Implementation of a Second-Order Stabilized CVFEM using Intrepid

Kara Peterson

Pavel Bochev Mauro Perego Suzey Gao

Sandia National Laboratories

Trilinos User's Group

October 28, 2014

SAND2014-19545C

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Outline

1. CVFEM for Advection Diffusion
2. Stabilization based on Scharfetter-Gummel Upwinding
3. Multi-scale CVFEM for Advection-Diffusion
4. Computational Results
5. Conclusions

CVFEM for Advection Diffusion

Advection-Diffusion Equation

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$

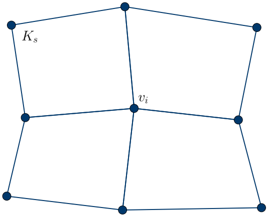
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$

$$\phi = g \quad \text{on } \Gamma$$

CVFEM for Advection Diffusion

Advection-Diffusion Equation

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$

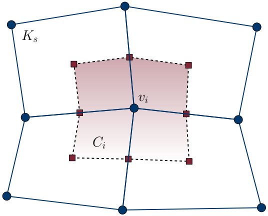


$$\phi_h(\mathbf{x}) = \sum_{p_i \in P(\Omega)} \phi_i N_i(\mathbf{x})$$

CVFEM for Advection Diffusion

Advection-Diffusion Equation

$$
\begin{aligned}
-\nabla \cdot F(\phi) &= f \quad \text{in } \Omega \\
F(\phi) &= (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega \\
\phi &= g \quad \text{on } \Gamma
\end{aligned}
$$



$$\phi_h(\mathbf{x}) = \sum_{p_i \in P(\Omega)} \phi_i N_i(\mathbf{x})$$

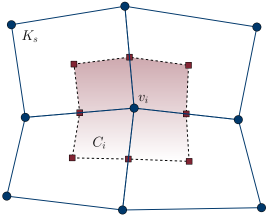
Integrate over control volumes

$$\int_{\partial C_i} F(\phi_h) \cdot \mathbf{n} dS = \int_{C_i} f dV \quad \forall p_i \in P(\Omega)$$

CVFEM for Advection Diffusion

Advection-Diffusion Equation

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$



$$\phi_h(\mathbf{x}) = \sum_{p_i \in P(\Omega)} \phi_i N_i(\mathbf{x})$$

Integrate over control volumes

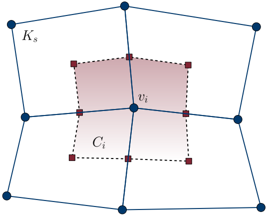
$$\int_{\partial C_i} F(\phi_h) \cdot \mathbf{n} dS = \int_{C_i} f dV \quad \forall p_i \in P(\Omega)$$

$$F(\phi_h) = \sum_{p_j \in P(\Omega)} \phi_j \left(\epsilon \nabla N_j(\mathbf{x}) - \mathbf{u} N_j(\mathbf{x})\right)$$

CVFEM for Advection Diffusion

Advection-Diffusion Equation

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$



$$\phi_h(\mathbf{x}) = \sum_{p_i \in P(\Omega)} \phi_i N_i(\mathbf{x})$$

Integrate over control volumes

$$\int_{\partial C_i} F(\phi_h) \cdot \mathbf{n} dS = \int_{C_i} f dV \quad \forall p_i \in P(\Omega)$$

Linear system:

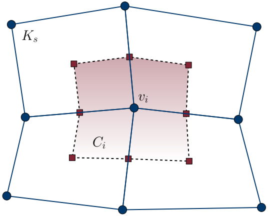
$$\mathbf{A}\boldsymbol{\phi} = \mathbf{f}$$

$$A_{ij} = \int_{\partial C_i} F(N_j) \cdot \mathbf{n} dS, \quad \mathbf{f}_i = \int_{C_i} f dV$$

