,u) \;+\; b \tag{53}$$

Surface Boundary Term

- Note that contributions to the boundary term,

$$b \;:=\; \int_{\partial\Omega} v\, u\, \mathbf{c} \cdot \hat{\mathbf{n}}\, dS \tag{54}$$

  will be zero on segments of $\partial\Omega$ under the following conditions,

  - on $\partial\Omega_D$ because $v(\mathbf{x}) \equiv 0$ on $\partial\Omega_D$,
  - on $\partial\Omega_N$ if $\mathbf{c} \cdot \hat{\mathbf{n}} = 0$ on $\partial\Omega_N$,
  - on $\partial\Omega_P$ if $\mathbf{c}$, v, $u \in X_p^N$.

- The last condition states that the velocity must be periodic if v and u are.
- We will refer to the domain as *closed* if $b = 0$.

- If the domain is closed, then $c()$ is skew symmetric,

$$c(v,u) \;\;=\;\; -c(u,v), \tag{55}$$

otherwise,

$$c(v,u) \;\;=\;\; -c(u,v) \;+\; b. \tag{56}$$

- In the context of advection-diffusion, we refer to $\partial\Omega_N$ as an *outflow boundary.*
- The outward flux of material on $\partial\Omega_N$ is

$$\int_{\partial\Omega_N} u\,\mathbf{c}\cdot\hat{\mathbf{n}}\,dS\,. \tag{57}$$

- As we will see, stability requires $\mathbf{c}(\mathbf{x})\cdot\hat{\mathbf{n}}(\mathbf{x}) \;\geq\; 0$ on $\partial\Omega_N$ in the advection-dominated case.
  - If $\mathbf{c}(\mathbf{x})\cdot\hat{\mathbf{n}}(\mathbf{x}) \;<\; 0$, you have "inflow coming in the outflow boundary."
  - The solution will blow up in just a few timesteps in this case.

Discrete Form of Advection-Diffusion

- Consider the weighted-residual approach to solving the advection-diffusion equation in the closed domain case,

$$\frac{\partial \tilde{u}}{\partial t} + \mathbf{c} \cdot \nabla \tilde{u} = \nu \nabla^2 \tilde{u} \; f \quad (+ \text{BC} / \text{IC}). \tag{58}$$

- Standard variational approach, *Find* $u \in X_0^N$ *such that,* $\forall \, v \in X_0^N$,

$$(v, u_t) + c(v,u) + \nu \, a(v,u) = (v, f) \tag{59}$$

- In matrix form, we have

$$B \frac{d\underline{u}}{dt} + C\underline{u} + \nu \, A\underline{u} = R\bar{B}\bar{\underline{f}}. \tag{60}$$

- With $\phi_i(\mathbf{x})$ representing our standard Lagrange cardinal function in $\mathbb{R}^d$, the advection matrix is

$$C_{ij} := c(\phi_i, \phi_j) = \int_\Omega \phi_i \, \mathbf{c} \cdot \nabla \, \phi_j \, dV. \tag{61}$$

- Note that, for *closed domains*, we have

$$C_{ij} := c(\phi_i, \phi_j) = -c(\phi_j, \phi_i) = -C_{ji}. \tag{62}$$

- Thus, C is *skew-symmetric:* $C = -C^T$.
- All skew-symmetric matrices have imaginary eigenvalues.
- If the mass matrix, B, is SPD, then it's also true that the eigenvalues of the advection evolution matrix, $L = -B^{-1}C$, are imaginary.
- As we saw last time, the stability region for BDF3/EXT3 (and RK3 and AB3) encompasses part of the imaginary axis and, for sufficiently small Δt, the evolution of the fully-discrete system will be *stable*.
- We'll look at fast (and stable!) evaluation of matrix-vector products $\underline{w} = C\underline{u}$ shortly.

More on Stability of Advection-Diffusion

- Recall that $c(v,u) \;=\; -c(u,v) \;+\; b$.

- In the closed domain case, $b = 0$, and we therefore have

$$c(u,u) = -c(u,u) \quad\Longrightarrow\quad 2\,c(u,u) = 0 \quad\Longrightarrow\quad c(u,u) = 0. \tag{63}$$

- Consider the evolution of the energy $E \;:=\; \int \tilde{u}^2\,dV$ for the homogeneous advection-diffusion problem ($f = 0$, $u \in X_0^N$) for the closed domain case.

