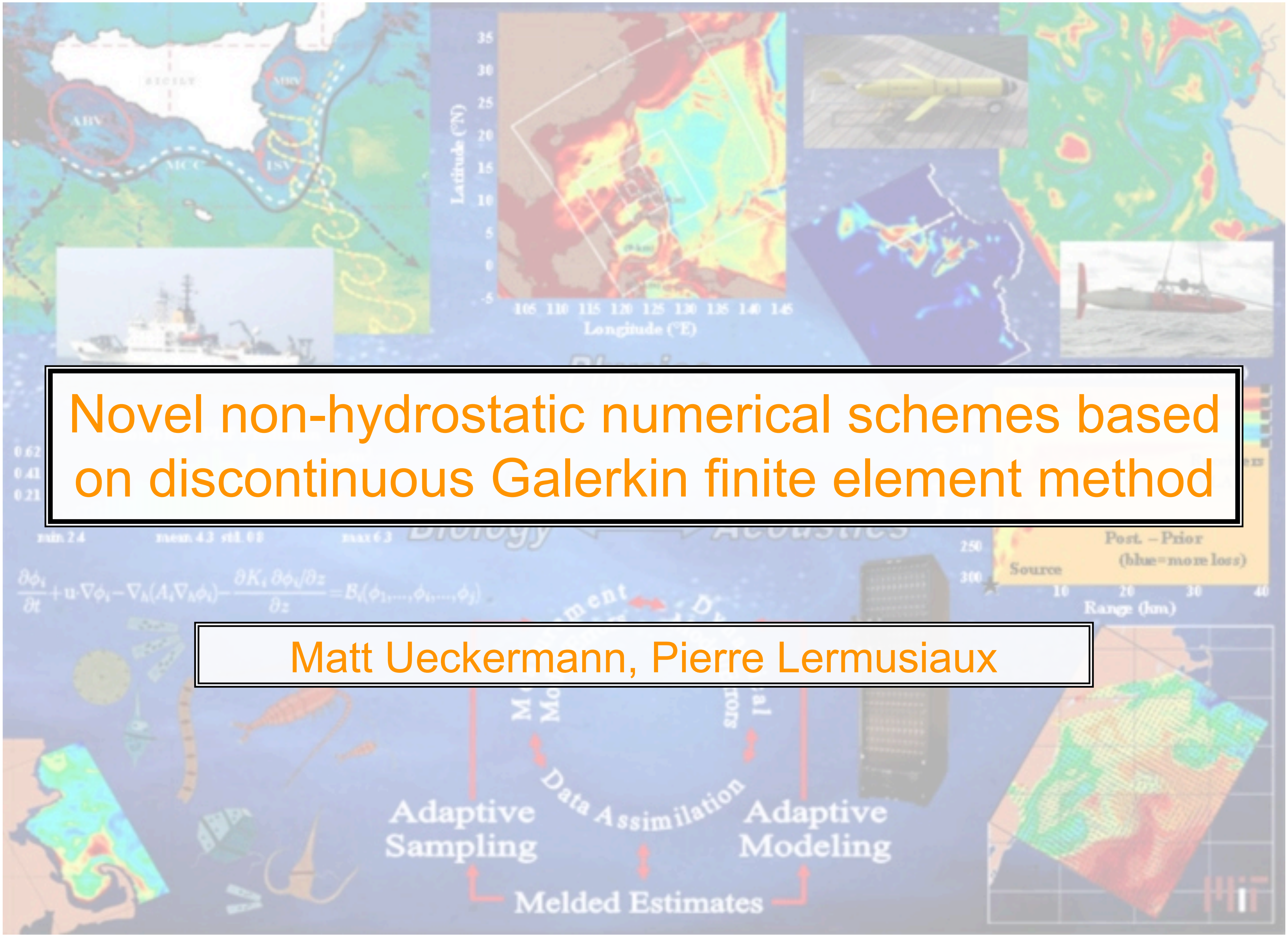


Novel non-hydrostatic numerical schemes based on discontinuous Galerkin finite element method

Matt Ueckermann, Pierre Lermusiaux

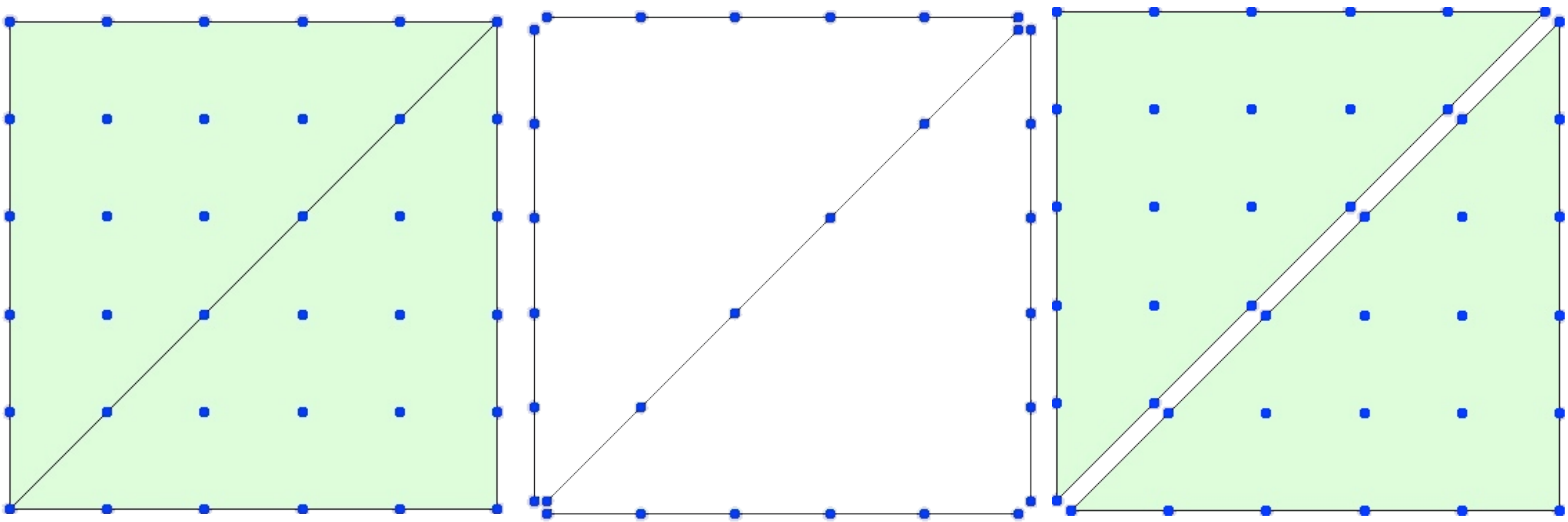
Outline

- Motivation
- Hybrid Discontinuous Galerkin (HDG) Methods
  - Formulation
  - Similarity to other methods (LDG, IP, CG)
  - Post-processing
- Simple Test cases: Convergence studies
- Projection method
  - Results for analytical stokes problem
  - Results for Lock-exchange problem
- Comments on GPU implementation of HDG algorithm
- Future work

Motivation

- Major complaint of DG methods:
  - Too many degrees of freedom (DOFs)!

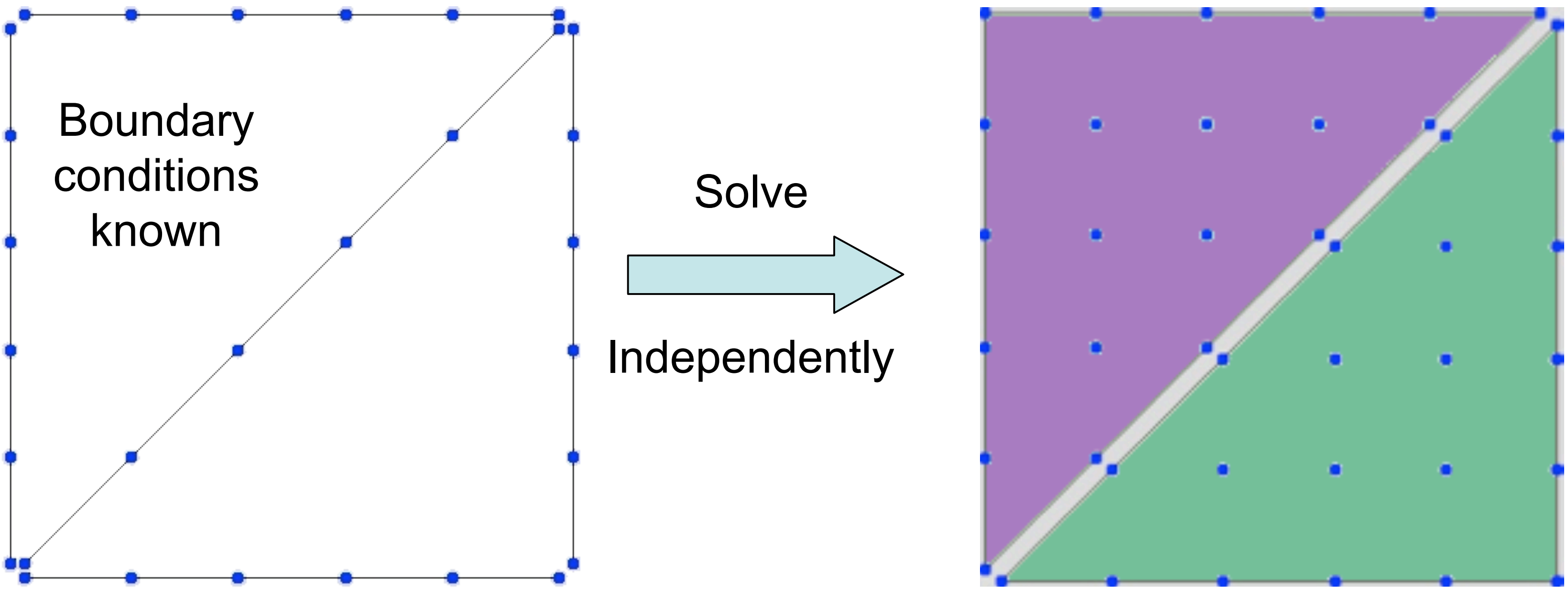
Continuous Galerkin | Hybrid Discontinuous Galerkin | Discontinuous Galerkin



No duplication of DOF | Duplication at corners, but no interior DOFs! | Duplication at edges

Hybrid Discontinuous Galerkin - Formulation

- Basic idea: if boundary value of element is known, the element can be independently solved.

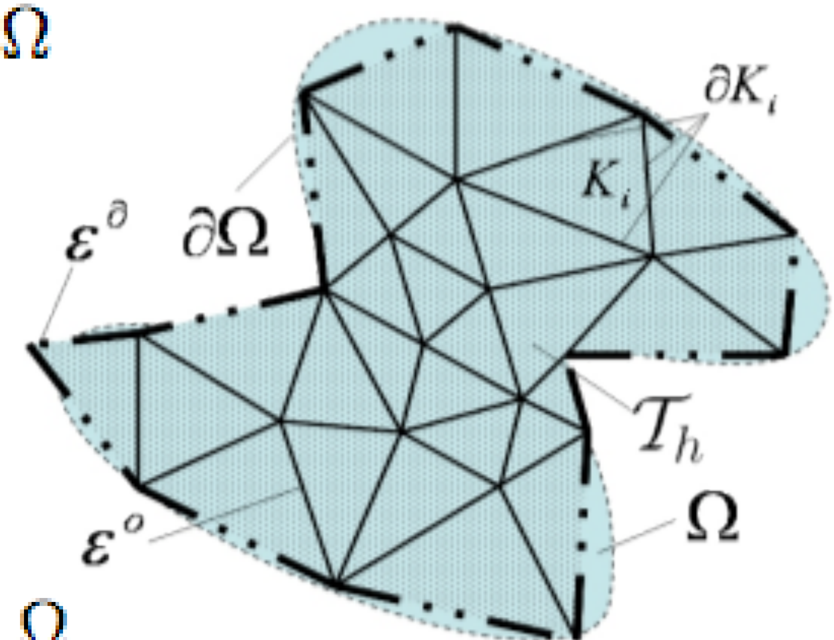


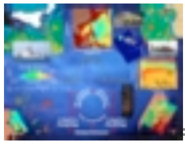
1. Nguyen, N. C., Peraire, J., and Cockburn, B. (2009). *An implicit high-order hybridizable discontinuous galerkin method for linear convection-diffusion equations.* Journal of Computational Physics, 228(9):3232−3254.

