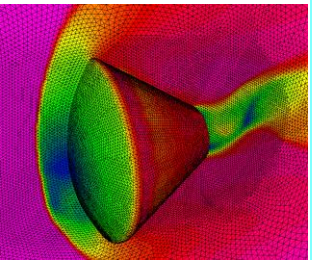
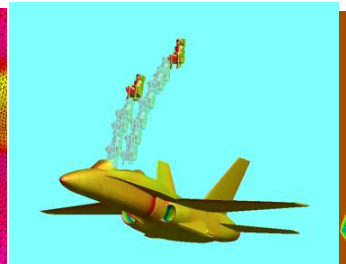
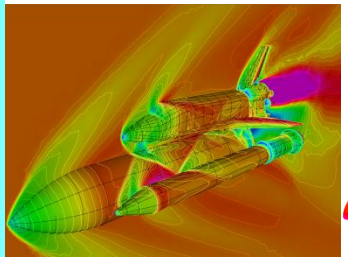
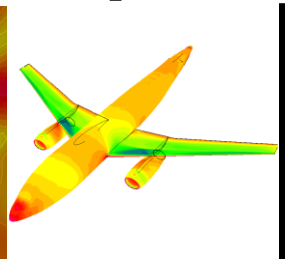
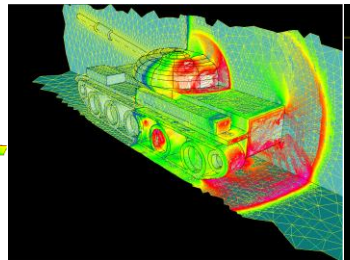
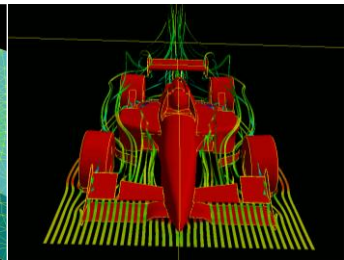
A Reconstructed Discontinuous Galerkin Method Based on a Hierarchical WENO Reconstruction for Compressible Flows on Hybrid Grids

Hong Luo

Department of Mechanical & Aerospace Engineering
North Carolina State University
Raleigh, NC 27695

Presented at
National Institute of Aerospace
May 28, 2013, Hampton, VA

Outline

- Background and Motivation
- Governing Equations
- Hierarchical Reconstructed Discontinuous Galerkin Method
  - Discontinuous Galerkin Method
  - WENO(P_1P_2): WENO reconstruction at P_2
  - HWENO(P_1P_2): WENO reconstruction at P_1
- Numerical Examples
- Concluding Remarks

Background

- Why DG ?
  - Several useful mathematical properties with respect to conservation, stability, and convergence.
  - Easy extension to higher-order (>2nd) schemes.
  - Well suited for complex geometries.
  - Easy adaptive strategies, allowing implementation of *hp*-refinement and hanging nodes.
  - Compact and highly parallelizable.
  - Accuracy for low Mach number flows.

Background

- Why not DG ?
  - High computing costs (more degrees of freedom)
    → CPU time
    → Storage requirements
  - Treatment of discontinuities (like all other high-order methods)
    → Sensitive to the implementation of limiters
    → Lead to loss of high-order accuracy

Background

To reduce high computing costs of the DG methods, Reconstructed DG (RDG(P_nP_m)) Schemes were introduced by Dumbser et al.

- P_n indicates that a piecewise polynomial of degree of n is used to represent a DG solution.
- P_m represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes and source terms.
- Provide a unified formulation for both finite volume and DG methods, and contain both classical finite volume and standard DG methods as two special cases of RDG(P_nP_m) schemes.

Background

Classification of the RDG(PnPm) Schemes

Order of Accuracy	Schemes
O(1)	RDG(P_0P_0) (DG(P_0))
O(2)	RDG(P_0P_1) RDG(P_1P_1) (DG(P_1))
O(3)	RDG(P_0P_2) RDG(P_1P_2) RDG(P_2P_2)
O(4)	RDG(P_0P_3) RDG(P_1P_3) RDG(P_2P_3) RDG(P_3P_3)
	:
O(M+1)	RDG(P_0P_m) … RDG(P_nP_m) … RDG(P_mP_m) FV New Class DG

Background

- Observation: The construction of an accurate and efficient reconstruction operator is crucial to the success of the RDG(PnPm) schemes.

- Objective: Develop a RDG method based on a hierarchical WENO reconstruction: HWENO(P_1P_2), for compressible flows with strong discontinuities on hybrid grids.
  - enhance the accuracy, and therefore reduce the high computational costs of the underlying DG methods
  - avoid the spurious oscillations in the vicinity of strong discontinuities, and therefore maintain the non-linear stability, and naturally linear stability.

- Attempt to address the two weakest links of the DG methods.

Governing Equations

Compressible Navier-Stokes Equations

$$\frac{\partial \mathbf{U}(x,t)}{\partial t} + \frac{\partial \mathbf{F}_k(\mathbf{U}(x,t))}{\partial x_k} = \frac{\partial \mathbf{G}_k(\mathbf{U}(x,t))}{\partial x_k}$$

F: inviscid flux vector
G: viscous flux vector
U: conservative variable state vector

DG(p) Method

$$\frac{d}{dt}\int_{\Omega_e}\mathbf{U}_h B_i d\Omega + \int_{\Gamma_e}\mathbf{F}_k(\mathbf{U}_h)\mathbf{n}_k B_i d\Gamma - \int_{\Omega_e}\mathbf{F}_k(\mathbf{U}_h)\frac{\partial B_i}{\partial x_k}d\Omega = \int_{\Gamma_e}\mathbf{G}_k(\mathbf{U}_h)\mathbf{n}_k B_i d\Gamma - \int_{\Omega_e}\mathbf{G}_k(\mathbf{U}_h)\frac{\partial B_i}{\partial x_k}d\Omega, \quad 1 \le i \le \mathrm{N}$$

B_i(x): basis functions of the polynomials of degree p, $1 \le i \le$ N.

N: dimension of the polynomial space p.

↑

Discontinuous Galerkin method of degree p DG(P) : O(h^{p+1})

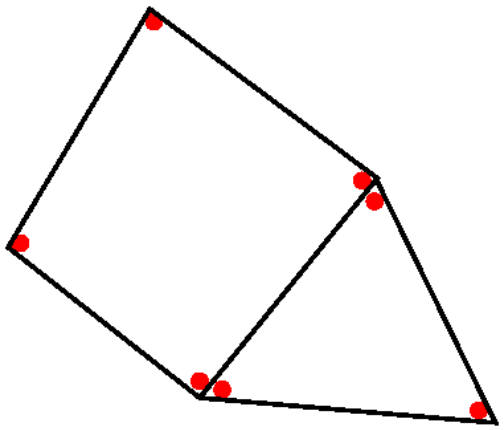
$\mathbf{F}_k(\mathbf{U}_h)\mathbf{n}_k = \mathbf{H}_k(\mathbf{U}^L_h, \mathbf{U}^R_h, \mathbf{n}_k)$ ← Numerical Riemann flux function

$$\mathbf{G}_k(\mathbf{U}_h, \frac{\partial \mathbf{U}_h}{\partial x_i})\mathbf{n}_k = \mathbf{H}_v(\mathbf{U}^L_h, \mathbf{U}^R_h, \frac{\partial \mathbf{U}^L_h}{\partial x_i}, \frac{\partial \mathbf{U}^R_h}{\partial x_i}, \mathbf{n})$$

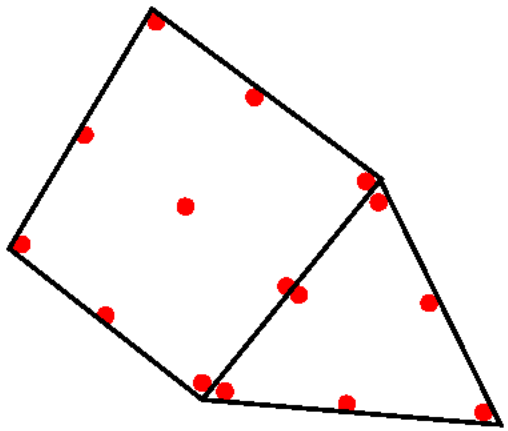
The computation of the viscous fluxes has to properly resolve the discontinuities at the interfaces.

Finite Element Basis (Nodal DGM)

$$\mathbf{U}_h = \sum_{i=1}^{N} \mathbf{U}_i(t) B_i(x),$$



Q1/P1



Q2/P2

$$\mathbf{M}_{N\times N} \frac{d}{dt} \begin{pmatrix} U_1 \\ U_2 \\ \bullet \\ \bullet \\ U_N \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ \bullet \\ \bullet \\ R_N \end{pmatrix}$$

Bi: Finite element shape functions.

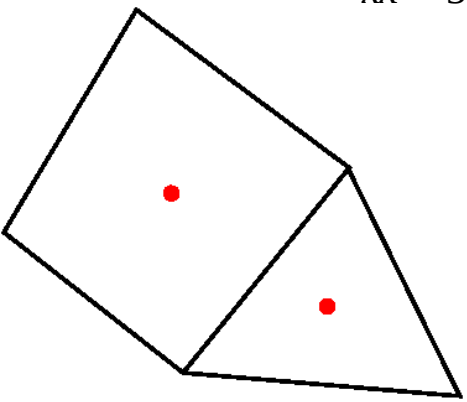
N : Dimension of polynomial space.

The unknowns are the conservative variables at the nodes.

Polynomial solutions are dependent on the shape of elements.

Taylor Basis (Modal DGM)

$$\mathbf{U}_h = \tilde{\mathbf{U}} B_1 + \mathbf{U}_x B_2 + \mathbf{U}_y B_3 + \mathbf{U}_z B_4$$
$$+ \mathbf{U}_{xx} B_5 + \mathbf{U}_{yy} B_6 + \mathbf{U}_{zz} B_7 + \mathbf{U}_{xy} B_8 + \mathbf{U}_{xz} B_9 + \mathbf{U}_{yz} B_{10}$$



$$\mathbf{U}_{\mathbf{x}} = \frac{\partial \mathbf{U}}{\partial x}\Big|_c \Delta x, \mathbf{U}_y = \frac{\partial \mathbf{U}}{\partial y}\Big|_c \Delta y, \mathbf{U}_z = \frac{\partial \mathbf{U}}{\partial z}\Big|_c \Delta z,$$

$$\mathbf{U}_{xx} = \frac{\partial^2 \mathbf{U}}{\partial x^2}\Big|_c \Delta x^2, \mathbf{U}_{yy} = \frac{\partial^2 \mathbf{U}}{\partial y^2}\Big|_c \Delta y^2, \mathbf{U}_{zz} = \frac{\partial^2 \mathbf{U}}{\partial z^2}\Big|_c \Delta z^2,$$

$$\mathbf{U}_{xy} = \frac{\partial^2 \mathbf{U}}{\partial x \partial y}\Big|_c \Delta x \Delta y, \mathbf{U}_{xz} = \frac{\partial^2 \mathbf{U}}{\partial x \partial z}\Big|_c \Delta x \Delta z, \mathbf{U}_{yz} = \frac{\partial^2 \mathbf{U}}{\partial y \partial z}\Big|_c \Delta y \Delta z$$

The unknowns are the cell-averaged conservative variables and their derivatives at the center of the cells, regardless of element shapes.

