Verification of Moving Mesh Discretizations

Krzysztof J. Fidkowski

Aerospace Engineering

University of Michigan ■ College of Engineering

High Order CFD Workshop

Kissimmee, Florida

January 6, 2018

How can we verify moving mesh results?

Goal:

Demonstrate accuracy of flow solutions on moving domains and meshes

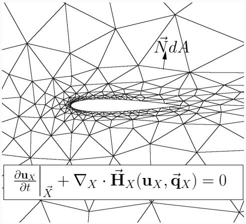
Considerations:

- Inviscid versus viscous term discretization
- Solution field norm versus scalar outputs
- Integration quadrature requirements
- Mapping singularities
- Geometric conservation
- Boundary condition consistency

The arbitrary Lagrangian-Eulerian (ALE) method

ALE Idea: solve transformed PDE on a static reference domain

Reference domain: $\vec{X}, \mathbf{u}_X, \vec{\mathbf{H}}_X$



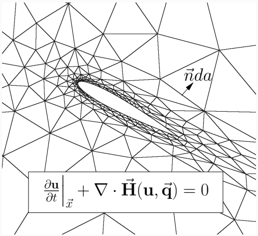
$\Rightarrow$

Mapping

$$
\begin{aligned}
\vec{X}, t &\Rightarrow \vec{x}(\vec{X}, t) \\
\mathcal{G} &= \frac{\partial \vec{x}}{\partial \vec{X}} \\
g &= \det(\mathcal{G}) \\
\mathbf{u}_X &= g\mathbf{u} \\
\vec{\mathbf{q}}_X &= g\mathcal{G}^T\vec{\mathbf{q}} \\
\vec{v}_G &= \frac{\partial \vec{x}}{\partial t} \\
\vec{\mathbf{H}}_X &= g\mathcal{G}^{-1}\vec{\mathbf{H}} - \mathbf{u}_X\mathcal{G}^{-1}\vec{v}_G \\
\vec{n}da &= g\mathcal{G}^{-T}\vec{N}dA \\
\vec{N}dA &= g^{-1}\mathcal{G}^T\vec{n}da
\end{aligned}
$$

$\Rightarrow$

Physical domain: $\vec{x}, \mathbf{u}, \vec{\mathbf{H}}$



Key definitions

$\vec{X}$	=	reference-domain coordinates	$\mathbf{u}$	=	physical state
$\vec{x}$	=	physical-domain coordinates	$\mathbf{u}_X$	=	reference state
g	=	determinant of Jacobian matrix	$\vec{\mathbf{H}}$	=	physical flux vector
$\vec{v}_G$	=	grid velocity, $\partial\vec{x}/\partial t$	$\vec{\mathbf{H}}_X$	=	reference flux vector

The transformed equations

- Integrate the evolution PDE over a time-varying volume $v(t)$, apply the divergence theorem, transform terms to reference space, and apply the divergence theorem again,

$$\left.\frac{\partial \mathbf{u}_X}{\partial t}\right|_{\vec{X}} + \nabla_X \cdot \vec{\mathbf{H}}_X(\mathbf{u}_X, \nabla_X \mathbf{u}_X) \quad = \quad \mathbf{0},$$

  where
  - $\mathbf{u}_X = g\mathbf{u}$
  - $\vec{\mathbf{H}}_X = g\mathcal{G}^{-1}\vec{\mathbf{H}} - \mathbf{u}_X\mathcal{G}^{-1}\vec{v}_G$
  - ∇_X is the gradient with respect to the reference coordinates
- The transformed flux, $\vec{\mathbf{H}}_X$, separates into inviscid and viscous contributions,

$$\vec{\mathbf{H}}_X = \vec{\mathbf{F}}_X + \vec{\mathbf{G}}_X, \quad \vec{\mathbf{F}}_X = g\mathcal{G}^{-1}\vec{\mathbf{F}} + \mathbf{u}_X\mathcal{G}^{-1}\vec{v}_G, \quad \vec{\mathbf{G}}_X = g\mathcal{G}^{-1}\vec{\mathbf{G}}$$

