An Examination of the Effects of Overset Interpolation Accuracy in the Context of a High-Order CFD Solver

Norman F. Foster

Applied Research Laboratory
The Pennsylvania State University

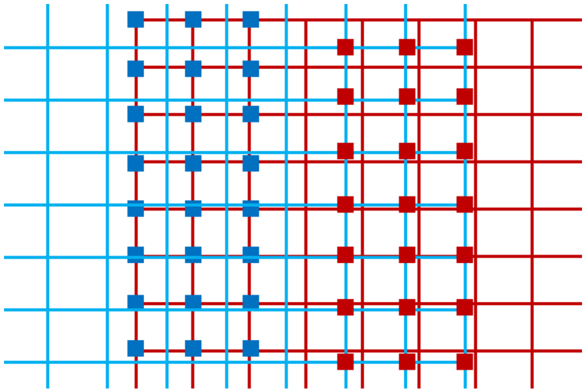
Background

- High order numerics (5th order and higher) has become more common in research, and is beginning to appear in industrial uses of CFD.
- Overset gridding and overset solver techniques is one of several enabling technologies to evaluating flow field around complex (and moving) geometries.
- Use of overset is still a tool for "experts" (or researchers), but is slowly penetrating into industry

Purpose of the Presentation

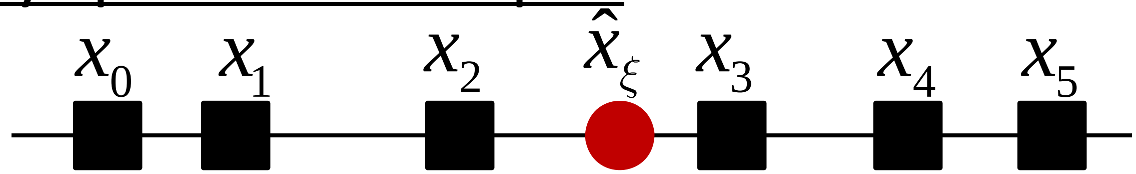
- Ignite discussion of the often overlooked issue of interpolation accuracy and its effects with high order solvers
- Present interesting findings through simple, and not so simple, computational examples

Limitations of this Study



- CFD on block structured overset meshes (for use in FV/FD codes)
- Two solvers:
  - In-house Penn State code (PSU)
  - OVERFLOW 2.2 (OF)
- Overset domain connectivity determined using Suggar++ and read into the solvers using DiRTLib
- Explicit isoparametric Lagrangian interpolation method to determine donor weights

For generally spaced source pts:

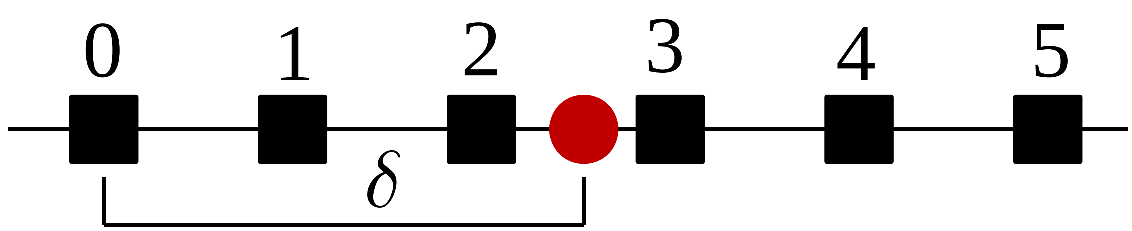


Example: N=6

Standard Lagrangian Interpolation (SLI)

$$f(\hat{x}) = \sum_{i=0}^{N-1} P_i(\hat{x}) f(x_i) \qquad P_i(\hat{x}) = \prod_{j=0, j\neq i}^{N-1} \frac{(\hat{x} - x_j)}{(x_i - x_j)}$$

If source pts are equally spaced, Δx=constant:

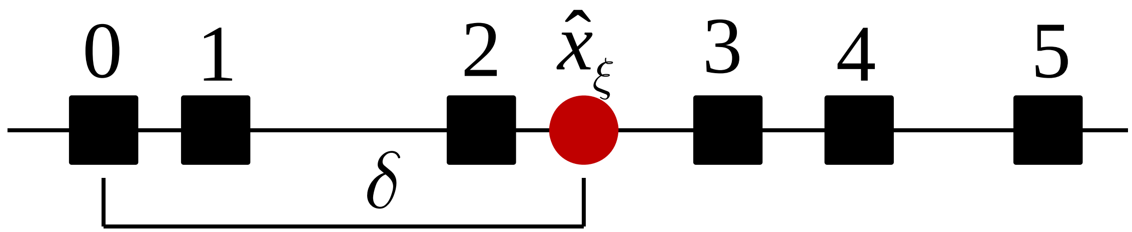


Example: N=6

Isoparametric Lagrangian Interpolation (ILI)

$$f(\delta) = \sum_{i=0}^{N-1} R_i(\delta) f(x_i) \qquad R_i(\delta) = \frac{(-1)^{N+i-1}}{[N-(i+1)]!\, i!} \prod_{l=0, l\neq i}^{N-1} (\delta - l)$$

Isoparametric Lagrangian Interpolation (ILI) For non-equally spaced source points.



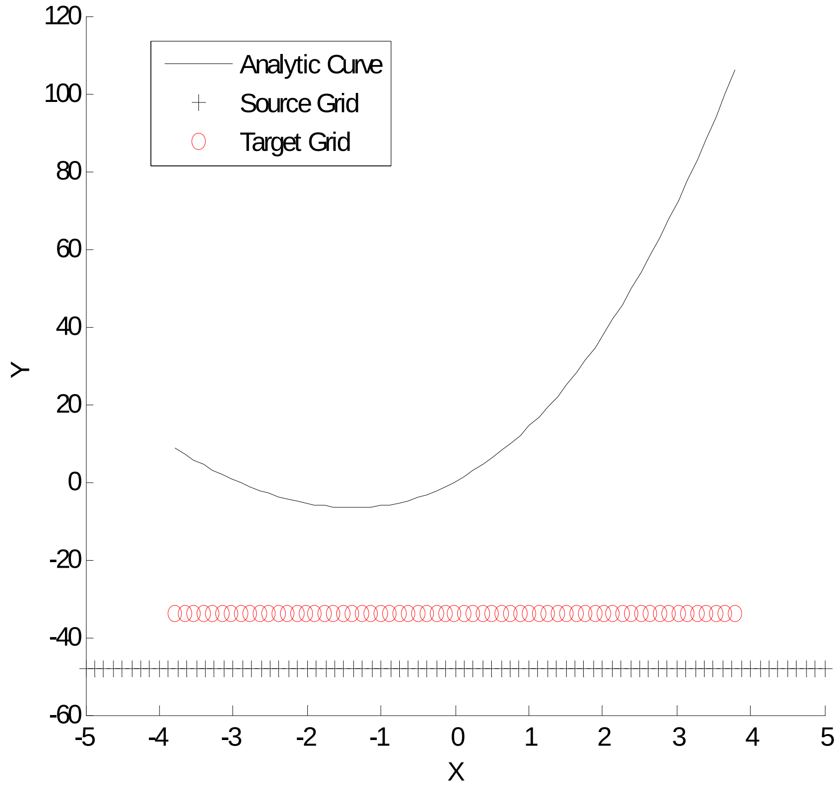
$$f(\delta) = \sum_{i=0}^{N-1} R_i(\delta) f(x_i) \qquad R_i(\delta) = \frac{(-1)^{N+i-1}}{[N-(i+1)]!\, i!} \prod_{l=0, l\neq i}^{N-1} (\delta - l)$$

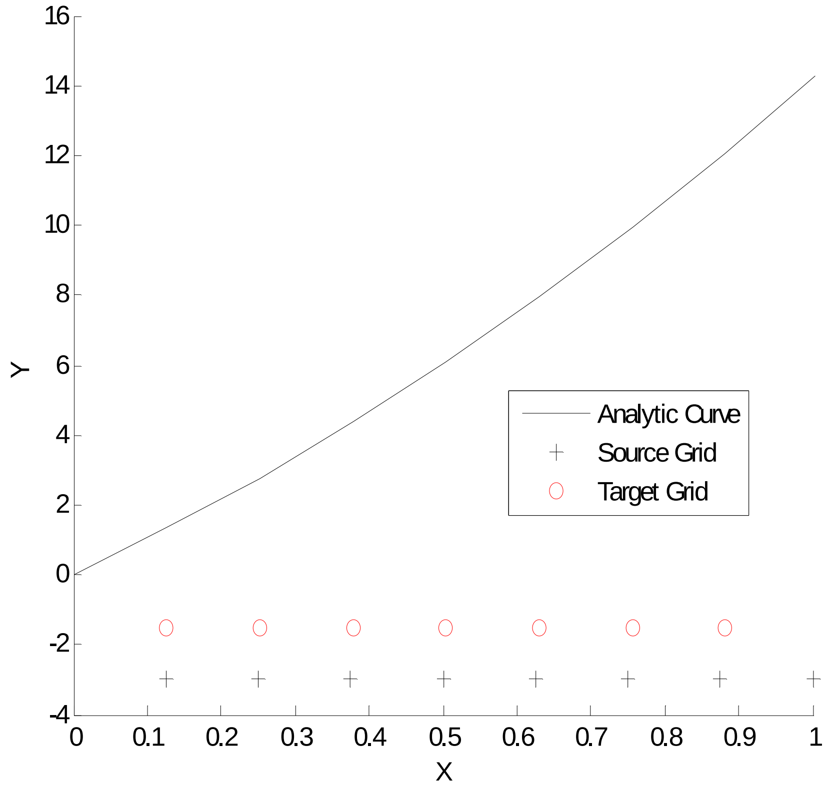
where δ is found by minimizing the functional:

$$F(x_i, \hat{x}, \delta) = \sum_{i=0}^{N-1} R_i(\delta) x_i - \hat{x} = 0$$

But what about loss of accuracy???