- For a nontrivial IC, $u(\mathbf{x}, t=0) = u^0$,

$$(v, u_t) \;=\; -c(v,u) \;-\; \nu\,a(v,u). \tag{64}$$

- Set $v = u$. On the left we have,

$$(u, u_t) \;=\; \int_\Omega u\frac{\partial u}{\partial t}\,dV \;=\; \int_\Omega \frac{1}{2}\frac{\partial u^2}{\partial t}\,dV \;=\; \frac{1}{2}\frac{d}{dt}\int_\Omega u^2\,dV \;=\; \frac{1}{2}\frac{d}{dt}(u,u) \;=\; \frac{1}{2}\frac{dE}{dt}. \tag{65}$$

- Equate this to the terms on the right,

$$\frac{1}{2}\frac{dE}{dt} \;=\; \underbrace{-\nu\,a(u,u)}_{<\,0} \;-\; \underbrace{c(u,u)}_{=\,0} \;<\; 0. \tag{66}$$

- As a result, *for closed domains,* this formulation is *stable.*

Stability in the Open-Domain Case

- With $c(v,u) = -c(u,v) + b$, the advective contribution on the rhs of the energy evolution becomes

$$-c(u,u) = -\frac{1}{2}\int_{\partial\Omega} u^2\,\mathbf{c}\cdot\hat{\mathbf{n}}\,dS \qquad (67)$$

  - This term will be ≤ 0 as long as $\mathbf{c}\cdot\hat{\mathbf{n}} \geq 0$.
  - That is, the outward normal component of the velocity $\mathbf{c}$ is either positive or zero.

- If $\mathbf{c}\cdot\hat{\mathbf{n}} < 0$, advection will contribute to a growth in energy for any nontrivial value of u on $\partial\Omega_N$,

$$\frac{1}{2}\frac{dE}{dt} = -\nu\,a(u,u) - \frac{1}{2}\int_{\partial\Omega} u^2\,\mathbf{c}\cdot\hat{\mathbf{n}}\,dS \qquad (68)$$

- If the diffusion is "large" (i.e., the Peclet number is small), the system might still be stable because the $-\nu a(u,u)$ term may dominate the boundary term.
- For advection-dominated problems, however, positive inlet velocities can be a disaster.
- Note that this is an *advection-based* instability, not a Navier-Stokes instability, *per se*, save that the NS equations do typically determine the velocity field.
- **Nek5000 Example:** Taylor-Green vortex.

Stability of Discrete System

- We argued earlier that $C = -C^T$, which implied the necessary stability condition if we use a common 3rd-order timestepper.

- That argument was based on the trivial observation that, for closed domains,

$$C_{ij} \;=\; c(\phi_i, \phi_j) \;=\; -c(\phi_j, \phi_i). \tag{69}$$

- This equality relies *crucially* on integration by parts.

- What happens if we use numerical quadrature?

- If the quadrature is exact then there is no issue.

- Typically, $\mathbf{c} \in \mathbb{P}_N$, which means that the integrand,

$$v\,\mathbf{c}\cdot\nabla u \;\in\; \mathbb{P}_N. \tag{70}$$

- Our $(N+1)$ point GLL quadrature is exact only up to $\mathbb{P}_{2N-1}$.

- Need to *over-integrate* or *dealias*.

- Define $(M+1)$ GL quadrature points, η_k, and weights, ω_k, and

$$\begin{aligned}
\mathbf{C}^M &= J\mathbf{C} && \text{diagonal matrix on } (\eta_k, \eta_l) && (71)\\
\mathbf{D}^M &= J\mathbf{D} && \text{diagonal matrix on } (\eta_k, \eta_l) && (72)\\
\underline{u}^M &= J\underline{u} && u(\mathbf{x}) \text{ interpolated to GL points} && (73)\\
\underline{v}^M &= J\underline{v} && u(\mathbf{x}) \text{ interpolated to GL points.} && (74)
\end{aligned}$$