DG(P_1) approximation

N =4, $$\mathbf{U}_h = \tilde{\mathbf{U}} B_1 + \mathbf{U}_x B_2 + \mathbf{U}_y B_3 + \mathbf{U}_z B_4$$

$$\frac{d}{dt}\int_{\Omega_e} \tilde{\mathbf{U}} d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h)\mathbf{n}_k \, d\Gamma = 0, \qquad i = 1$$

$$\begin{pmatrix} \int B_2 B_2 d\Omega & \int B_2 B_3 d\Omega & \int B_2 B_4 d\Omega \\ \int B_3 B_2 d\Omega & \int B_3 B_3 d\Omega & \int B_3 B_4 d\Omega \\ \int B_4 B_2 d\Omega & \int B_4 B_3 d\Omega & \int B_4 B_4 d\Omega \end{pmatrix} \begin{pmatrix} \frac{d\mathbf{U}_x}{dt} \\ \frac{d\mathbf{U}_y}{dt} \\ \frac{d\mathbf{U}_z}{dt} \end{pmatrix} + \begin{pmatrix} \int_{\Gamma_e} F_k n_k B_2 d\Gamma - \int_{\Omega_e} F_k \frac{\partial B_2}{\partial x_k} d\Omega \\ \int_{\Gamma_e} F_k n_k B_3 d\Gamma - \int_{\Omega_e} F_k \frac{\partial B_3}{\partial x_k} d\Omega \\ \int_{\Gamma_e} F_k n_k B_4 d\Gamma - \int_{\Omega_e} F_k \frac{\partial B_4}{\partial x_k} d\Omega \end{pmatrix} = 0$$

Features:

- A finite volume code can be easily converted to a DG code.
- Same approximate polynomial solution for any shapes of elements:
  - Can be easily extended and implemented on arbitrary meshes.
- Cell-averaged variables and their derivatives are handily available:
  - Make implementation of WENO reconstruction easy and efficient
- Hierarchic basis
  - Make implementation of *p*-multigrid methods and *p*-refinement easy and efficient

WENO reconstruction at P_2: WENO(P_1P_2)

- Objective:
  - Reconstruct a quadratic polynomial solution (P_2) from the underlying discontinuous linear polynomial DG polynomial solution (P_1) based on a WENO reconstruction.

2-exactness Least-squares Reconstruction

- From a linear polynomial DG solution in any cell i

$$\mathbf{U}_i = \tilde{\mathbf{U}}_i + \mathbf{U}_{\mathbf{x}i} B_2 + \mathbf{U}_{\mathbf{y}\mathbf{i}} B_3 + \mathbf{U}_{zi} B_4$$

- Reconstruct a quadratic polynomial solution U^R

$$\begin{aligned}\mathbf{U}_i^R = \tilde{\mathbf{U}}_i^R &+ \mathbf{U}_{xi}^R B_2 + \mathbf{U}_{yi}^R B_3 + \mathbf{U}_{zi}^R B_4 \\ &+ \mathbf{U}_{xxi}^R B_5 + \mathbf{U}_{yyi}^R B_6 + \mathbf{U}_{zzi}^R B_7 \\ &+ \mathbf{U}_{xyi}^R B_8 + \mathbf{U}_{xzi}^R B_9 + \mathbf{U}_{yzi}^R B_{10}\end{aligned}$$

- 10 degrees of freedom

Requirements for Reconstruction

- Conservation
- Compactness
  - Maintain the compactness of the underlying DG method
  - Necessary for unstructured arbitrary grids
  - Stencils involve only Von Neumann neighborhood (adjacent face-neighboring cells)

Least-squares Reconstruction (P_1P_2)

- Requiring conservation and reconstructed first derivatives equal to the ones of the underlying DG solution leads

$$\tilde{\mathbf{U}}_i^R = \tilde{\mathbf{U}}_i \,, \quad \mathbf{U}_{xi}^R = \mathbf{U}_{xi} \,, \quad \mathbf{U}_{yi}^R = \mathbf{U}_{yi} \,, \quad \mathbf{U}_{zi}^R = \mathbf{U}_{zi}$$

due to the judicious choice of Taylor-basis in the DG formulation

- Six second derivatives only need to be reconstructed

Least-squares Reconstruction (P_1P_2)

$$\mathbf{U}_j = \tilde{\mathbf{U}}_i + \mathbf{U}_{xi} B_2 + \mathbf{U}_{yi} B_3 + \mathbf{U}_{zi} B_4$$

$$+ \mathbf{U}^R_{xxi} B_5 + \mathbf{U}^R_{yyi} B_6 + \mathbf{U}^R_{zzi} B_7 + \mathbf{U}^R_{xyi} B_8 + \mathbf{U}^R_{xzi} B_9 + \mathbf{U}^R_{yzi} B_{10}$$

$$\frac{\partial \mathbf{U}}{\partial x}\Big|_j = \mathbf{U}_{xi} \frac{1}{\Delta x_i} + \mathbf{U}^R_{xxi} \frac{B_2}{\Delta x_i} + \mathbf{U}^R_{xyi} \frac{B_3}{\Delta x_i} + \mathbf{U}^R_{xzi} \frac{B_4}{\Delta x_i}$$

$$\frac{\partial \mathbf{U}}{\partial y}\Big|_j = \mathbf{U}_{yi} \frac{1}{\Delta y_i} + \mathbf{U}^R_{yyi} \frac{B_3}{\Delta y_i} + \mathbf{U}^R_{xyi} \frac{B_2}{\Delta y_i} + \mathbf{U}^R_{yzi} \frac{B_4}{\Delta y_i}$$

$$\frac{\partial \mathbf{U}}{\partial z}\Big|_j = \mathbf{U}_{zi} \frac{1}{\Delta z_i} + \mathbf{U}^R_{zzi} \frac{B_4}{\Delta z_i} + \mathbf{U}^R_{xzi} \frac{B_2}{\Delta z_i} + \mathbf{U}^R_{yzi} \frac{B_3}{\Delta z_i}$$

Least-squares Reconstruction (P_1P_2)

- Similar equations can be written for all face-neighboring cells, which leads to a non-square matrix.
- The number of face-neighboring cells for a tetrahedral cell is 4. As a result, the size of resulting non-square matrix is 16×6.
- This over-determined linear system of 16 equations for 6 unknowns can be solved in the least-squares sense.

Instability Issues of RDG Method in 3D

- The RDG method suffers from linear instability, similar to RDG(P_0P_1)
  - Instability occurs even for linear equations and in smooth flows.
  - Reconstruction stencils only involve von Neumann neighborhood, i.e., adjacent face-neighboring cells

- To maintain the linear stability
  - Augment stencils in the reconstruction
    → Destroy the compactness of the underlying DGM.
  - Use non-linear stability enforcement to achieve linear stability
    → Limiters in the case of RDG(P_0P_1) method
    → ENO/WENO reconstructions.

Hermite WENO Reconstruction

- On a tetrahedral cell *i*, a convex combination of the least-squares reconstructed 2nd order derivatives at the cell itself and its four face-neighboring cells

$$\frac{\partial^2 U}{\partial x_i \partial x_j}\Big|_i = \sum_{k=1}^{5} w_k \frac{\partial^2 U}{\partial x_i \partial x_j}\Big|_k$$

w_k: weighting function

$$w_k = \frac{(\varepsilon + o_k)^{-\gamma}}{\sum_{i=1}^{5} (\varepsilon + o_i)^{-\gamma}}$$

o_k : oscillation indicator

$$o_k = [\int_{\Omega_i} (\frac{\partial^2 U}{\partial x_i \partial x_j}\Big|_k)^2 d\Omega]^2$$

ε → a small positive number

γ → an integer parameter

Hermite WENO Reconstruction

- Central stencil

  → the least-squares reconstructed polynomial at the cell itself

- Biased stencils

  → the least-squares reconstructed polynomials on its four face-neighboring cells

WENO reconstruction at P_1 : HWENO(P_1P_2)

- Observation:

  Although the WENO(P_1P_2) method does not introduce any new oscillatory behavior for the reconstructed curvature terms (second derivatives) due to the WENO reconstruction, it cannot remove inherent oscillations in the underlying DG(P1) solutions, leading to non-linear instability.

- Objective:

  Reconstruct and modify the linear part (first derivatives) of the resulting quadratic polynomial solution (P_2) in order to ensure non-linear instability for flows with strong discontinuities using WENO reconstruction.

WENO reconstruction at P_1 : HWENO(P_1P_2)

- For a cell i, , the following four stencils (i,j1,j2,j3), (i,j1,j2,j4), (ie,j1,j3,j4), and (i,j2,j3,j4), where j1, j2, j3, and j4 designate the four adjacent face-neighboring cells of the cell i, are chosen to construct a Lagrange polynomial such that

$$\mathbf{U}_j = \tilde{\mathbf{U}}_i + \mathbf{U}^R_{xi} B_2 + \mathbf{U}^R_{yi} B_3 + \mathbf{U}^R_{zi} B_4$$
$$+ \mathbf{U}^R_{xxi} B_5 + \mathbf{U}^R_{yyi} B_6 + \mathbf{U}^R_{zzi} B_7 + \mathbf{U}^R_{xyi} B_8 + \mathbf{U}^R_{xzi} B_9 + \mathbf{U}^R_{yzi} B_{10}$$

WENO reconstruction at P_1 : HWENO(P_1P_2)

- The following four stencils (i,j1), (i,j2), (i,j3), and (i,j4) are chosen to construct a Hermite polynomial such that

$$\frac{\partial \mathbf{U}}{\partial x}\Big|_j = \mathbf{U}^R_{xi}\frac{1}{\Delta x_i} + \mathbf{U}^R_{xxi}\frac{B_2}{\Delta x_i} + \mathbf{U}^R_{xyi}\frac{B_3}{\Delta x_i} + \mathbf{U}^R_{xzi}\frac{B_4}{\Delta x_i}$$

$$\frac{\partial \mathbf{U}}{\partial y}\Big|_j = \mathbf{U}^R_{yi}\frac{1}{\Delta y_i} + \mathbf{U}^R_{yyi}\frac{B_3}{\Delta y_i} + \mathbf{U}^R_{xyi}\frac{B_2}{\Delta y_i} + \mathbf{U}^R_{yzi}\frac{B_4}{\Delta y_i}$$

$$\frac{\partial \mathbf{U}}{\partial z}\Big|_j = \mathbf{U}^R_{zi}\frac{1}{\Delta z_i} + \mathbf{U}^R_{zzi}\frac{B_4}{\Delta z_i} + \mathbf{U}^R_{xzi}\frac{B_2}{\Delta z_i} + \mathbf{U}^R_{yzi}\frac{B_3}{\Delta z_i}$$