Example: Interpolate a cubic function using uniform source grid

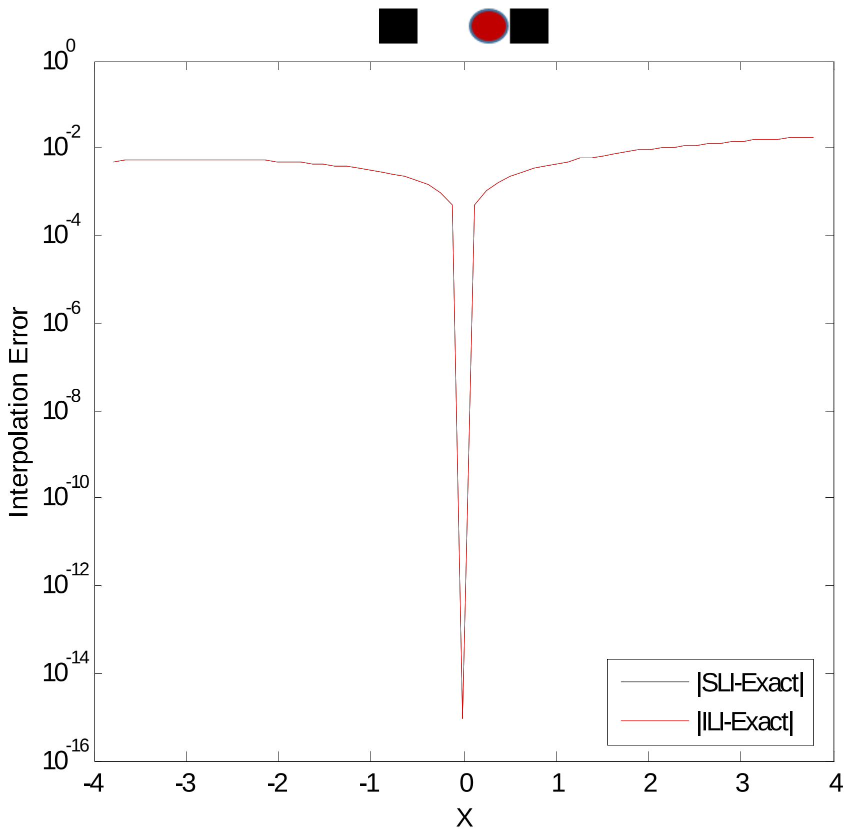




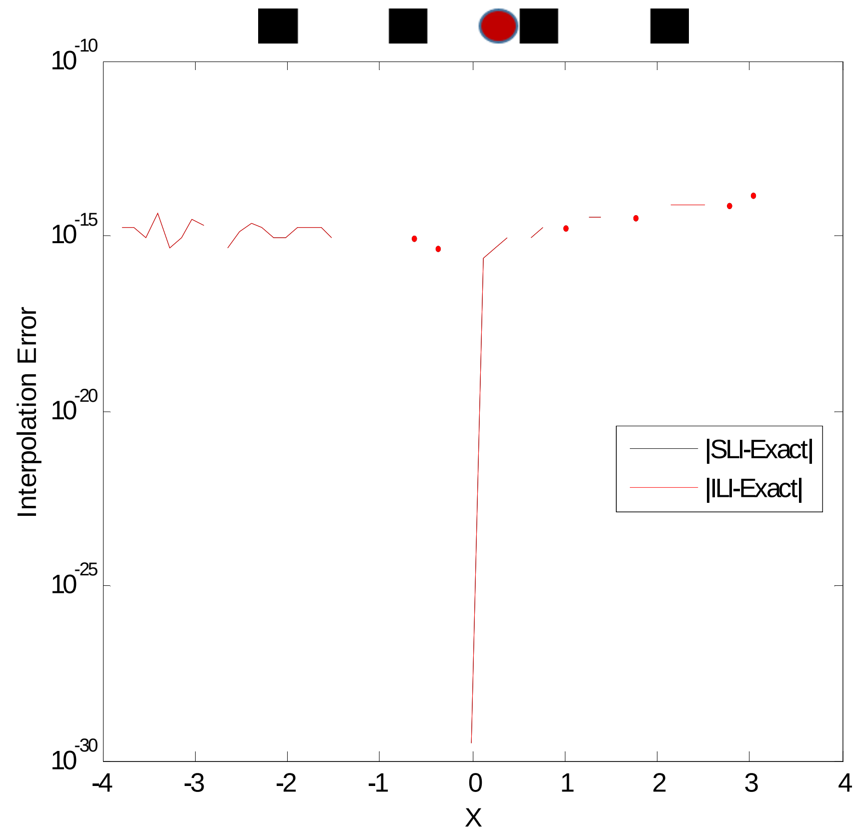
Example: Interpolate a cubic function using uniform source grid

