Spectral Element Method

Background and Details

A compilation by

Dr. Jacques C. Richard

JC-Richard@CSU.edu

Spectral Element Method

- Like Finite Element Method
- But with Spectral Functions
- Infinitely differentiable global functions of SEM vs. local character of FEM functions.
- Adaptive mesh
- Polynomials of high and differing degrees
- Non-conforming spectral element method presented here is as described by Fischer; Patera; van de Vosse and Minev; Bernadi and Maday, etc.

SEM Discretization

- Polynomial approximation for velocity two degrees higher than that for pressure
- Avoids spurious pressure modes.
- Like solving eqs. on a staggered grid where $\mathbf{u}$ and p are solved on different grids but coupled (e.g., via interpolation)

SEM Approach

- Temporal discretization of Navier-Stokes eqs. based on high-order operator splitting methods
  - Splitting problem into convection & diffusion
  - Some combination of integration schemes for convection operator or for time-dependent terms that may be high order
  - With some degree of polynomial for SEM discretization of diffusion terms giving high-order in space
- Coupled w/SEM spatial discretization to yield sequence of symmetric positive definite (SPD) sub-problems to be solved at each time step.

Current Models

- SEM for unsteady incompressible viscous flow
- Navier-Stokes eqs.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \bullet \nabla \mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}$$

$$\nabla \bullet \mathbf{u} = 0$$

Initial and Boundary Conditions

- Ic: $\mathbf{u}(\mathbf{x},0)=\mathbf{u}^0(\mathbf{x})$
- bc's: $\mathbf{u} = \mathbf{u}_v$ on $\partial\Omega_v$,

  $\nabla u_i \bullet \hat{\mathbf{u}}_\mathbf{n} = 0$ on $\partial\Omega_o$ or $\nabla u_i \bullet \hat{\mathbf{n}} = 0$
  - $\hat{\mathbf{u}}_\mathbf{n}$ is an outward pointing normal on boundary
  - Subscripts v and o denote parts of boundary w/either "velocity" or "outflow" bc's

SEM Algorithm

- The convective term is expressed as a material derivative, which is discretized using a stable m^{th} order backward-difference scheme (m=2 or 3)
- For m=2, $$\frac{\tilde{\mathbf{u}}^{n-2} - 4\tilde{\mathbf{u}}^{n-1} + 3\tilde{\mathbf{u}}^{n}}{2\Delta t} = S(\tilde{\mathbf{u}})$$
- where RHS represents a linear symmetric Stokes problem to be solved implicitly and $\tilde{\mathbf{u}}^{n-2}$ is a velocity field that is computed as the explicit solution to a pure convection problem over time interval $[t^{n-2}, t^{n}]$.

SEM Algorithm

- Sub-integration of convection term permits values of Δt corresponding to convective Courant numbers CFL = $max_{\Omega} c\Delta t/\Delta r$ = 1-5
- Significantly reduces number of (computationally expensive) Stokes solves

Operator Splitting

- Splitting leads to unsteady Stokes problem to be solved at each time step in Ω:

  $\mathcal{H}\,\mathbf{u}^n + \nabla p^n = \mathbf{f}^n$

  $\nabla \bullet \mathbf{u}^n = 0$

  where $\mathcal{H} = (-\nabla^2/\mathrm{Re} + c_0 / \Delta t)$ is the Helmholtz operator,

  c_0 is an order unity constant

  $\mathbf{f}^n$ incorporates treatment of non-linear terms

SEM Algorithm

- Stokes discretization (w/o n) based on following variational form: Find $(\mathbf{u}, p)$ in $X \times Y$ such that

$$\frac{1}{\mathrm{Re}}(\nabla \mathbf{u}, \nabla \mathbf{v}) + \frac{3}{2\Delta t}(\mathbf{u}, \mathbf{v}) - (p, \nabla \bullet \mathbf{v}) = (\mathbf{f}, \mathbf{v})$$

$$(\nabla \bullet \mathbf{u}, q) = 0$$

- $\forall$ $(\mathbf{v},\mathrm{q}) \in X \times Y$, I.e., as weights in $X \times Y$.
- Inner products: $(l,g)=\int_\Omega l(\mathbf{x})\, g(\mathbf{x})\, d\,\mathbf{x}$

Proper Subspaces

- The proper subspaces for **u**, **v**, and p, q are:

  $X=\{\mathbf{v} : v_i \in H^1{}_0(\Omega), i=1,\ldots,d, \ \mathbf{v} = 0 \text{ on } \partial\Omega_v\}$, d=2 if 2D...