Hermite WENO Reconstruction

- On a tetrahedral cell *i*, a convex combination of these eight reconstructed 1st derivatives and the first derivatives at the cell itself is used to modify the first derivatives

$$\frac{\partial U}{\partial x_i}\Big|_i = \sum_{k=1}^{9} w_k \frac{\partial U}{\partial x_i}\Big|_k$$

w_k: weighting function

$$w_k = \frac{(\varepsilon + o_k)^{-\gamma}}{\sum_{i=1}^{9} (\varepsilon + o_i)^{-\gamma}}$$

o_k : oscillation indicator

$$o_k = [\int_{\Omega_i} (\frac{\partial U}{\partial x_i}\Big|_k)^2 d\Omega]^{1/2}$$

ε → a small positive number

γ → an integer parameter

Hermite WENO Reconstruction

- Central stencil

  → the gradient from the DG solution itself at cell itself

- Biased stencils

  → the eight reconstructed gradients

Cost Analysis (Tetrahedral Grids)

•

	RDG(P_0P_1)	RDG(P_1P_1)	RDG(P_1P_2)	RDG(P_2P_2)
Number of quadrature points for boundary integrals	1	3	4	7
Number of quadrature points for domain integrals	0	4	5	11
Reconstruction	Yes	No	Yes	No
Order of Accuracy	$O(h^2)$	$O(h^2)$	$O(h^3)$	$O(h^3)$
Storage for Implicit Diagonal Matrix	25 words Per element	400	400	2500

Design Goal

FVFLO	RDGFLO
Unstructured triangular/tetrahedral grid	Unstructured arbitrary grid
Finite volume/finite element formulation	Reconstructed Discontinuous Galerkin formulation
h-adaptation	*hp*-adaptation
None	Error Estimation Uncertainty Quantification

RDGFLO

- Physics
  - Compressible flow for all speeds
  - Inviscid, Laminar, Turbulent (RDG(P_0P_1))
  - Chemically reactive flows (RDG(P_0P_1))

- Numerics
  - Unstructured Hybrid Mesh(tetrahedral, pyramidal, prismatic, and hexahedral)
  - Reconstructed Discontinuous Galerkin Formulation
  - Implicit (SGS, GMRES+LU-SGS)/Explicit (RK) Time Integration
  - *p*-multigrid
  - HLLC, LDFSS, AUSM for Inviscid Fluxes
  - BR2, RDG for Viscous Fluxes
  - BGK for Inviscid+Viscou Fluxes

- Parallelization
  - MPI

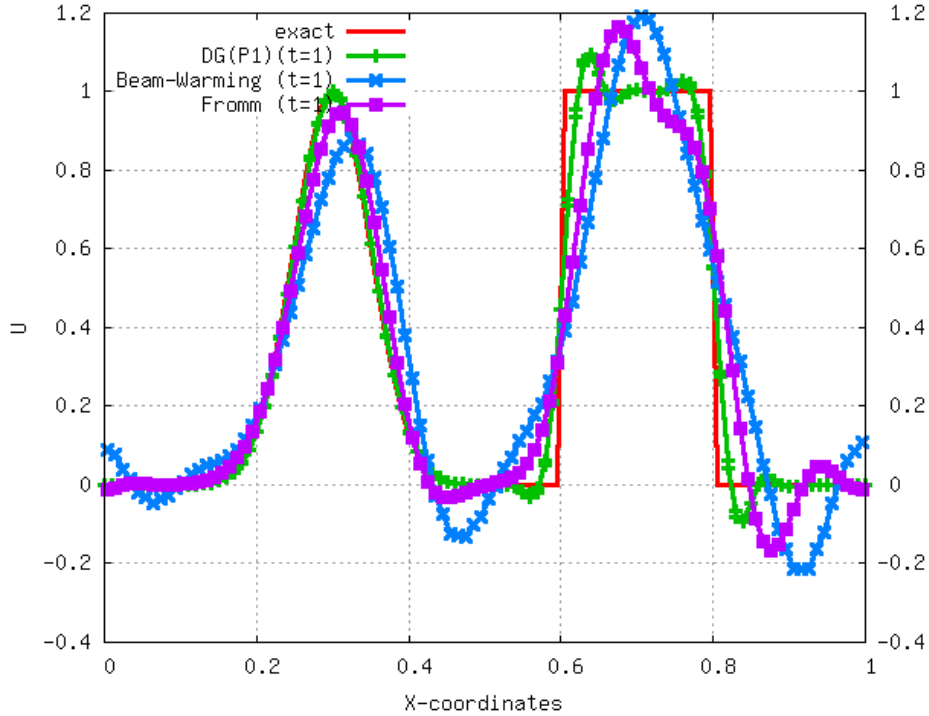
Numerical Examples

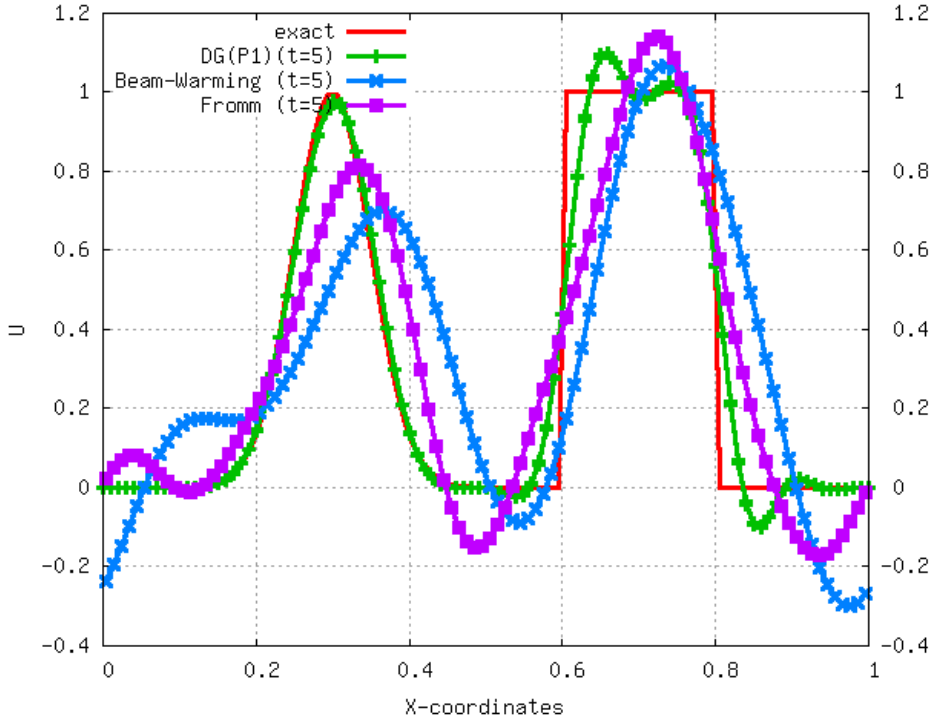
- Strengths of the RDG method
  - Accuracy
  - Robustness
  - Essentially oscillation-free property

Strengths of the RDG method

Accuracy Demonstration

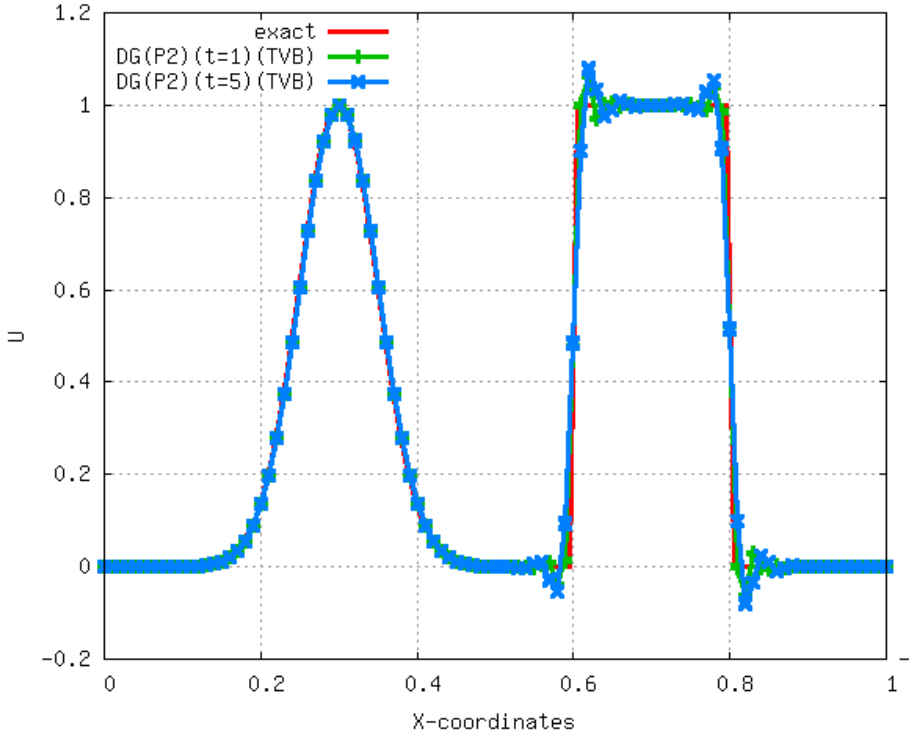
Convection of a Gaussian and a square wave

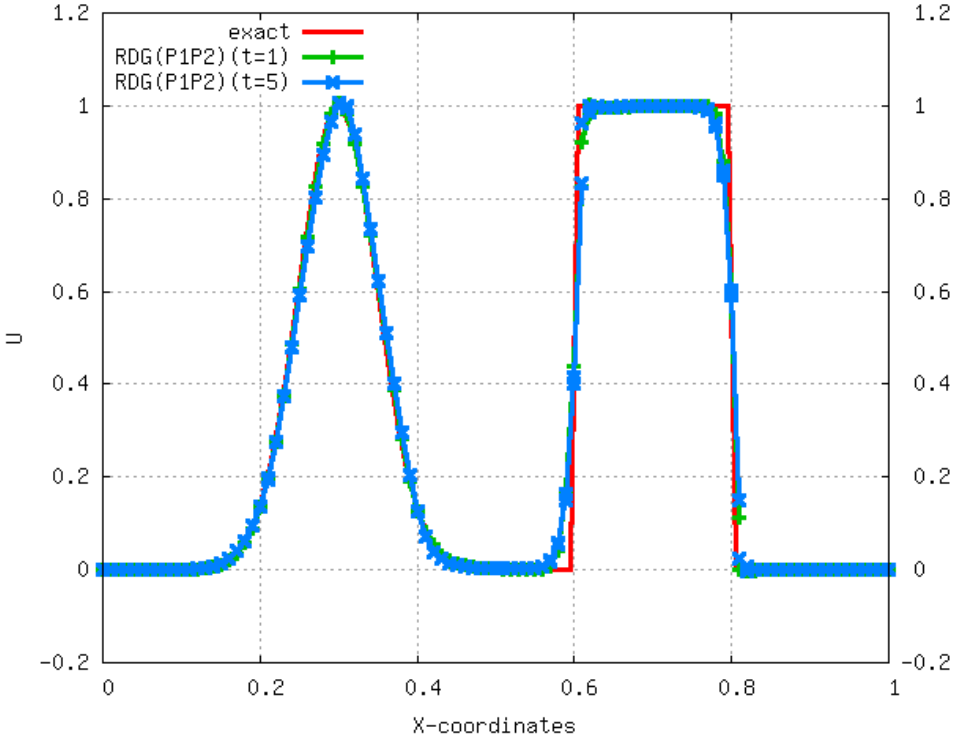




The superior dissipation and dispersion property of DG !