CVFEM for Advection Diffusion

Advection-Diffusion Equation

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$



$$\phi_h(\mathbf{x}) = \sum_{p_i \in P(\Omega)} \phi_i N_i(\mathbf{x})$$

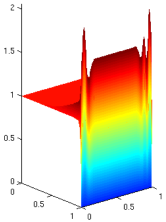
Integrate over control volumes

$$\int_{\partial C_i} F(\phi_h) \cdot \mathbf{n} dS = \int_{C_i} f dV \quad \forall p_i \in P(\Omega)$$

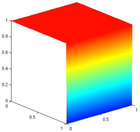
Linear system:

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{f}$$

$$A_{ij} = \int_{\partial C_i} F(N_j) \cdot \mathbf{n} dS, \quad \mathbf{f}_i = \int_{C_i} f dV$$



Unstabilized CVFEM



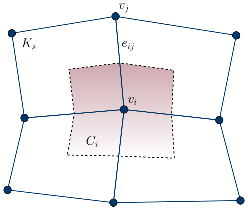
Correct Solution

Stabilization

Multi-dimensional Scharfetter-Gummel Upwinding

- Assume that $F_{ij} \approx F(\phi) \cdot \mathbf{t}_{ij}$ is constant along $\mathbf{e}_{ij}$
- 1-d boundary value problem for F_{ij}

$$\frac{dF_{ij}}{ds} = 0; \quad F_{ij} = \bar{\epsilon}_{ij}\frac{d\phi}{ds} - \bar{\mathbf{u}}_{ij}\phi(s)$$
$$\phi(0) = \phi_i \quad \text{and} \quad \phi(h_{ij}) = \phi_j$$



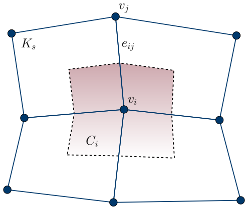
Bochev, Peterson, Gao (2013) "A new control volume finite element method for the stable and accurate solution of the drift-diffusion equations on general unstructured grids",*CMAME* 254, 126-145.

Stabilization

Multi-dimensional Scharfetter-Gummel Upwinding

- Assume that $F_{ij} \approx F(\phi) \cdot \mathbf{t}_{ij}$ is constant along $\mathbf{e}_{ij}$
- 1-d boundary value problem for F_{ij}

$$\frac{dF_{ij}}{ds} = 0; \quad F_{ij} = \bar{\epsilon}_{ij}\frac{d\phi}{ds} - \overline{\mathbf{u}}_{ij}\phi(s)$$
$$\phi(0) = \phi_i \quad \text{and} \quad \phi(h_{ij}) = \phi_j$$



- Edge flux

$$F_{ij} = \frac{h_{ij}\overline{\mathbf{u}}_{ij}}{2}\Big(\phi_j(\coth(\beta_{ij}) - 1) - \phi_i(\coth(\beta_{ij}) + 1)\Big), \quad \beta_{ij} = \frac{\overline{\mathbf{u}}_{ij}h_{ij}}{2\bar{\epsilon}_{ij}}$$

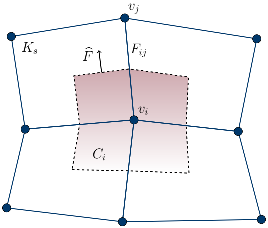
Bochev, Peterson, Gao (2013) "A new control volume finite element method for the stable and accurate solution of the drift-diffusion equations on general unstructured grids",*CMAME* 254, 126-145.