ALE implementation

- To minimize code intrusion, express the reference-space fluxes in terms of the physical fluxes

$$\begin{aligned}\vec{\mathbf{F}}_X &= g\mathcal{G}^{-1}\vec{\mathbf{F}} - \mathbf{u}_X\mathcal{G}^{-1}\vec{v}_G = g\mathcal{G}^{-1}\left(\vec{\mathbf{F}} - \mathbf{u}\vec{v}_G\right)\\ \vec{\mathbf{G}}_X &= g\mathcal{G}^{-1}\vec{\mathbf{G}} = -g\mathcal{G}^{-1}\mathbf{K}\vec{\mathbf{q}} = -\underbrace{\mathcal{G}^{-1}\mathbf{K}\mathcal{G}^{-T}}_{\mathbf{K}_X}\vec{\mathbf{q}}_X\end{aligned}$$

- Linearizations must be performed w.r.t reference states
- Viscous stabilization does not change (same as physical)!
- Boundary conditions (e.g. no slip) must incorporate the boundary velocity, $\vec{v}_G$.

Outputs

- General form of boundary flux integral outputs:

$$J = \int_{\partial\Omega} \mathbf{o}^T \widehat{\mathbf{H}} \cdot \vec{n}\, da,$$

- $\mathbf{o} \in \mathbb{R}^s$ is a weight function that defines the output
- For 2D Navier-Stokes:

$$\begin{array}{lcl} \mathbf{o} = [0; \cos\alpha; \sin\alpha; 0] & \Rightarrow & \text{Drag} \\ \mathbf{o} = [0; -y; x; 0] & \Rightarrow & \text{Moment about origin} \\ \mathbf{o} = [0; v_{Gx}; v_{Gy}; 0] & \Rightarrow & \text{Power} \end{array}$$

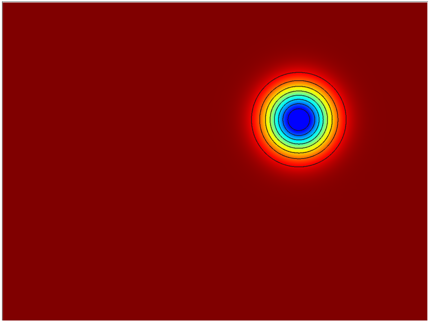
- For example, if $\vec{v}_G = \vec{v}_0 + \vec{\omega} \times \vec{r}$, where $\vec{r}$ is a position vector relative to some origin of rotation,

$$J = \int_{\partial\Omega} (\vec{v}_0 + \vec{\omega} \times \vec{r}) \cdot \vec{f}_{\text{surf}}\, da = \underbrace{\int_{\partial\Omega} \vec{v}_0 \cdot \vec{f}_{\text{surf}}\, da}_{\vec{v}_0 \cdot \vec{F}_{\text{net}}} + \underbrace{\int_{\partial\Omega} \vec{\omega} \cdot (\vec{r} \times \vec{f}_{\text{surf}})\, da}_{\vec{\omega} \cdot \vec{T}_{\text{net}}}$$

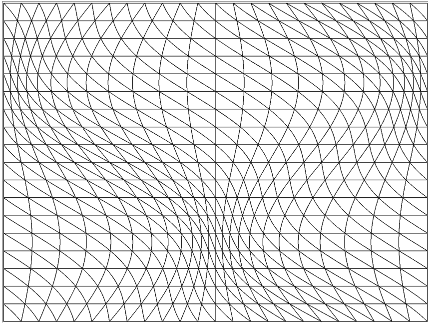
Inviscid field norm checK: Euler vortex

- Analytical vortex solution to the Euler equations
- Sinusoidal interior mesh deformation for testing

$$
\begin{aligned}
x(t) &= X + 2\sin(2\pi X/20)\sin(2\pi Y/15)\sin(2\pi t) \\
y(t) &= Y + 1.5\sin(2\pi X/20)\sin(2\pi Y/15)\sin(4\pi t)
\end{aligned}
$$