Stencil size

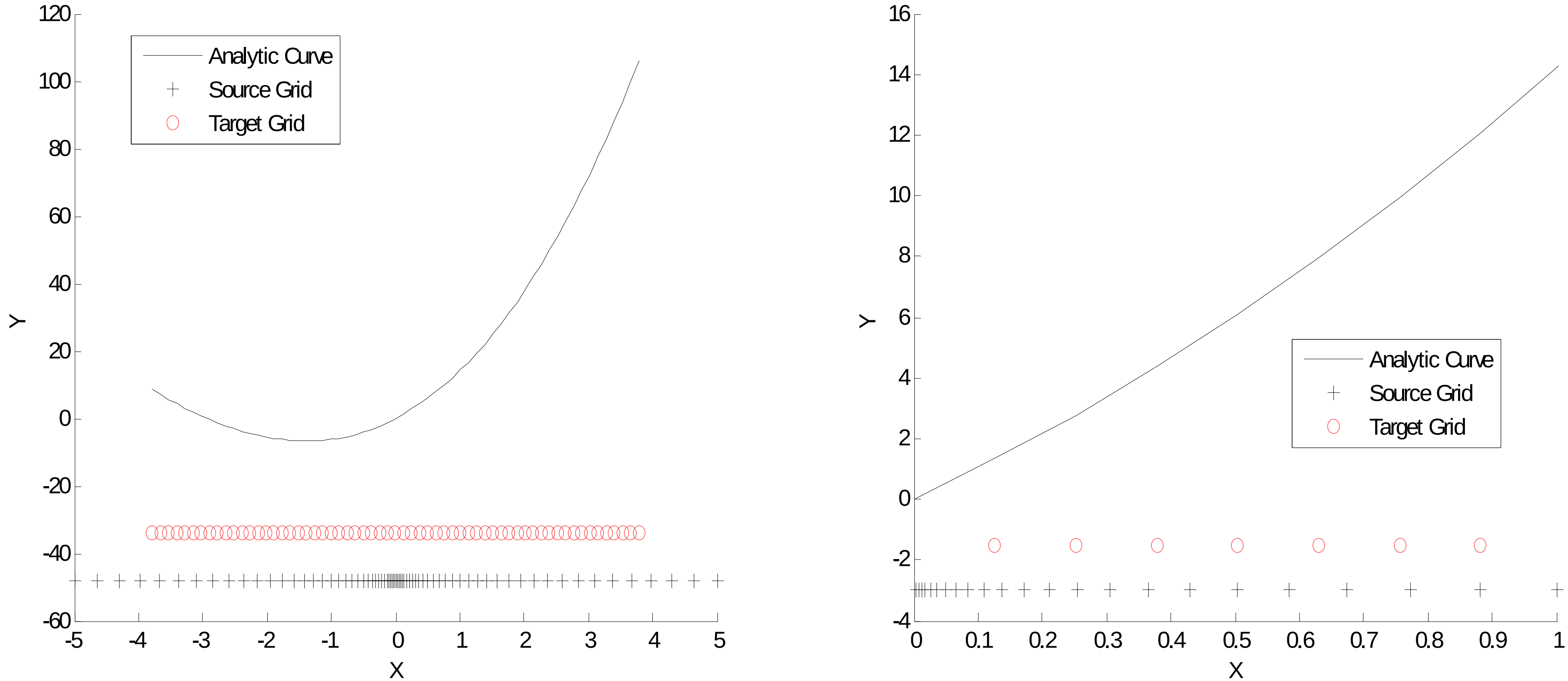
N2



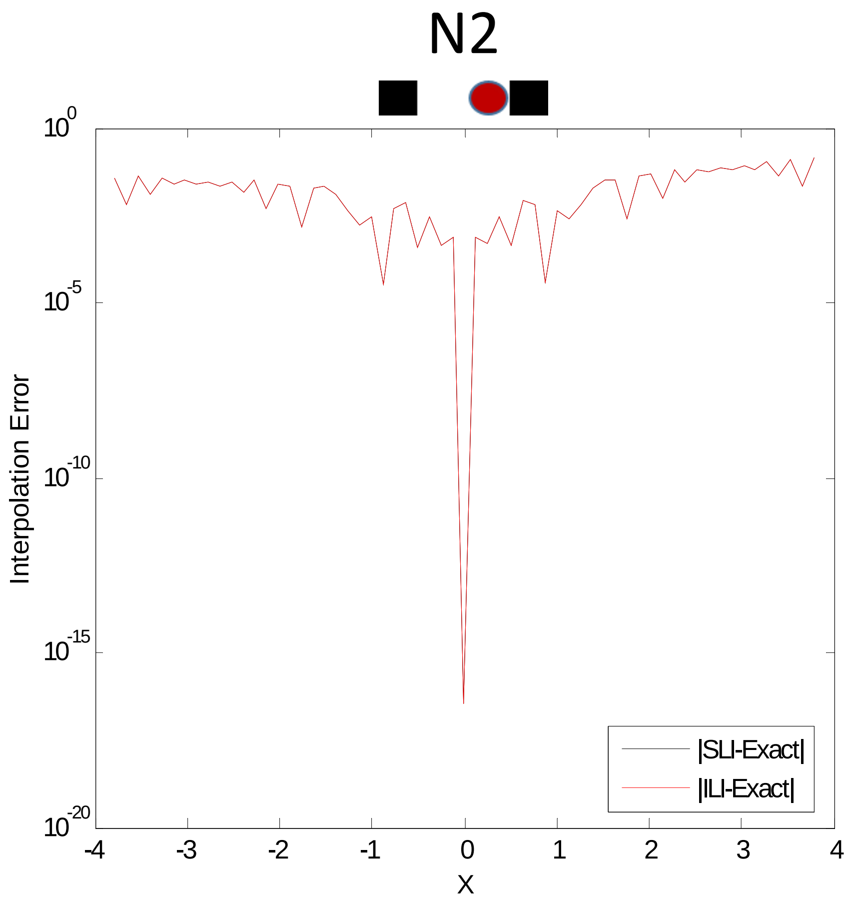
N4

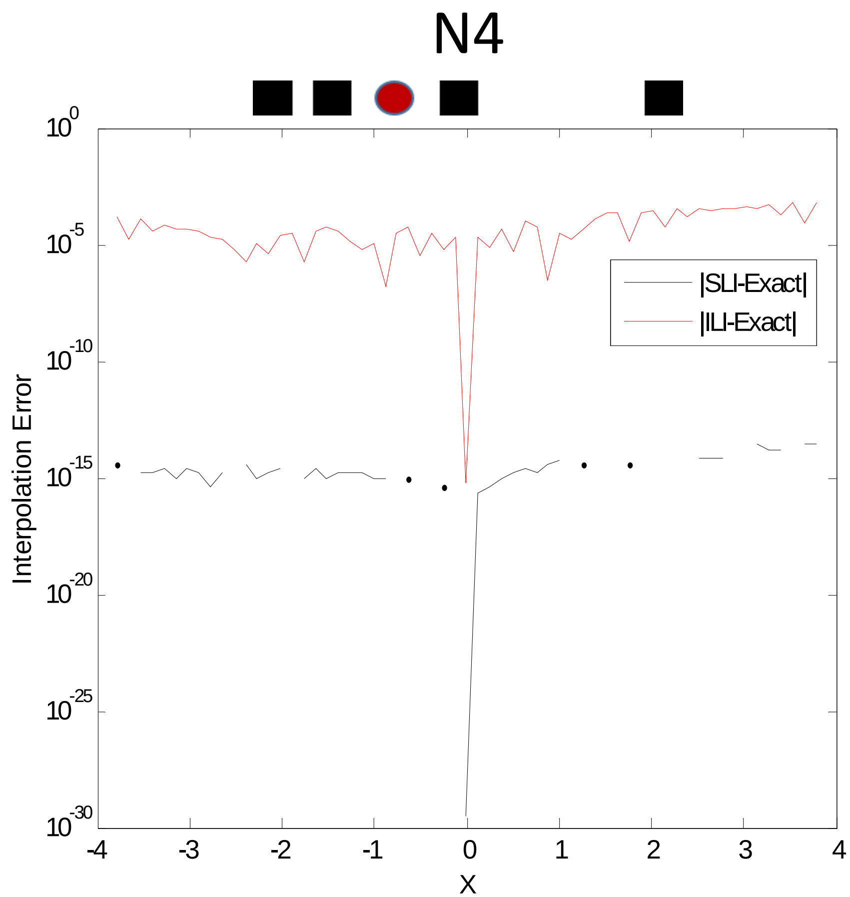


Example: Interpolate a cubic function using quadratic-ly stretched source grid



Example: Interpolate a cubic function using quadraticly stretched source grid



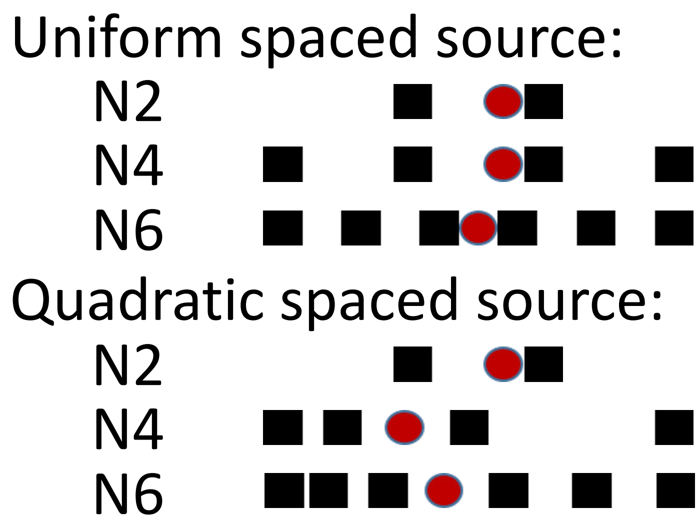


Examination of 1D interpolation error using different…

function types,

Function Number	Equation
1	$f(x) = 10x$
2	$f(x) = 4x^2 + 10x$
3	$f(x) = 0.2x^3 + 4x^2 + 10x$
4	$f(x) = 0.6x^4 + 0.2x^3 + 4x^2 + 10x$
5	$f(x) = 0.8x^5 + 0.6x^4 + 0.2x^3 + 4x^2 + 10x$
6	$f(x) = 5e^{-x^2/16}$
7	$f(x) = 5(\sin x)/x$

donor stencils,



and interpolation methods.

Standard Lagrangian (SLI)

$$f(\hat{x}) = \sum_{i=0}^{N-1} P_i(\hat{x}) f(x_i)$$

$$P_i(\hat{x}) = \prod_{j=0, j \neq i}^{N-1} \frac{(\hat{x} - x_j)}{(x_i - x_j)}$$

Isoparametric Lagrangian (ILI)

$$f(\delta) = \sum_{i=0}^{N-1} R_i(\delta) f(x_i)$$

$$R_i(\delta) = \frac{(-1)^{N+i-1}}{[N - (i+1)]!\, i!} \prod_{l=0, l \neq i}^{N-1} (\delta - l)$$

Interpolation error from uniform source grid

		Isoparametric Lagrangian		Standard Lagrangian	
Basis Function Number	Stencil size, N	Maximum Difference Error:	Maximum Relative Error (%):	Maximum Difference Error:	Maximum Relative Error (%):
1	2	7.11E-15	1.89E-14	7.11E-15	1.89E-14
1	4	7.11E-15	2.11E-14	7.11E-15	2.11E-14
1	6	7.11E-15	2.19E-14	7.11E-15	2.19E-14
2	2	1.17E-02	5.50E-02	1.17E-02	5.50E-02
2	4	1.42E-14	3.32E-14	1.42E-14	3.32E-14
2	6	1.42E-14	3.95E-14	1.42E-14	3.95E-14
3	2	1.84E-02	1.67E-02	1.84E-02	1.67E-02
3	4	1.42E-14	1.86E-14	1.42E-14	1.86E-14
3	6	2.84E-14	3.07E-14	2.84E-14	3.07E-14
4	2	1.71E-01	6.95E-02	1.71E-01	6.95E-02
4	4	6.01E-05	4.14E-05	6.01E-05	4.14E-05
4	6	4.26E-14	4.43E-14	4.26E-14	4.43E-14
5	2	1.46E+00	1.54E-01	1.46E+00	1.54E-01
5	4	1.58E-03	1.68E-04	1.58E-03	1.68E-04
5	6	1.71E-13	3.66E-14	1.71E-13	3.66E-14
6	2	2.99E-04	1.44E-02	2.99E-04	1.44E-02
6	4	6.04E-07	2.91E-05	6.04E-07	2.91E-05
6	6	1.27E-09	5.48E-08	1.27E-09	5.48E-08
7	2	1.82E-03	2.10E-01	1.82E-03	2.10E-01
7	4	3.62E-06	4.19E-04	3.62E-06	4.19E-04
7	6	8.61E-09	9.97E-07	8.61E-09	9.97E-07

Some observations:

For polynomial functions:

- N should be > function order

For non-polynomial functions:

- Error reduces with higher N

Interpolation error from quadratic source grid

<u>Some observations:</u>

For polynomial functions:

- Error reduces with higher N

For non-polynomial functions:

- Error reduces with higher N
- ILI sometimes better than SLI

What kind of function is a flow variable (i.e u,v,w,p,e,ρu,ρv)?

		Isoparametric Lagrangian		Standard Lagrangian	
Basis Function Number	Stencil size, N	Maximum Difference Error:	Maximum Relative Error (%):	Maximum Difference Error:	Maximum Relative Error (%):
1	2	7.11E-15	1.94E-14	7.11E-15	1.94E-14
1	4	1.00E-11	3.92E-10	1.07E-14	3.16E-14
1	6	1.00E-11	3.92E-10	1.07E-14	3.75E-14
2	2	8.81E-02	8.56E-02	8.81E-02	8.56E-02
2	4	2.46E-04	2.72E-04	1.42E-14	2.68E-14
2	6	8.12E-08	1.65E-06	1.42E-14	2.68E-14
3	2	1.38E-01	1.20E-01	1.38E-01	1.20E-01
3	4	6.30E-04	5.45E-04	2.84E-14	3.26E-14
3	6	1.26E-06	1.09E-06	2.84E-14	3.26E-14
4	2	1.29E+00	4.86E-01	1.29E+00	4.86E-01
4	4	1.79E-02	6.77E-03	2.77E-03	1.04E-03
4	6	1.49E-04	5.62E-05	5.68E-14	3.44E-14
5	2	1.10E+01	1.04E+00	1.10E+01	1.04E+00
5	4	2.61E-01	2.46E-02	7.33E-02	6.92E-03
5	6	4.22E-03	3.98E-04	2.27E-13	3.07E-14
6	2	2.26E-03	1.05E-01	2.26E-03	1.05E-01
6	4	2.23E-05	5.97E-04	2.77E-05	1.28E-03
6	6	6.02E-07	2.46E-05	3.36E-07	1.56E-05
7	2	1.37E-02	1.47E+00	1.37E-02	1.47E+00
7	4	1.21E-04	1.30E-02	1.66E-04	1.78E-02
7	6	3.58E-06	1.04E-03	2.36E-06	2.53E-04

1-D Told Us …

- If your function is a polynomial, then polynomial interpolation works great! (Duh)
- If you are using Isoparametic Lagrangian Interpolation (ILI), then you should use a stencil wider than the order of the function.
- If the function is not a polynomial, then usually wider the stencil the better.
- When using ILI for flow variables, all this is interpreted as:
  Use an overset stencil width (N) wider than the order of the scheme.

All these points will apply to 3-D

Solvers

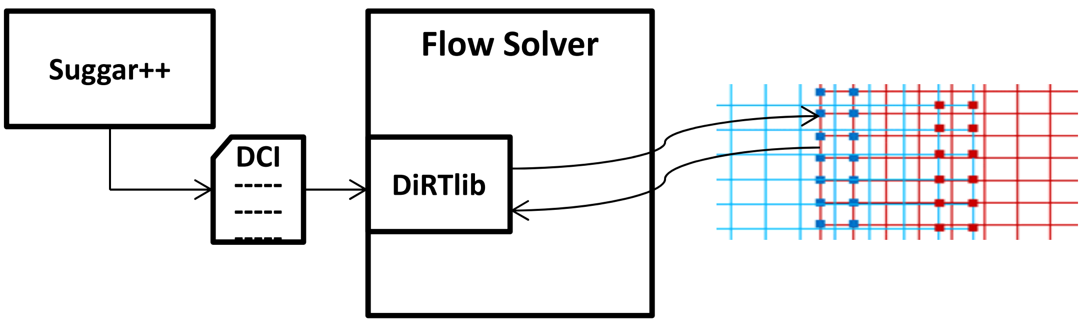
- Penn State (PSU) in-house code:
  - Density based finite volume
  - Roe flux based upwind scheme
  - Up to 7th order accurate inviscid flux differencing
  - Tailored for incompressible flows via preconditioning
  - Overset via DiRTlib, also supports block-to-block

- OVERFLOW 2.2 (OF)
  - Density based finite difference
  - Multiple spatial and time integration schemes
  - Up to 6th order accurate inviscid flux differencing
  - Widely used for compressible flows, also has low Mach preconditioning
  - Overset built-in (or XINTOUT), *but has been modified to use DiRTlib in order to read high-order DCI files*

DiRTlib/Suggar++*

- DiRTlib is a library that can be linked to OVERFLOW (or any equipped solver) that encapsulates overset interpolation and communication to the solver by making calls to solver interface routines that are specific to the solver.
  - Gets data from solver memory for use in donor interpolations
  - Puts data into solver memory for use at fringes
  - Fills solver IBLANK array to define holes and fringes

- Suggar++ is a general overset grid assembly code that provides a domain connectivity information (DCI) file to DiRTlib.