  $Y= L^2(\Omega)$

  – L^2 is the space of square integrable functions on Ω; $\int_\Omega v^2 \mathrm{dV} = \int_\Omega v^2 \mathrm{d}^3\mathbf{r}$

  – $H^1{}_0$ is the space of functions in L^2 that vanish on the boundary ($_0$) and whose first derivative (^{1}) is also in L^2; $\int_\Omega(\partial v/\partial \mathrm{r})^2 d\mathrm{V} = \int_\Omega(\partial v/\partial \mathrm{r})^2 d^3\mathbf{r}$

- Spatial discretization proceeds by restricting **u**, **v**, and p, q to compatible finite-dimensional velocity and pressure subspaces: $X^N \subset X$ and $Y^N \subset Y$

SEM Algorithm

- Stokes discretization is then written as:

Find $(\mathbf{u}, p)$ in $X^N \times Y^N$ such that

$$\frac{1}{\mathrm{Re}}(\nabla\mathbf{u},\nabla\mathbf{v})_{GL} + \frac{3}{2\Delta t}(\mathbf{u},\mathbf{v})_{GL} - (p,\nabla \bullet \mathbf{v})_G = (\mathbf{f},\mathbf{v})_{GL}$$

$$(\nabla \bullet \mathbf{u}, q)_G = 0$$

- $\forall$ $(\mathbf{v},q) \in X^N \times Y^N$, I.e., as weights in $X^N \times Y^N$.
- Subscripts $(.,.)_{GL}$ and $(.,.)_G$ refer to Gauss-Lobatto-Legendre (GL) and Gauss-Legendre (G) quadrature

Sub-Domains

- In SEM, bases for X^N and Y^N are defined by tessellating domain into K non-overlapping sub-domains $\Omega = \cup^{K}_{k=1} \Omega^k$
- Within each sub-domain, functions are represented in terms of tensor-product polynomials on a reference sub-domain, e.g., $\Omega_{\mathrm{ref}} := [-1,1]^d$.

Mapping Sub-Domain to “Reference Sub-Domain”

- Each Ω^k is image of ref. sub-domain under mapping: $\mathbf{x}^k(\mathbf{r}) \in \Omega^k \Rightarrow \mathbf{r} \in \Omega_{\mathrm{ref}}$
- With well-defined inverse:

  $\mathbf{r}^k(\mathbf{x}) \in \Omega_{\mathrm{ref}} \Rightarrow \mathbf{x} \in \Omega^k$
- I.e., each sub-domain is a deformed quadrilateral in $\mathbf{R}^2$ (2D) or deformed parallelepiped in $\mathbf{R}^3$ (3D)
- Intersection of closure of any two sub-domains is void, a vertex, an entire edge (2D), or an entire face (3D)

Conforming/Non-Conforming SEM

- For conforming case $\Gamma^{kl} = \Omega^k \cap \Omega^l$ for $k \neq l$ is void, a single vertex, or an entire edge.
- For non-conforming case, Γ^{kl} may be a subset of either $\partial\Omega^k$ or $\partial\Omega^l$ but must coincide with an entire edge of the elements.
- Function continuity, $\mathbf{u} \in H^1_0(\Omega)$, enforced by matching Lagrangian basis functions on sub-domain interfaces.
- The velocity space is thus conforming, even for the nonconforming meshes (by 1st bullet)

Handling Pressure

- To avoid spurious pressure modes, Maday, Patera and Rønquist, and, Bernardi and Maday suggest different approximation spaces for velocity and pressure:

  $X^N = X \cap \mathbf{P}_{N,K}(\Omega)$

  $Y^N = Y \cap \mathbf{P}_{N\text{-}2,K}(\Omega)$

  where

$$\mathbf{P}_{N,K}(\Omega)=\{v(\mathbf{x}^k(\,\mathbf{r}\,))|_{\Omega^k} \in \mathbf{P}_N(r_1) \otimes \ldots \otimes \mathbf{P}_N(r_d),\ k=1,..,K\ \}$$

  and $\mathbf{P}_N(r)$ is space of all polynomials of degree $\leq N$

Space Dimensions

- Dimension of Y^N is $K(N\text{-}1)^d$ since continuity is enforced for functions in Y^N
- Dimension of X^N is $dK(N+1)^d$ because
  - functions in X^N must be continuous across sub-domain interfaces
  - Dirichlet bc's on $\partial\Omega_v$

Function Spaces

- Velocity Space: Basis chosen for $\mathbf{P}_N(r)$ is set of Lagrangian interpolants on Gauss-Lobatto-Legendre (GL) quadrature pts. in ref. domain: $\xi_i \in [-1,1]$, $i=0,\ldots,N$
- Pressure Space: Basis chosen for $\mathbf{P}_{N-2}(r)$ is set of Lagrangian interpolants on Gauss-Legendre (G) quadrature pts. in ref. domain: $\eta_i \in\ ]-1,1[$, $i=1,\ldots,N-1$
- Basis for velocity is continuous across sub-domain interfaces but basis for pressure is not

SEM Algorithm Subspaces

- Could also write $X_N := [Z_N H^1{}_0(\Omega^k)]^{\mathrm{d}}$ and $Y_N := Z_{N\text{-}2}$ where $Z_N := \{\, v \in L^2(\Omega) \mid v_\Omega \in \mathbf{P}_N(\Omega^k) \,\}$
  - I.e., v belongs to space of functions in L^2
  - $v_{|\Omega}{}^k$ belongs to space of polynomials of degree $\leq N$ in k^{th} element's size sub-space Ω^k
  - And these both define the space Z_N
- $\mathbf{P}_N(\Omega^k)$ is a space of functions for k^{th} element Ω^k whose image is a tensor-product polynomial of degree $\leq N$ in a ref. solution domain $\Omega_{\text{ref}} := [-1,1]^d$.

SEM Algorithm Quadrature

- Subscripts $(.,.)_{GL}$ and $(.,.)_{G}$ referred to Gauss-Lobatto-Legendre (GL) and Gauss-Legendre (G) quadrature which are:
- $\int^1_{-1} f(x)dx = w_1 f(-1) + w_N f(1) + \sum^N_i w_i f(x_i)$

Gauss-Lobatto-Legendre (GL) Quadrature

- $\int^1_{-1} f(x)dx = w_1 f(-1) + w_N f(1) + \sum^n_i w_i f(x_i)$ where

$$w_i^{GL} = \frac{2N}{(1-x_i^2)L''_{N-1}(x_i)L'_N(x_i)} = \frac{2}{N(N-1)[L_{N-1}(x_i)]^2}$$

- L_n are the *Legendre* polynomials,
- Gauss-Lobatto points are zeroes of L'_N or $(1-x^2)$ L'_N & at endpoints (-1,1)

$$w_{1,N}^{GL} = \frac{2}{N(N-1)}$$

Gauss-Lobatto-Legendre (GL) Quadrature

- w/error

$$E = \frac{N(N-1)^3 2^{2N-1}[(N-2)!]^4}{(2N-1)[(2N-2)!]^3} f^{(2N-2)}(\xi)$$

- for $\xi \in$ (-1,1)
- The weights may also be written as

$$w_i^{GL} = \rho_i = \frac{2}{N(N+1)} \frac{1}{[L_N(x_i)]^2}$$

Gauss-Legendre (G) Quadrature

- Same as Gauss-Legendre-Lobatto
- But w/o endpoints (not used for prescribed function values at boundaries)
- Weights are

$$w_i^G = \sigma_i = \frac{2}{(1-x_i^2)[L_{N+1}(x_i)]^2}$$

- Where L_N are the *Legendre* polynomials,
- Gauss points (interior points) are zeroes of L_{N+1}