Convection of a Gaussian and a square wave

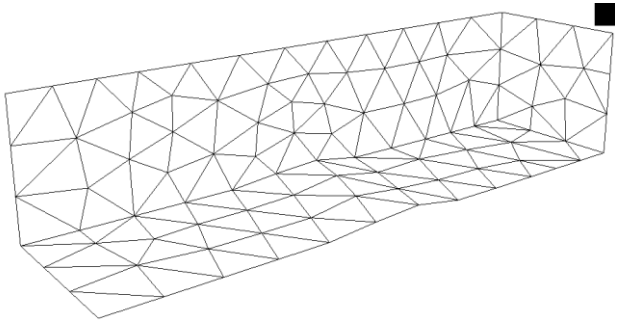




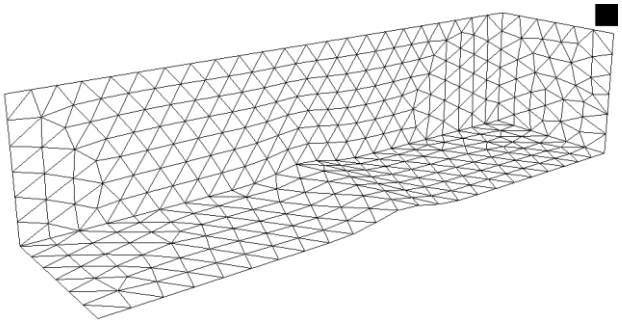
Note the high accuracy and oscillation-free of the RDG !

A Subsonic Flow through a Channel with a Smooth Bump (M_∞=0.5, α=0°)

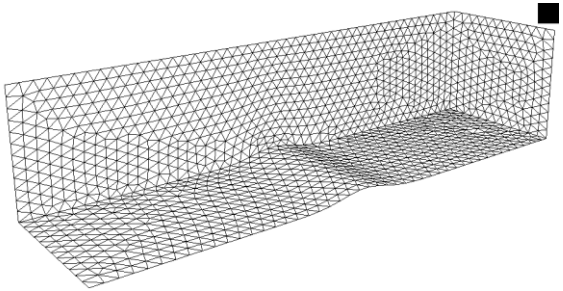
- Access the order of accuracy of the RDG(P_1P_1), WENO(P_1P_2) and HWENO(P_1P_2) methods for internal flows.
- Entropy production is served as the error measurement.



889 cells
254 pts
171 boundary pts

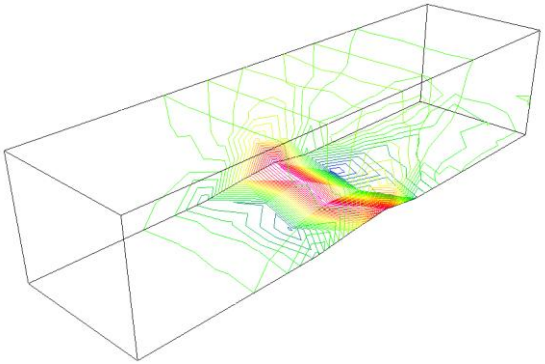


6986 cells
1555 pts
691 boundary pts

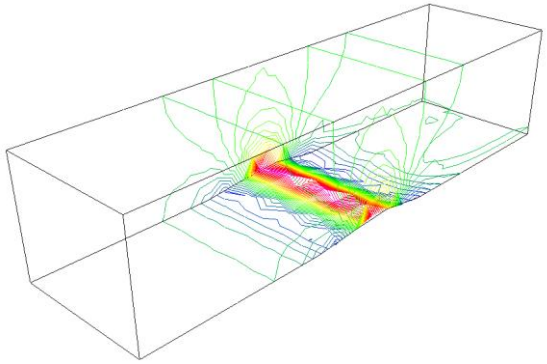


449522 cells
81567 pts
10999 boundary pts

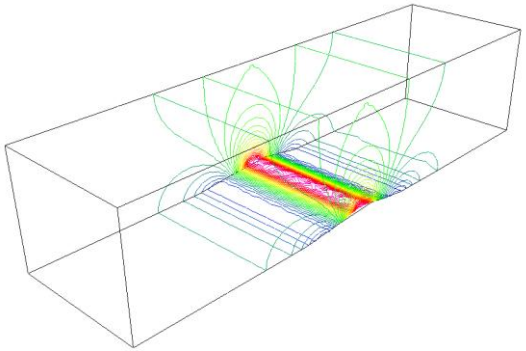
Computed Velocity Contours by HWENO(P_1P_2)



Coarse Grid



Medium Grid



Fine Grids

Convergence Study for different RDG methods

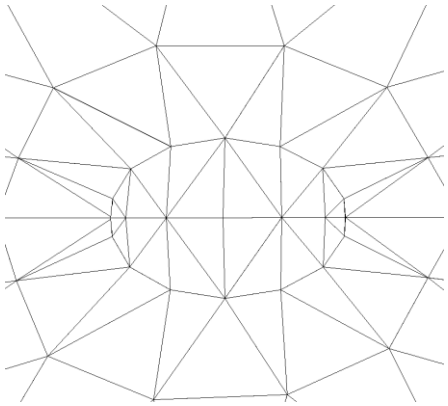
L2-error and order of convergence for the RDG(P_1P_1), WENO(P_1P_2), and HWENO(P_1P_2) methods

	RDG(P_1P_1)		WENO(P_1P_2)		HWENO(P_1P_2)	
Length Scale	L^2-error	Order	L2-error	Order	L^2-error	Order
6.552E-2	2.438E-3		2.183E-3		2.220E-3	
3.295E-2	7.356E-4	1.744	2.794E-4	2.992	2.851E-4	2.987
1.650E-2	1.807e-4	2.032	4.539E-05	2.626	4.565E-5	2.647

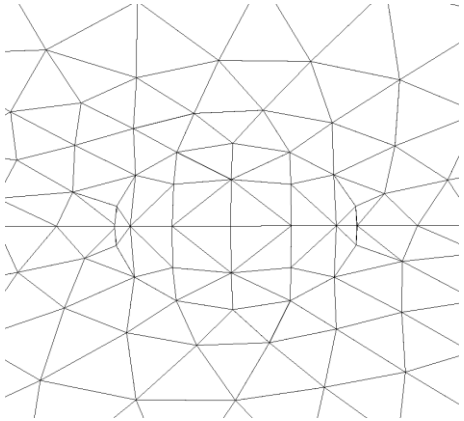
Both WENO(P_1P_2) and HWENO(P_1P_2) deliver the designed 3rd order of convergence !!

A Subsonic Flow past a Sphere (M_∞=0.5)

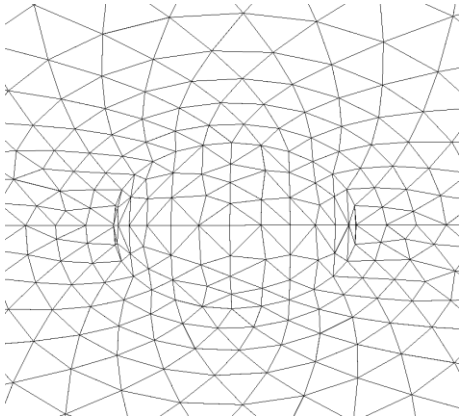
- Access the order of accuracy of the RDG(P_1P_1), WENO(P_1P_2) and HWENO(P_1P_2) methods for external flows.
- Entropy production is served as the error measurement.



535 cells
167 points
124 boundary pts

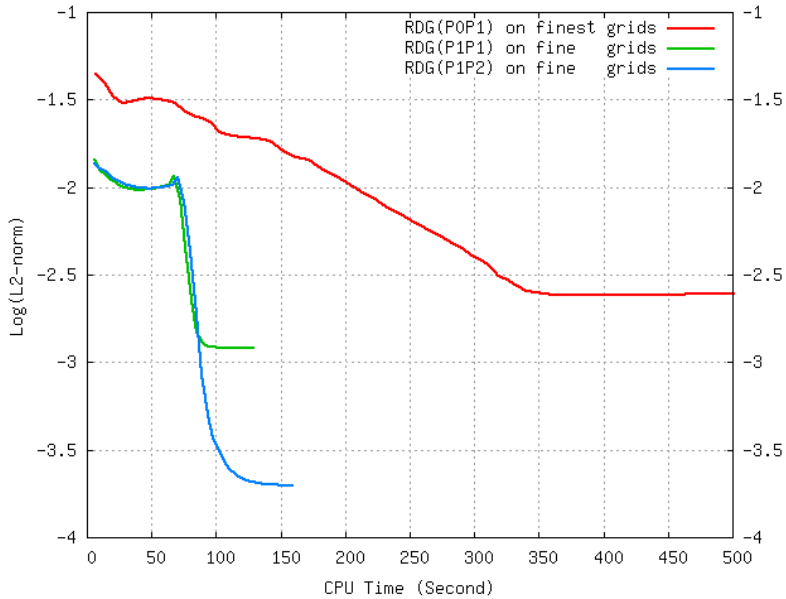


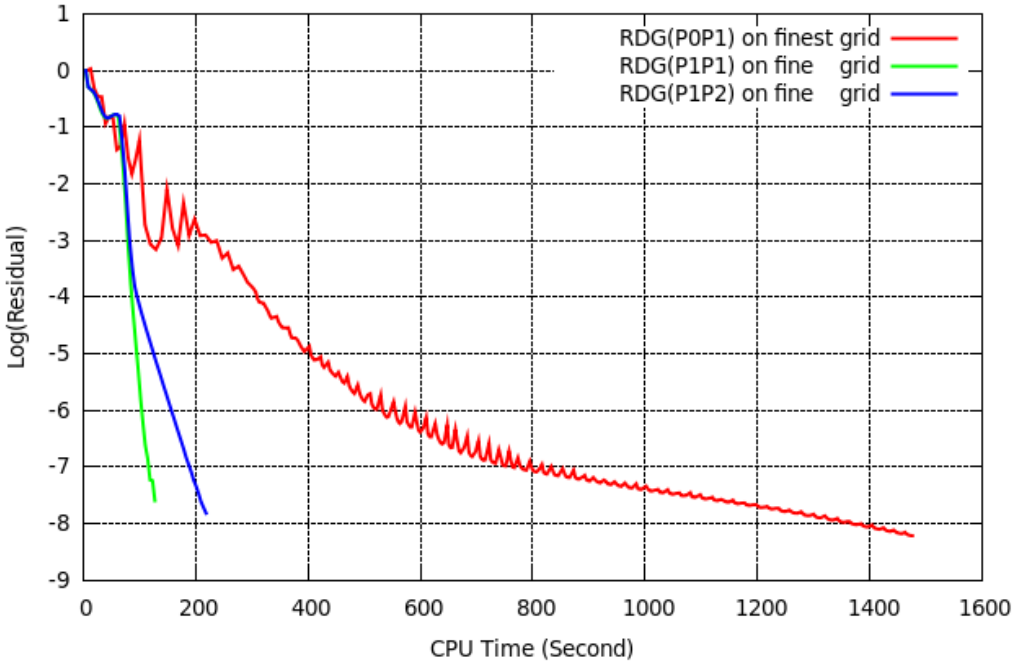
62426 cells
598 points
322 boundary pts



16467 cells
3425 points
1188 boundary pts

Efficiency Comparison for Different RDG Methods





L2 norm versus CPU time

Convergence history versus CPU time (Second)