Stabilization

Multi-dimensional Scharfetter-Gummel Upwinding

- Assume that $F_{ij} \approx F(\phi) \cdot \mathbf{t}_{ij}$ is constant along $\mathbf{e}_{ij}$
- 1-d boundary value problem for F_{ij}

$$\frac{dF_{ij}}{ds} = 0; \quad F_{ij} = \bar{\epsilon}_{ij}\frac{d\phi}{ds} - \overline{\mathbf{u}}_{ij}\phi(s)$$
$$\phi(0) = \phi_i \quad \text{and} \quad \phi(h_{ij}) = \phi_j$$



- Edge flux

$$F_{ij} = \frac{h_{ij}\overline{\mathbf{u}}_{ij}}{2}\Big(\phi_j(\coth(\beta_{ij}) - 1) - \phi_i(\coth(\beta_{ij}) + 1)\Big), \quad \beta_{ij} = \frac{\overline{\mathbf{u}}_{ij}h_{ij}}{2\bar{\epsilon}_{ij}}$$

- Expand into primary cell using $H(curl)$-conforming finite elements

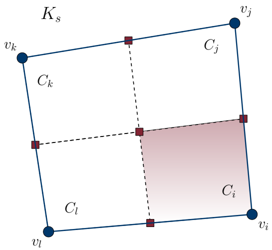
$$\widehat{F}(\phi_h) = \sum_{e_{ij} \in E(\Omega)} F_{ij}\overrightarrow{W}_{ij}$$

Bochev, Peterson, Gao (2013) "A new control volume finite element method for the stable and accurate solution of the drift-diffusion equations on general unstructured grids",*CMAME* 254, 126-145.

Operator Matrix Assembly

Loop over primary cell elements

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j) \cdot \mathbf{n} dS$$

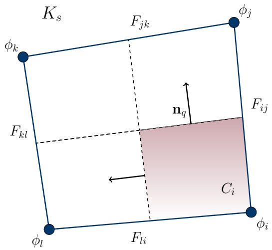


Operator Matrix Assembly

Loop over primary cell elements

- Compute edge flux $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m$
- Compute control volume side normals $(\mathbf{n}_q)$
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume sides (x_q)

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j) \cdot \mathbf{n} dS$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I1

Operator Matrix Assembly

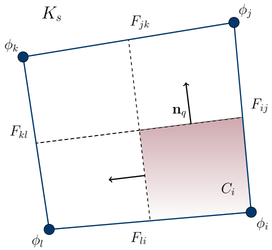
Loop over primary cell elements

- Compute edge flux $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m$
- Compute control volume side normals $(\mathbf{n}_q)$
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume sides (x_q)
- Fill element operator with ϕ_j coefficients

Contributions to C_i boundary integral from $\mathbf{x}_q$

$$A_{ii} \mathrel{+}= \alpha_i^{ij}\overrightarrow{W}_{ij}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_i^{li}\overrightarrow{W}_{li}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{ij} \mathrel{+}= \alpha_j^{ij}\overrightarrow{W}_{ij}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_j^{jk}\overrightarrow{W}_{jk}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{ik} \mathrel{+}= \alpha_k^{jk}\overrightarrow{W}_{jk}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_k^{kl}\overrightarrow{W}_{kl}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{il} \mathrel{+}= \alpha_l^{kl}\overrightarrow{W}_{kl}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_l^{li}\overrightarrow{W}_{li}(\mathbf{x}_q)\cdot\mathbf{n}_q$$

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j)\cdot\mathbf{n}dS$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I1

Operator Matrix Assembly

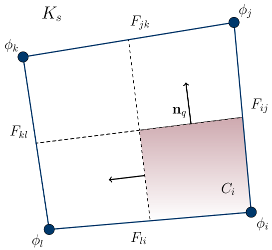
Loop over primary cell elements

- Compute edge flux $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m$
- Compute control volume side normals ($\mathbf{n}_q$)
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume sides (x_q)
- Fill element operator with ϕ_j coefficients

Contributions to C_i boundary integral from $\mathbf{x}_q$

$$A_{ii} \mathrel{+}= \alpha_i^{ij}\overrightarrow{W}_{ij}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_i^{li}\overrightarrow{W}_{li}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{ij} \mathrel{+}= \alpha_j^{ij}\overrightarrow{W}_{ij}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_j^{jk}\overrightarrow{W}_{jk}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{ik} \mathrel{+}= \alpha_k^{jk}\overrightarrow{W}_{jk}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_k^{kl}\overrightarrow{W}_{kl}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{il} \mathrel{+}= \alpha_l^{kl}\overrightarrow{W}_{kl}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_l^{li}\overrightarrow{W}_{li}(\mathbf{x}_q)\cdot\mathbf{n}_q$$

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j)\cdot\mathbf{n}dS$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I1

Note that edge flux expressions are independent of nodal basis.