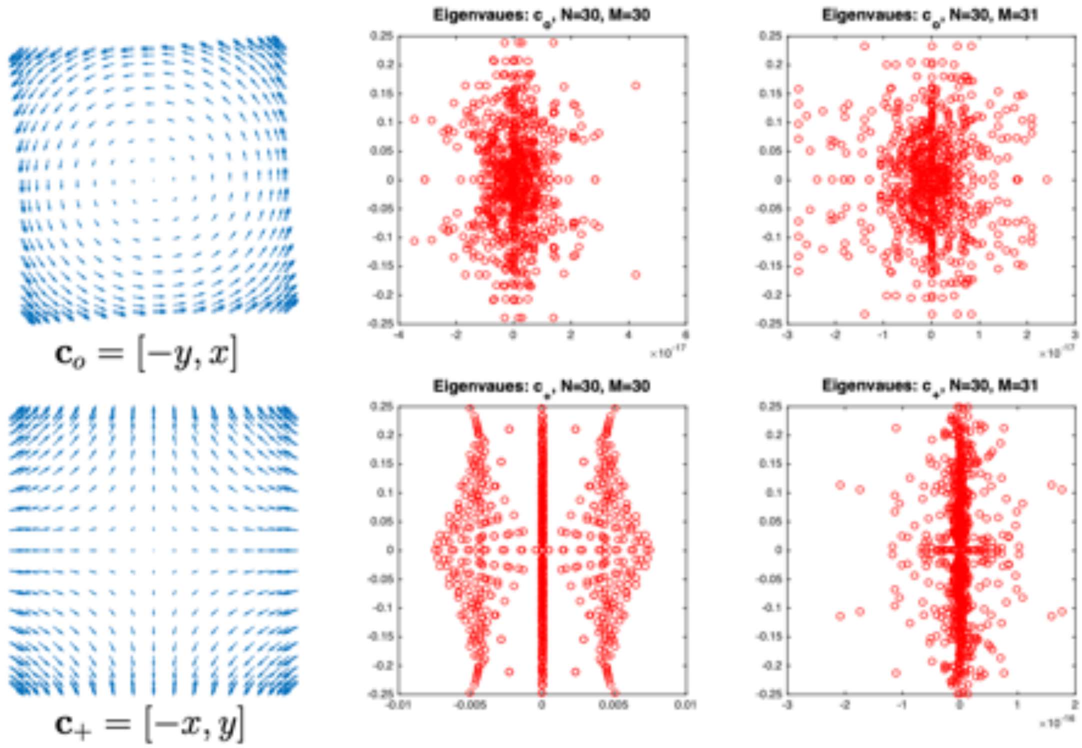
- Then,

$$C \;:=\; J^T(\mathbf{C}^M)^T\,\mathbf{B}^M\,\mathbf{D}^M\,J. \tag{75}$$

  will yield exact integration and therefore $C = -C^T$.

Dealiasing Example[a]

Here, we look at the eigenvalues for two velocity fields $\mathbf{c}_o = [-y, x]^T$ and $\mathbf{c}_+ = [-x, y]^T$, which satisfy $\mathbf{c} \in \mathbb{P}_1$. Both cases will yield skew-symmetric advection matrices $C = -C^T$ if we take $M = N + 1$ using either GL or GLL quadrature. It is also straightforward to show that C for the case of plane-rotation, $\mathbf{c}_o$, will be skew symmetric with $M = N$ and GLL quadrature. For $\mathbf{c}_+$, which corresponds to a straining field, C will not be skew symmetric. This case is unstable.



[a]This topic was explored in [Malm *et al.*,2012].

Fully-Discrete System for Unsteady Advection-Diffusion

- Starting with the original PDE, we have

$$\frac{\partial u}{\partial t} + \mathbf{c} \cdot \nabla u \;=\; \nu \nabla^2 u \;+\; f, \tag{76}$$

  with a given initial condition $u^0(\mathbf{x})$ and BCs to be discussed below.

- Applying BDFk/EXTk semi-implicit timestepping at time t^n leads to

$$\beta_0 u^n \;-\; \Delta t\, \nu\, \nabla^2 u^n \;=\; -\sum_{j=1}^{k} \left[\beta_j\, u^{n-j} \;+\; \Delta t\, \alpha_j\, (\mathbf{c} \cdot \nabla u \;-\; f)^{n-j} \right] \;=:\; h^n(\mathbf{x}). \tag{77}$$

- Note that the advection+forcing term is viewed as a vector (i.e., a function), which is evaluated once and stored at time t^{n-j} so that it does not need to be recomputed.

- This approach also accommodates a time-dependent advecting field, $\mathbf{c}^n$.

- Notice also that the *rhs*, h^n, is likely discontinuous—there is no *a priori* reason to expect or to demand that it be continuous or that it satisfy any particular set of boundary conditions.