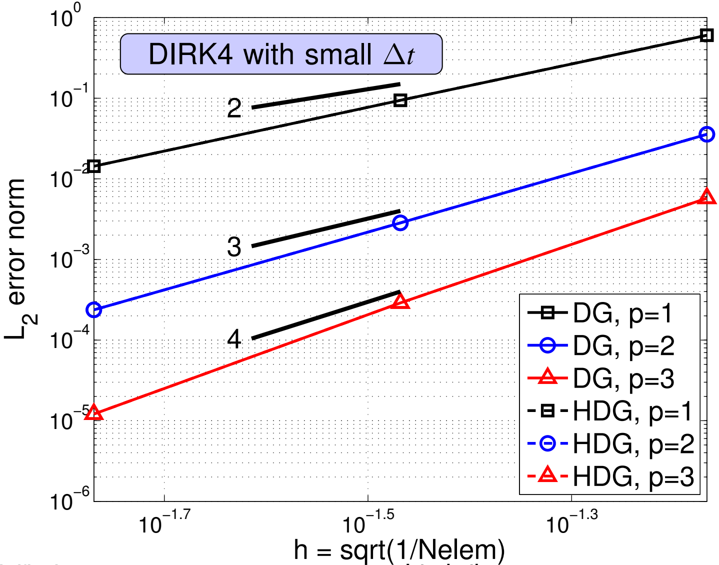
Pressure at final time



Deformed mesh

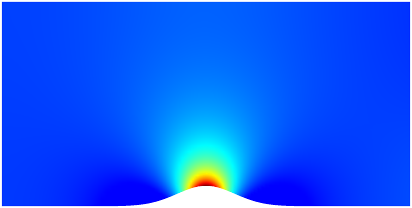
Euler vortex results

Look at L_2 error norm at end of the simulation

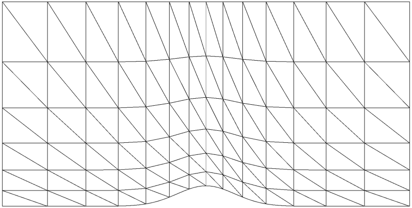


Inviscid boundary output: bump

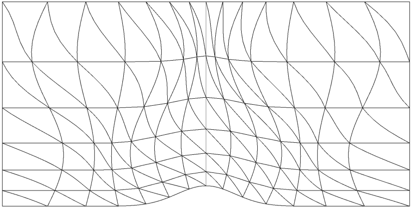
M=0.2. structured mesh, sinusoidal mesh motion