Operator Matrix Assembly

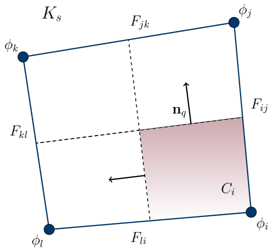
Loop over primary cell elements

- Compute edge flux $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m$
- Compute control volume side normals ($\mathbf{n}_q$)
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume sides (x_q)
- Fill element operator with ϕ_j coefficients

Contributions to C_i boundary integral from $\mathbf{x}_q$

$$A_{ii} \mathrel{+}= \alpha_i^{ij}\overrightarrow{W}_{ij}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_i^{li}\overrightarrow{W}_{li}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{ij} \mathrel{+}= \alpha_j^{ij}\overrightarrow{W}_{ij}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_j^{jk}\overrightarrow{W}_{jk}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{ik} \mathrel{+}= \alpha_k^{jk}\overrightarrow{W}_{jk}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_k^{kl}\overrightarrow{W}_{kl}(\mathbf{x}_q)\cdot\mathbf{n}_q$$
$$A_{il} \mathrel{+}= \alpha_l^{kl}\overrightarrow{W}_{kl}(\mathbf{x}_q)\cdot\mathbf{n}_q + \alpha_l^{li}\overrightarrow{W}_{li}(\mathbf{x}_q)\cdot\mathbf{n}_q$$

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j)\cdot\mathbf{n}dS$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I1

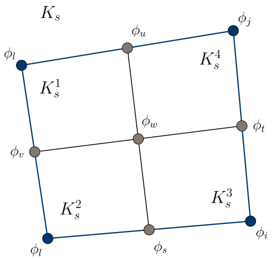
Note that edge flux expressions are independent of nodal basis.

Method works well, but is only 1st order accurate

Stabilization

Second-order Scharfetter-Gummel Upwinding

- Divide each element into four sub-elements



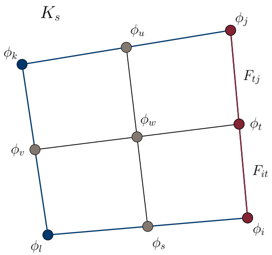
Bochev, Peterson, Perego "A multi-scale control-volume finite element method for advection-diffusion equations",*IJNMF* in review.

Stabilization

Second-order Scharfetter-Gummel Upwinding

- Divide each element into four sub-elements
- Assume that $F_s \approx F(\phi) \cdot \mathbf{t}_s = A + Bs$
- 1-d boundary value problem along segment

$$F_s(s) = -\overline{\mathbf{u}}_s \phi(s) + \overline{\epsilon}_s \frac{d\phi}{ds}$$
$$\phi(0) = \phi_i, \quad \phi(h_s/2) = \phi_t \quad \text{and} \quad \phi(h_s) = \phi_j$$
$$F_{it} = F_s(h_s/4) \quad F_{tj} = F_s(3h_s/4)$$



Bochev, Peterson, Perego "A multi-scale control-volume finite element method for advection-diffusion equations",*IJNMF* in review.