2. Cockburn, B., Gopalakrishnan, J., and Lazarov, R. (2009). *Unified hybridization of discontinuous galerkin, mixed, and continuous galerkin methods for second order elliptic problems*. Siam Journal on Numerical Analysis, 47(2):1319−1365.

- Consider:

$$
\begin{aligned}
\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{q} &= 0, \quad \text{in } (0,T) \times \Omega \\
\mathbf{q} + \kappa \nabla u &= 0, \quad \text{in } \Omega \\
u &= g_D, \quad \text{on } \partial\Omega_D \\
\mathbf{q} \cdot \hat{\mathbf{n}} &= g_N, \quad \text{on } \partial\Omega_N \\
u &= u_0, \quad \text{in } (0,0) \times \Omega
\end{aligned}
$$





Hybrid Discontinuous Galerkin - Formulation

- Finite Element (FE) spaces:

$$W_h^p = \{ w \in L^2(\Omega) : w|_K \in \mathcal{P}^p(K), \forall K \in \mathcal{T}_h \}$$

$$V_h^p = \{ \mathbf{v} \in (L^2(\Omega))^d : \mathbf{v}|_K \in (P^p(K))^d, \forall K \in \mathcal{T}_h \}$$

- FE formulation

$$\int_K \kappa^{-1} \mathbf{q}_h \cdot \mathbf{v} dK + \int_{\partial K} \hat{u}_h \mathbf{v} \cdot \mathbf{n} d\partial K - \int_K u_h \nabla \cdot \mathbf{v} dK = 0$$

$$\int_K \frac{\partial u_h}{\partial t} w dK + \int_{\partial K} \hat{\mathbf{q}}_h \cdot \hat{\mathbf{n}} w \partial K - \int_K \mathbf{q}_h \cdot \nabla w dK = 0$$

Hybrid Discontinuous Galerkin - Formulation

- Finite Element (FE) spaces:

$$W_h^p = \{ w \in L^2(\Omega) : w\,|_K \in \mathcal{P}^p(K), \forall K \in \mathcal{T}_h \}$$

$$V_h^p = \{ \mathbf{v} \in (L^2(\Omega))^d : \mathbf{v}\,|_K \in (P^p(K))^d, \forall K \in \mathcal{T}_h \}$$

- FE formulation

$$\int_K \kappa^{-1} \mathbf{q}_h \cdot \mathbf{v} dK + \int_{\partial K} \hat{u}_h \mathbf{v} \cdot \mathbf{n} d\partial K - \int_K u_h \nabla \cdot \mathbf{v} dK = 0$$

$$\int_K \frac{\partial u_h}{\partial t} w dK + \int_{\partial K} \hat{\mathbf{q}}_h \cdot \hat{\mathbf{n}} w \partial K - \int_K \mathbf{q}_h \cdot \nabla w dK = 0$$

$$\hat{\mathbf{q}}_\mathbf{h} = \{\{\mathbf{q}_h\}\} - C_{11} [\![u_h \hat{\mathbf{n}}]\!] + \mathbf{C}_{12} [\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!]$$

$$\hat{u}_h = \{\{u_h\}\} - \mathbf{C}_{12} \cdot [\![u_h \hat{\mathbf{n}}]\!] - C_{22} [\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!]$$

- Finite Element (FE) spaces:

$$W_h^p = \{w \in L^2(\Omega) : w|_K \in \mathcal{P}^p(K), \forall K \in \mathcal{T}_h\}$$

$$V_h^p = \{\mathbf{v} \in (L^2(\Omega))^d : \mathbf{v}|_K \in (P^p(K))^d, \forall K \in \mathcal{T}_h\}$$

- FE formulation

$$\int_K \kappa^{-1}\mathbf{q}_h \cdot \mathbf{v} dK + \int_{\partial K} \hat{u}_h \mathbf{v} \cdot \mathbf{n} d\partial K - \int_K u_h \nabla \cdot \mathbf{v} dK = 0$$

$$\int_K \frac{\partial u_h}{\partial t} w dK + \int_{\partial K} \hat{\mathbf{q}}_h \cdot \hat{\mathbf{n}} w \partial K - \int_K \mathbf{q}_h \cdot \nabla w dK = 0$$

$$\hat{\mathbf{q}}_\mathbf{h} = \{\{\mathbf{q}_h\}\} - C_{11}[u_h \hat{\mathbf{n}}] + \mathbf{C}_{12}[\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!]$$

$$\hat{u}_h = \{\{u_h\}\} - \mathbf{C}_{12} \cdot [\![u_h \hat{\mathbf{n}}]\!] - C_{22}[\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!]$$

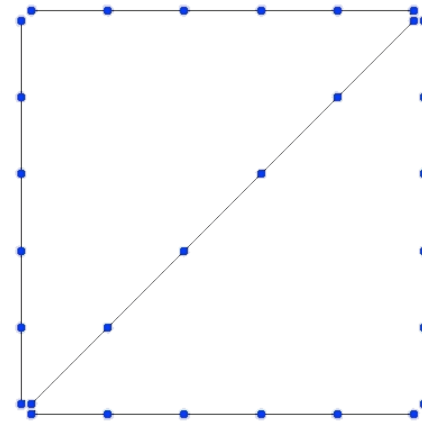
Couples *u* and **q** solutions. Inefficient?

Hybrid Discontinuous Galerkin - Formulation

- Additional FE space

$$M_h^p = \{\mu \in L^2(\varepsilon_h) : \mu|_e \in \mathcal{P}^p(e), \forall e \in \varepsilon_h\}$$

$$\varepsilon_h = \bigcup \partial K_i,$$



- Flux Definition(1)

$$\hat{\mathbf{q}}_h = \mathbf{q}_h + \tau(u_h - \hat{u}_h) \qquad \hat{u}_h = \begin{cases} Pg_D, & \text{on } \varepsilon_h^\partial \\ \lambda_h, & \text{on } \varepsilon_h^\circ \end{cases}$$

- Final Equations

$$\begin{aligned} \int_{\mathcal{T}_h} \kappa^{-1}\mathbf{q}_h \cdot \mathbf{v} d\mathcal{T}_h - \int_{\mathcal{T}_h} u_h \nabla \cdot \mathbf{v} d\mathcal{T}_h + \int_{\partial\mathcal{T}_h} \lambda_h \hat{\mathbf{n}} \cdot \mathbf{v} d\partial\mathcal{T}_h &= -\int_{\partial\mathcal{T}_h} g_D \hat{\mathbf{n}} \cdot \mathbf{v} d\partial\mathcal{T}_h \\ \int_{\mathcal{T}_h} \frac{\partial u_h}{\partial t} w d\mathcal{T}_h - \int_{\mathcal{T}_h} \mathbf{q}_h \cdot \nabla w d\mathcal{T}_h + \int_{\partial\mathcal{T}_h} \hat{\mathbf{q}} \cdot \hat{\mathbf{n}} w d\partial\mathcal{T}_h &= \int_{\mathcal{T}_h} f w d\mathcal{T}_h \\ \int_{\varepsilon_h} [\![\hat{\mathbf{q}}]\!] \mu \partial\varepsilon_h &= \int_{\varepsilon_h^\partial} g_N \mu \partial\Gamma_N \end{aligned}$$

- Algebraic Equations

$$\begin{bmatrix} A & -B^T & C^T \\ B & D & E \\ C & E^T & H \end{bmatrix} \begin{bmatrix} \mathbf{Q} \\ U \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ F \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{Q} \\ U \end{bmatrix} = \boxed{\begin{bmatrix} A & -B^T \\ B & D \end{bmatrix}^{-1}} \left(\begin{bmatrix} 0 \\ F \end{bmatrix} - \begin{bmatrix} C^T \\ E \end{bmatrix} \Lambda \right)$$

Block Diagonal! Element-local inversion

$K = \lambda F$ - Global system of unknowns

Hybrid Discontinuous Galerkin - Formulation

- Algebraic Equations -- Simple implementation

$$K = \lambda F$$

$$K = -\begin{bmatrix} C & E^T \end{bmatrix} \begin{bmatrix} A & -B^T \\ B & D \end{bmatrix}^{-1} \begin{bmatrix} C^T \\ E \end{bmatrix} + H$$

$$F = -\begin{bmatrix} C & E^T \end{bmatrix} \begin{bmatrix} A & -B^T \\ B & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ F \end{bmatrix}$$

- May also implement this by constructing the K matrix <u>directly</u>.
  - Considerably faster
  - Much more difficult to implement

Hybrid Discontinuous Galerkin - Comparison

- HDG in the standard form:

$$\hat{\mathbf{q}}_h = \{\{\mathbf{q}_h\}\} - C_{11}[\![u_h \hat{\mathbf{n}}]\!] + \mathbf{C}_{12}[\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!]$$

$$\hat{u}_h = \{\{u_h\}\} - \mathbf{C}_{12} \cdot [\![u_h \hat{\mathbf{n}}]\!] - C_{22}[\![\mathbf{q}_h \cdot \hat{\mathbf{n}}]\!]$$

$$C_{11} = \frac{\tau^+ \tau^-}{\tau^+ + \tau^-}, \quad \mathbf{C}_{12} = \frac{1}{2}\left(\frac{[\![\tau \hat{\mathbf{n}}]\!]}{\tau^+ + \tau^-}\right), \quad C_{22} = \frac{1}{\tau^+ + \tau^-}$$

<u>Compared to LDG/IP:</u>

- HDG: Sparser matrix, Fewer globally coupled degrees of freedom,
  - Always for 2D(2)
  - For higher than 2^{nd} order basis in 3D(2)
- What about expense of local solvers?

<u>Compared to CG/Mixed-method</u>

- $\tau \to \infty$: CG $\qquad \tau \to 0$: Mixed methods (redefine u, $\mathbf{q}$ spaces)

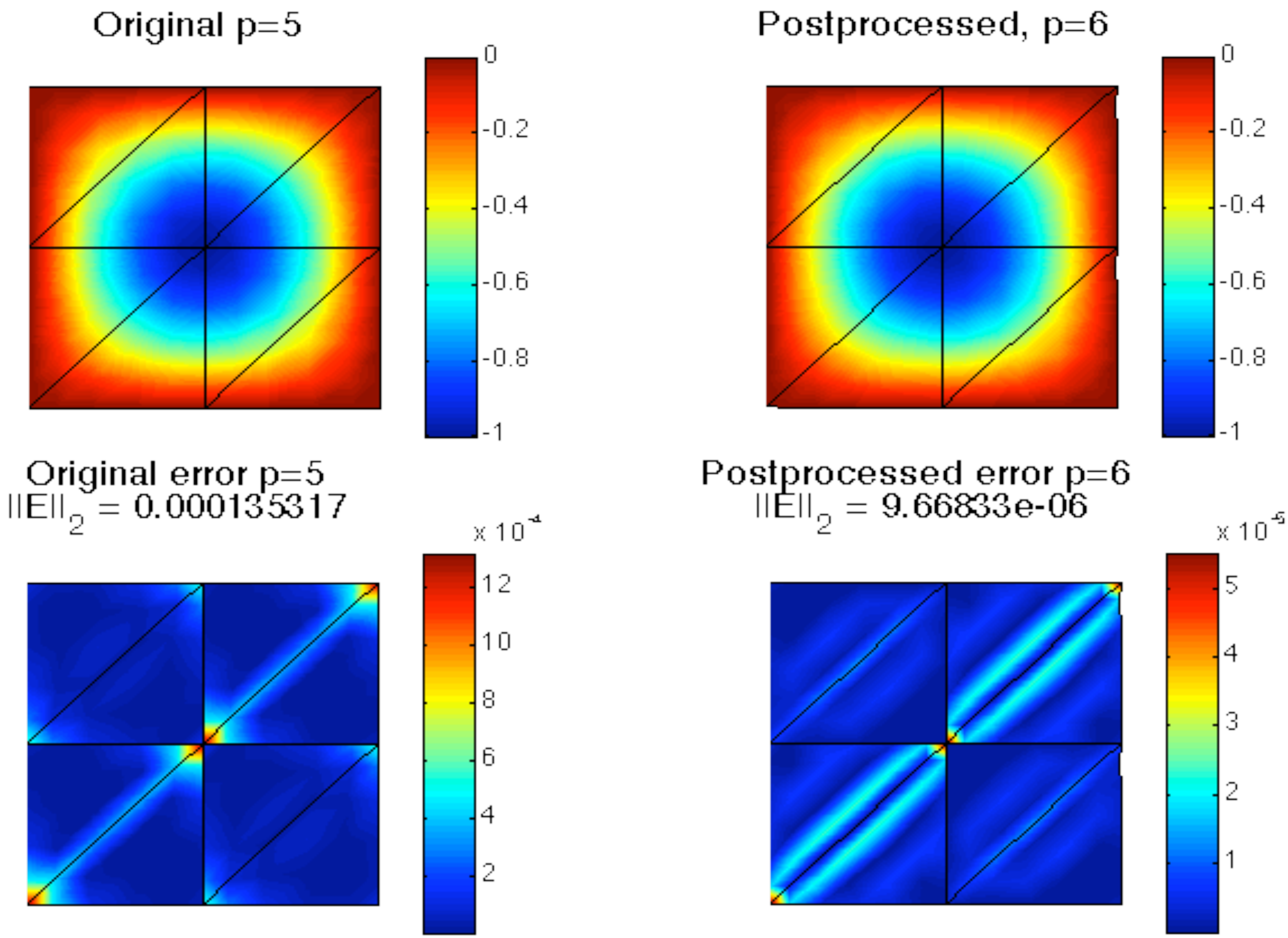
Hybrid Discontinuous Galerkin - Post-Processing