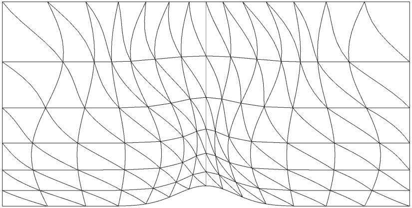
Mach contours



mesh at $t = 0.0$



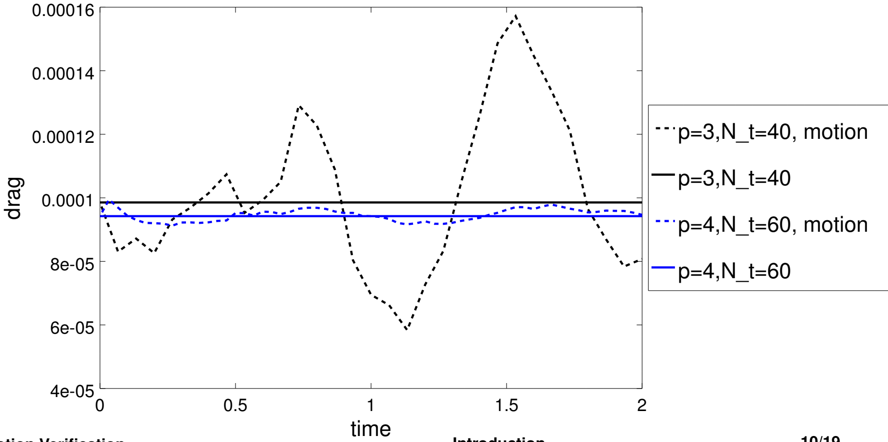
mesh at $t = 0.5$



mesh at $t = 1.5$

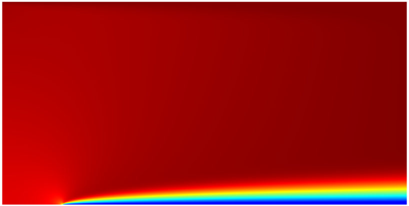
Inviscid boundary output: bump

- Monitor instantaneous drag output on the bump
- Compare moving mesh results to a static mesh
- Coarse ($p = 3$, $N_t = 40$) and fine ($p = 4$, $N_t = 60$) solutions
- **Verification:** error decreases with increasing resolution

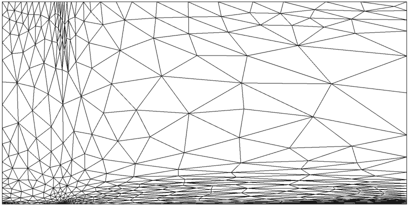


Viscous boundary output: flat plate

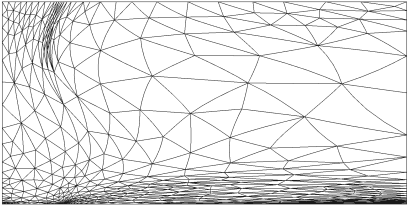
M=0.2, $Re = 1000$, adapted mesh, sinusoidal mesh motion



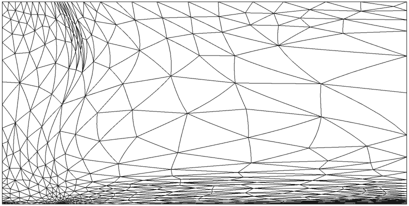
Mach contours



mesh at $t = 0.0$



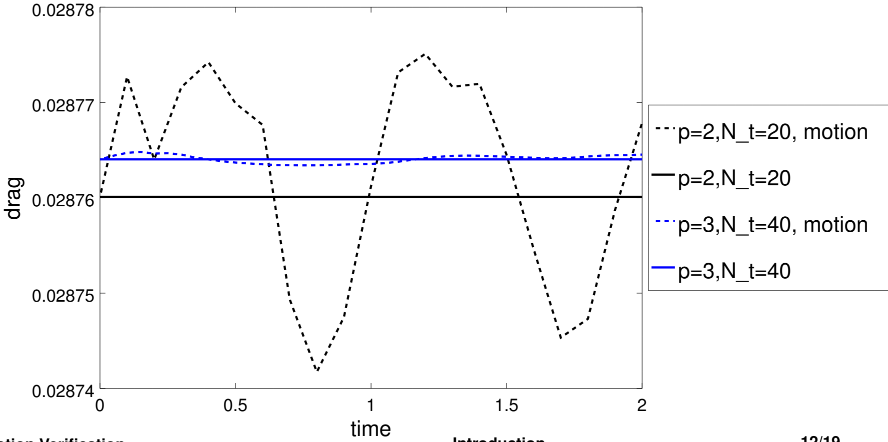
mesh at $t = 0.5$



mesh at $t = 1.5$

Viscous boundary output: flat plate

- Monitor instantaneous drag output on the flat plate
- Compare moving mesh results to a static mesh
- Coarse ($p = 2$, $N_t = 20$) and fine ($p = 3$, $N_t = 40$) solutions
- **Verification:** error decreases with increasing resolution