Interpolation Polynomials

- Basis functions are Legendre-Gauss-Lobatto-Lagrange interpolation polynomials:

$$h_i = \frac{-1}{N(N+1)L_N(x_i)} \frac{(1-x^2)L_N'(x)}{x-x_i}$$

2D Affine Mappings

- In $f\,(\mathbf{x}^k(\,\mathbf{r}\,))$, $\mathbf{r} \in \Omega_{\mathrm{ref}}$, define:
  $\mathbf{x}^k(\,\mathbf{r}\,) = \mathbf{x}^k(r_1,r_2) = (x^k{}_{0,1} + L^k{}_1\, r_1/2,\; x^k{}_{0,2} + L^k{}_2\, r_2/2)$
  where $x^k{}_{0,i}$ and $L^k{}_j$ represent local translation and dilation constants
- Evaluation of elemental integrals for general curvilinear coordinates is facilitated by these mappings of physical ($\mathbf{x}$) system into local ($\mathbf{r}$) system

2D Affine Mappings

- Derivatives in elemental integrals can be expressed in local (**r**) coordinates w/Jacobian transformation (in indicial notation): $$\frac{\partial}{\partial x_i} = J^{-1}_{i\alpha}\frac{\partial}{\partial r_\alpha}$$
- With Jacobian: $J = \begin{bmatrix} x_{1,r_1} & x_{2,r_1} \\ x_{1,r_2} & x_{2,r_2} \end{bmatrix}$
- Jacobian determinant: $|J| = x_{1,r_1}x_{2,r_2} - x_{2,r_1}x_{1,r_2}$
- And inverse Jacobian: $J^{-1} = \frac{1}{|J|}\begin{bmatrix} x_{2,r_2} & -x_{2,r_1} \\ -x_{1,r_2} & x_{1,r_1} \end{bmatrix}$

2D Affine Mappings

- Using $\mathbf{x}^k(r_1,r_2) = (x^k_{0,1} + L^k_1\, r_1/2,\; x^k_{0,2} + L^k_2\, r_2/2)$
- The Jacobian is: $J = \begin{bmatrix} x_{1,r_1} & x_{2,r_1} \\ x_{1,r_2} & x_{2,r_2} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} L_1^k & 0 \\ 0 & L_2^k \end{bmatrix}$
- Its determinant is: $|J| = x_{1,r_1}\, x_{2,r_2} - x_{2,r_1}\, x_{1,r_2} = \frac{L_1^k L_2^k}{4}$
- And inverse Jacobian is:

$$J^{-1} = \frac{1}{|J|}\begin{bmatrix} x_{2,r_2} & -x_{2,r_1} \\ -x_{1,r_2} & x_{1,r_1} \end{bmatrix} = \begin{bmatrix} \frac{2}{L_1^k} & 0 \\ 0 & \frac{2}{L_2^k} \end{bmatrix}$$

Elemental Integrals

- Using the affine mappings, the integrals can be evaluated as (e.g.):

  $(v_i, f_i)^k = \int^1_{-1} \int^1_{-1} v^k_i f^k_i \, |J|^k \, dr_1 \, dr_2$
- Numerical integration rules for element Ω_k with *GL* is

  $\int_{\Omega k} g \, dV = \rho_m \, \rho_n \, |J^k(\xi_m , \xi_n) | \, g^k (\xi_m , \xi_n)$

  for all $g^k \in C^0(\Omega_k)$

Quadrature Implementation

- Lagrangian bases makes quadrature implementation convenient
- Let $f^k(\mathbf{r}) := f(\mathbf{x}^k(\mathbf{r}))$, $\mathbf{r} \in \Omega_{\mathrm{ref}}$
- In $\mathbf{R}^2$ ($\mathbf{R}^3$ follows readily from tensor product form):