- HDG convergence for $\tau \sim \mathcal{O}(1)$
  - <u>Both</u> *u* and **q** converge at order *p*+1
- Allows for local post-processing of solution

$$
\begin{aligned}
\int_K \kappa \nabla u_h^* \cdot \nabla w dK &= -\int_K \mathbf{q} \cdot \nabla w dK \quad \forall w \in \mathcal{P}^{p+1} \\
\int_K \nabla u_h^* dK &= \int_K u_h dK
\end{aligned}
$$

- Only required at final time

Test Case Examples - Steady Diffusive Problem

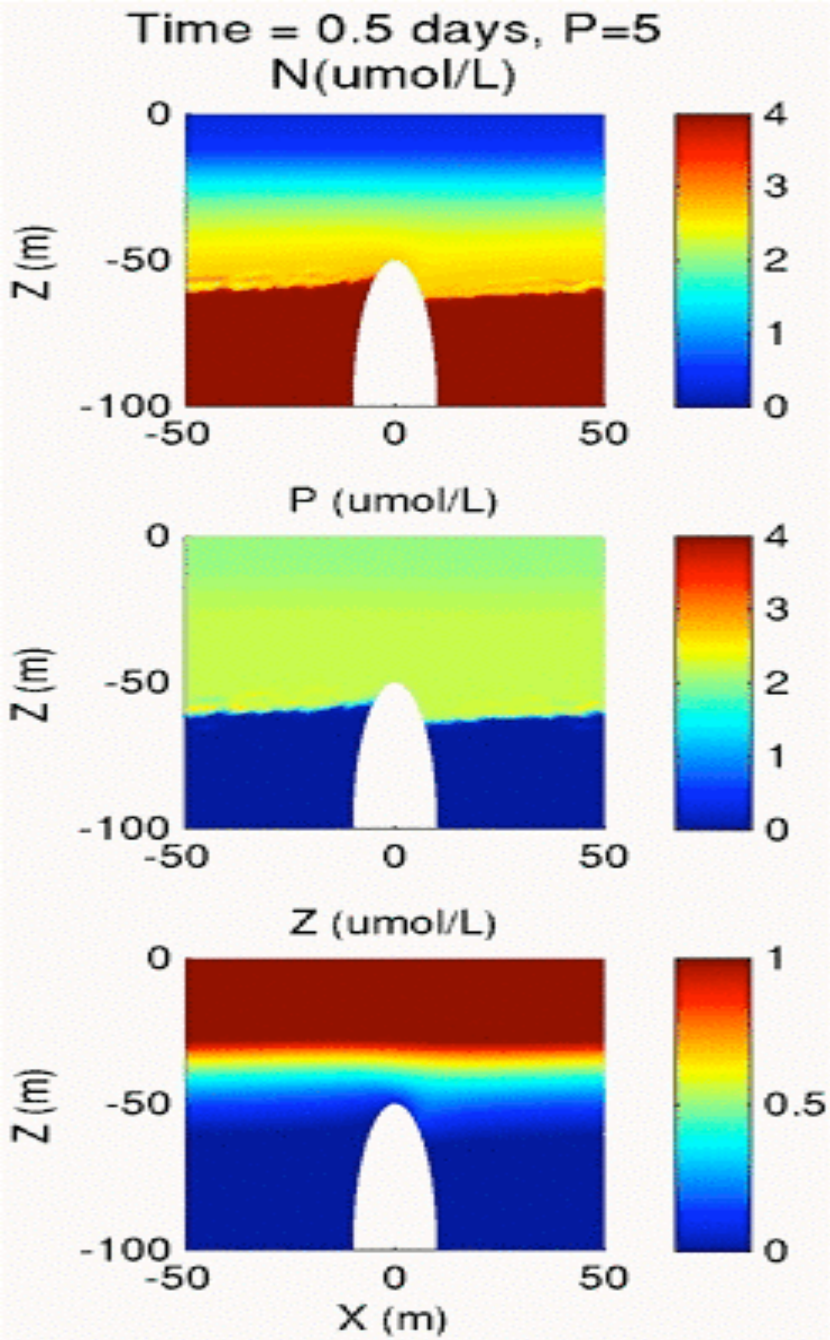


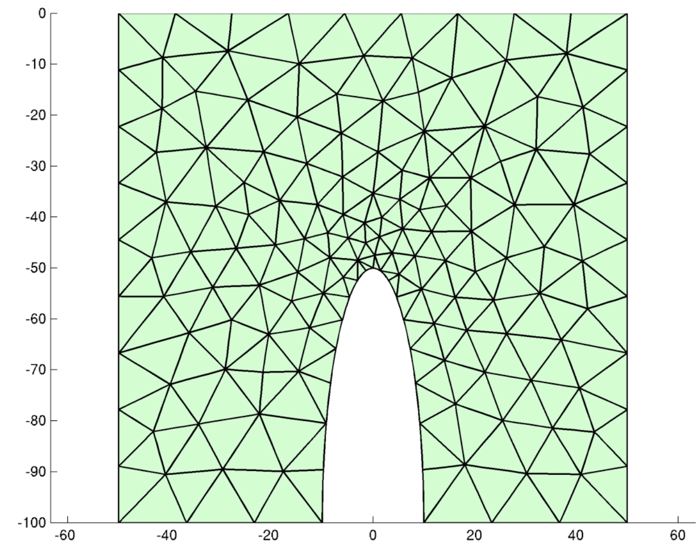
$$u(x,y) = -sin(\pi x)sin(\pi y)$$

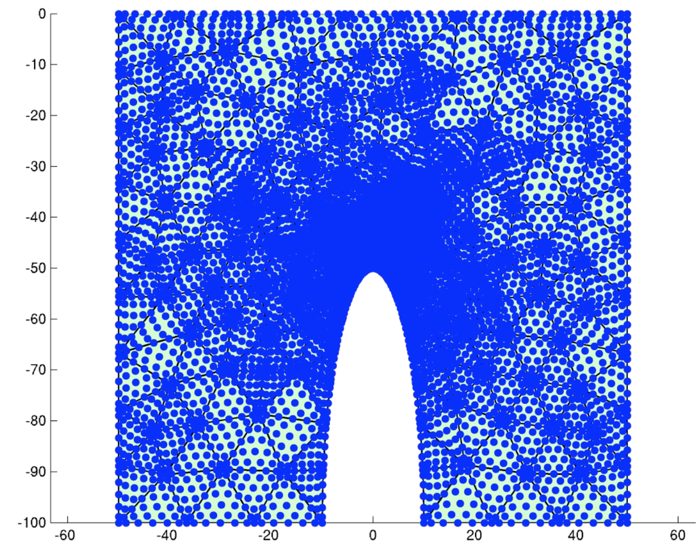
Test Case Examples - Steady Diffusive Problem

Degree(p)	Mesh(n)	u_h Error	u_h Order	q_h Error	q_h Order	U_h^* Error	U_h^* Order
1	3	2.661e-01	0	8.533e-01	0	3.158e-02	0
	5	7.350e-02	1.86	2.510e-01	1.77	4.315e-03	2.87
	9	1.883e-02	1.97	6.559e-02	1.94	5.579e-04	2.95
	17	4.731e-03	1.99	1.662e-02	1.98	7.073e-05	2.98
2	3	4.442e-02	0	1.235e-01	0	4.858e-03	0
	5	5.952e-03	2.9	1.688e-02	2.87	3.225e-04	3.91
	9	7.565e-04	2.98	2.162e-03	2.97	2.029e-05	3.99
	17	9.491e-05	2.99	2.722e-04	2.99	1.266e-06	4
3	3	7.049e-03	0	1.857e-02	0	7.181e-04	0
	5	4.686e-04	3.91	1.251e-03	3.89	2.352e-05	4.93
	9	2.975e-05	3.98	7.967e-05	3.97	7.424e-07	4.99
	17	1.867e-06	3.99	5.004e-06	3.99	2.362e-08	4.97

Test Case Examples - Unsteady ADR







Projection Method

- Solution scheme: Incremental pressure correction scheme[3]
  - Uses Pressure Poisson Equation (PPE)

$$\frac{2\Delta t}{3}(\tilde{u}^{k+1} - 4u^k + u^{k+1}) - \nu\nabla^2\tilde{u}^{k+1} + \nabla p^k = f(t^{k+1}), \quad \tilde{u}^{k+1}|_\Gamma = 0$$

$$\nabla^2(p^{k+1} - p^k) = \frac{3}{2\Delta t}\boxed{\nabla \cdot \tilde{u}^{k+1}}$$

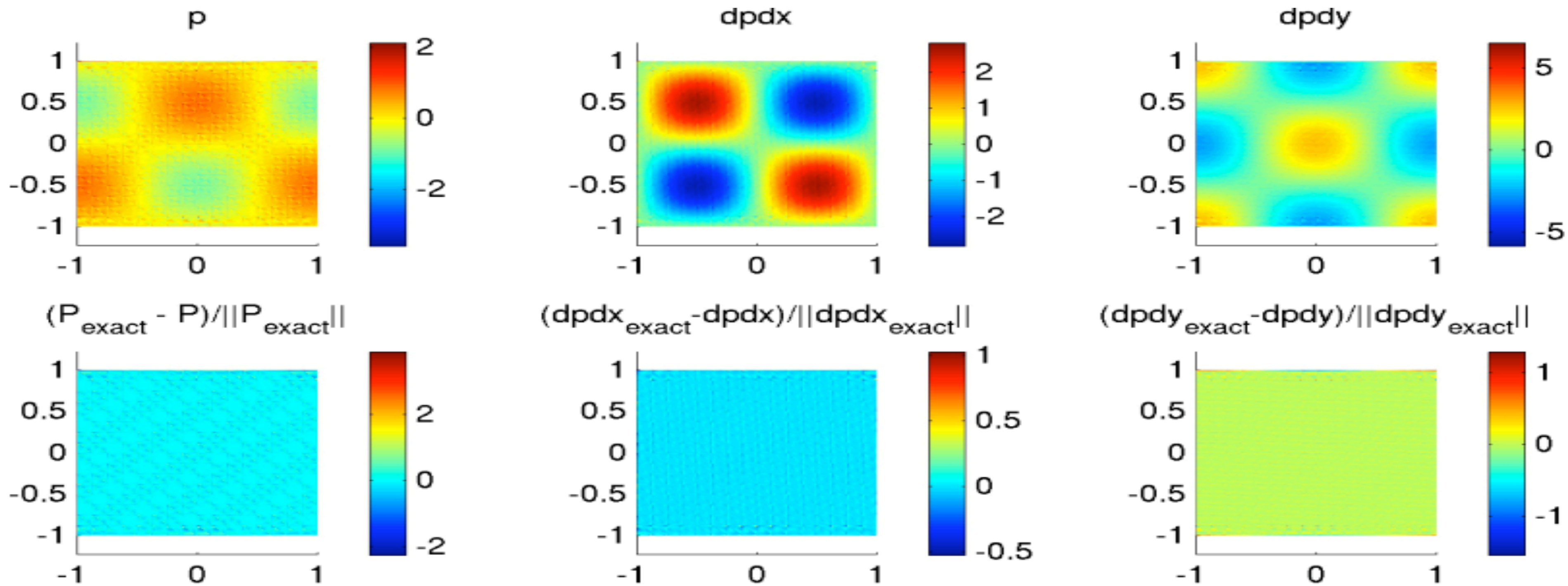
$$u^{k+1} = \tilde{u}^{k+1} - \frac{2\Delta t}{3}\boxed{\nabla(p^{k+1} - p^k)}$$