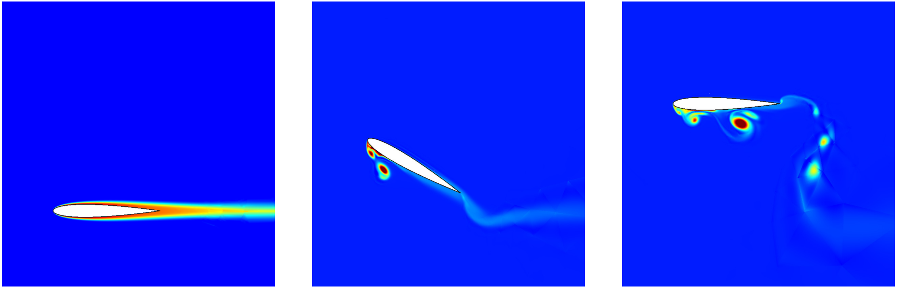


NACA 0012 airfoil in pitch/plunge motion

- Smooth plunge, $h(t)$, and pitch, $\theta(t)$, motion, $0 \leq t \leq 1$

$$h(t) = \frac{1 - \cos \pi t}{2}, \quad \theta(t) = \frac{\pi}{6} \frac{1 - \cos 2\pi t}{2}.$$

- NACA 0012, $Re_c = 5000$, $M = 0.2$, steady-state IC

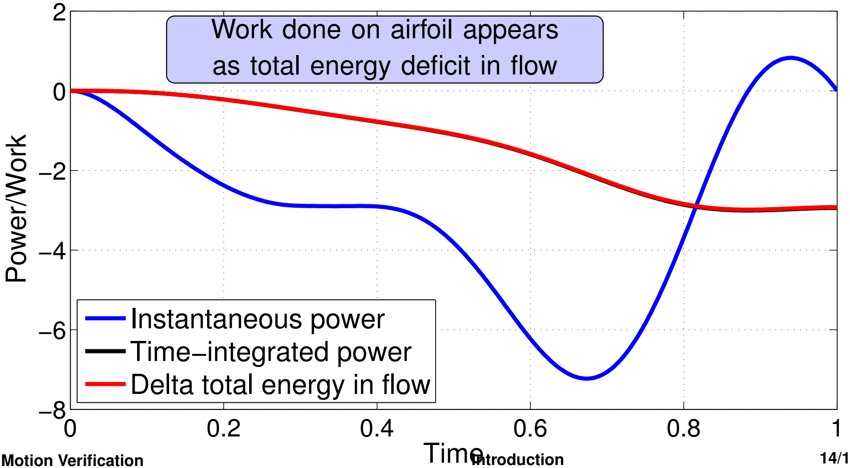


$t = 0$ $\qquad$ $t = 0.5$ $\qquad$ $t = 1.0$

Outputs

$$\text{Impulse} = \int_0^1 F_y\,dt = \int_0^1 \int_{\text{airfoil}} \hat{y} \cdot \vec{f}_{\text{surf}}\,ds\,dt$$