Convergence Study for different RDG methods

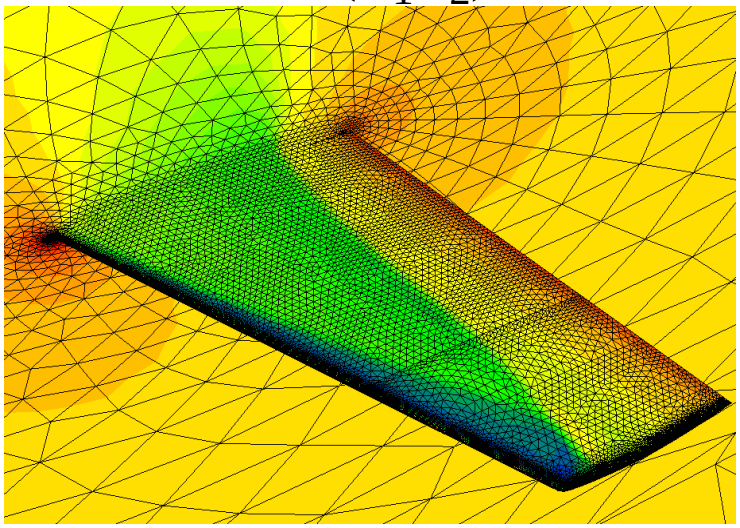
L2-error and order of convergence for the RDG(P_1P_1), WENO(P_1P_2), and HWENO(P_1P_2) methods

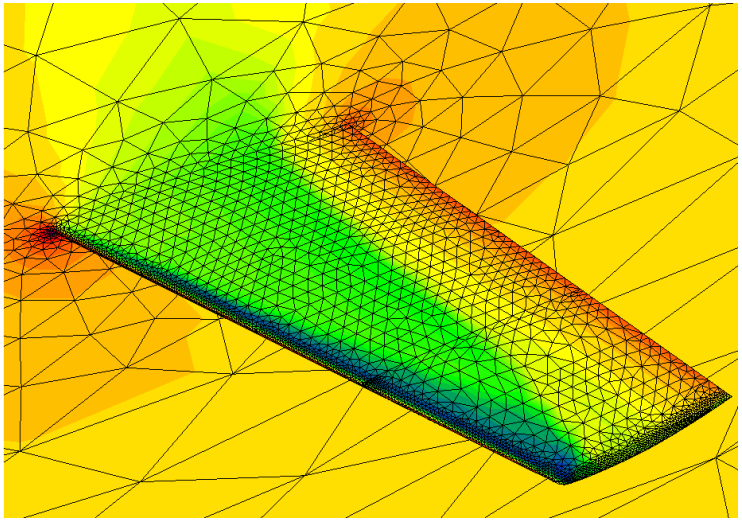
	RDG(P_1P_1)		WENO(P_1P_2)		HWENO(P_1P_2)	
Length scale	L^2-error	Order	L^2-error	Order	L^2-error	Order
7.760E-2	1.783E-2		1.052E-2		1.117E-2	
4.688E-2	5.010E-3	2.519	1.317E-3	4.124	1.503E-3	3.980
2.476E-2	1.232E-3	2.198	1.978E-4	2.964	2.201E-4	3.009

Both WENO(P_1P_2) and HWENO(P_1P_2) deliver the designed 3rd order of convergence !!

Transonic Flow past an ONERA M6 Wing (M_∞=0.84, α=3.06°)

- Access the accuracy and non-oscillatory property of the HWENO(P_1P_2) method for flows with discontinuities.

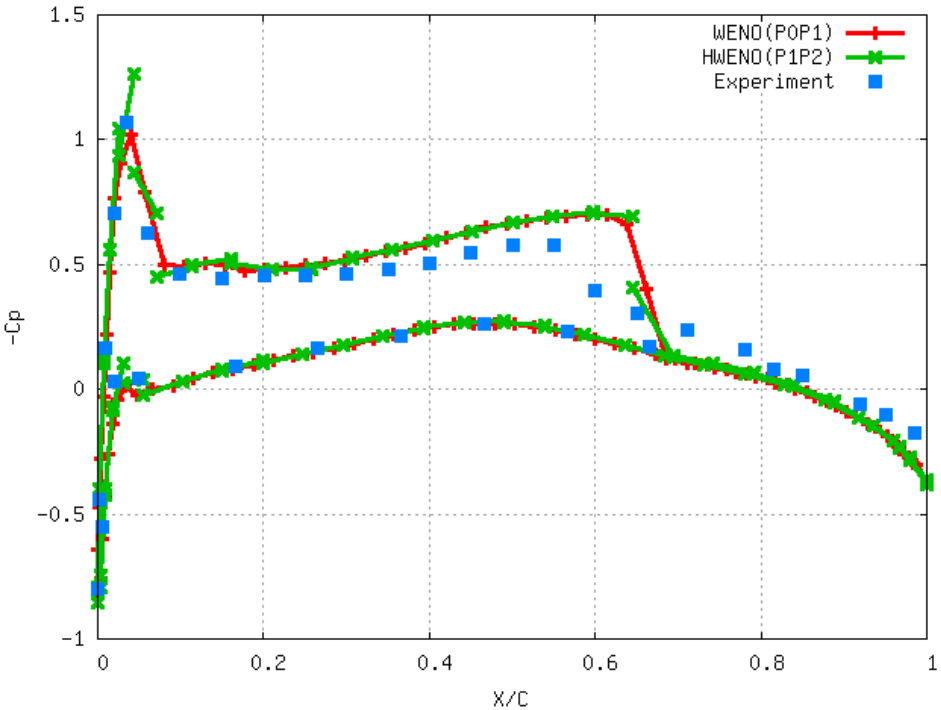


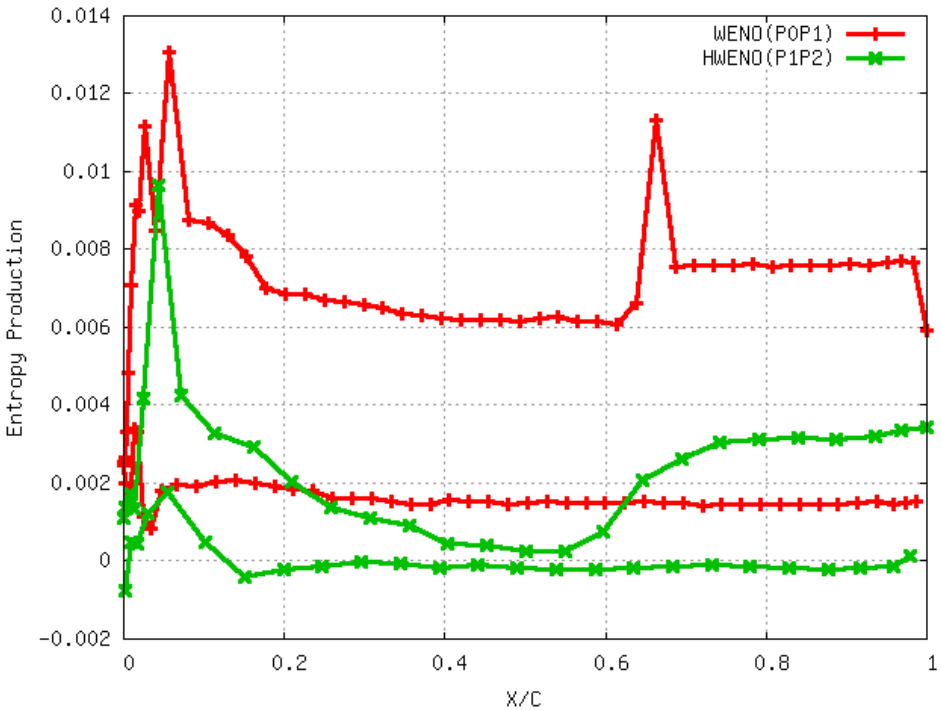


Computed Pressure Contours

WENO(P_0P_1)	HWENO(P_1P_2)
nelem = 593,169	nelem = 95,266
npoin = 110,282	npoin = 18,806
nboun = 19,887	nboun = 5,287

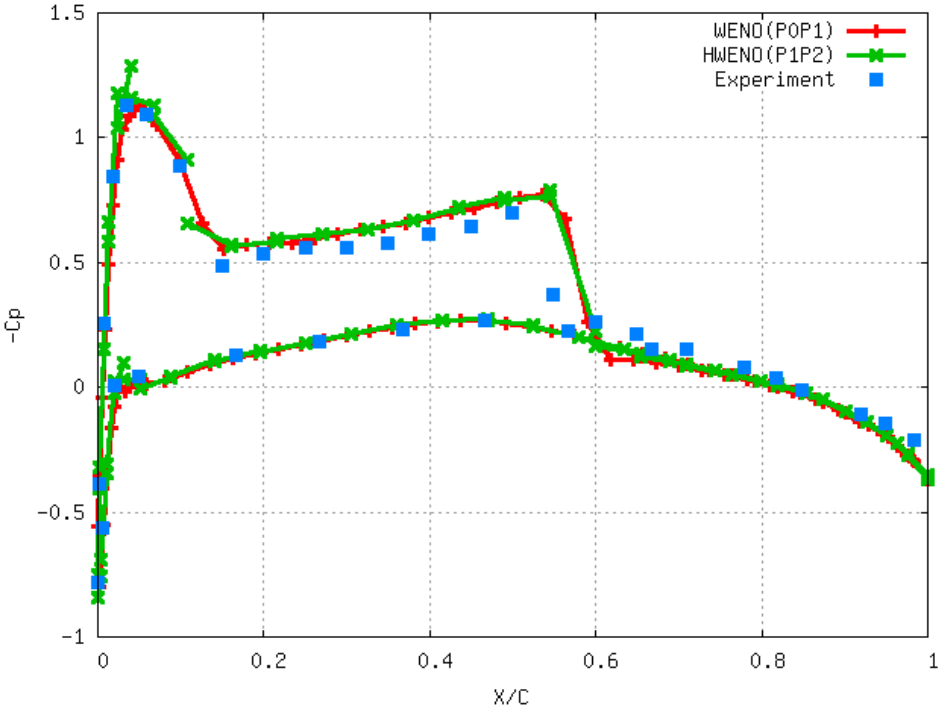
Computed Pressure Coefficient and Entropy Production Distributions at different spanwise locations