*Noack, Boger

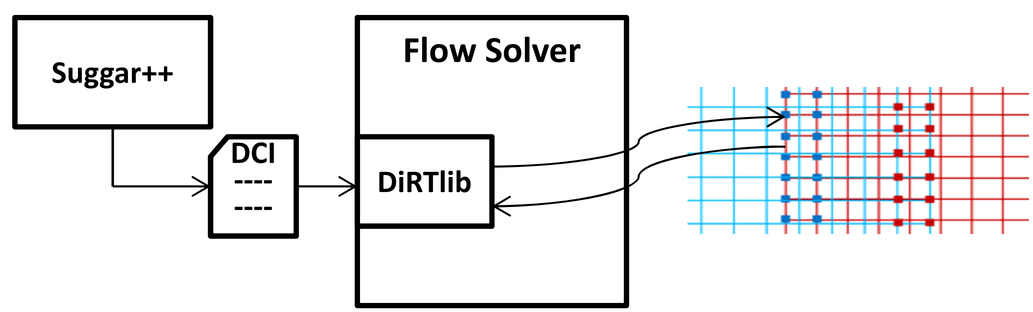
Relevant Features of DiRTlib/Suggar++

Suggar++

- Supports the specification of an *arbitrary* number of fringe layers as is required for the solver to maintain its high-order spatial discretization
- Can provide standard and high order (Lagrangian) weights

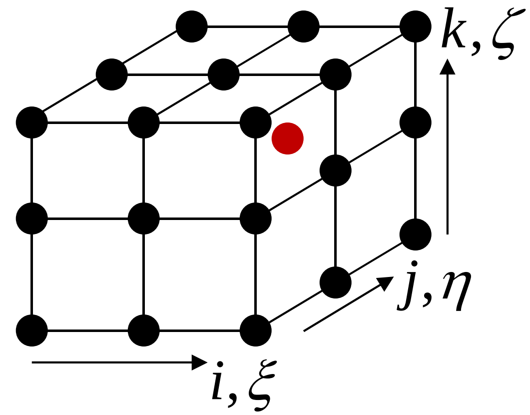
DiRTlib

- Supports arbitrary number of fringe layers
- Makes no assumption on the number of weights



Standard Lagrangian Interpolation (SLI) In Three Dimensions

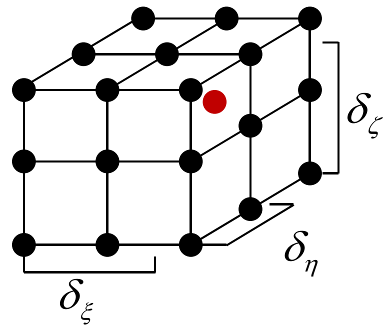
$$f(\bar{\hat{x}}) = \sum_{k=0}^{N_\zeta-1} \sum_{j=0}^{N_\eta-1} \sum_{i=0}^{N_\xi-1} P_{ijk}(\bar{\hat{x}}) f(\bar{x}_{ijk})$$



$$P_{ijk}(\bar{\hat{x}}) = P_i(\hat{x})P_j(\hat{y})P_k(\hat{z})$$
$$= \prod_{l=0,l\neq i}^{N_\xi-1} \frac{(\hat{x}-x_l)}{(x_i-x_l)} \prod_{m=0,m\neq j}^{N_\eta-1} \frac{(\hat{y}-y_m)}{(y_j-y_m)} \prod_{n=0,n\neq k}^{N_\zeta-1} \frac{(\hat{z}-z_n)}{(z_k-z_n)}$$

Note: Depending on orientation of the source points, *P* may be undefined.

Isoparametric Lagrangian Interpolation (ILI) In Three Dimensions



$$f(\bar{x}) = \sum_{k=0}^{N_\zeta - 1} \sum_{j=0}^{N_\eta - 1} \sum_{i=0}^{N_\xi - 1} R^{\xi}_{ijk}(\delta_\xi) R^{\eta}_{ijk}(\delta_\eta) R^{\zeta}_{ijk}(\delta_\zeta) f(\bar{x})$$

$$R_i(\delta) = \alpha_i \prod_{j=0, j \neq i}^{N-1} (\delta - j) \qquad \alpha_i = \frac{(-1)^{N+i-1}}{[N-(i+1)]!\, i!}$$

where δ is found by minimizing the functionals:

$$F(\bar{x}, \bar{\hat{x}}, \bar{\delta}) = \sum_{k=0}^{N_\zeta - 1} \sum_{j=0}^{N_\eta - 1} \sum_{i=0}^{N_\xi - 1} R^{\xi}_{ijk}(\delta_\xi) R^{\eta}_{ijk}(\delta_\eta) R^{\zeta}_{ijk}(\delta_\zeta) \bar{x} - \bar{\hat{x}} = 0$$

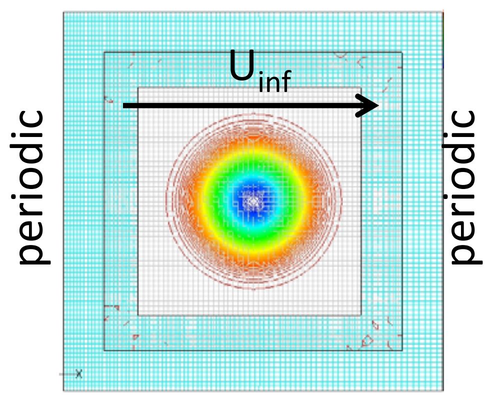
This is the high order Lagrangian technique implemented in Suggar++.

Convecting Inviscid Vortex

(PSU: incompressible; OF: isentropic, M=0.2)

Description:

- Prescribe an initial inviscid vortex subject to a uniform cross flow
- Domain: Uniform background grid, with a series of different inset grids
- Boundary Conditions: Free stream, periodic BCs in flow direction
- Numerics: 3rd, 5th order accurate with N2, N4, and N6 overset interpolation



Domain width=10
DTPHYS=0.01

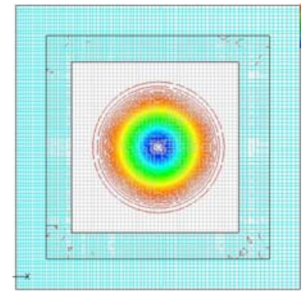
Steps for one cycle=1000

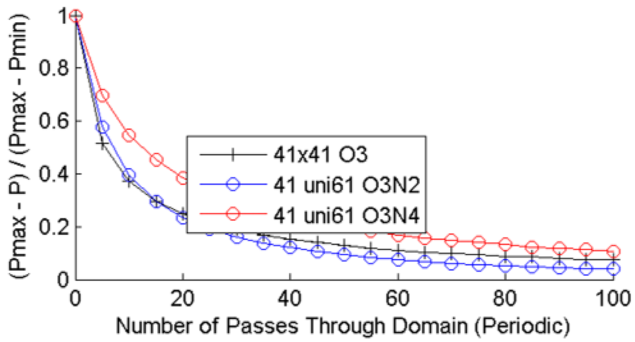
Purpose:

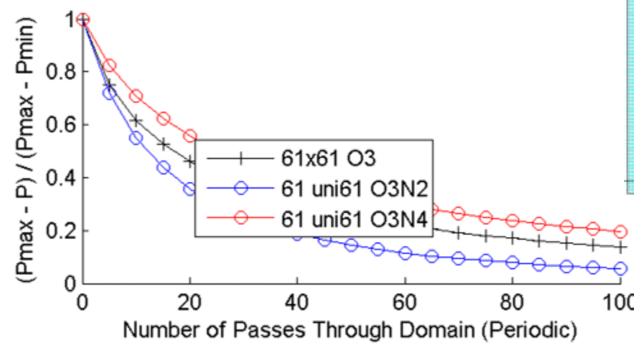
- Examine differences in accuracy (vortex dissipation and location) and between standard and high-order interpolation.

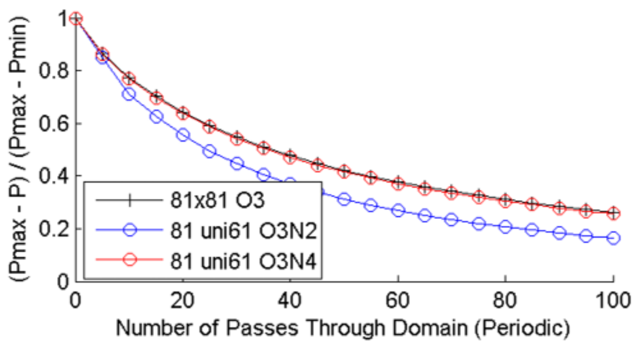
2D Convecting Inviscid Vortex, Code: PSU

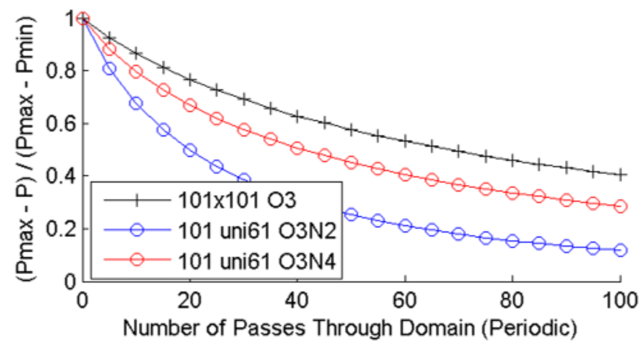
Vortex Dissipation, Uniform Inset Grid, 3rd Order

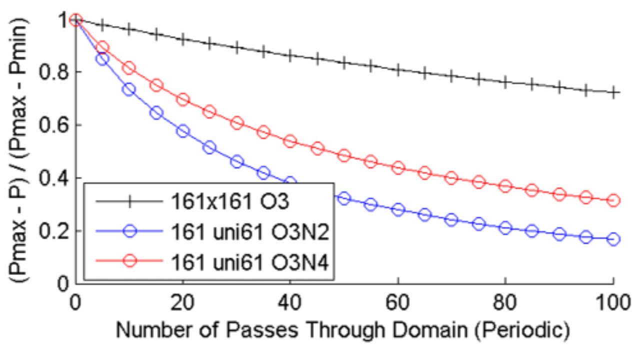






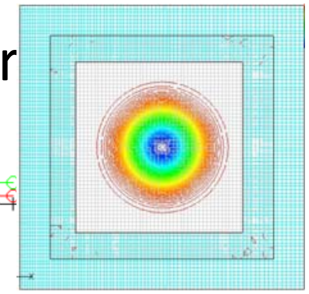


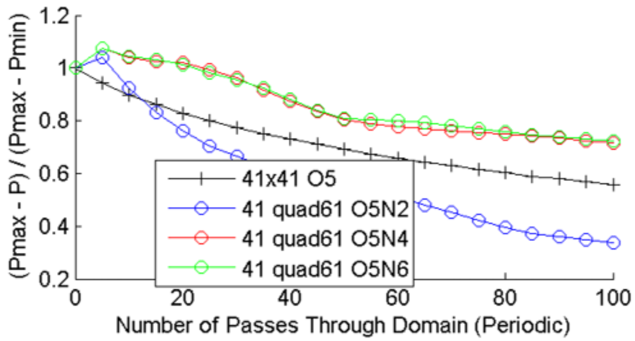


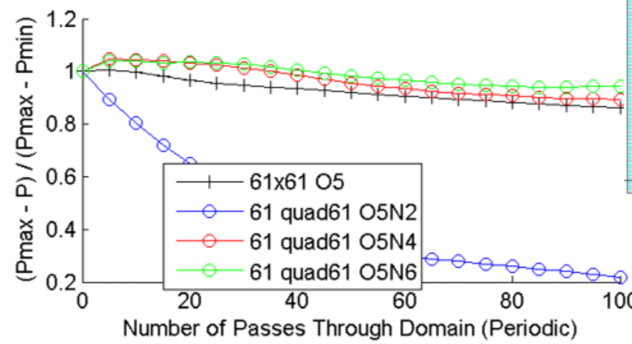


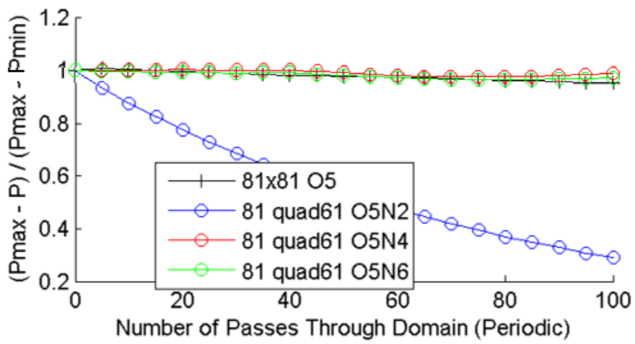
2D Convecting Inviscid Vortex, Code: PSU