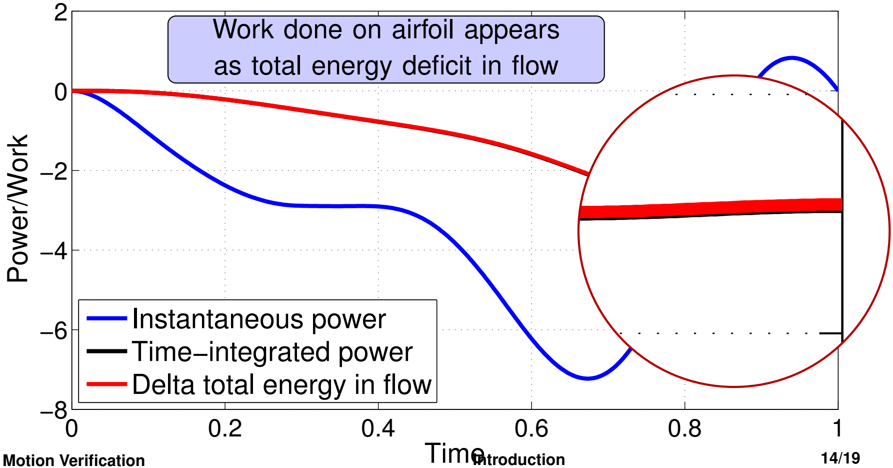
$$\text{Work} = \int_0^1 \left[\vec{v}_0 \cdot \vec{F}_{\text{net}} + \vec{\omega} \cdot \vec{T}_{\text{net}}\right] dt = \int_0^1 \int_{\text{airfoil}} \vec{v}_G \cdot \vec{f}_{\text{surf}}\,ds\,dt$$



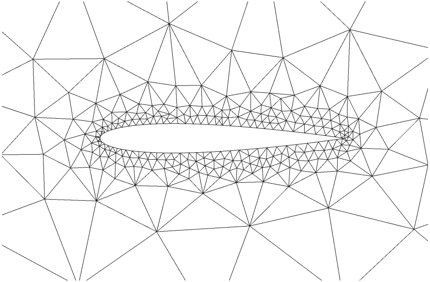
Outputs

$$\text{Impulse} = \int_0^1 F_y \, dt = \int_0^1 \int_{\text{airfoil}} \hat{y} \cdot \vec{f}_{\text{surf}} \, ds \, dt$$

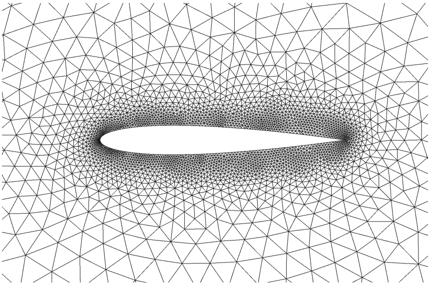
$$\text{Work} = \int_0^1 \left[\vec{v}_0 \cdot \vec{F}_{\text{net}} + \vec{\omega} \cdot \vec{T}_{\text{net}} \right] dt = \int_0^1 \int_{\text{airfoil}} \vec{v}_G \cdot \vec{f}_{\text{surf}} \, ds \, dt$$



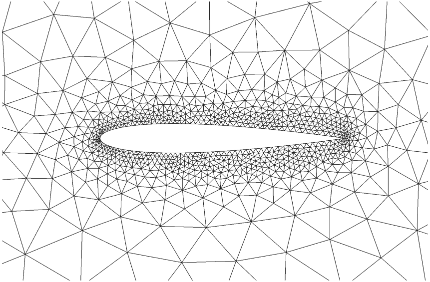
Worskhop runs meshes: quasi-uniform refinement



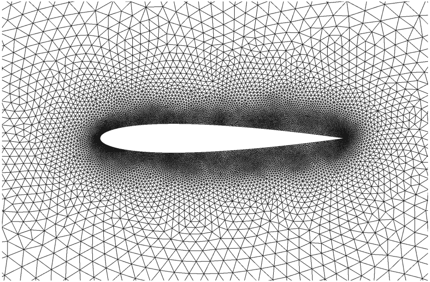
Mesh 0: 664 elements (farfield at $100c$)



Mesh 2: 7389 elements



Mesh 1: 2137 elements



Mesh 3: 26771 elements

Worskhop runs: details

- Considered spatial orders $p = 1, 2, 3, 4, 5$
- ESDIRK5 (5th order) time stepping
- Number of time steps depends on mesh and order:

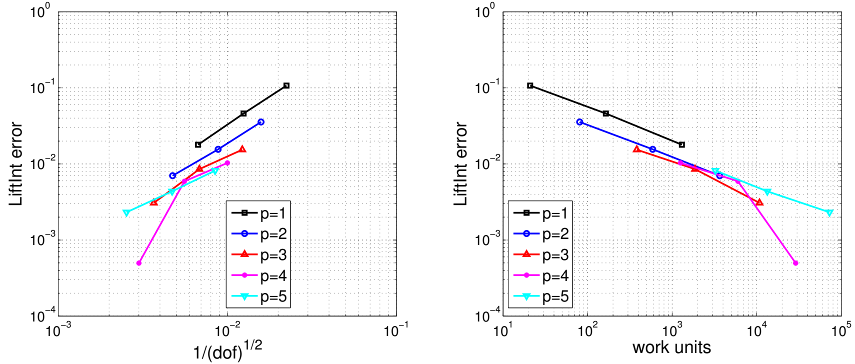
	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
Mesh 0	30	60	120	180	240
Mesh 1	60	120	180	240	320
Mesh 2	120	180	240	320	400

(these values were determined empirically)

- Error computed relative to a “truth” solution from a $p = 5, N_t = 800$, ESDIRK5 run on Mesh 3

Worskhop result: case 1, pure heaving

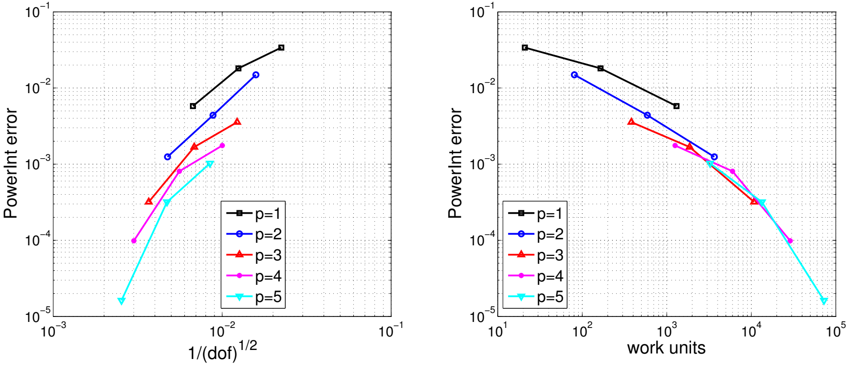
$$h(t) = \frac{t^2(3-t)}{4}, \quad \theta(t) = 0$$



Truth lift integral = -2.33124734

Worskhop result: case 1, pure heaving

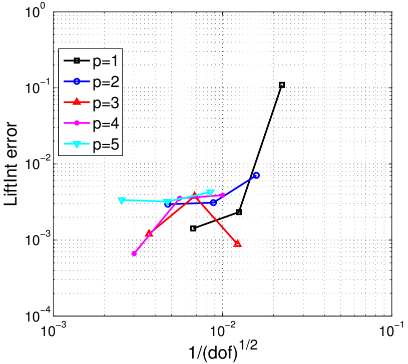
$$h(t) = \frac{t^2(3-t)}{4}, \quad \theta(t) = 0$$