$$(f,g)_{GL} = \sum_k \sum_{i=0}^{N} \sum_{j=0}^{N} f^k(\xi_i,\xi_j) \cdot g^k(\xi_i,\xi_j) \cdot \left| J^k(\xi_i,\xi_j) \right| \cdot \rho_i \rho_j$$

$$(f,g)_{G} = \sum_k \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} f^k(\eta_i,\eta_j) \cdot g^k(\eta_i,\eta_j) \cdot \left| J^k(\eta_i,\eta_j) \right| \cdot \sigma_i \sigma_j$$

where $J^k(\mathbf{r})$ is Jacobian from transformation $\mathbf{x}^k(\mathbf{r})$

Polynomial Representation

- Every scalar in $\mathbf{P}_{N,K}(\Omega)$ is represented in the form $f(\mathbf{x})|_{\Omega^k} = \sum^N_{i=0} \sum^N_{j=0} f^k_{ij}\, h_i(\, r_1\,)\, h_j(\, r_2\,)$
- where $h_i(\, r\,) \in \mathbf{P}_N(r)$ is the Lagrange polynomial satisfying $h_i(\, \xi_j\,) = \delta_{ij}$
- For each sub-domain, a natural ordering, f^k_{ij}, $i, j \in \{\, 0,\ldots,N\, \}^2$ is associated w/vector $\underline{f}^k$
- And, in turn, natural ordering, f^k_{ij}, $k \in \{\, 0,\ldots,K\, \}^2$ is associated w/the $K(N+1)^2 + 1$ vector $\underline{f}_L$

Discrete Stokes System

- Inserting SEM basis

  $f(\mathbf{x}^k(\,\mathbf{r}\,))|_{\Omega}{}^k = \sum^N{}_{i=0} \sum^N{}_{j=0} f^k{}_{ij}\, h_i(\,r_1\,)\, h_j(\,r_2\,)$

  into

  $$\frac{1}{\mathrm{Re}}(\nabla\mathbf{u},\nabla\mathbf{v})+\frac{3}{2\Delta t}(\mathbf{u},\mathbf{v})-(p,\nabla\bullet\mathbf{v})=(\mathbf{f},\mathbf{v})$$

  $$(\nabla\bullet\mathbf{u},q)=0$$

  yields $\mathcal{H}\,\underline{\mathbf{u}}^n$ - $D^T\,\underline{p}^n = B\,\underline{\mathbf{f}}^n$, $D\,\underline{\mathbf{u}}^n = 0$

  where

  - ✓ $\mathcal{H} = A/\mathrm{Re} + B/\Delta t$ = discrete equivalent of Helmholtz operator;
  - ✓ A = discrete Laplacian,
  - ✓ B = mass matrix associated with the velocity mesh (diagonal);
  - ✓ D = discrete divergence operator

Discrete Stokes System

- A pressure correction step is then needed:

$$E\ \delta \underline{p} = -\ D\ \underline{\mathbf{u}}'$$

$$\underline{\mathbf{u}}^n = \underline{\mathbf{u}}^n + \Delta t\ B^{-1}\ D^T\ \delta \underline{p} + O(\Delta t^2\ )$$

where $E = \Delta t\ D\ B^{-1}\ D^T$ is the Stokes Schur complement governing the pressure in the absence of the viscous term

Discrete Stokes System

- Define unassembled mass matrix to be block-diagonal matrix $B_L \equiv diag(\ B^k\ )$
- Where each local mass matrix is expressed as tensor-product of 1D operators:

$$B^k = \left(\frac{L_1^k L_2^k}{4} \right) B^* \otimes B^*$$

- Where B^*=$diag(\ \rho_i\ )$, i=$0,\ldots N$

Discrete Stokes System

- Express

$$(f,g)_{GL} = \sum_{k} \sum_{i=0}^{N} \sum_{j=0}^{N} f^k(\xi_i,\xi_j) \cdot g^k(\xi_i,\xi_j) \cdot \left| J^k(\xi_i,\xi_j) \right| \cdot \rho_i \rho_j$$

in terms of mass matrices as

$$\forall\, f,g \in \mathbf{P}_{N,K}(\Omega)\ (f,g)_{GL} = \sum_k (\underline{f}^k)^T B^k \underline{g}^k = \underline{f}_L^T B_L\, \underline{g}_L$$