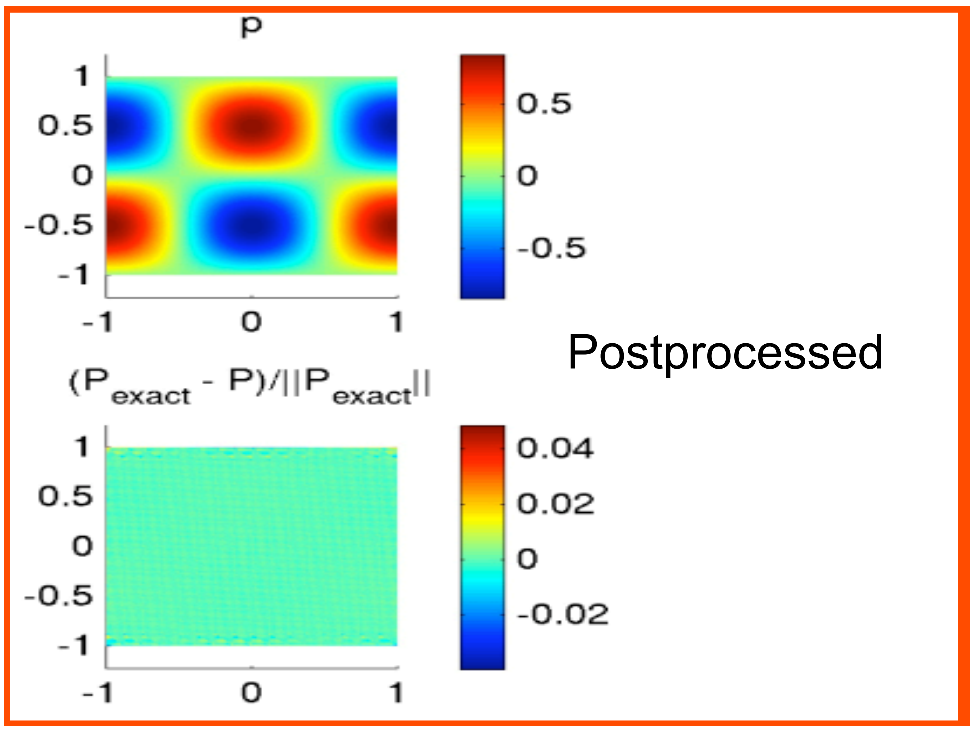
$$f(t^k + 1) = -u^{\star,k+1} \cdot \nabla u^{\star,k+1} + \vec{g}\rho^{\star,k+1}$$

$$(.)^{\star,k+1} = 2(.)^k - (.)^{k-1}$$

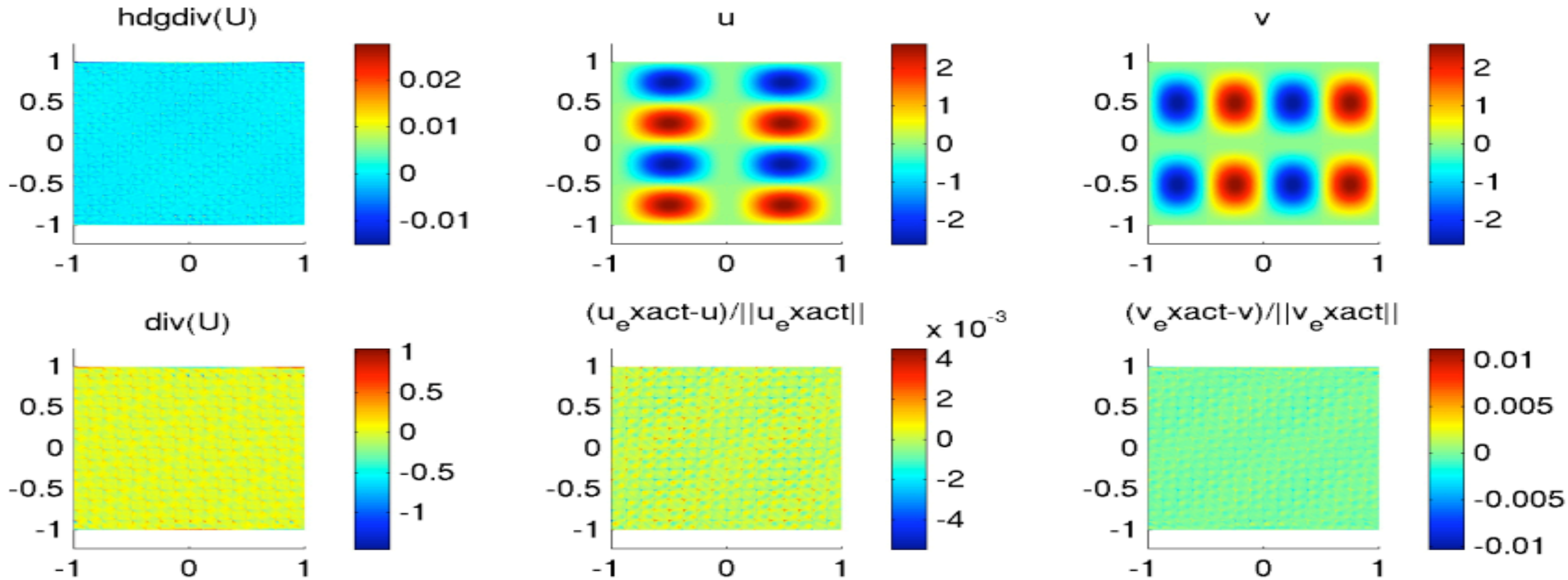
- HDG provides accurate derivative terms

3. Guermond, J.L., Minev, P., and Shen, Jie. (2006). *An overview of projection methods for incompressible flows*. Comput. Methods Appl. Mech. Engrg, 195:6011–6045.

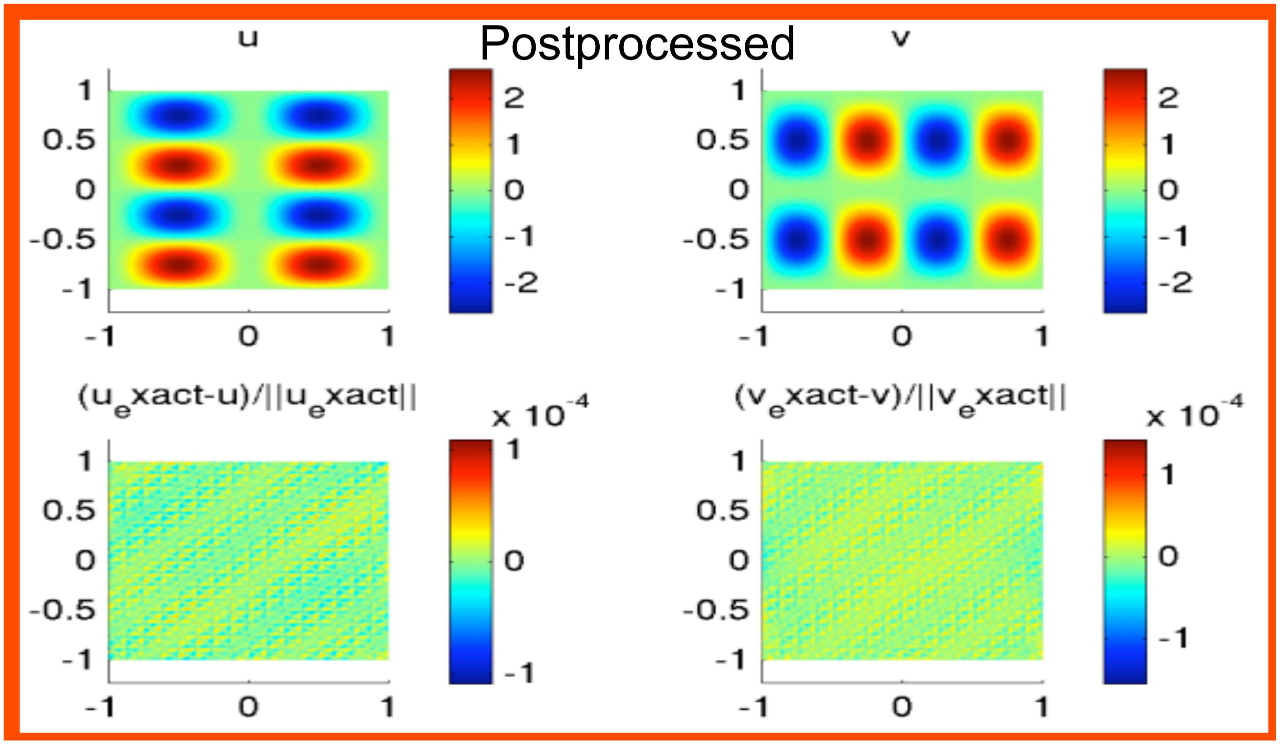


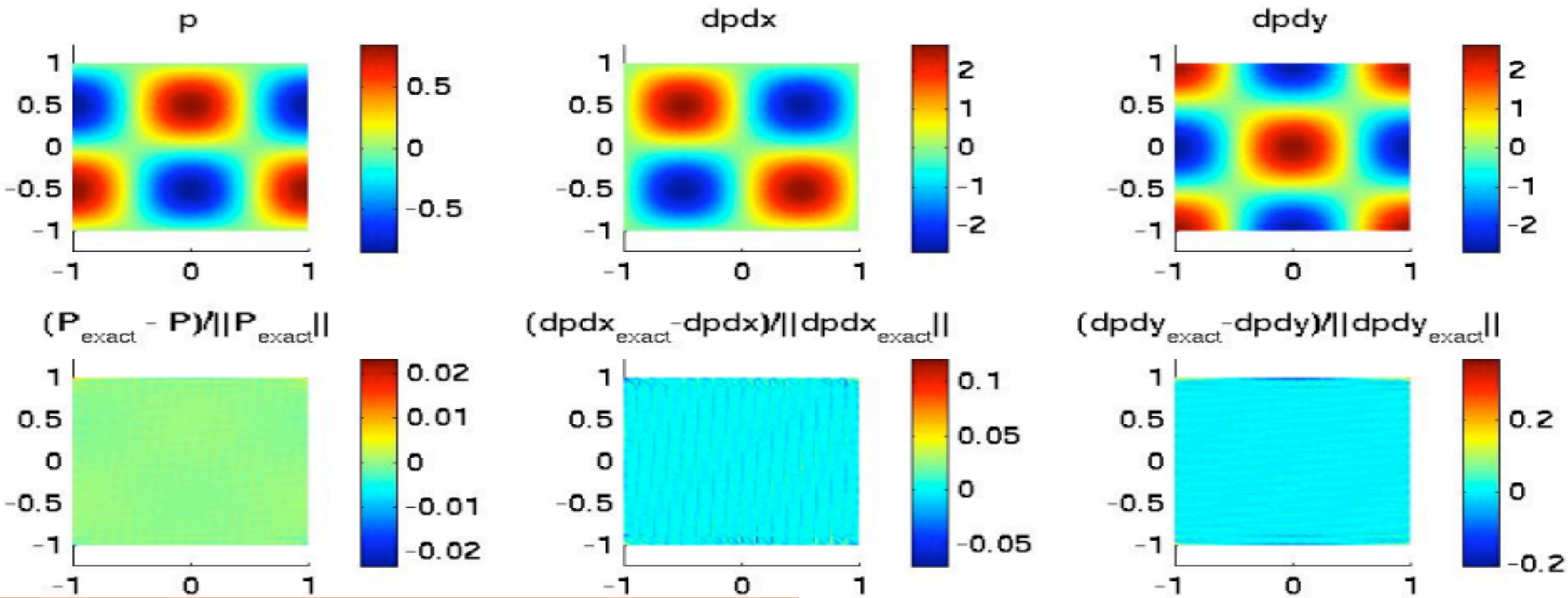


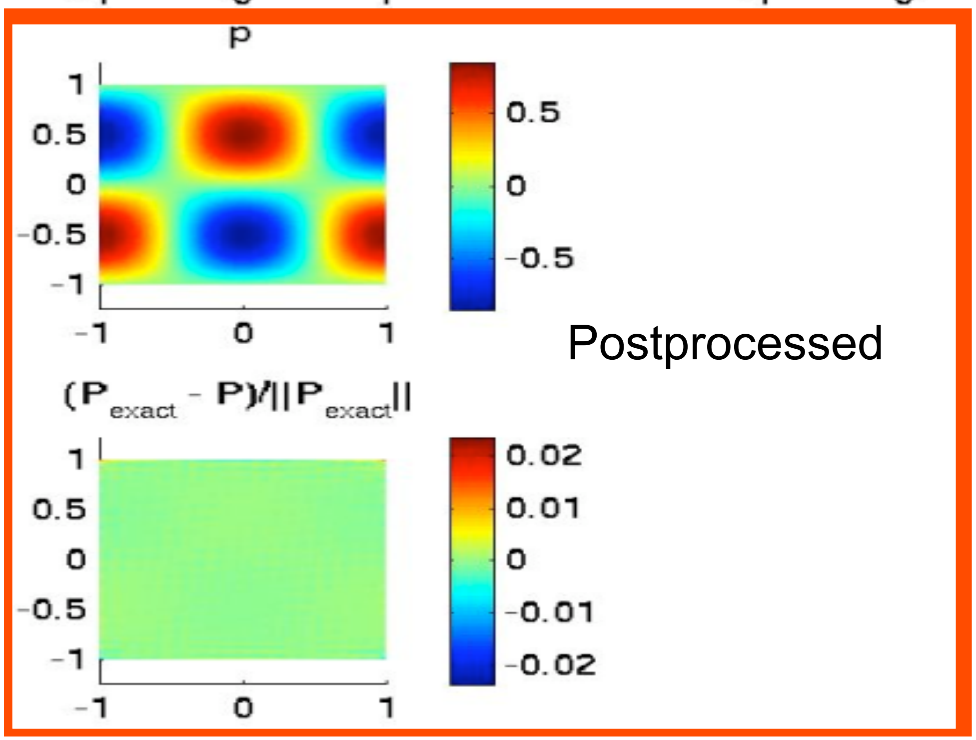
- Tau = 1
- Pressure postprocessing has significant effect