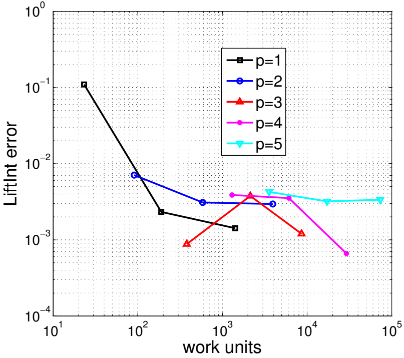


Truth power integral = -1.3834990

Worskhop result: case 2, flow aligning

$$h(t) = \frac{t^2(3-t)}{4}, \quad \theta(t) = \frac{\pi}{3}t^2(t^2 - 4t + 4)$$

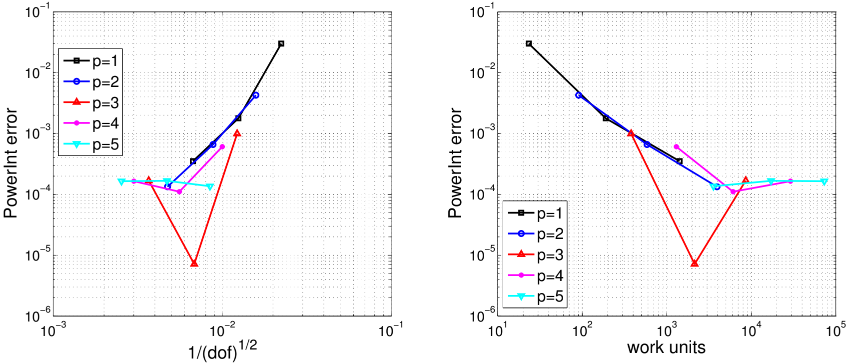




Truth lift integral = 0.610038583

Worskhop result: case 2, flow aligning

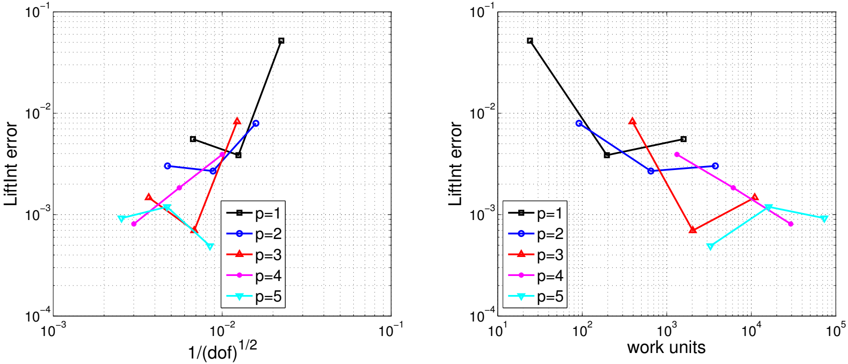
$$h(t) = \frac{t^2(3-t)}{4}, \quad \theta(t) = \frac{\pi}{3}t^2(t^2 - 4t + 4)$$



Truth power integral = -0.204899808

Worskhop result: case 3, energy extracting

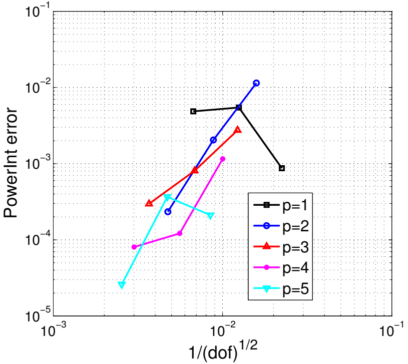
$$h(t) = \frac{t^2}{16}(-8t^3 + 51t^2 - 111t + 84), \quad \theta(t) = \frac{4\pi}{9}t^2(t^2 - 4t + 4)$$

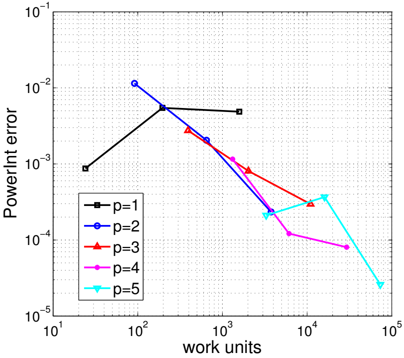


Truth lift integral = 1.6704741181

Worskhop result: case 3, energy extracting

$$h(t) = \frac{t^2}{16}(-8t^3 + 51t^2 - 111t + 84), \quad \theta(t) = \frac{4\pi}{9}t^2(t^2 - 4t + 4)$$





Truth power integral = 0.3637653632