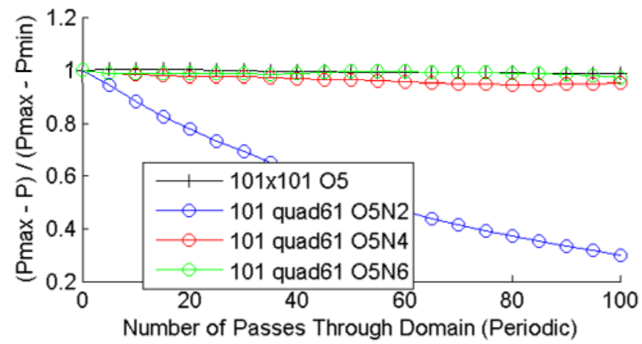
Vortex Dissipation, Quadratic Inset Grid, 5th Order

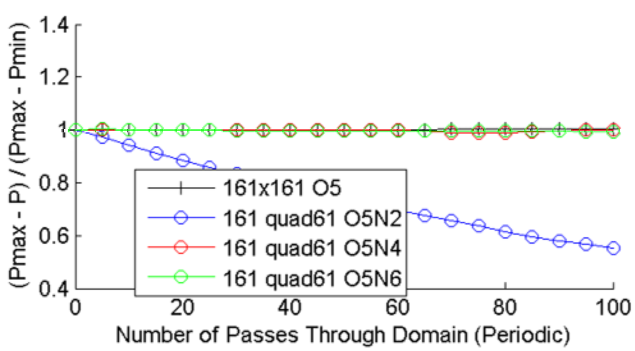






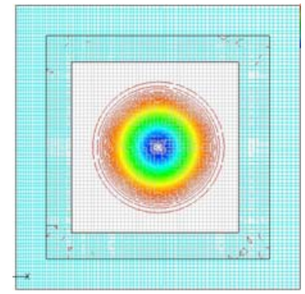


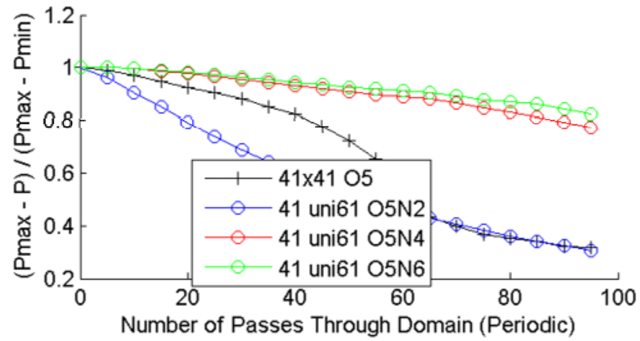


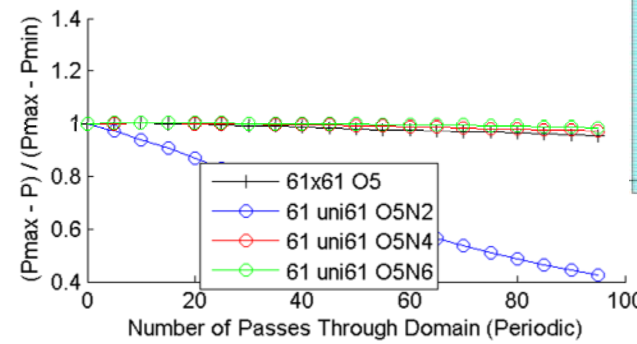


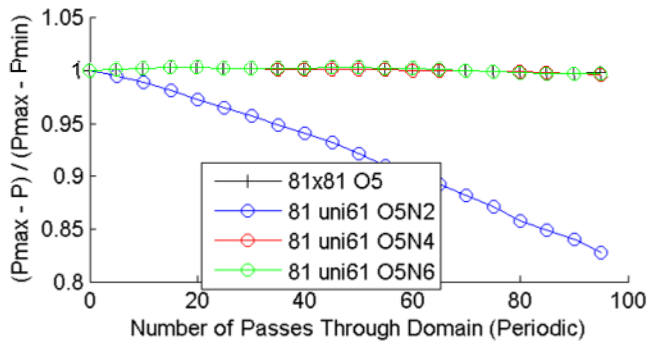
2D Convecting Inviscid Vortex, Code: OF

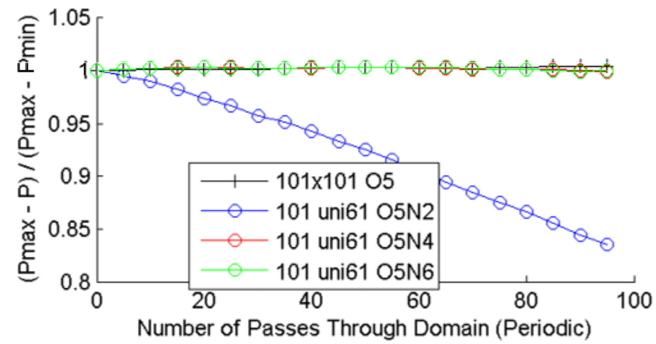
Vortex Dissipation, Uniform Inset Grid, 5th Order

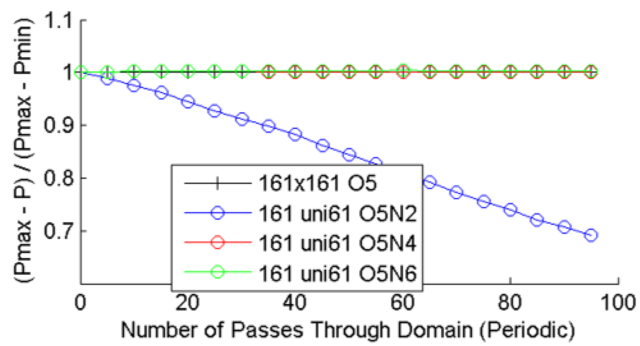






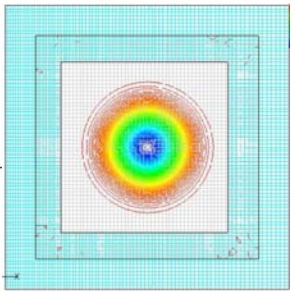


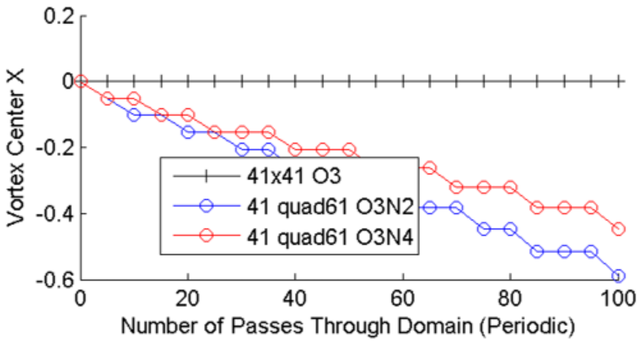


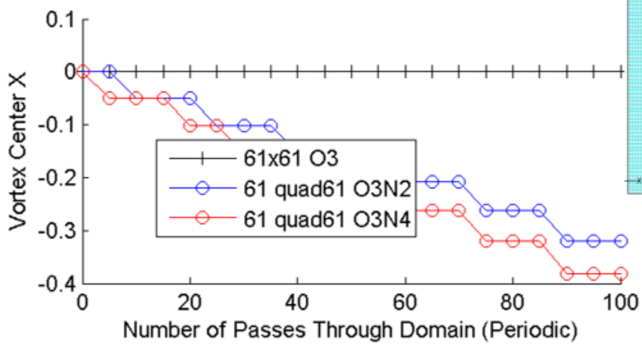


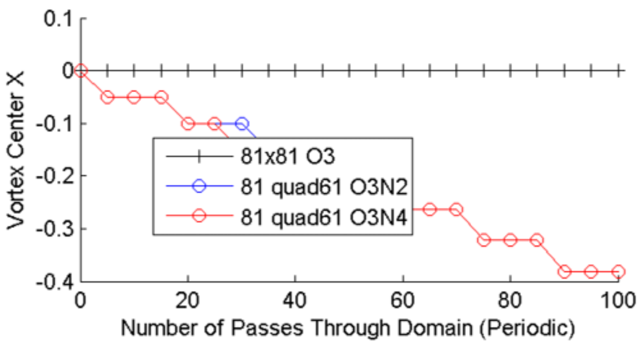
2D Convecting Inviscid Vortex, Code: PSU

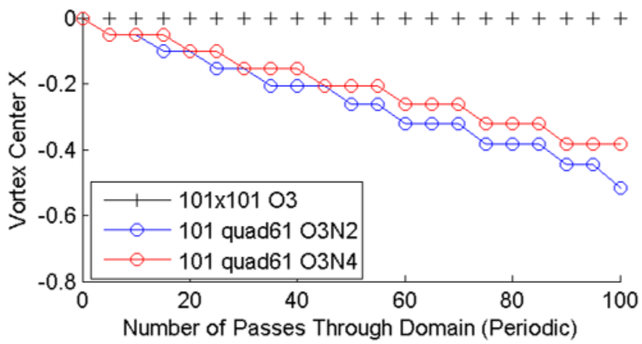
Vortex Location, Quadratic Inset Grid, 3rd Order