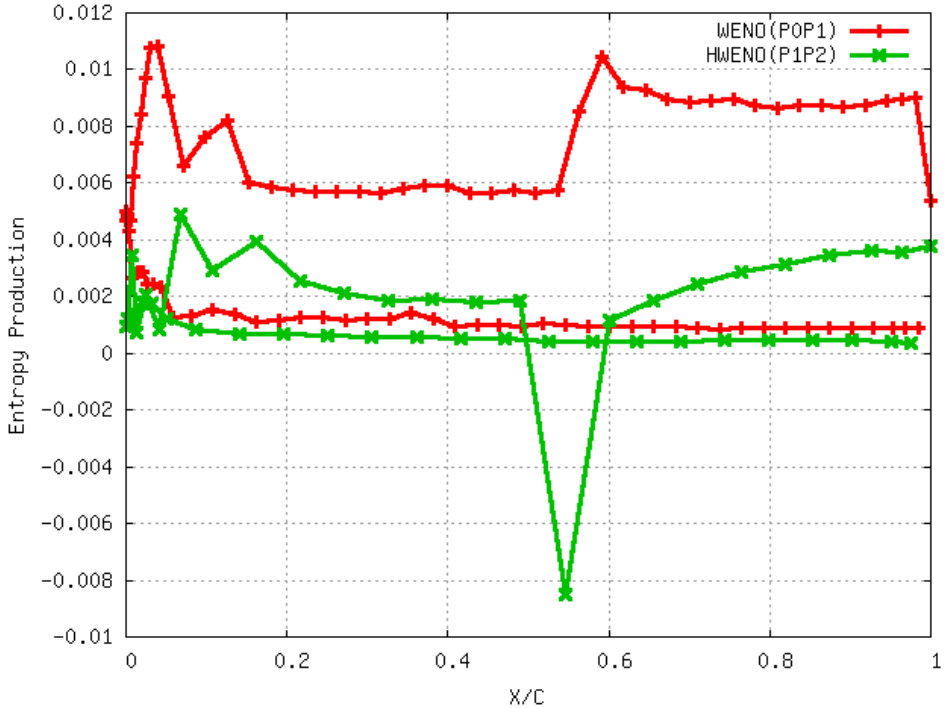




η=0.20

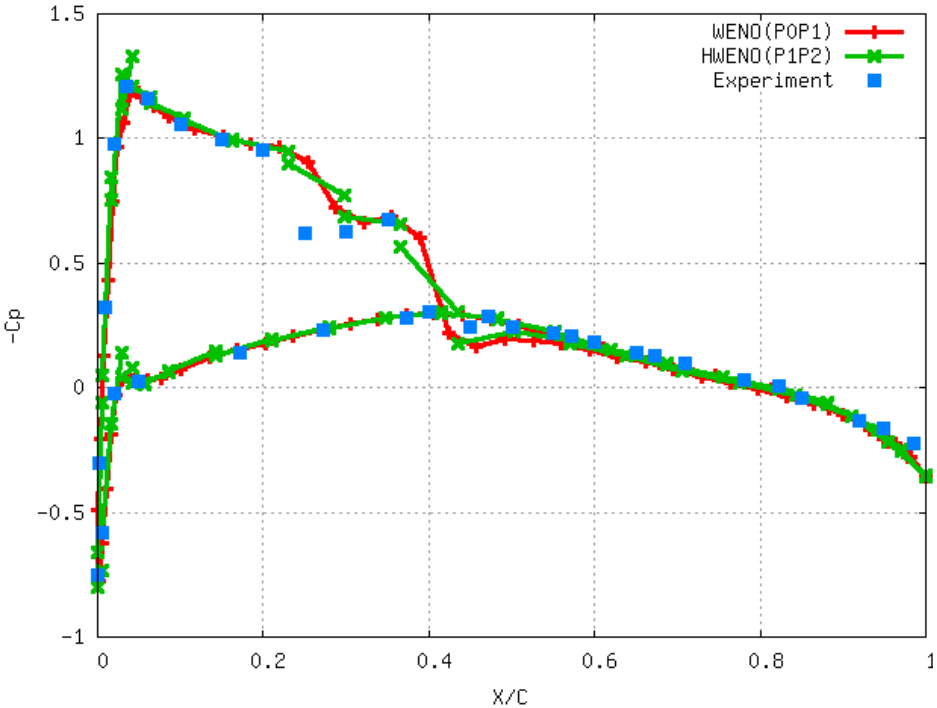
Computed Pressure Coefficient and Entropy Production Distributions at different spanwise locations

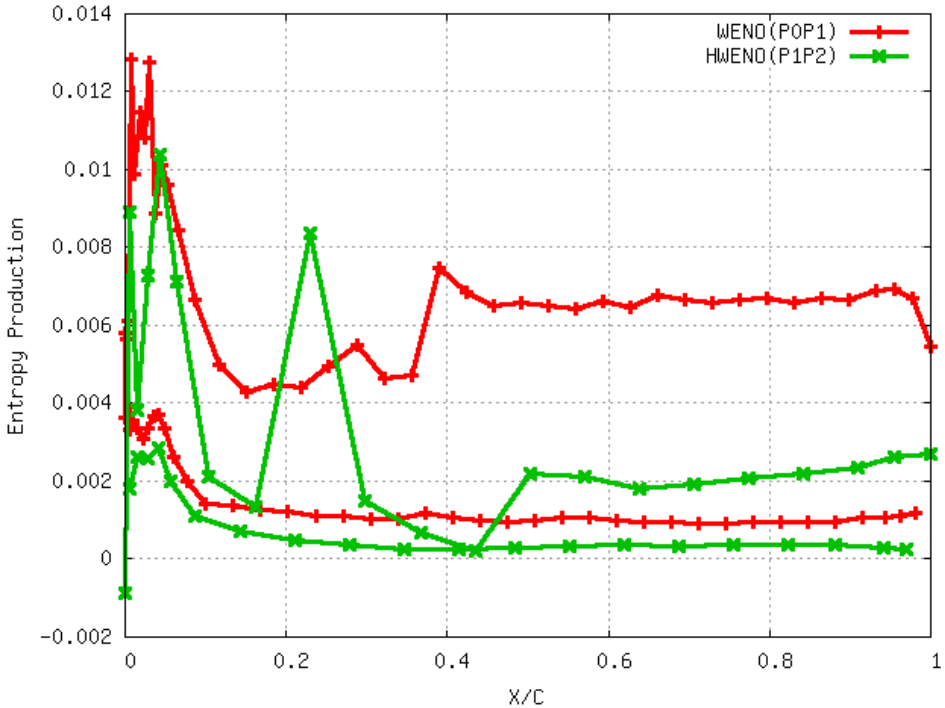




η=0.44

Computed Pressure Coefficient and Entropy Production Distributions at different spanwise locations

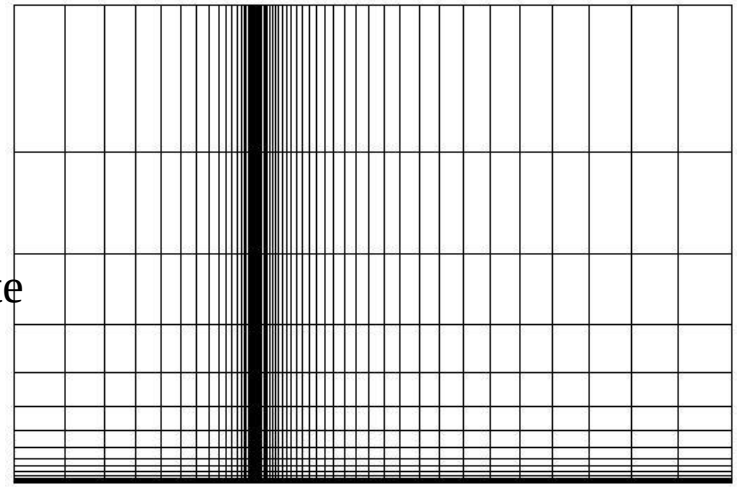




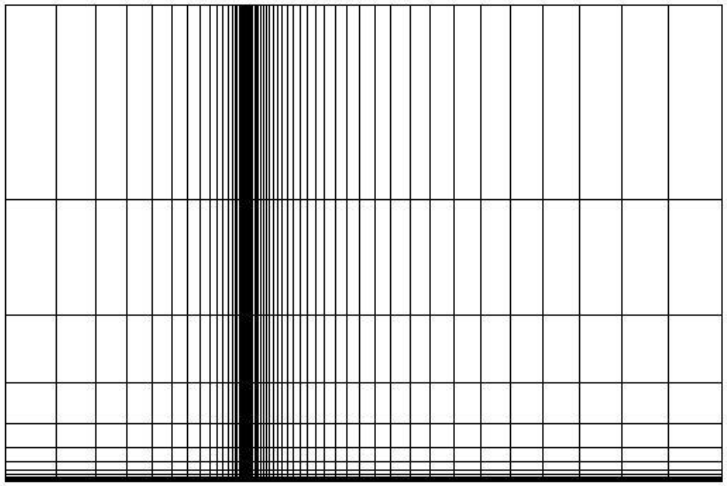
η=0.80

Blasius Boundary Layer (M=0.2, Re=100,000)