- Boundary conditions will be imposed when solving the elliptic problem for u^n, represented by the operator on the left.

- Spatial discretization of (77) follows our standard variational approach,

  Find $u^n \in X_0^N$ *such that,* $\forall\, v \in X_0^N$,

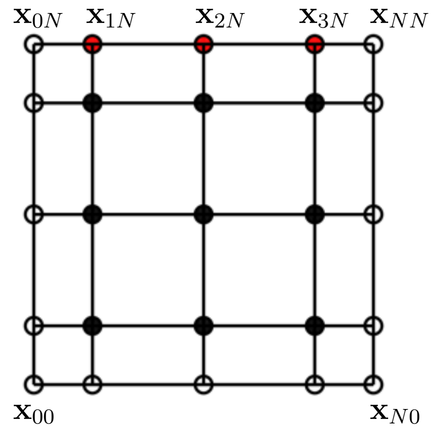
$$\beta_0(v, u^n) \;+\; \nu\, \Delta t\, a(v, u^n) \;=\; (v, h^n). \tag{78}$$

- In matrix form, we have the following Helmholtz problem,

$$H\underline{u}^n \;:=\; (\,\beta_0 B \;+\; \nu\, \Delta t\, A)\, \underline{u}^n \;\;=\;\; R\, \bar{B}\, \underline{\bar{h}}^n. \tag{79}$$

- If $\nu = \nu(\mathbf{x})$, then we must embed it into the bilinear form $a(.,.)$ as we did earlier with $p(\mathbf{x})$.
- Notice that the rhs function, $h(\mathbf{x})$ is evaluated at all points, including the Dirichlet conditions.
- As always, the restriction matrix applied to $\bar{B}\underline{\bar{h}}^n$ is coming from the test function, $v \in X_0^N$.

Inhomogeneous Boundary Conditions in 2D/3D



- Consider initially 2D with

$$
\begin{aligned}
\textit{(i)} \quad \tilde{u}|_{\partial\Omega_D} &= \tilde{u}_b \quad \text{on } \partial\Omega_D & (80)\\
\textit{(ii)} \quad \nabla u \cdot \hat{\mathbf{n}} &= g(\mathbf{x}) \quad \text{on } \partial\Omega_N. & (81)
\end{aligned}
$$

- For *(i)*, write $u = u_0 + u_b$, where $u_b \in X_b^N$.

  $\Longrightarrow u_b$ is continuous.

  $u_b = \tilde{u}_b$ on $\partial\Omega_D$.

- In the figure above, let $\partial\Omega_D$ be the left, right, and lower sides, depicted by the open circles on the GLL points, and let $\partial\Omega_N$ be the upper surface. Where $\partial\Omega_N \cap \partial\Omega_D$ we typically let the Dirichlet condition dominate, so all corner vertices in this example are Dirichlet.

- Here, the 3 red dots correspond to degrees-of-freedom on $\partial\Omega_N$ where we impose the inhomogeneous Neumann (i.e., *flux*) condition, (81).

Inhomogeneous Dirichlet Conditions

- *Any function* in the interpolation space X^N that satisfies $u_b(\mathbf{x}_j)|_{\partial\Omega_D}\ \tilde{u}_b(\mathbf{x}_j)|_{\partial\Omega_D}$ is assumed to be a suitable approximation to the boundary data.

- We typically take $u_b(\mathbf{x})$ to be the interpolant in X^N that has the correct values on $\partial\Omega_D$ and is zero elsewhere.

- In the context of timesteppers at time level t^n, we instead use

$$u_b = u^{n-1} \quad \text{in } \Omega\backslash\partial\Omega_D \tag{82}$$

$$u_b = \tilde{u}_b^n \quad \text{on } \partial\Omega_D. \tag{83}$$

- This approach is particularly advantageous when using iterative solvers as it provides an initial guess that is only $O(\Delta t)$ from the desired solution, u^n.

- In particular, if u is approaching a steady state, then $u_0^n \longrightarrow 0$, and very few, if any, iterations will be required to compute the solution.