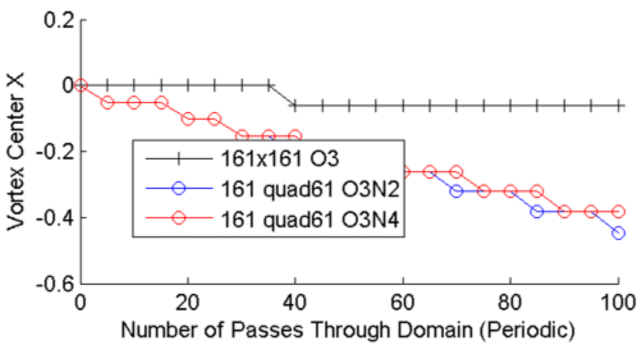






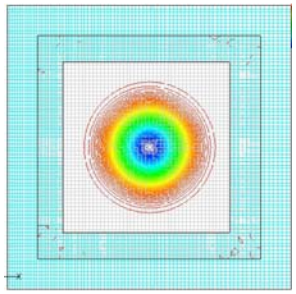


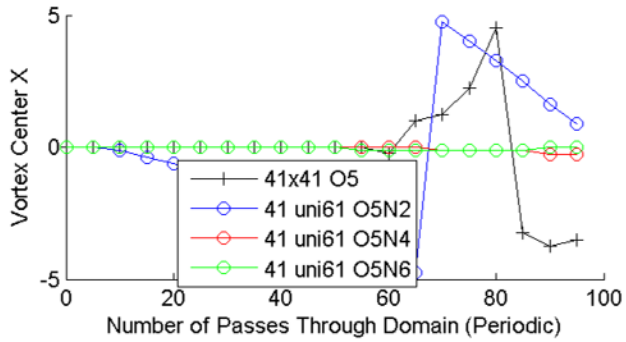


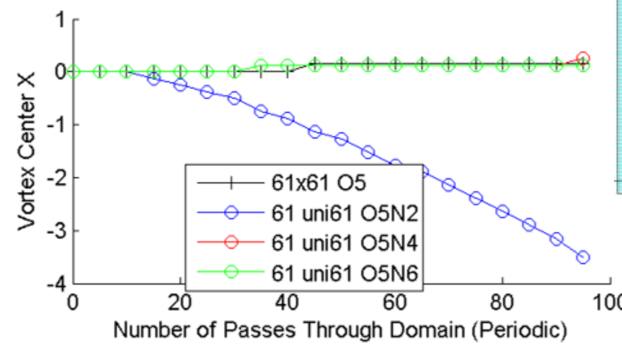


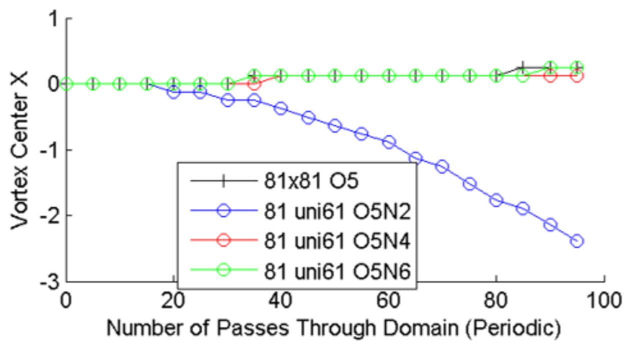
2D Convecting Inviscid Vortex, Code: OF

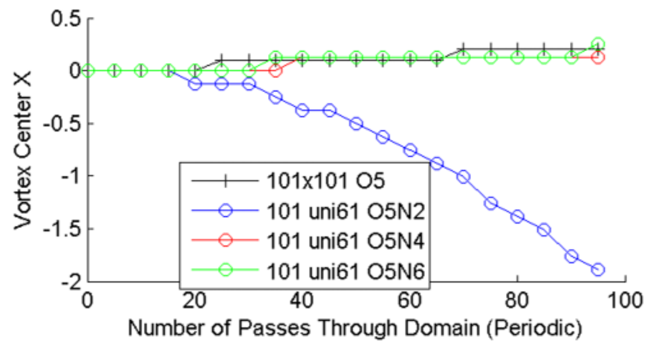
Vortex Location, Uniform Inset Grid, 5th Order

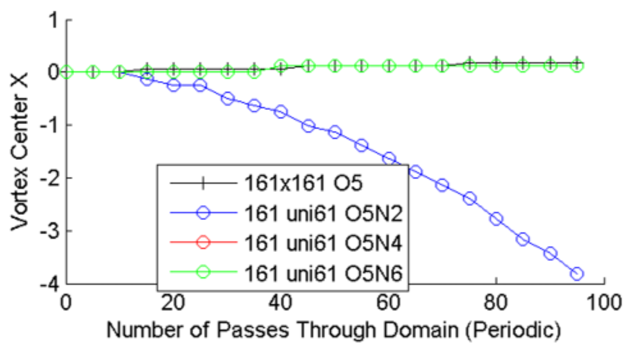








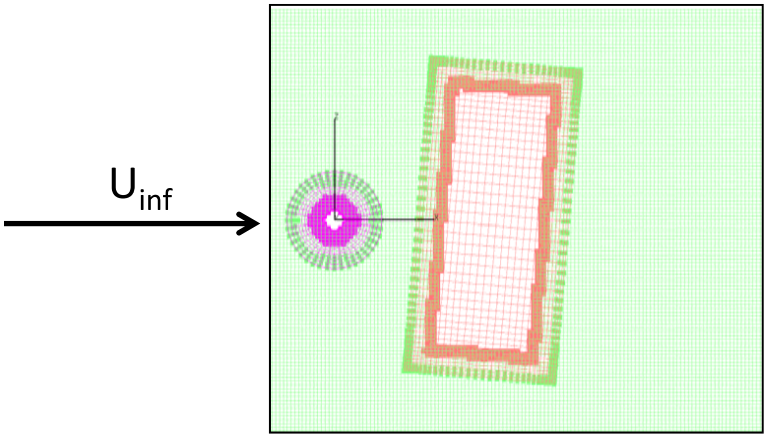




2D Cylinder in Cross Flow With Overset Patch, Code:OF

Description:

- Canonical Karman vortex shedding flow (laminar)
- Domain: Uniform background grid with and without oblique hole, covered by non-uniform overset patch grid. 3 fringe layers
- Numerics: WENOM, 2nd order time(10 Newton sub its), ADI P-C



Re_D = 150
M = 0.2
Laminar

Purpose:

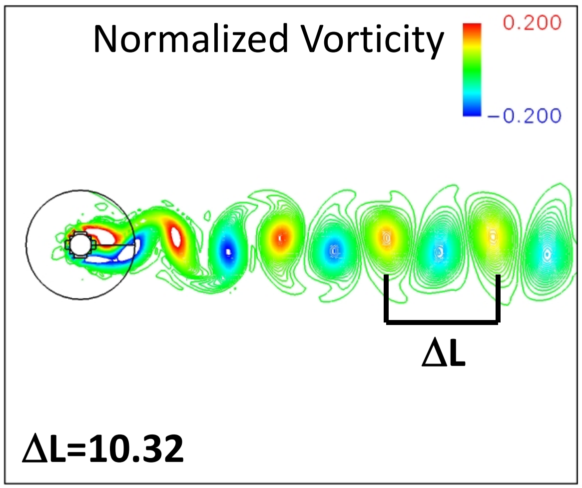
- Compare the vortex street downstream of the patch, with the baseline vortex street without the hole/patch

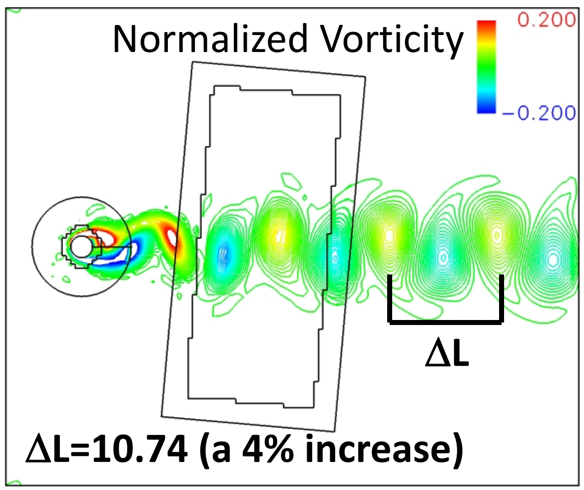
2D Cylinder in Cross Flow With Overset Patch, Code: OF

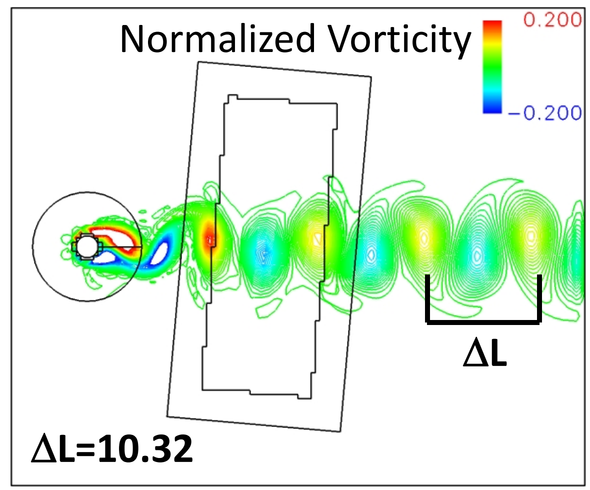
Baseline Without Downstream Patch
5^{th} Order Space (WENOM)
2^{nd} Order Time
3 Fringe Layers
High Order Interpolation (N=6)

With Downstream Patch
5^{th} Order Space (WENOM)
2^{nd} Order Time
3 Fringe Layers
Standard Interpolation (N=2)

With Downstream Patch
5^{th} Order Space (WENOM)
2^{nd} Order Time
3 Fringe Layers
High Order Interpolation (N=6)







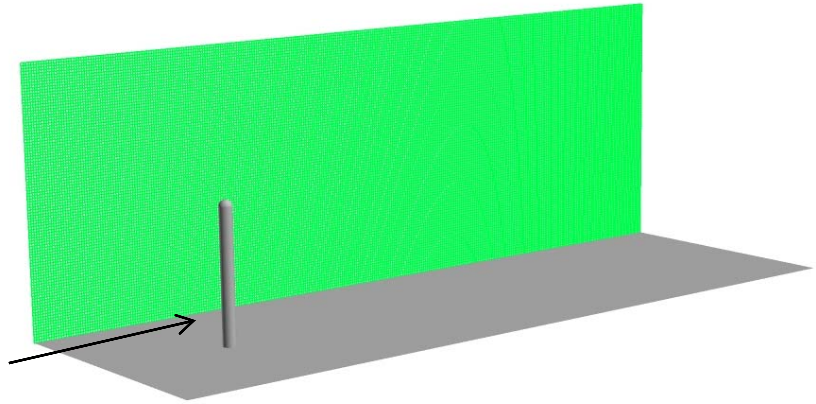
Conclusions:

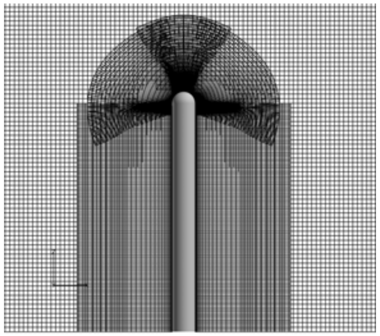
- Grid obliqueness and stretching accentuates the need to use high-order interpolation *with* 3 fringe layers to preserve WENOM accuracy

3D Cylindrical Column in Cross Flow, Codes: PSU,OF

Description:

- Column standing on solid surface

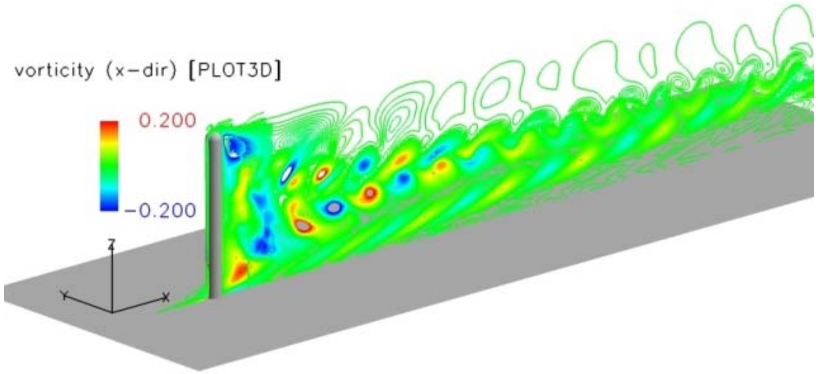




$Re_D = 150$
PSU: Incompress
OF: M = 0.2
Laminar

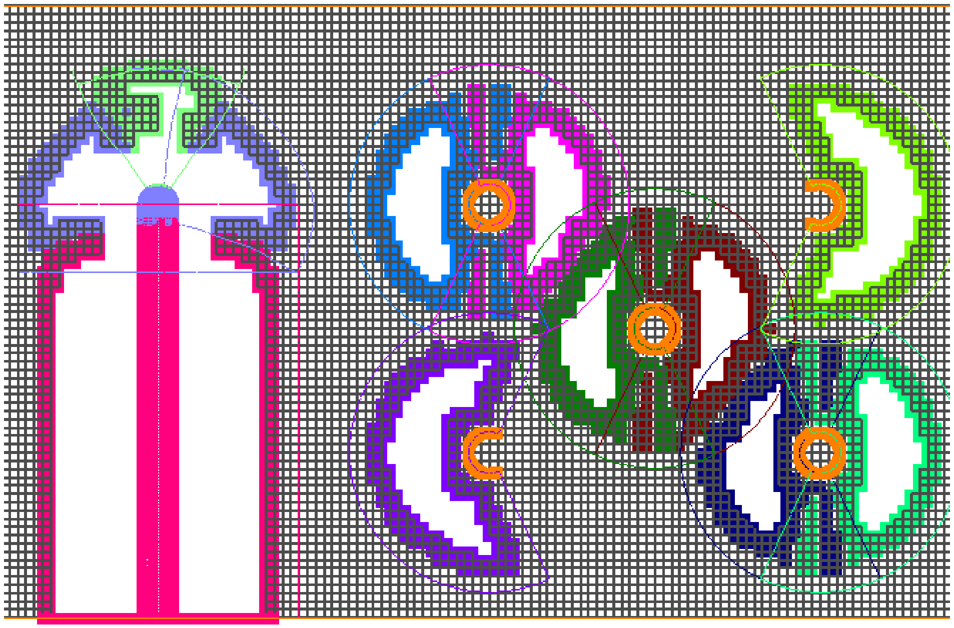
Purpose:

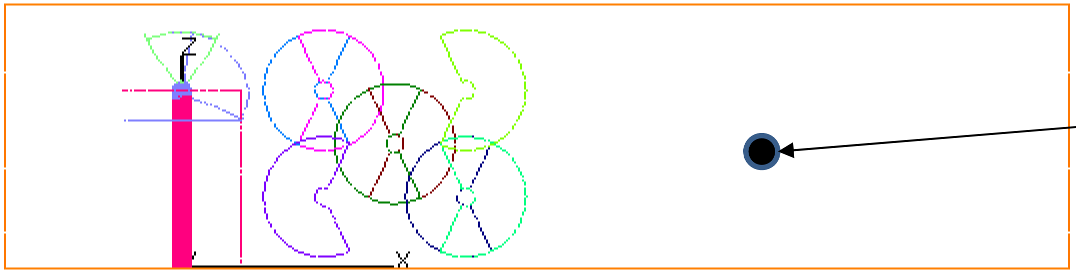
- Examine changes in shedding frequency due to overset treatment



3D Cylindrical Column in Cross Flow

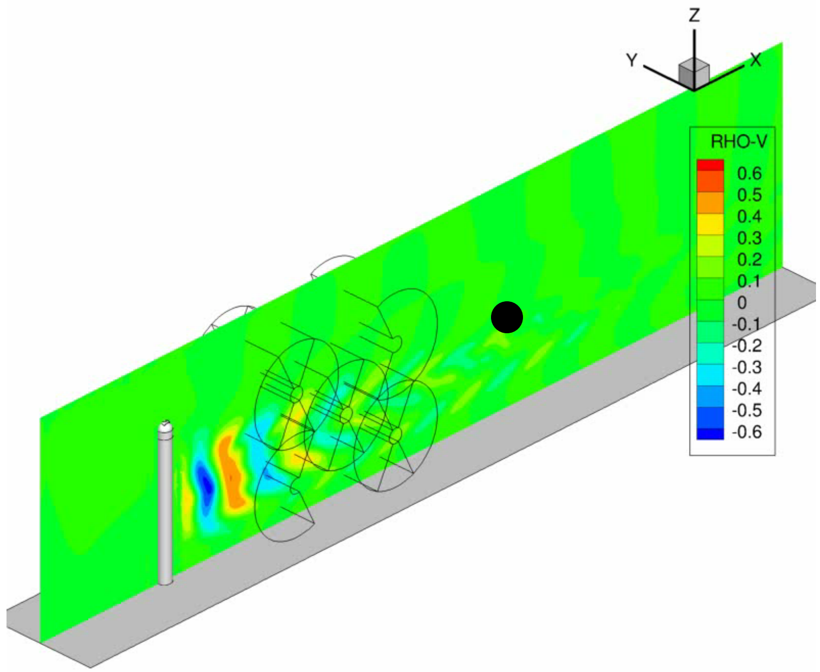
Intentionally introduced extraneous overset grids in order to create multiple overset boundaries





Examine flow at probe point

3D Cylindrical Column in Cross Flow



Frequency of oscillating lateral (v) velocity:

5th Order, Overset Stencil=6 (O5N6): 0.1389 Hz

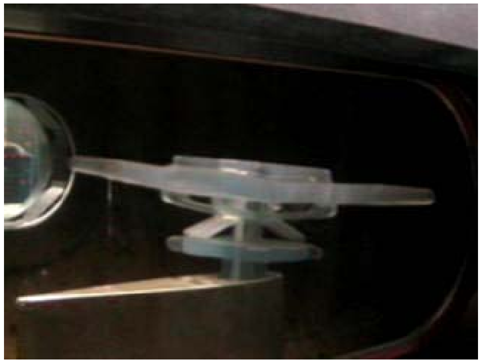
5th Order, Overset Stencil=2 (O5N2): 0.1417 Hz

A 2% change with only several overset grids!

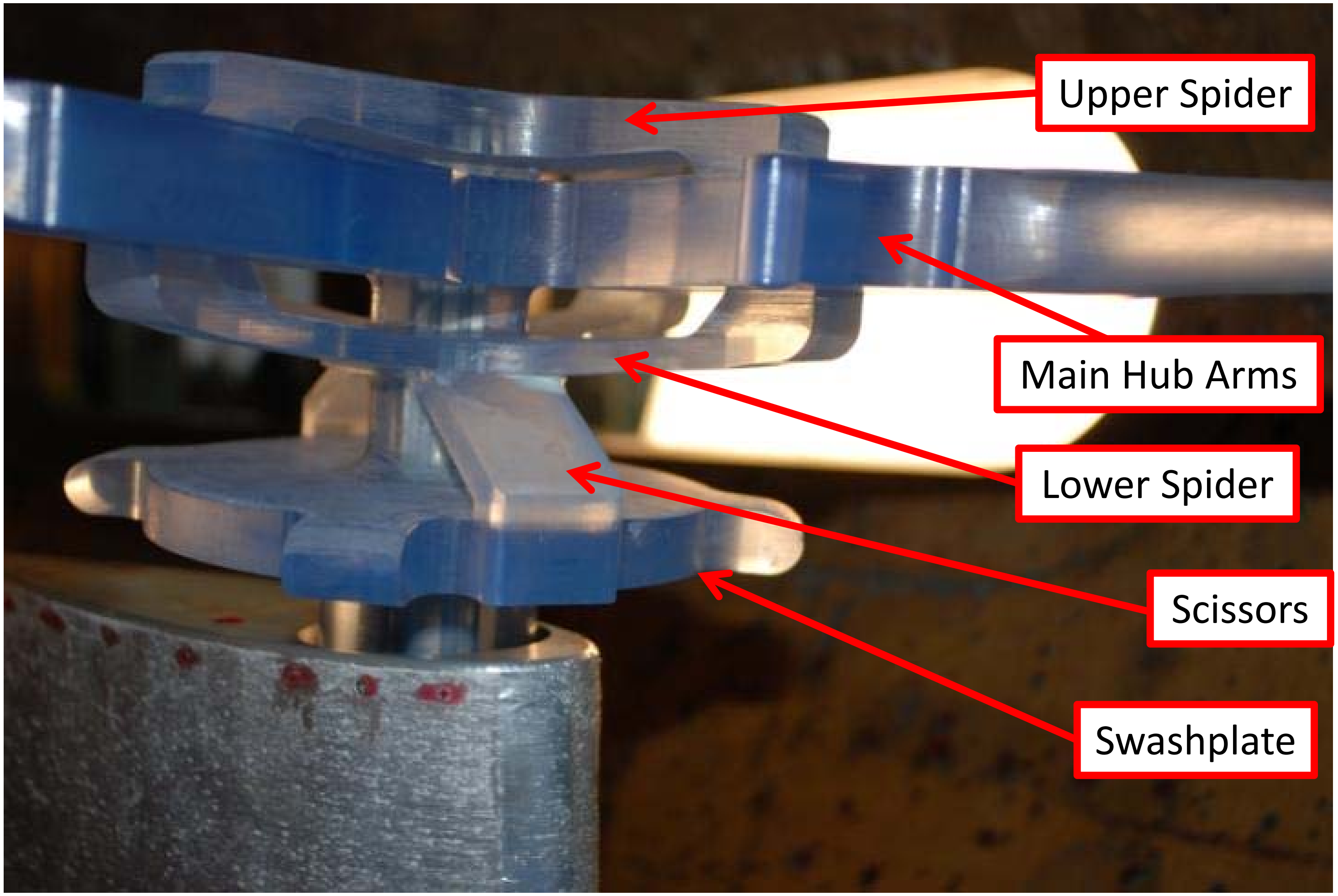
3D Rotor Hub, Code: PSU

Spinning notional scaled rotor tested in Water Tunnel at PSU. Free stream = 6.5 m/s