Discrete Stokes System

- Similarly, for bilinear form (∇f, ∇g):

  $\forall\, f,g \in \mathbf{P}_{N,K}(\Omega)$ $(f,g)_{GL} = \sum_k (\underline{f}^k)^T A^k \underline{g}^k = \underline{f}_L^T A_L \underline{g}_L$

- Here $A^L \equiv diag(A^k)$ is the unassembled stiffness matrix and A^k is the local stiffness matrix:

$$A^k = \left(\frac{L_2^k}{L_1^k}\right) B^* \otimes A^* + \left(\frac{L_1^k}{L_2^k}\right) A^* \otimes B^*$$

- A^* is a 1D stiffness matrix defined in terms of spectral differentiation matrix D^*:

$$A^*_{ij} = \sum_{l=0}^{N} D^*_{li}\, \rho_l\, D^*_{lj}, \quad i, j \in \{ 0,\ldots,N \}^2$$

$$D^*_{ij} = \left.\frac{dh_j}{dr}\right|_{r=\xi_i}$$

Computing A^k

- Whereas A^* is full, A^k is sparse due to using diagonal mass matrix B^*
- Computational stencil of A^k is a cross, much like finite difference stencil
- For deformed sub-domains, A^k is generally full with $(N+1)^d$ non-zero entries
- Action of A^k upon a vector can be efficiently computed in $O(N^{d+1})$ operations if tensor-product form is retained in favor of its explicit formation

Computing f

- Local sub-domain operators (A_L and B_L) incorporated into global $n_v \times n_v$ system matrices through "direct stiffness" summation assembly procedure which maps vectors from their local representation, f_L to global form, f
- I.e., let Q be global-to-local mapping operator that transfers basis coefs. from global to local ordering: $f_L = Q f$

Computing f

- Local sub-domain operators (A_L and B_L) incorporated into global $n_v \times n_v$ system matrices by defining index set $q_{ijk} \in \{1, \ldots, n_v\}$ which maps vectors from their local representation, f_L to global form, f
- Index set has repeated entries for any node (i, j, k) that is physically coincident w/another (i', j', k'),
- I.e., $q_{ijk} = q_{i'j'k'}$ *iff* $\mathbf{x}^k(r_i, r_j) = \mathbf{x}^{k'}(r_{i'}, r_{j'})$
  or $\mathbf{x}^k{}_{ij} = \mathbf{x}^{k'}{}_{i'j'} \Rightarrow u^k{}_{ij} = u^{k'}{}_{i'j'}$

Computing Index Maps

- Index map can be represented in matrix form as prolongation operator Q which maps from set of global indices to local index set
- Q is a $K(N+1)^d \times n_v$ is a Boolean matrix w/a single "1" in each row and zeroes elsewhere
- If $m=(k - 1) \bullet (N + 1)^2 + j \bullet (N + 1) + i + 1$ is position of f^k_{ij} in f_L and $q = q_{ijk}$ is the corresponding global index
- Then m^{th} column of Q^T is unit vector $\underline{\hat{e}}_q$, I.e., the q^{th} column of the identity matrix

Computing Index Maps

- Application of Q to a vector implies distribution whereas application of Q^T to a vector implies summation, or gathering of information
- Q^T is sometimes referred to as the "direct-stiffness-summation" operator

Discrete Stokes System

- A direct consequence of unique mapping property $q_{ijk} = q_{i'j'k'}$ *iff* $\mathbf{x}^k(r_i, r_j) = \mathbf{x}^{k'}(r_{i'}, r_{j'})$ and use of Lagrangian basis is that

$$\forall\, f,g \in \mathbf{P}_{N,K}(\Omega) \cap H^1,$$

$$(\nabla f, \nabla g)_{GL} = \underline{f}^T Q^T A_L\, Q\, \underline{g}$$

- Define $Q^T A_L\, Q$ as Neumann Laplacian operator - it has a null-space of dimension unity corresponding to constant mode
- Define associated Dirichlet operator as $M^T Q^T A_L\, Q\, M$ where M is the diagonal mask matrix having ones on the diagonal at points $q_{ijk} : \mathbf{x}^k_{ij} \in \Omega \cup \partial\Omega_0$ and zeroes elsewhere