- With the substitution

$$u \;=\; u_0 + u_b, \tag{84}$$

the weighted residual statement for (78) becomes

Find $u_0 \in X_0^N$ *such that* $\forall\, v \,\in\, X_0^N$,

$$\beta_0(v, u_0) \;+\; \nu\,\Delta t\, a(v, u_0) \;=\; (v, h^n) \;-\; (\beta_0(v, u_0) \;+\; \nu\,\Delta t\, a(v, u_0))\,. \tag{85}$$

- In matrix form,

$$H \underline{u}_0 \;:=\; (\,\beta_0 B \;+\; \nu\,\Delta t\, A)\, \underline{u}_0 \;\;=\;\; R\,\left(\bar{B}\,\underline{\bar{h}}^n \;-\; \bar{H}\,\underline{\bar{u}}_b\right), \tag{86}$$

where

$$\bar{H} \;:=\; \beta_0 \bar{B} \;+\; \nu\,\Delta t\, \bar{A}, \tag{87}$$

is the unrestricted Helmholtz matrix.

- Once $\underline{u}_0$ is computed, we recover the full solution (at all points),

$$\underline{\bar{u}}^n \;\;=\;\; R^T \underline{u}_0 \;+\; \underline{\bar{u}}_b. \tag{88}$$

Inhomogeneous Neumann Conditions

- Recall the basic sequence for the variational approach to discretizing the Poisson equation with diffusivity ν,

  Find $u \in X_b^N$ *such that* $\forall\, v \,\in\, X_0^N$,

$$
\begin{aligned}
a(v,u) &= a(v,\tilde{u}) && (89)\\
a(v,u-u_b) &= a(v,\tilde{u}-u_b) && (90)\\
a(v,u_0) &= a(v,\tilde{u}) \,-\, a(v,u_b) && (91)\\
&= (v,f) \,+\, \int_{\partial\Omega} v\,\underbrace{\nabla\tilde{u}\cdot\hat{\mathbf{n}}}_{g} - (v,u_b). && (92)
\end{aligned}
$$

- In matrix form we have,

$$
A\underline{u}_0 \;=\; R\left(\bar{B}\,\underline{\bar{f}} \,-\, \bar{A}\underline{\bar{u}}_b \,+\, R_N^T\,(Area)\,\underline{g}\right). \qquad (93)
$$

- Here, we have introduced some new discrete operators,

  - R_N^T – prolongate from $\partial\Omega_N$ to $\bar{\Omega}$
  - $Area_f$ – diagonal("surface area per node" on face f).
  - $\mathcal{J}_f$ – "surface Jacobian".

- Here, R_N^T is just an abstract concept that allows us to write (93) in compact form (but it *is* convenient in matlab).

- The "$Area_f$" array is defined at every point $\mathbf{x}_i$ of face f as,

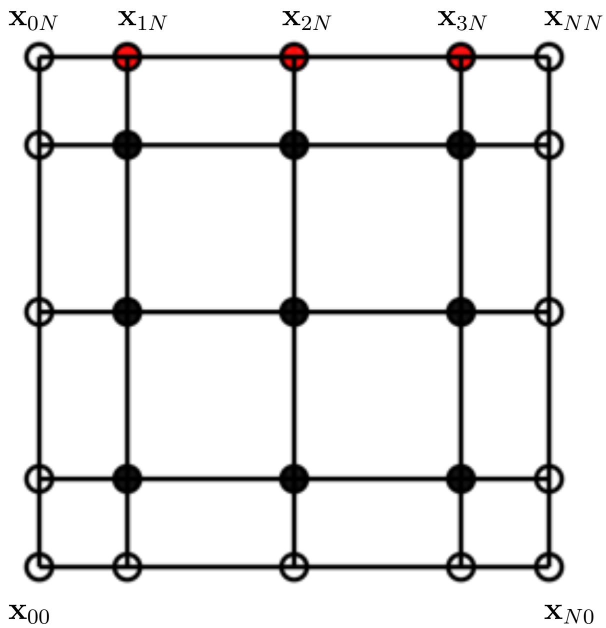
$$
Area_{f,i} \;=\; \mathcal{J}_{s,i}\,\rho_i, \qquad (94)
$$

  where ρ_i is the local surface quadrature weight (in $\hat{\Omega}$) and $\mathcal{J}_{f,i}$ is the local surface Jacobian on face f.

- $\mathcal{J}_{f,i}$ corresponds to the amount of area covered on $\partial\Omega$ when one traverses a corresponding patch of $\partial\hat{\Omega}$.

- Note that $\sum_i Area_{f,i}$ is equal to the area (in physical space) of face f.