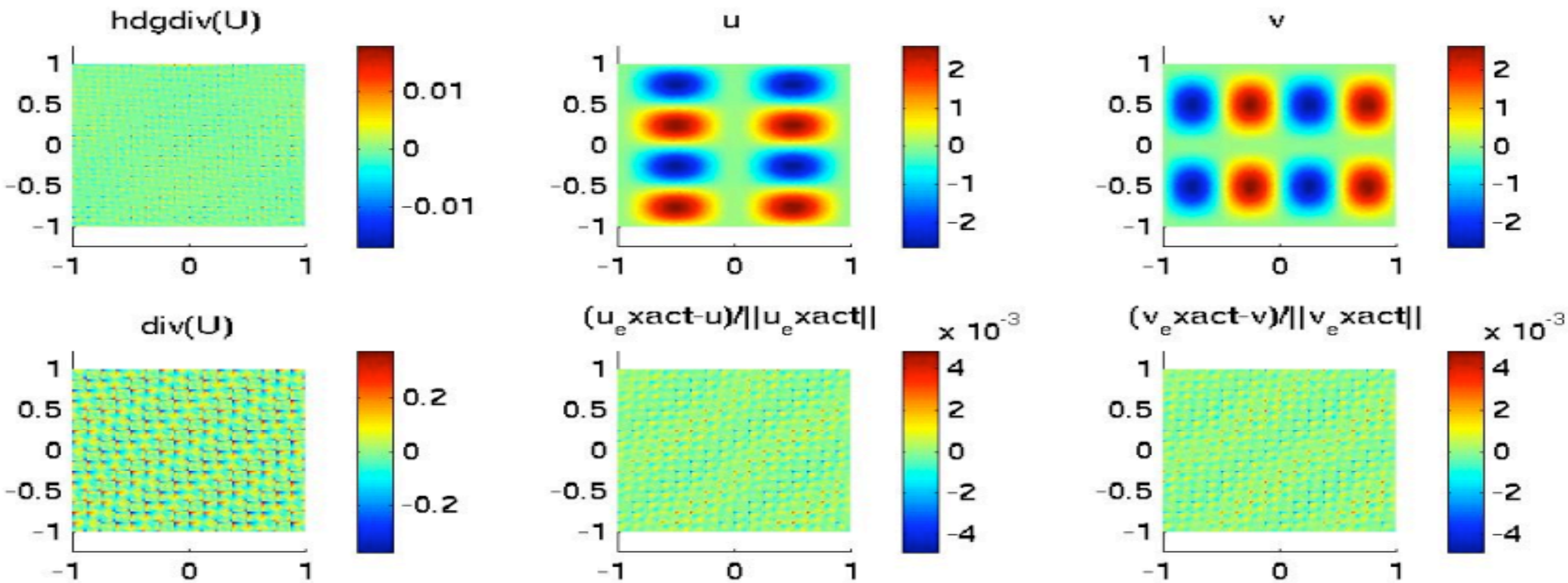
- Tau = 1
- Velocity postprocessing has significant effect
  - Reduce error by one order
- Divergence field very noisy

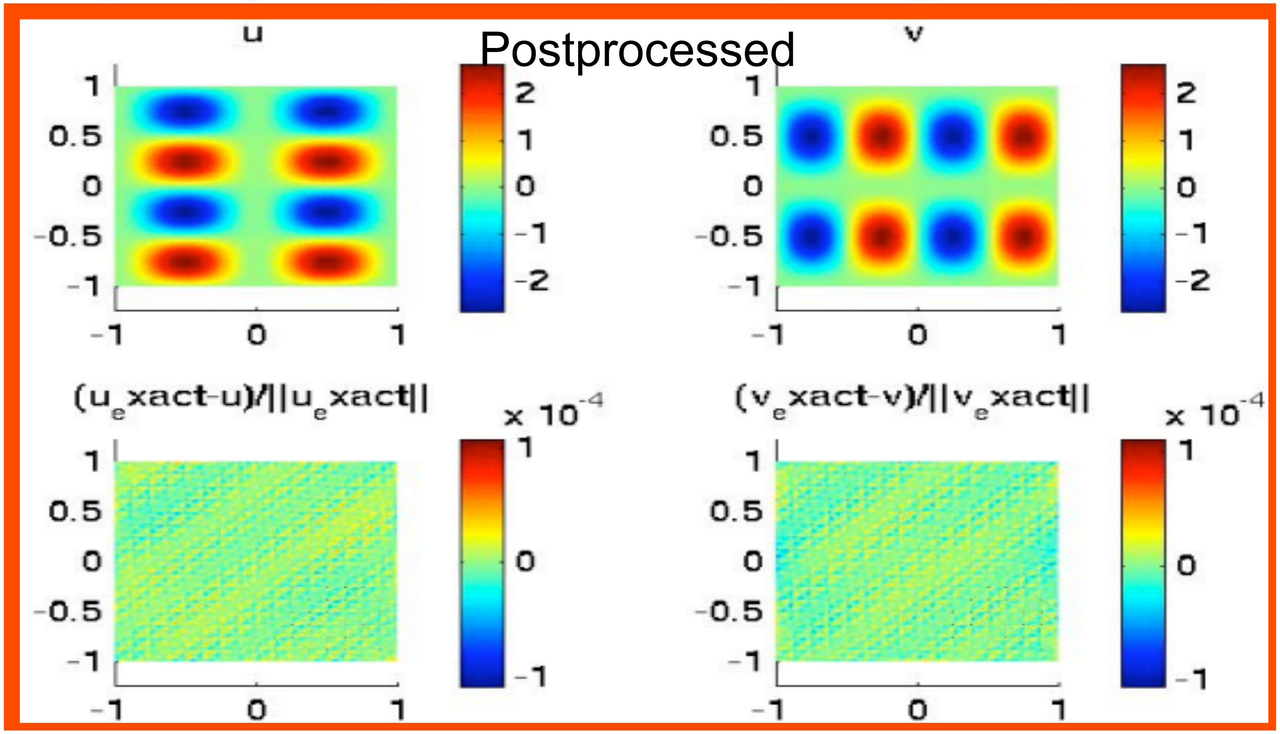






- Tau = 10,000
- Pressure postprocessing has no effect

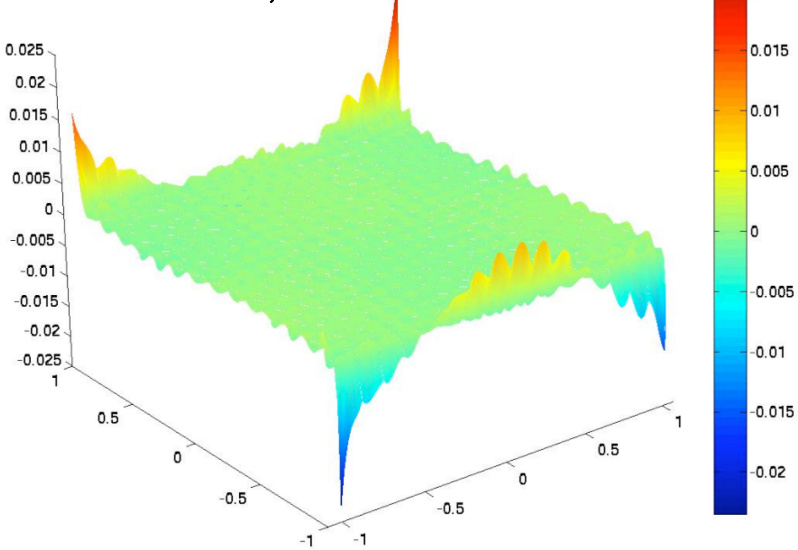




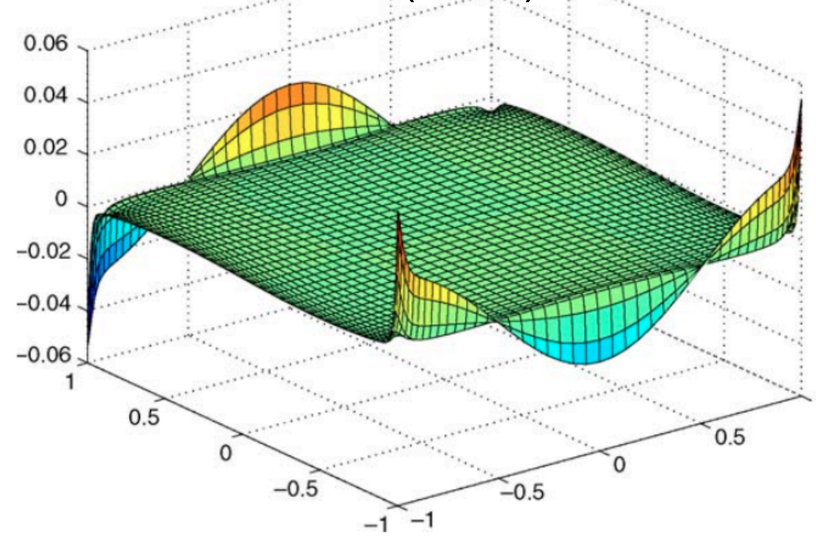
- Tau = 10,000
- Velocity postprocessing reduces error

Stokes: Pressure error comparison

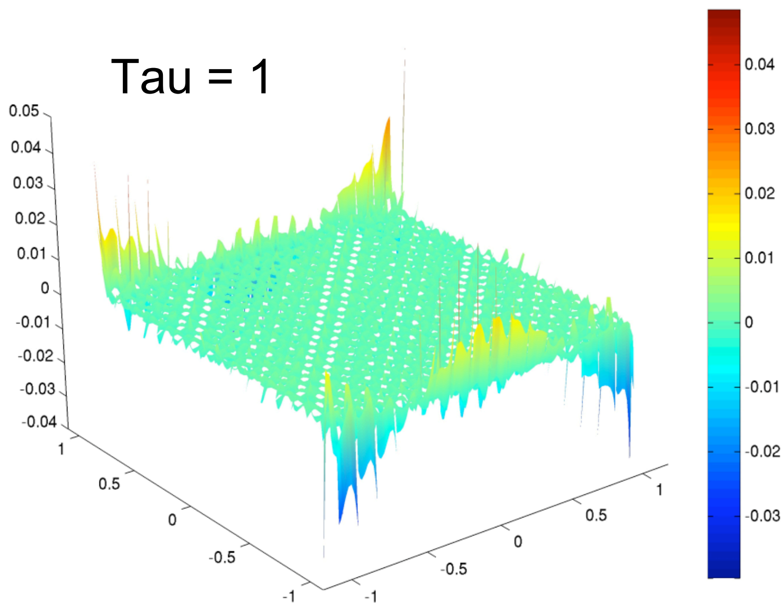
Tau = 10,000



Guermond (2006)



Tau = 1



- Tau = 10,000 pressure error compares well with literature
- Tau = 1 pressure error very noisy
- Numerical boundary Pressure boundary layer