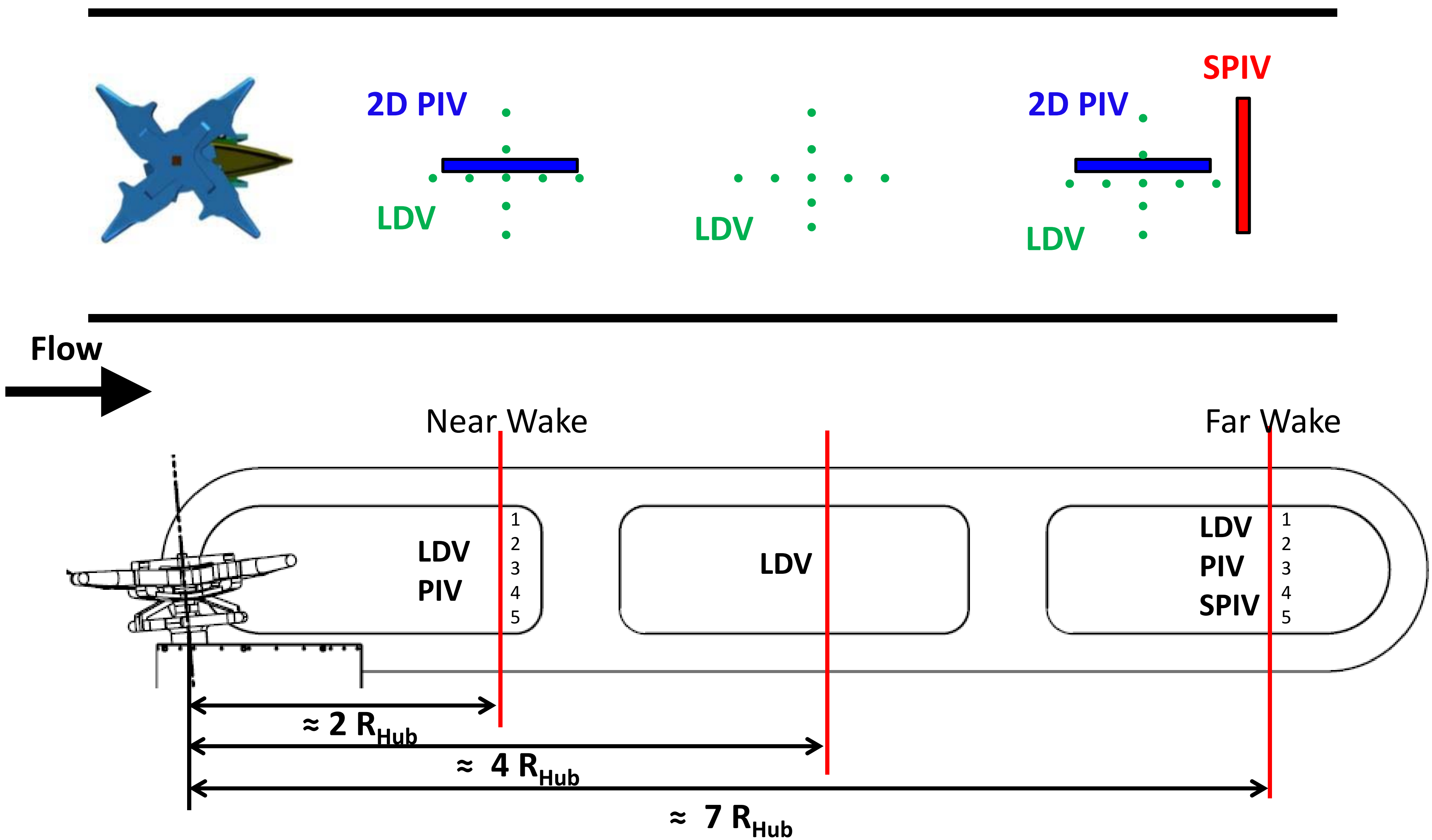
- Interest in flow structures at locations downstream where empennage would be



3D Rotor Hub Close-up



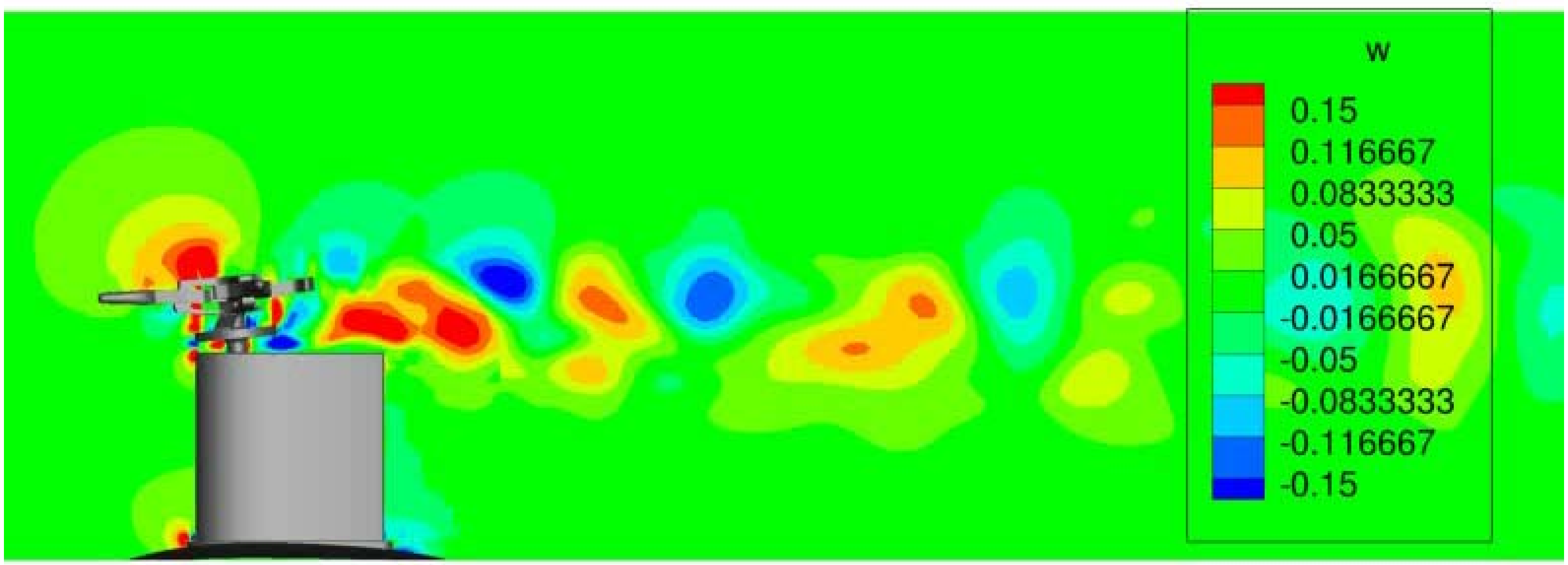
3D Rotor Hub Test



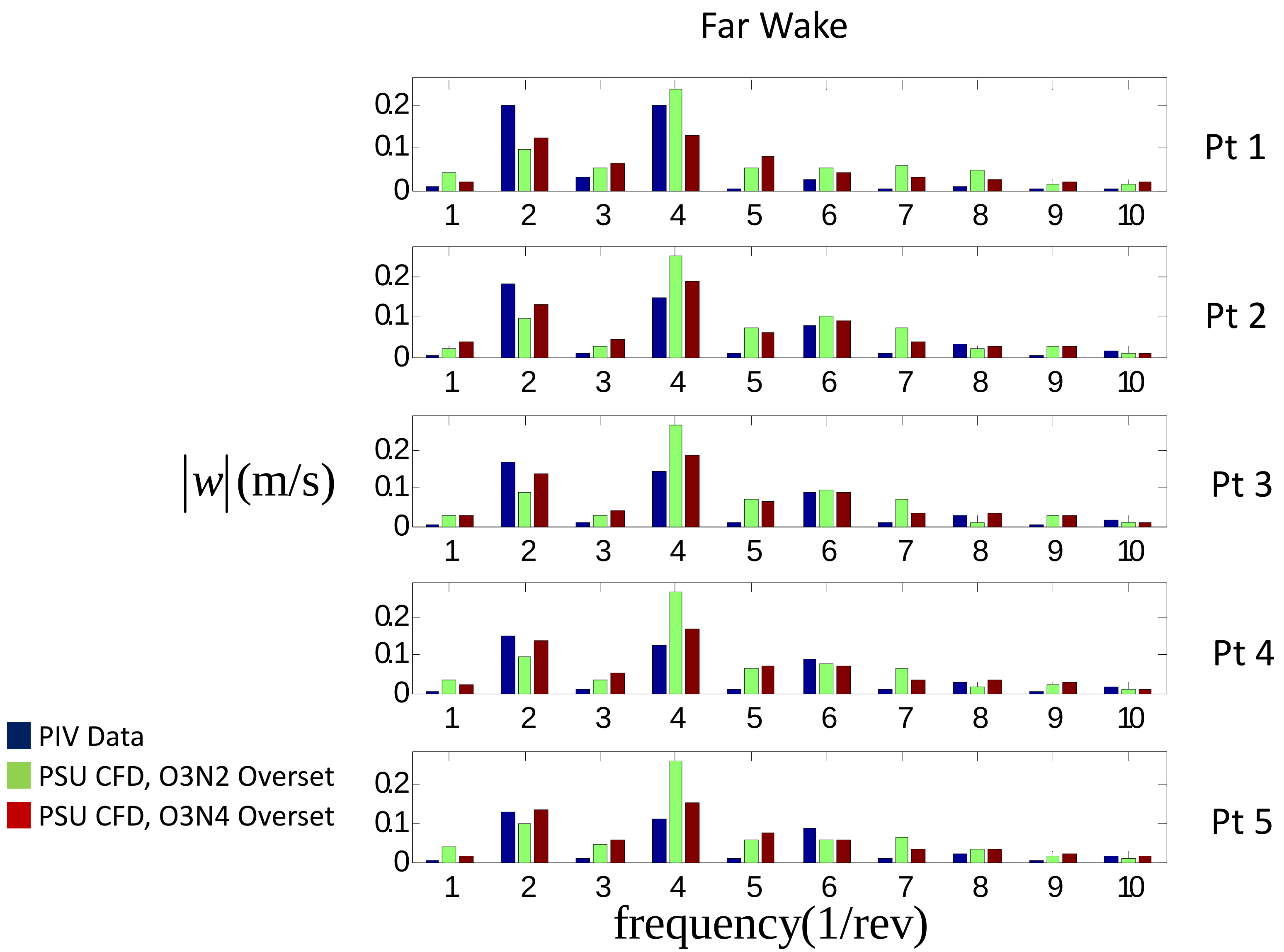
3D Rotor Hub CFD

Rotor in tunnel simulated using PSU and OF codes

- 99 Structured overset blocks
- 3rd order upwind bias
- Overset interpolation using N2 and N4



Wake Oscillating Vertical Velocity Comparisons

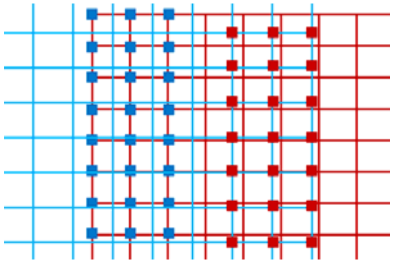


High-Order Overset is Great, But it ….

- Requires more overlap in the grid system (to keep donor qual=1)
  - For 3rd order upwind bias, standard (N2) overset needs:

    2 fringe layers, overlap=5
  - For 5th order (e.g. WENO), high-order (N6) overset needs:

    3 fringe layers, <u>overlap=9</u>!!!!
- Requires more computer time (solver and Suggar++)
- Can be susceptible to under/over shoots (just like any polynomial interpolation can/will do)



Summary

- Using simple cases, it was determined that:
  - ➢High-order interpolation better preserves flow structures across overset boundaries
  - ➢More fringe layers (3) with standard interpolation is *not* sufficient. High-order interpolation *and* more fringe layers is required to preserve accuracy of high-order numerics
- With increasing order of accuracy of the interpolation (*N*), a larger donor stencil is needed, which in turn requires more overlap in the overset gridding and more runtime requirements

Thank you !