Stabilization

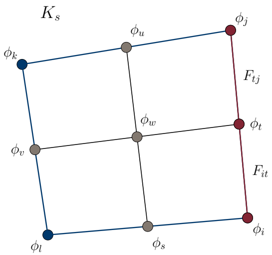
Second-order Scharfetter-Gummel Upwinding

- Divide each element into four sub-elements
- Assume that $F_s \approx F(\phi) \cdot \mathbf{t}_s = A + Bs$
- 1-d boundary value problem along segment

$$F_s(s) = -\overline{\mathbf{u}}_s \phi(s) + \overline{\epsilon}_s \frac{d\phi}{ds}$$

$$\phi(0) = \phi_i, \quad \phi(h_s/2) = \phi_t \quad \text{and} \quad \phi(h_s) = \phi_j$$

$$F_{it} = F_s(h_s/4) \quad F_{tj} = F_s(3h_s/4)$$



- Edge flux

$$F_{it} = F_{it}^{1st}(\phi_i, \phi_t) + \gamma_{it}(\phi_i, \phi_t, \phi_j)$$

$$F_{tj} = F_{tj}^{1st}(\phi_t, \phi_j) + \gamma_{tj}(\phi_i, \phi_t, \phi_j)$$

Bochev, Peterson, Perego "A multi-scale control-volume finite element method for advection-diffusion equations",*IJNMF* in review.

Stabilization

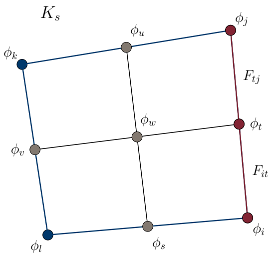
Second-order Scharfetter-Gummel Upwinding

- Divide each element into four sub-elements
- Assume that $F_s \approx F(\phi) \cdot \mathbf{t}_s = A + Bs$
- 1-d boundary value problem along segment

$$F_s(s) = -\overline{\mathbf{u}}_s \phi(s) + \overline{\epsilon}_s \frac{d\phi}{ds}$$

$$\phi(0) = \phi_i, \quad \phi(h_s/2) = \phi_t \quad \text{and} \quad \phi(h_s) = \phi_j$$

$$F_{it} = F_s(h_s/4) \quad F_{tj} = F_s(3h_s/4)$$



- Edge flux

$$F_{it} = F_{it}^{1st}(\phi_i, \phi_t) + \gamma_{it}(\phi_i, \phi_t, \phi_j)$$

$$F_{tj} = F_{tj}^{1st}(\phi_t, \phi_j) + \gamma_{tj}(\phi_i, \phi_t, \phi_j)$$

- Expand into primary cell using $H(curl)$-conforming finite elements

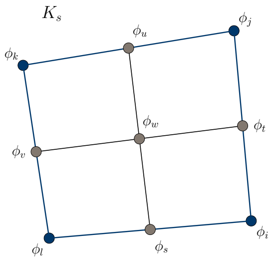
$$\widehat{F}(\phi_h) = \sum_{e_{ij} \in E(\Omega)} F_{ij} \overrightarrow{W}_{ij}$$

Bochev, Peterson, Perego "A multi-scale control-volume finite element method for advection-diffusion equations",*IJNMF* in review.

Operator Matrix Assembly

Loop over macro elements

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j) \cdot \mathbf{n} dS$$

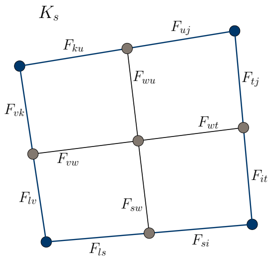


Operator Matrix Assembly

Loop over macro elements

- Compute edge flux nodal coefficients:
  $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m + \alpha_p^{nm}\phi_p$

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j) \cdot \mathbf{n} dS$$

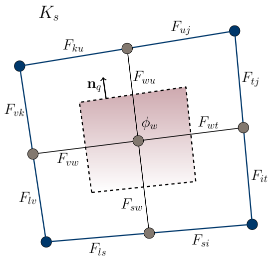


Operator Matrix Assembly

Loop over macro elements

- Compute edge flux nodal coefficients:
  $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m + \alpha_p^{nm}\phi_p$
- Compute control volume side normals ($\mathbf{n}_q$)
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume edges (x_q)

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j) \cdot \mathbf{n} dS$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I2