Stokes: Convergence

Order	Grid	u	v	P	$\partial_x u$	$\partial_y v$	$\partial_x P$	$\partial_y P$
1	2-3	1.13	1.13	3.45	2.37	2.37	2.84	2.85
	3-5	0.67	0.67	0.58	-0.96	-0.96	-0.28	-0.28
	5-9	1.73	1.73	2.15	1.10	1.10	0.98	0.98
	9-17	1.88	1.88	3.08	1.76	1.76	2.03	2.03
2	2-3	0.48	0.49	1.61	-0.64	-0.64	0.56	0.56
	3-5	1.65	1.65	2.71	1.17	1.17	1.86	1.86
	5-9	2.42	2.42	3.49	2.13	2.13	2.58	2.59
	9-17	2.90	2.90	3.32	2.88	2.88	2.39	2.40
3	2-3	0.26	0.26	2.37	0.43	0.43	1.10	1.10
	3-5	2.76	2.76	4.05	2.36	2.36	2.68	2.68
	5-9	3.99	3.99	4.19	3.93	3.93	3.01	3.01
	9-17	3.88	3.88	4.56	3.87	3.87	3.84	3.65

Table 1: Convergence of Stokes test case before postprocessing

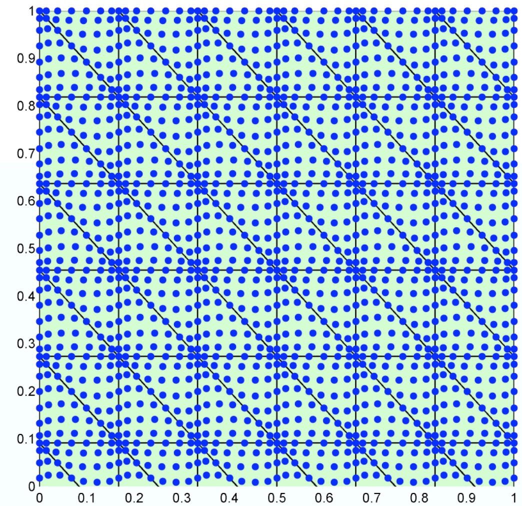
Stokes: Convergence

Order	Grid	u	v	P	$\partial_x u$	$\partial_y v$	$\partial_x P$	$\partial_y P$
1	2-3	2.76	2.76	3.19	2.37	2.37	2.86	2.85
	3-5	1.28	1.28	0.67	-0.19	-0.19	-0.29	-0.29
	5-9	2.76	2.76	2.06	1.57	1.57	0.98	0.98
	9-17	2.86	2.86	3.17	1.87	1.87	2.03	2.03
2	2-3	1.42	1.42	1.56	0.11	0.11	0.56	0.56
	3-5	3.19	3.19	2.76	1.92	1.92	1.86	1.86
	5-9	3.35	3.35	3.67	2.27	2.27	2.58	2.58
	9-17	3.92	3.92	3.25	2.88	2.88	2.39	2.40
3	2-3	1.31	1.31	2.38	-0.35	-0.35	1.10	1.10
	3-5	3.76	3.76	4.16	2.30	2.30	2.68	2.68
	5-7	4.98	4.98	4.16	3.92	3.92	3.01	3.01
	9-17	4.90	4.90	4.57	3.90	3.90	3.84	3.65

Table 1: Convergence of Stokes test case after postprocessing

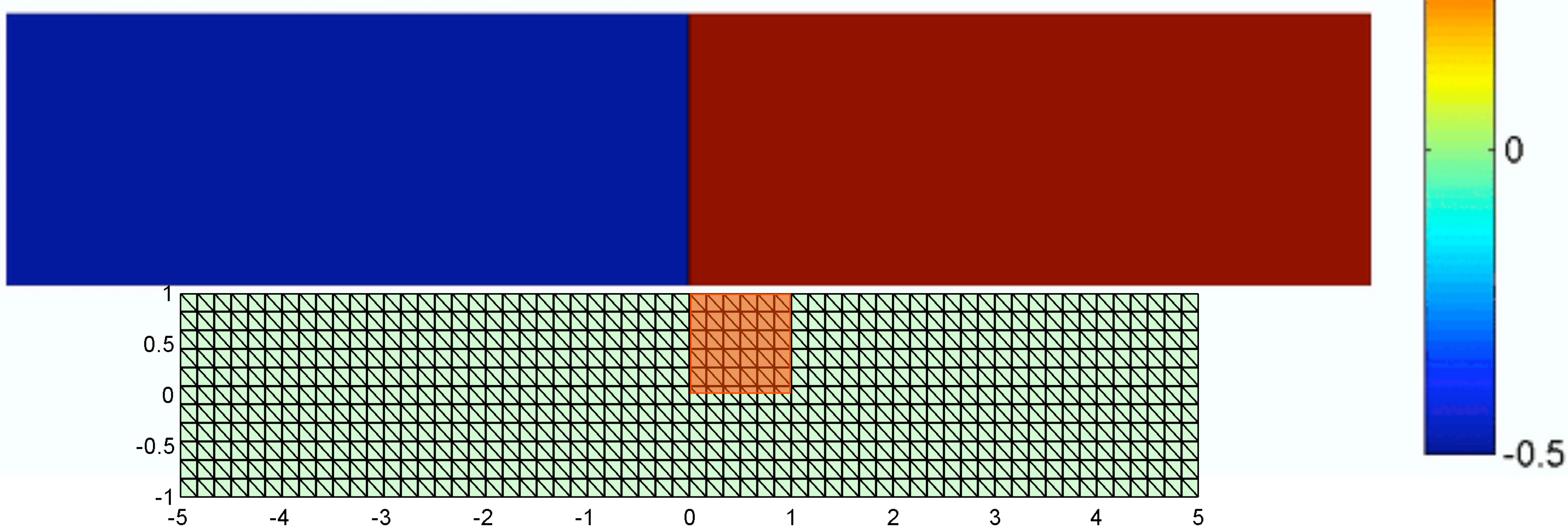
Lock Exchange Problem

- 37,000 DOF, 14,000 HDG unknowns
- 13.5 hrs
- 1320 Elements
- *p*=6
- Gr = 1.25×10^6, Sc=0.71



time: 0

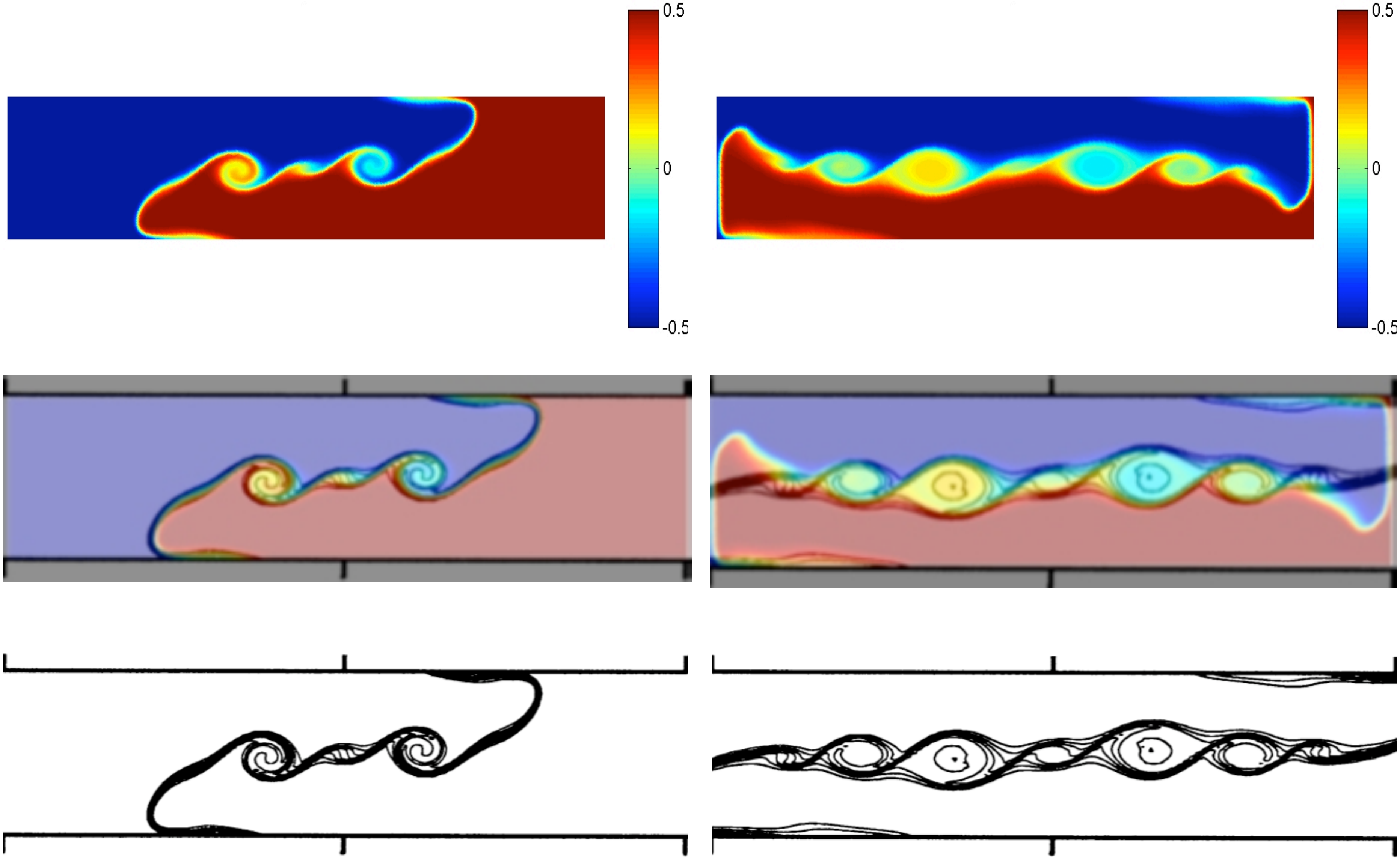
ρ



4. Hartel, C., Meinburg, E., and Freider, N. (2000). *Analysis and direct numerical simulations of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries.* J. Fluid. Mech, 418:189-212.

Lock Exchange Problem

Time = 5 Time = 10



Implementation issues: GPU's

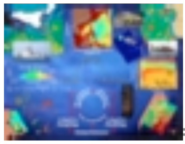
- Local HDG solver has three steps
  1. Build matrix and right-hand side vector: Local operation
  2. Solve global system of equations: Global operation
  3. Reconstruct solution: Local operation

- Local steps (1, 3) very well-suited to parallel architectures such as GPU
- Local read patterns very well-suited to coalesced memory access
- Write pattern less localized (to form matrix)