61x17x1
40 cells on the plate



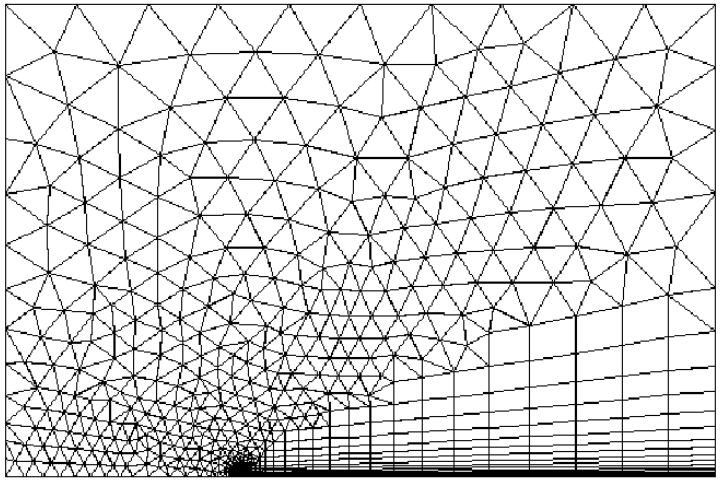
Eta =1.2
dy= 0.1291E-2

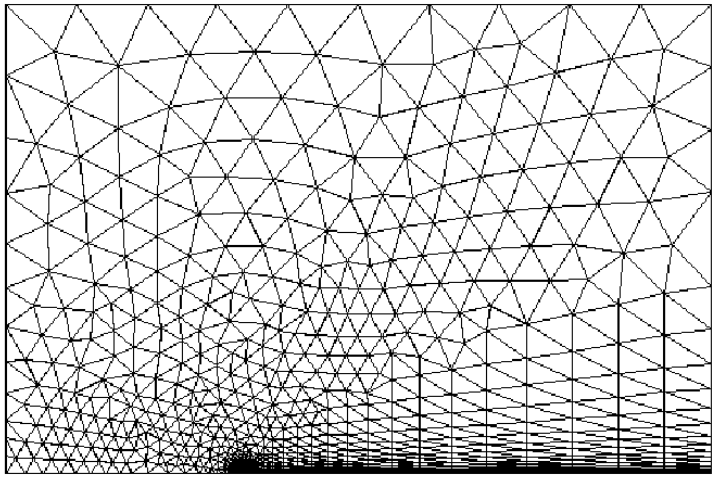


Eta=1.3
dy= 0.155869E-3

Npoin=1800
Nboun=210
62 grid points on the flat plate.
Dy=0.3464E-03 at LE,

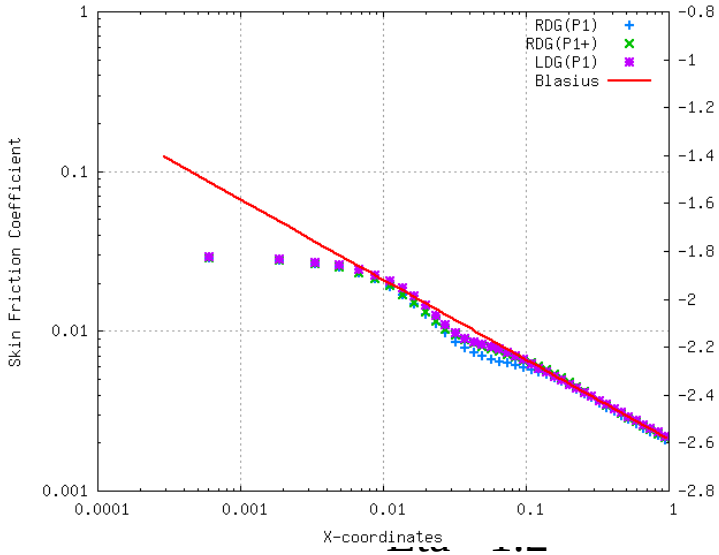
Dy=0.82649E-03 at TE

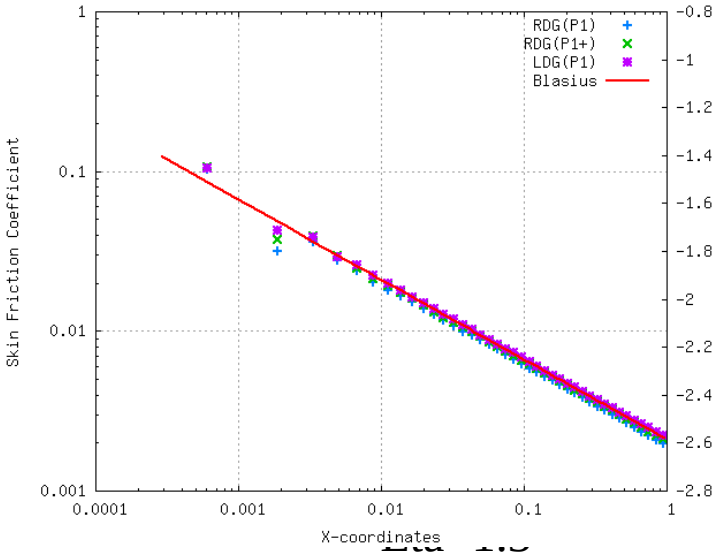




Computed skin friction coefficients (RDG(P1))

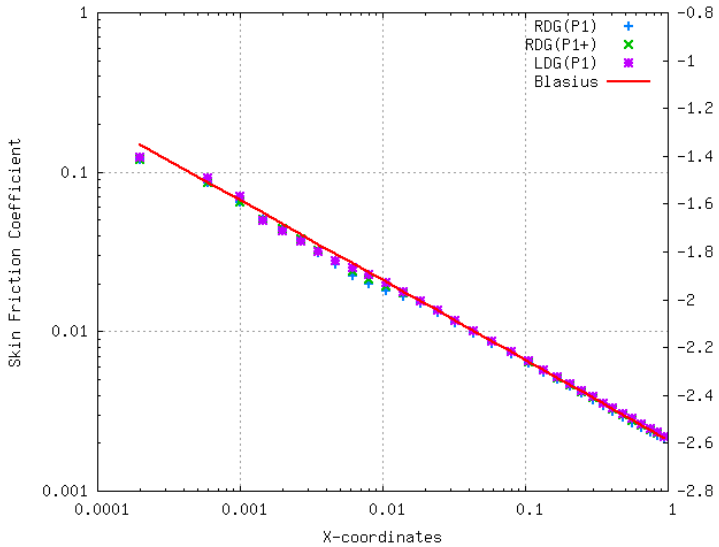
Eta=1.2

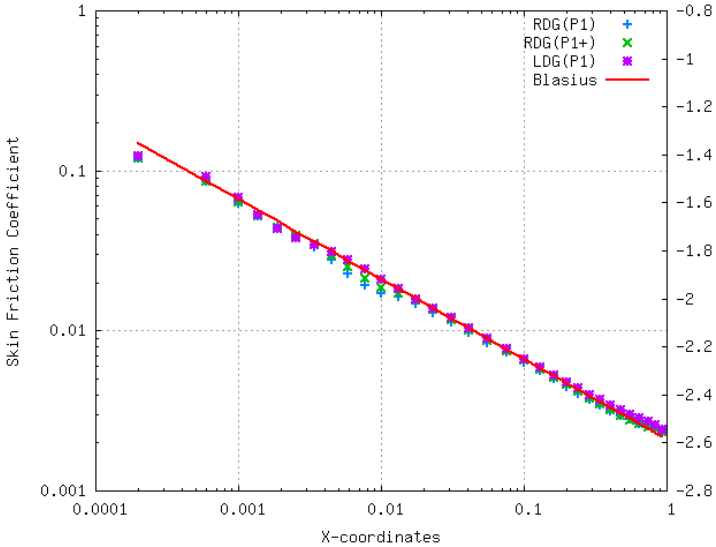




Eta=1.3

Hybrid

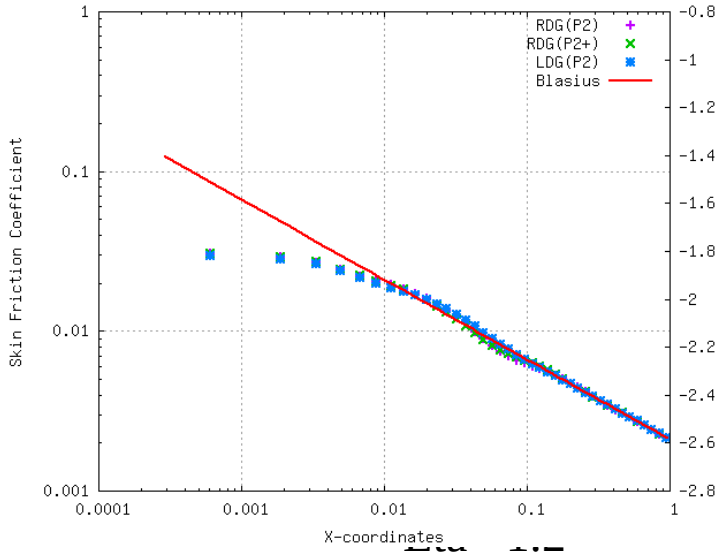


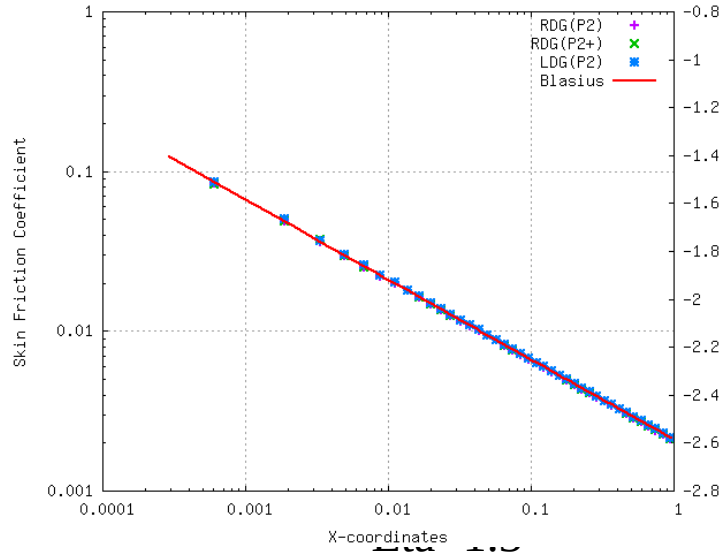


Triangular

Computed skin friction coefficients (RDG(P2))