Operator Matrix Assembly

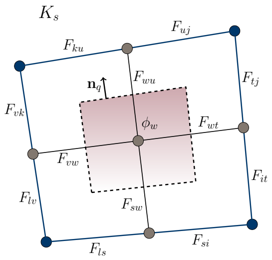
Loop over macro elements

- Compute edge flux nodal coefficients:
  $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m + \alpha_p^{nm}\phi_p$
- Compute control volume side normals ($\mathbf{n}_q$)
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume edges (x_q)
- Fill element operator with ϕ_j coefficients

Example contributions to C_w boundary integral from $\mathbf{x}_q$

$$A_{wi} \mathrel{+}= \alpha_i^{si}\overrightarrow{W}_{si}(x_q)\cdot\mathbf{n}_q + \alpha_i^{ls}\overrightarrow{W}_{ls}(x_q)\cdot\mathbf{n}_q + \alpha_i^{it}\overrightarrow{W}_{it}(x_q)\cdot\mathbf{n}_q + \alpha_i^{tj}\overrightarrow{W}_{tj}(x_q)\cdot\mathbf{n}_q$$

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j)\cdot\mathbf{n}dS$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I2

Operator Matrix Assembly

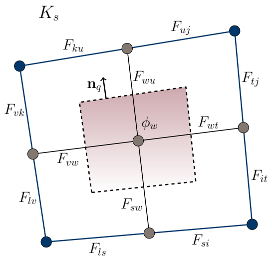
Loop over macro elements

- Compute edge flux nodal coefficients: $F_{nm} = \alpha_n^{nm}\phi_n + \alpha_m^{nm}\phi_m + \alpha_p^{nm}\phi_p$
- Compute control volume side normals ($\mathbf{n}_q$)
- Compute $\overrightarrow{W}_{nm}$ at integration points on control volume edges (x_q)
- Fill element operator with ϕ_j coefficients

$$A_{ij} = \int_{C_i} \widehat{F}(\phi_j) \cdot \mathbf{n} dS$$

Example contributions to C_w boundary integral from $\mathbf{x}_q$

$$A_{wi} \mathrel{+}= \alpha_i^{si}\overrightarrow{W}_{si}(x_q)\cdot\mathbf{n}_q + \alpha_i^{ls}\overrightarrow{W}_{ls}(x_q)\cdot\mathbf{n}_q$$
$$+\alpha_i^{it}\overrightarrow{W}_{it}(x_q)\cdot\mathbf{n}_q + \alpha_i^{tj}\overrightarrow{W}_{tj}(x_q)\cdot\mathbf{n}_q$$



$\overrightarrow{W}$: Intrepid_HCURL_QUAD_I2

Edge flux expressions are independent of nodal basis.

Assembly with Intrepid

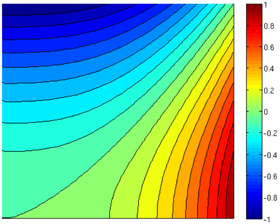
Intrepid provides the following capabilities:

- H(Curl)-conforming basis function definitions
  - $\overrightarrow{W}$: Intrepid_HCURL_QUAD_I1
  - $\overrightarrow{W}$: Intrepid_HCURL_QUAD_I2
- Mappings from reference to physical space
- Routines to compute control volume side normals