Discrete Stokes System

- With operators Q and M the following problems are equivalent:

  For $f \in \mathbf{P}_{N,K}(\Omega)$

  Find $u \in X^N{}_0$ such that $(\nabla v, \nabla u)_{GL} = (v, f)_{GL}$, $\forall\ v \in X^N{}_0$

  Find $\underline{u} \in R(M)$ such that $\underline{v}^T M^T Q^T A_L\, Q\, M\, \underline{u} = M\, Q^T B_L \underline{f}_L$, $\forall\ \underline{v} \in R(M)$
- Here $R()$ is the range of argument and $\underline{f}_L$ is the vector of nodal values of $f(\mathbf{x})$
- Direct stiffness-summation operator ensures that solution will lie in H^1 while mask M enforces homogeneous Dirichlet bc: u=0 on $\partial\Omega_v$

Laplacian and Mass Matrices

- Define discrete Laplacian and mass matrices as:

$$A = M\,Q^T A_L\,Q\,M$$
$$B = M\,Q^T B_L\,Q\,M$$

- Both treated as invertible and SPD
- But this is not strictly true due to null space associated w/boundaries (**u**=0 bc on some boundaries)

Stokes Operators

- Using

$$(f,g)_G = \sum_{k} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} f^k(\eta_i,\eta_j) \cdot g^k(\eta_i,\eta_j) \cdot \left| J^k(\eta_i,\eta_j) \right| \cdot \sigma_i \sigma_j$$

contribution to $(q,\nabla \cdot \mathbf{u})_G = \sum_{l=1}^{d} \left(q, \frac{\partial u_l}{\partial x_l} \right)_G$ from single element in $\mathbf{R}^2$ is

$$\sum_{l=1}^{d} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} q^k\left(\eta_i,\eta_j\right) \cdot \frac{\partial u_l^k}{\partial x_l}\left(\eta_i,\eta_j\right) \cdot \left| J^k\left(\eta_i,\eta_j\right) \right| \cdot \sigma_i \sigma_j$$

Stokes Operators

- Contribution from q represented by Lagrangian interpolants on Gauss points:

  $q^k\,(\eta_i,\ \eta_j) = q^k{}_{ij}$

- Derivative of velocity must be interpolated giving rise to matrix form

$$(q, \nabla \cdot \mathbf{u})_G = \sum_{k=1}^{K} \left(q^k\right)^T \left(D_1^k u_1^k + D_2^k u_2^k\right)$$

Stokes Operators

- For affine mappings case, local derivative matrices are define as

$$D_1^k = \left(\frac{L_2^k}{2}\right) I^* \otimes D^* \qquad\qquad D_2^k = \left(\frac{L_1^k}{2}\right) D^* \otimes I^*$$

  where $I^*_{ij} = \sigma_i\, h_j(\eta_i)$ is the 1D interpolation matrix mapping from Gauss-Lobatto points to Gauss points

- and the weighted 1D differentiation matrix interpolated onto the Gauss points is $D_{ij}^* = \sigma_i \left.\frac{dh_j}{dr}\right|_{r=\eta_i}$

Stokes Problem in Matrix Form

- Let $D_i \equiv D_{L,i}\, Q\, M,\ i=1,\ldots,d$

  with $D_{L,i} \equiv diag(\, D^k{}_i\,)$
- In $\mathbf{R}^2$, matrix form of Stokes problem is

$$\begin{bmatrix} H & & -D_1^T \\ & H & -D_2^T \\ -D_1 & -D_2 & 0 \end{bmatrix} \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{f}_1 \\ \underline{f}_2 \\ \underline{f}_p \end{pmatrix}$$