- In the present case, $\mathcal{J}_f$ on $\partial\Omega_N$ would be $\frac{L_x}{2}$ for a domain $\Omega = [0 : L_x] \times [0 : L_y]$.



- In more complex domains we have direct formulas for computing this Jacobian.

- We'll start with a general deformed domain in 2D, then consider a 2D manifold in $\mathbb{R}^3$.

(Sidebar on surface Jacobians, from class notes...)

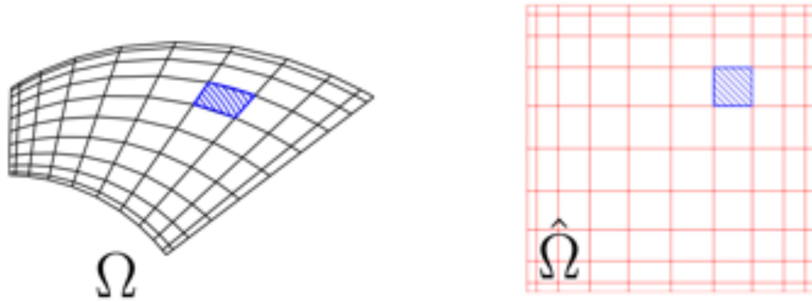


Figure 14: Example of mapped domain (left) from reference element, $\hat{\Omega} = [-1,1]^2$ (right).

As the cells become infitesimally small, the blue area is given by the *cross product*,

$$\begin{aligned} dA &= \frac{\partial \mathbf{x}}{\partial r} dr \times \frac{\partial \mathbf{x}}{\partial s} ds = \mathcal{J}\, dr\, ds \qquad (598) \\ &= \left[\frac{\partial x}{\partial r}\frac{\partial y}{\partial s} - \frac{\partial x}{\partial s}\frac{\partial y}{\partial r} \right] dr\, ds \\ &= \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} \end{vmatrix} dr\, ds. \end{aligned}$$

That is, $\mathcal{J}_{pq}$ is the determinant of the 2×2 metric tensor at each quadrature point (r_p, s_q). If we accept some (exponentially small) quadrature error, we can have a diagonal mass matrix by evaluating $\mathcal{J}$ at the GLL points,

$$B = \mathrm{diag}(\mathcal{J}_{pq}\rho_p\rho_q) = \mathcal{J}(\hat{B} \otimes \hat{B}), \qquad (599)$$

Here, we treat $\mathcal{J}$ as a diagonal matrix, following our standard practice with GLL quadrature. For M-point quadrature rules, (599) will be replaced by a full mass matrix,

$$\tilde{B} = J^T \left[\mathrm{diag}(\mathcal{J}^M_{pq}\rho^M_p\rho^M_q)\right] J = (\hat{J}^T \otimes \hat{J}^T)\mathcal{J}^M(B^M \otimes B^M)(\hat{J} \otimes \hat{J}). \qquad (600)$$

Here, B^M=diag(ρ^M_j) and $\hat{J}$ is the interpolation matrix from the N+1 GLL points to the M quadrature points, and $\mathcal{J}^M$ is the Jacobian interpolated to the $M \times M$ arrary of quadrature points.

(From DFM'02, Chapter 4...)

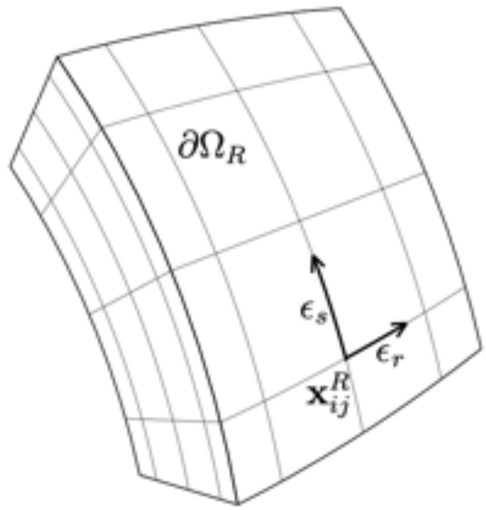


Figure 4.4.3: Description of surface geometry in $I\!R^3$.