Manufactured Solution

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$

$$\phi(x, y) = x^3 - y^2$$
$$\mathbf{u} = (-\sin \pi/6, \cos \pi/6)$$



	CVFEM-MS		CVFEM-SG		FEM-SUPG	
	L^2 error	H^1 error	L^2 error	H^1 error	L^2 error	H^1 error
Grid*	$\epsilon = 1 \times 10^{-3}$					
32	1.57e-3	6.05e-2	4.24e-3	7.48e-2	2.09e-4	3.61e-2
64	3.93e-4	2.89e-2	2.07e-3	4.91e-2	4.85e-5	1.80e-2
128	8.98e-5	1.24e-2	9.78e-4	3.07e-2	1.11e-5	9.02e-3
Rate	2.06	1.14	1.06	0.642	2.12	1.00
Grid	$\epsilon = 1 \times 10^{-5}$					
32	1.69e-3	6.60e-2	4.73e-3	7.90e-2	2.30e-4	3.61e-2
64	4.54e-4	3.45e-2	2.52e-3	5.48e-2	5.78e-5	1.80e-2
128	1.18e-4	1.76e-2	1.30e-3	3.83e-2	1.45e-5	9.02e-3
Rate	1.92	0.955	0.933	0.521	1.99	1.00

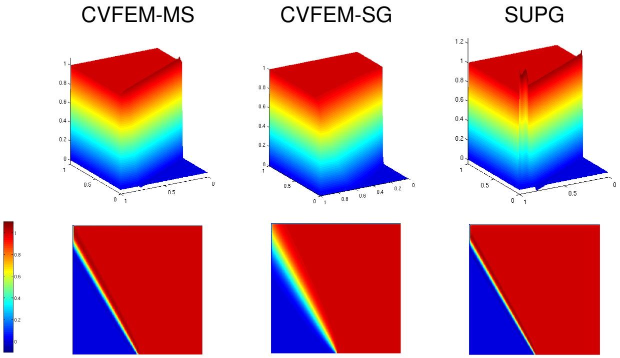
* For CVFEM-MS the size corresponds sub-elements rather than macro-elements.

Skew Advection Test

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$

$$g = \begin{cases} 0 & \text{on } \Gamma_L \cup \Gamma_T \cup (\Gamma_B \cap \{x \leq 0.5\}) \\ 1 & \text{on } \Gamma_R \cup (\Gamma_B \cap \{x > 0.5\}) \end{cases}$$

$$\mathbf{u} = (-\sin \pi/6, \cos \pi/6) \quad \epsilon = 1.0 \times 10^{-5}$$



CVFEM-MS	CVFEM-SG	SUPG
min = -0.0445	min = 0.00	min = -0.0459
max = 1.077	max = 1.004	max = 1.251

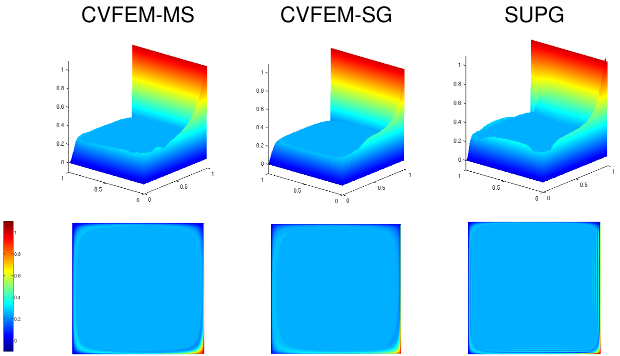
Double Glazing Test

$$-\nabla \cdot F(\phi) = f \quad \text{in } \Omega$$
$$F(\phi) = (\epsilon \nabla \phi - \mathbf{u}\phi) \quad \text{in } \Omega$$
$$\phi = g \quad \text{on } \Gamma$$

$$\epsilon = 1.0 \times 10^{-5}$$

$$g = \begin{cases} 0 & \text{on } \Gamma_L \cup \Gamma_T \cup (\Gamma_B \cap \{x \leq 0.5\}) \\ 1 & \text{on } \Gamma_R \cup (\Gamma_B \cap \{x > 0.5\}) \end{cases}$$

$$\mathbf{u} = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{pmatrix}$$



Conclusions

Multi-scale CVFEM offers a stable and robust method for solving advection-diffusion equations

- Stabilization uses 2nd-order Nedelec elements to lift 2nd-order edge fluxes into element
- Works on unstructured grids
- Does not require heuristic stabilization parameters
- Relatively straightforward to implement using tools in the Intrepid library