- Negates the problem with "expensive" local solutions

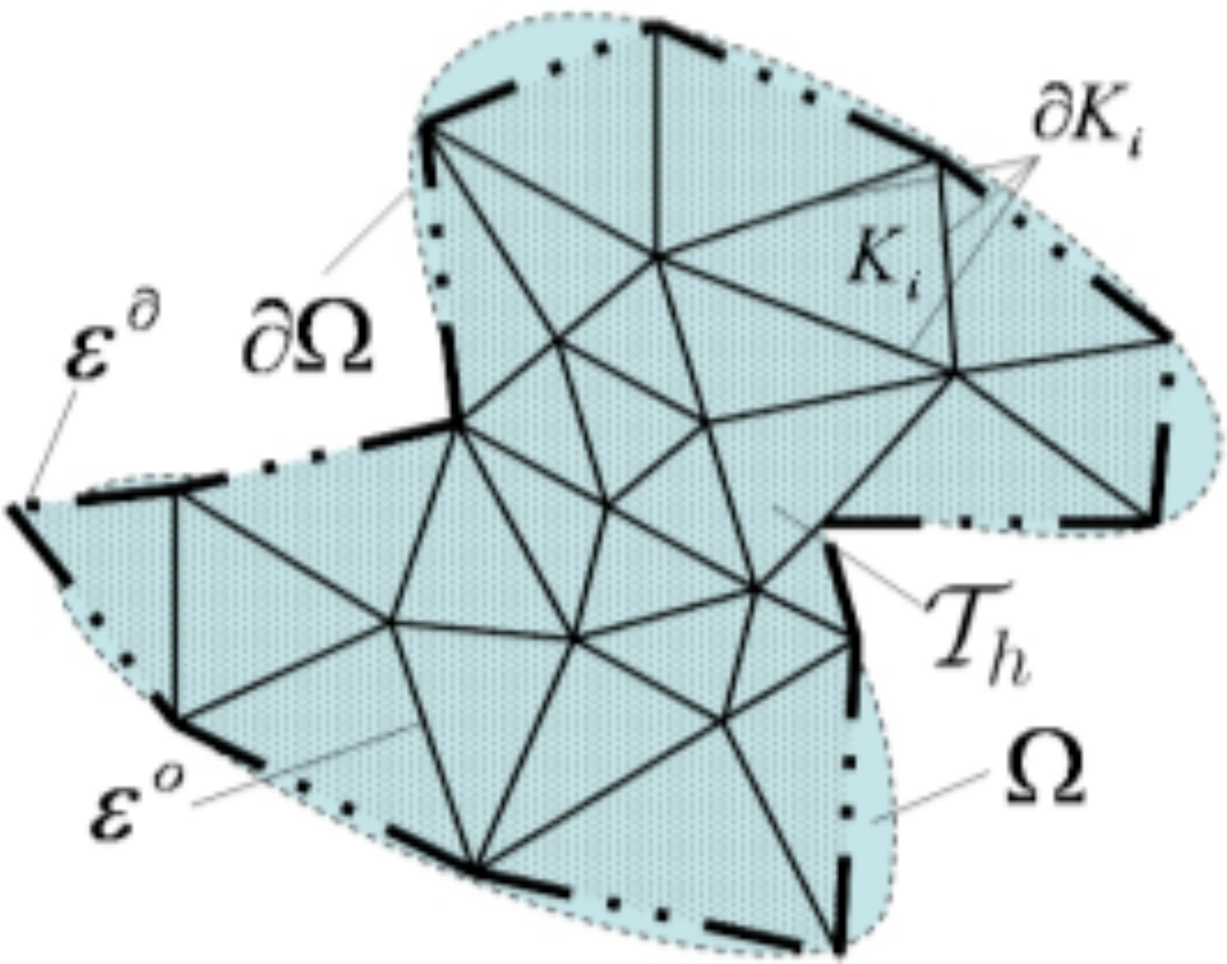
Future work

- Investigate methods to reduce tau in PPE
- Test different algorithms for solving Boussinesq Equations
- Can we do better?
  - Some ideas to further reduce global DOF
- Investigate adaptive/multiscale strategies
  - Octree-structured (2*p*+1 accurate post-processed solution)
  - Unstructured
  - Nesting
- 2D Non-hydrostatic dynamical studies
- Implement efficient 3D code
  - Exploit HDG parallelism: domain decomposition
    - MPI?
    - GPU?



Thank You!

Notation



∂K_i
K_i
$\mathcal{E}^\partial$
$\partial\Omega$
$\mathcal{T}_h$
$\mathcal{E}^o$
Ω

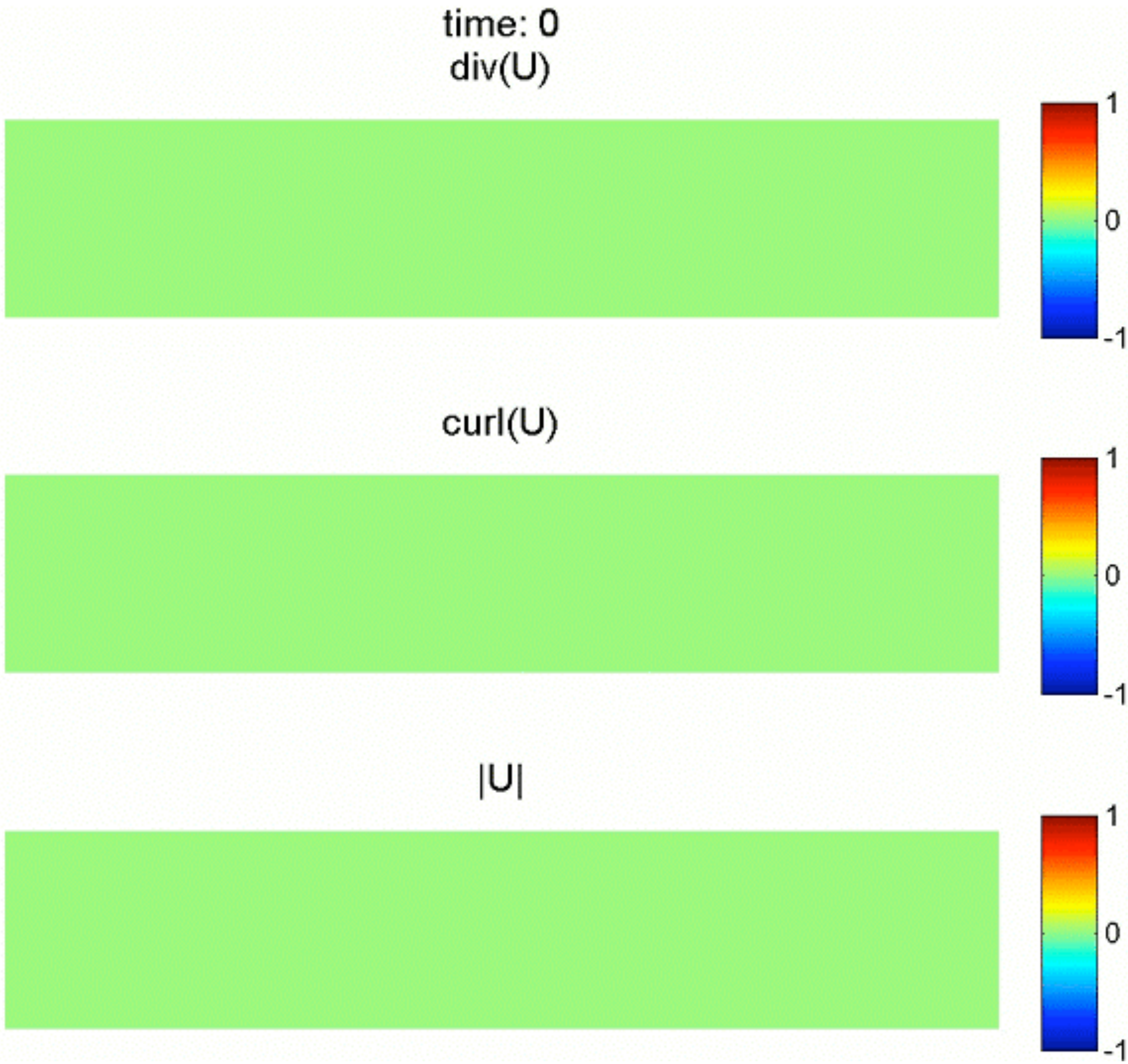
Stokes: Convergence tempora

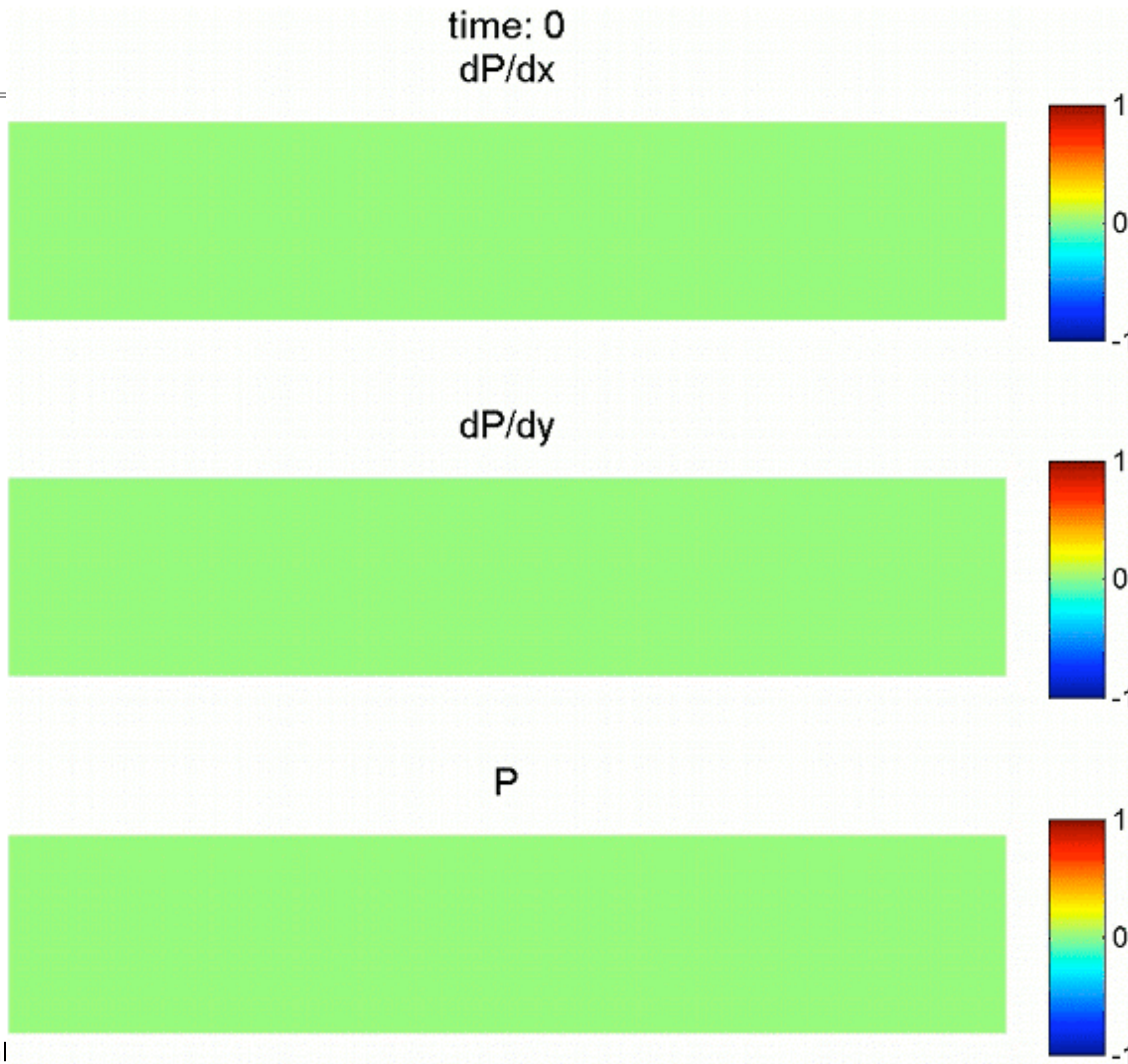
dt	u	v	P	$\partial_x u$	$\partial_y v$	$\partial_x P$	$\partial_y P$
0.05-0.025	2.11	2.12	1.74	2.06	1.53	1.25	0.58
0.025-0.0125	2.04	2.05	1.65	1.92	1.42	1.02	0.62
0.0125-0.00625	1.92	1.89	1.64	1.29	1.45	0.94	1.17
0.00625-0.003125	1.22	1.39	2.07	0.35	1.61	1.74	1.97

Table 1: Time convergence after postprocessing

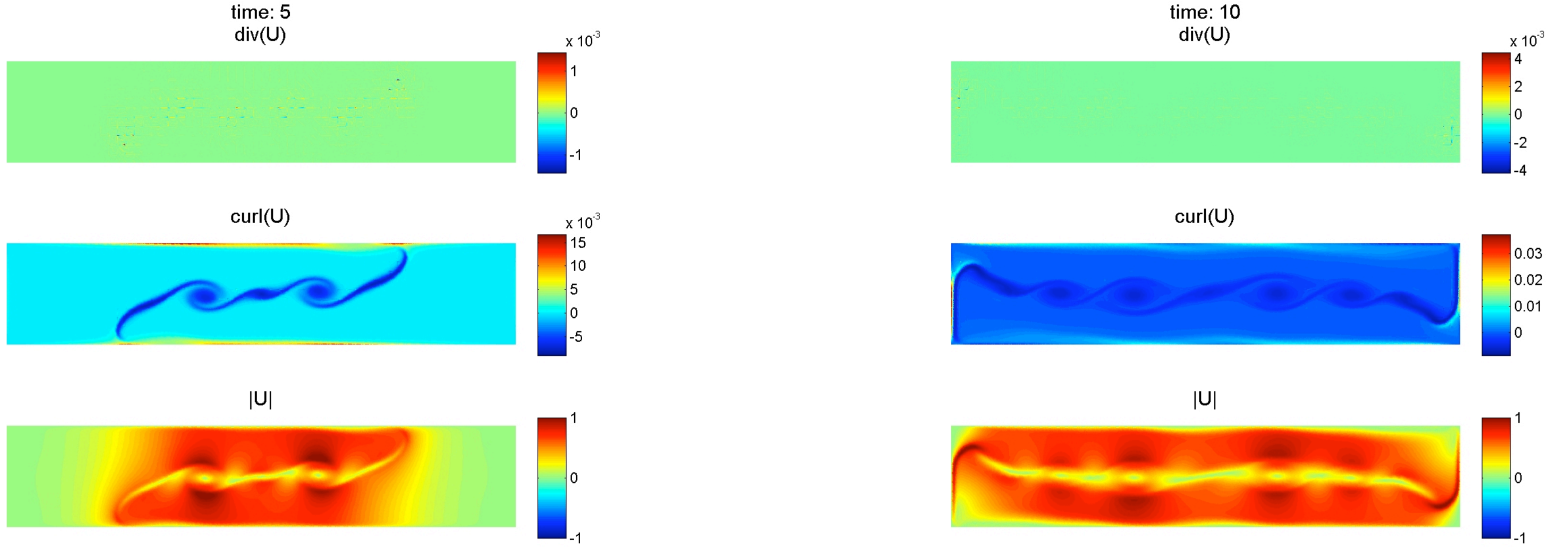
dt	u	v	P	$\partial_x u$	$\partial_y v$	$\partial_x P$	$\partial_y P$
0.05-0.025	1.81	1.93	1.74	2.06	1.53	1.25	0.58
0.025-0.0125	0.64	0.95	1.65	1.90	1.41	1.02	0.62
0.0125-0.00625	0.07	0.14	1.64	1.16	1.44	0.94	1.17
0.00625-0003125	0.04	0.04	2.10	0.27	1.50	1.74	1.97