The surface Jacobian is determined by noting that an infinitesimal displacement dr on $\partial\widehat{\Omega}_R$ gives rise to a corresponding displacement,

$$\epsilon_r := \frac{\partial \mathbf{x}}{\partial r} dr = \left(\frac{\partial x}{\partial r}, \frac{\partial y}{\partial r}, \frac{\partial z}{\partial r} \right)^T dr \,,$$

on $\partial\Omega_R$, while an orthogonal displacement, ds, yields $\epsilon_s = \frac{\partial \mathbf{x}}{\partial s} ds$. It is clear from Fig. 4.4.3 that an infinitesimal area on the physical surface is given by

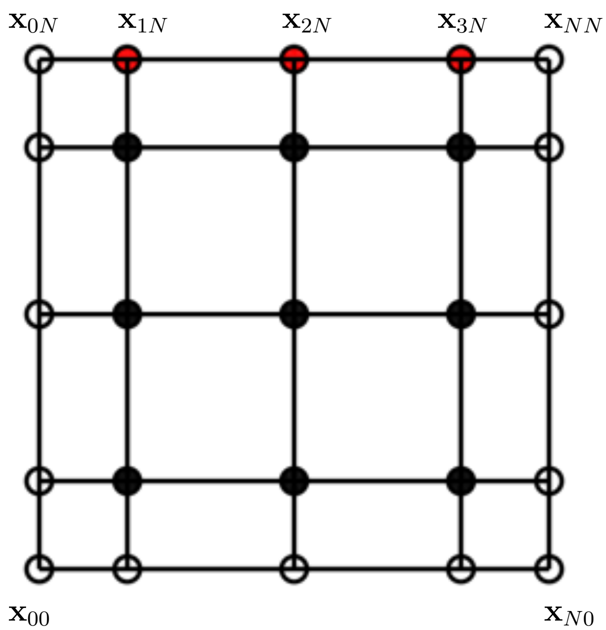
$$dA = \|\epsilon_r \times \epsilon_s\| = \left\| \frac{\partial \mathbf{x}}{\partial r} \times \frac{\partial \mathbf{x}}{\partial s} \right\| dr\, ds \,,$$

where $\|.\|$ denotes the standard Euclidean norm. Thus, the surface Jacobian at a point $\mathbf{x}_{ij}^R$ is

$$\tilde{J}_{ij}^R := \left\| \left. \frac{\partial \mathbf{x}}{\partial r} \right|_{ij}^R \times \left. \frac{\partial \mathbf{x}}{\partial s} \right|_{ij}^R \right\| .$$

The associated unit normal in physical space is

$$\hat{\mathbf{n}}_{ij} := \frac{1}{\tilde{J}_{ij}^R} \left(\left. \frac{\partial \mathbf{x}}{\partial r} \right|_{ij}^R \times \left. \frac{\partial \mathbf{x}}{\partial s} \right|_{ij}^R \right) .$$



- Returning to our inhomogeneous flux example, consider the following vectors,

$$\underline{\bar{u}}_b = \begin{pmatrix} u_{b,00} \\ \vdots \\ u_{b,N0} \\ \hline \vdots \\ \hline u_{b,0N} \\ \vdots \\ u_{b,NN} \end{pmatrix}, \qquad \underline{u}_0 = \begin{pmatrix} u_{0,11} \\ \vdots \\ u_{0,3N} \\ \hline \vdots \\ \hline u_{0,1N} \\ \vdots \\ u_{0,3N} \end{pmatrix}, \qquad \underline{g} = \begin{pmatrix} g_{1N} \\ g_{2N} \\ g_{3N} \end{pmatrix}. \tag{95}$$

- We see that $\underline{g}$ comprises values of $g(\mathbf{x})$ at the *red points*, which are the only points on $\partial\Omega$ where any of the test functions, $v \in X_0^N$, are nonzero.

- With inhomogeneous Dirichlet and Neumann conditions, the system to be solved will be,

$$A\underline{u}_0 \;=\; R\bar{B}\,\underline{\bar{f}} \;-\; R\bar{A}\,\underline{\bar{u}}_b \;+\; R\left(\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline \\ \vdots \\ \\ \hline 0 \\ \frac{L_x}{2}\rho_1 g_{1N} \\ \frac{L_x}{2}\rho_2 g_{2N} \\ \frac{L_x}{2}\rho_3 g_{3N} \\ 0 \end{array}\right). \tag{96}$$

- Here, we have multiplied $\underline{g}$ by $R\,R_N^T\,Area_f \;=\; R\,R_N^T\,\mathcal{J}_f\hat{B}$ to properly lift the surface flux boundary data to the full domain, followed by restriction.

- The restriction matrix R of course comes from the test function v.