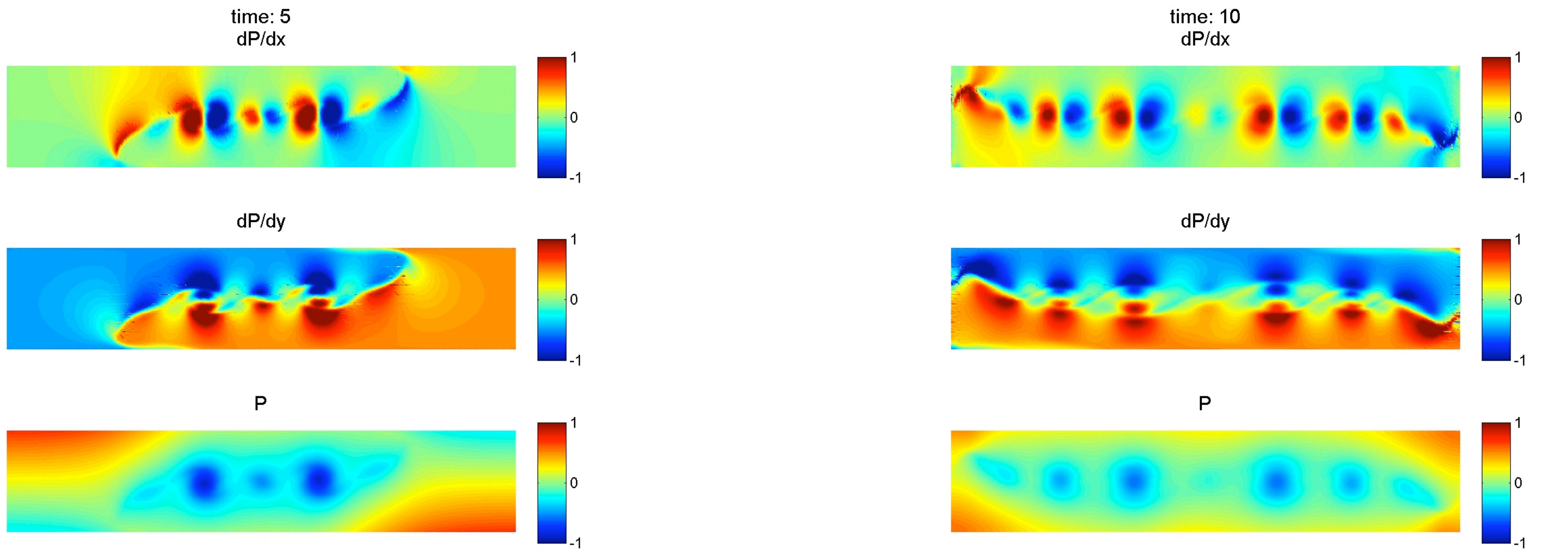
Table 1: Time convergence before postprocessing

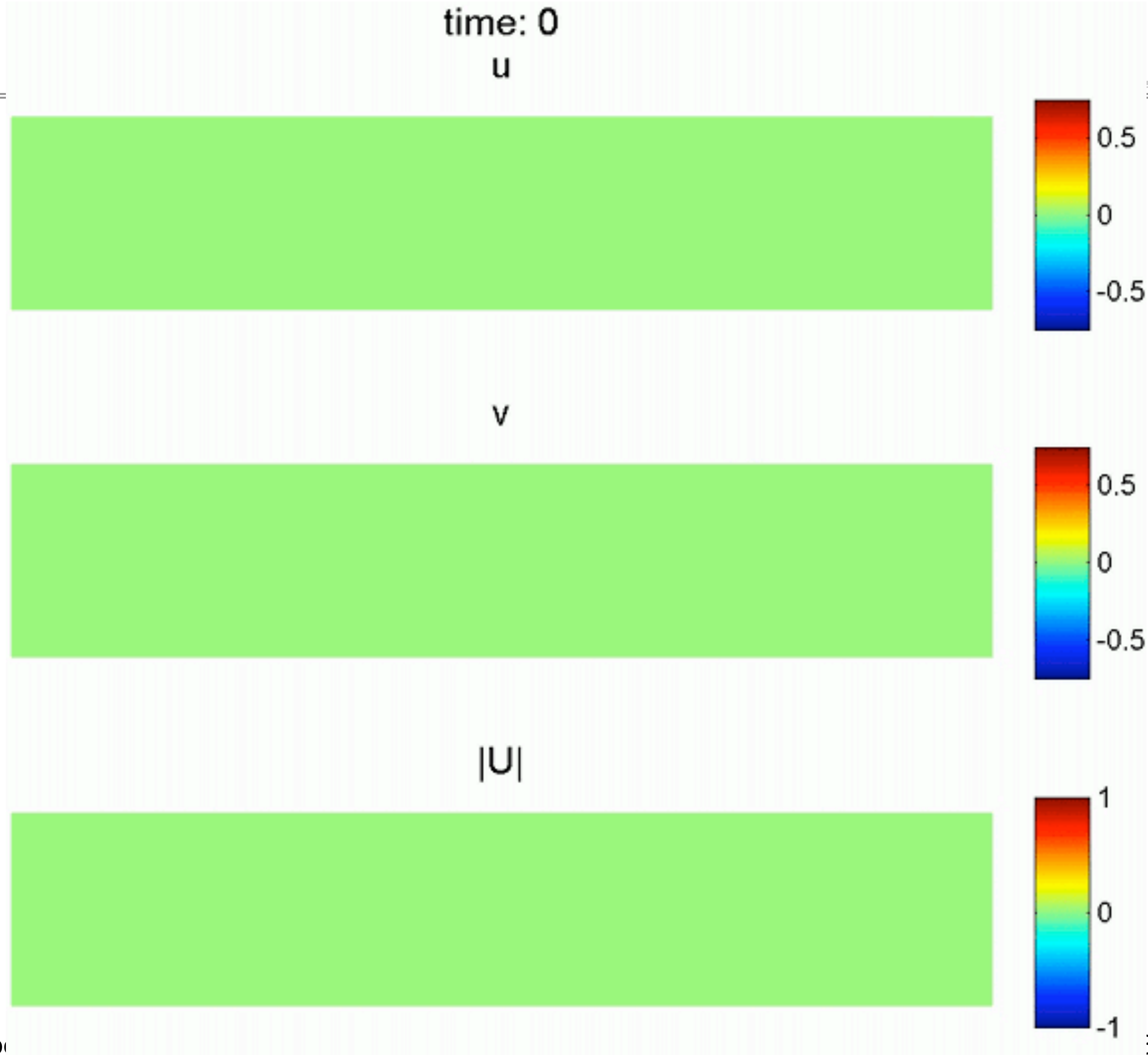

time: 0
div(U)
1
0
-1
curl(U)
1
0
-1
|U|
1
0
-1


time: 0
dP/dx
1
0
-1
dP/dy
1
0
-1
P
1
0
-1

Lock Exchange: Vorticity and Pressure






time: 0
u
0.5
0
-0.5
v
0.5
0
-0.5
|U|
1
0
-1

Preconditioners with HDG

- Have compared LDG descretization with HDG
- Solved matrixes using various preconditioners
- HDG at least 3x faster than LDG for p=4 using similar preconditioners

HDG, solve system 3 times

Slash	**GmresR**				**Qmr**				**BiCgstab**				**Summary**	
Time	***Precond.***	***Time***	***Iter.***	***Resid.***	***Precond.***	***Time***	***Iter.***	***Resid.***	***Precond.***	***Time***	***Iter.***	***Resid.***	***Min Time***	***Min MV***
dt=0.2		dt=0.2				dt=0.2				dt=0.2			dt=0.2	
0.4219	None	0.6563	188	9.0E-07	None	0.5000	142	9.2E-07	None	0.4531	90	5.0E-09	0.4531	180
0.4063	Upper	0.2656	107	8.2E-07	Upper	0.5156	58	6.1E-07	**Upper**	0.2813	30	4.6E-08	0.2656	**60**
0.4375	Lower	0.2656	105	7.4E-07	Lower	0.4844	55	6.4E-07	**Lower**	0.2500	30	2.4E-08	0.2500	***60***
0.4375	Jacobi	0.3438	131	9.7E-07	Jacobi	0.3750	86	6.6E-07	Jacobi	0.3750	60	9.3E-10	0.3438	120
0.4375	ILU0	0.1563	88	9.5E-07	ILU0	0.3750	31	8.1E-07	**ILU0**	0.3438	30	1.8E-12	0.1563	**60**
0.4375	BlockILU	0.2969	119	9.7E-07	BlockILU	0.5000	73	7.4E-07	**BlockILU**	0.2500	30	7.2E-07	0.2500	**60**
0.4375	BlockJacobi	0.4063	119	9.7E-07	BlockJacobi	0.7344	73	8.0E-07	**BlockJacobi**	0.3594	30	7.2E-07	0.3594	**60**
0.4375	BlockJacobi2	0.6563	119	9.7E-07	BlockJacobi2	1.2344	73	8.0E-07	**BlockJacobi2**	2.0000	30	7.2E-07	0.6563	**60**
0.4219	GS	0.3750	99	8.9E-07	GS	0.54688	49	4.6E-07	GS	3.75	30	3.2E-09	0.3750	**60**
0.4219	Multigrid	3.2656	95	5.4E-07	Multigrid	28.0156	26	2.4E-03	**Multigrid**	86.7969	30	3.1E-10	3.2656	**60**

LDG, solve system once

dt=0.2		dt=0.2				dt=0.2				dt=0.2			dt=0.2	
0.5938	None	0.4688	66	9.6E-07	None	0.8438	50	7.3E-07	None	0.4063	30	1.2E-07	0.4063	60
0.6094	Upper	0.3438	42	7.5E-07	Upper	1.0313	32	6.1E-07	Upper	0.4219	20	8.2E-07	0.3438	40
0.5781	Lower	0.2969	37	9.3E-07	Lower	0.7031	26	3.9E-07	Lower	0.4063	20	1.2E-08	0.2969	37
0.5938	Jacobi	0.3750	49	7.3E-07	Jacobi	0.7344	43	9.5E-07	Jacobi	0.5313	30	2.5E-07	0.3750	49
0.5938	ILU0	0.1563	28	4.9E-07	**ILU0**	0.6094	10	5.1E-07	**ILU0**	0.2969	10	4.4E-11	0.1563	**20**
0.6094	BlockILU	0.5781	41	8.9E-07	BlockILU	1.4688	35	3.6E-07	BlockILU	1.0938	30	3.6E-09	0.5781	41
0.5938	BlockJacobi	0.5938	41	8.9E-07	BlockJacobi	1.6094	35	3.1E-07	BlockJacobi	1.1094	30	3.6E-09	0.5938	41
0.6094	BlockJacobi2	1.0625	41	8.9E-07	BlockJacobi2	2.5000	35	3.1E-07	BlockJacobi2	9.4844	30	3.6E-09	1.0625	41
0.5938	GS	0.7344	33	7.3E-07	GS	1.92188	19	7.7E-07	GS	19.84375	20	3.7E-10	0.7344	33
0.6094	Multigrid	1.0781	28	8.7E-07	Multigrid	10.9219	43	1.4E-01	**Multigrid**	29.9063	10	1.0E-12	1.0781	**20**
													0.1563	**20**

Test Case Examples - Steady advection-diffusion