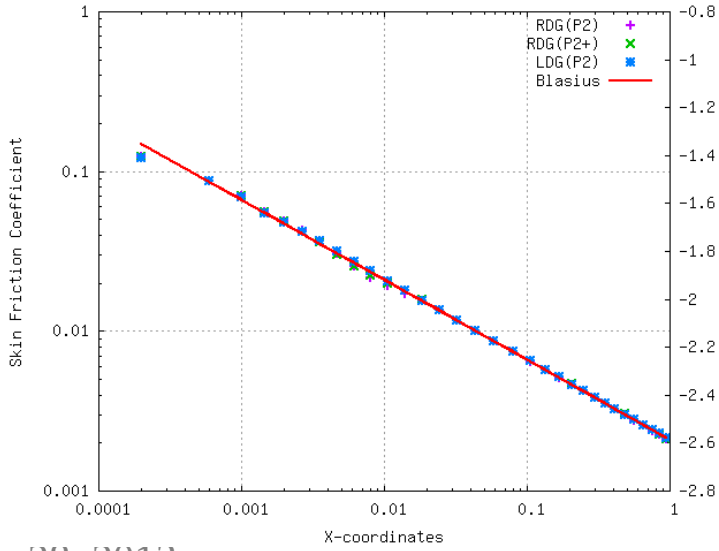
Eta=1.2

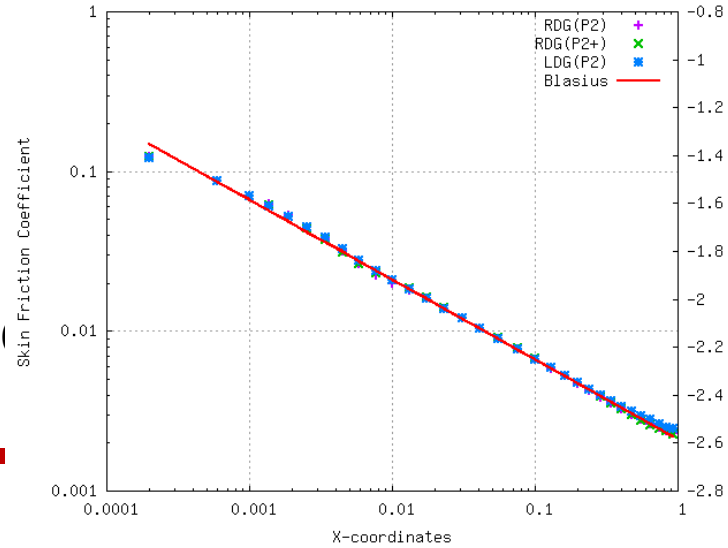




Eta=1.3

Hybrid

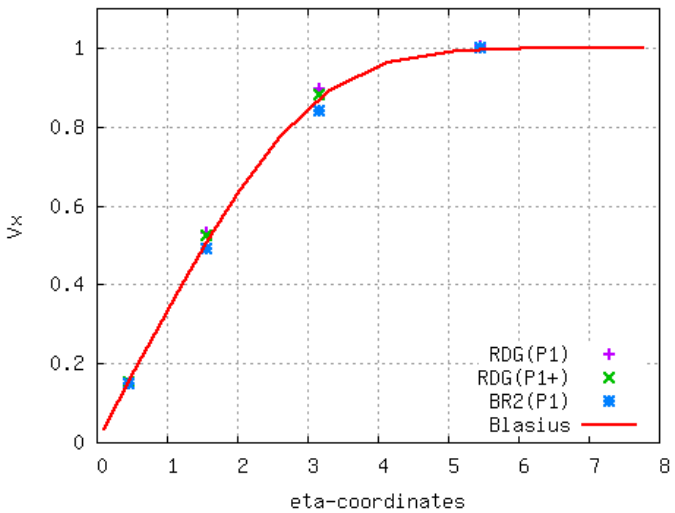


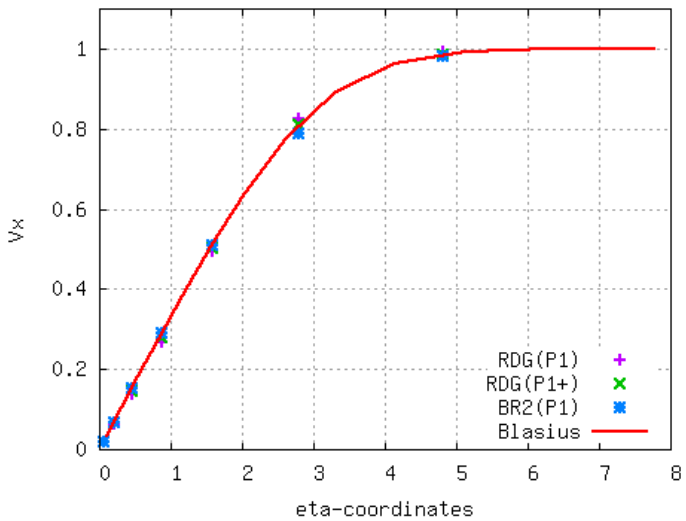


Triangular

Computed X-velocity profiles at x=0.2(RDG(P1))

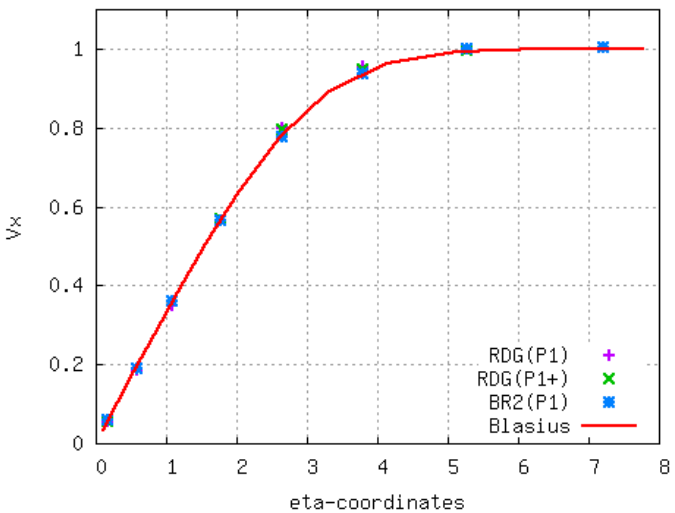
Eta=1.2

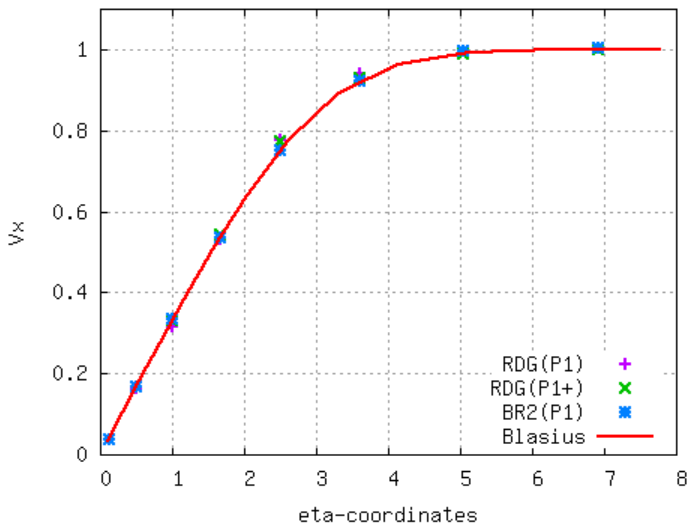




Eta=1.3

Hybrid

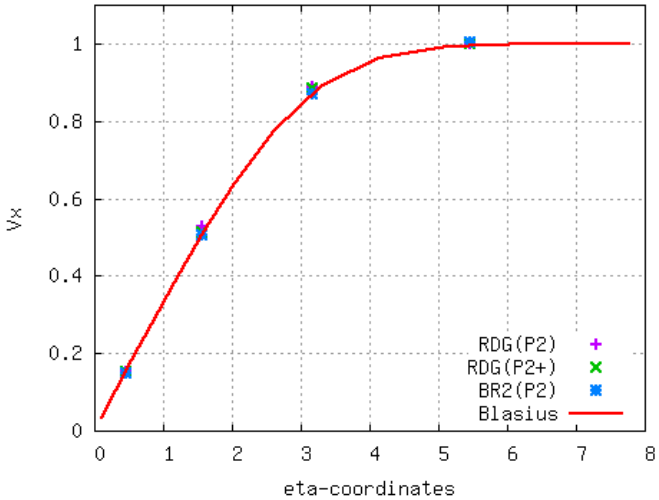


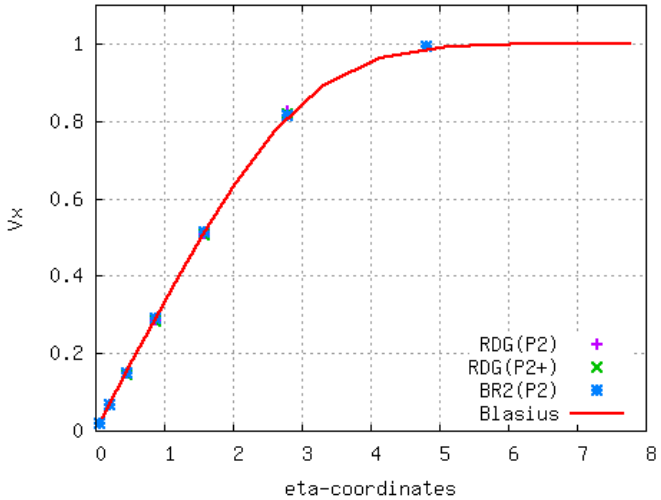


Triangular

Computed X-velocity profiles at x=0.2(RDG(P2))

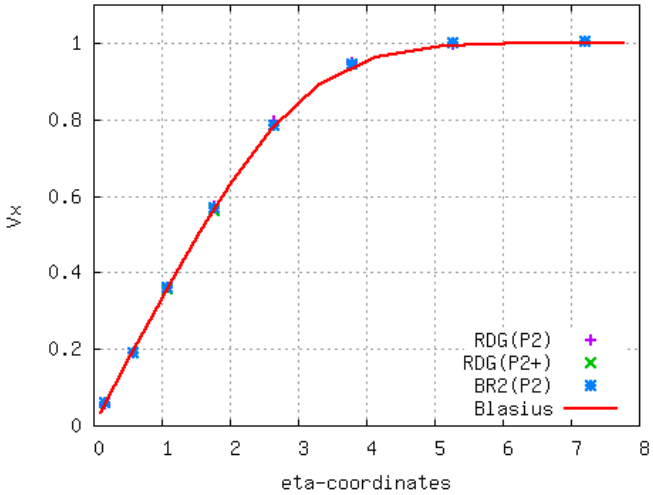
Eta=1.2

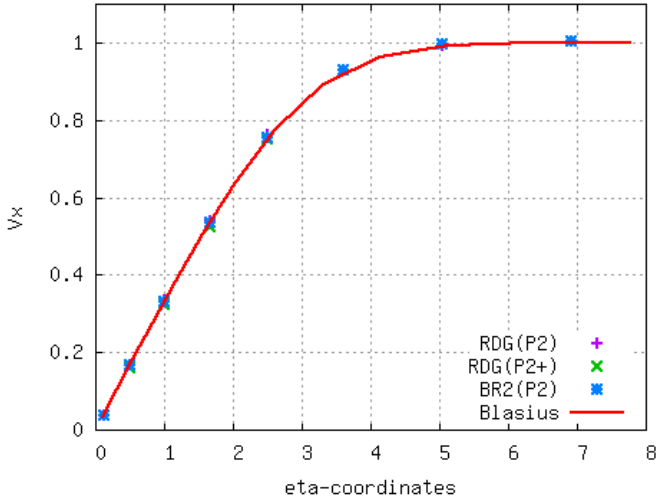




Eta=1.3

Hybrid

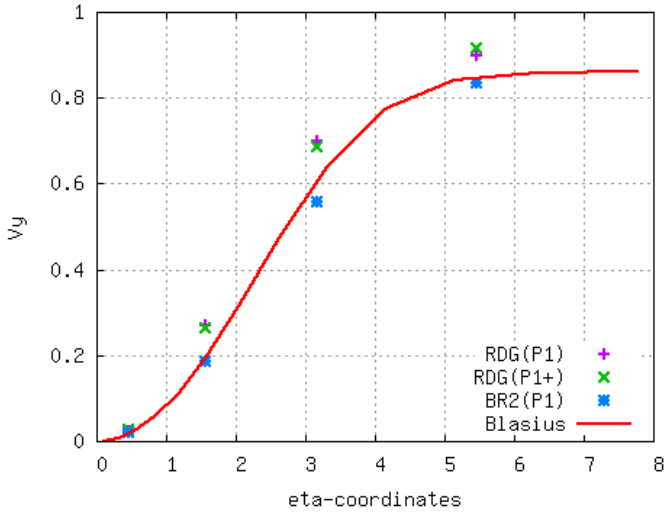


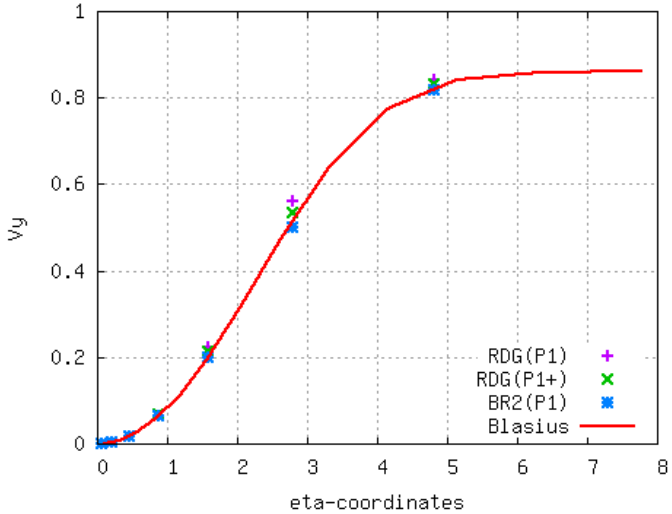


Triangular

Computed Y-velocity profiles at x=0.2(RDG(P1))

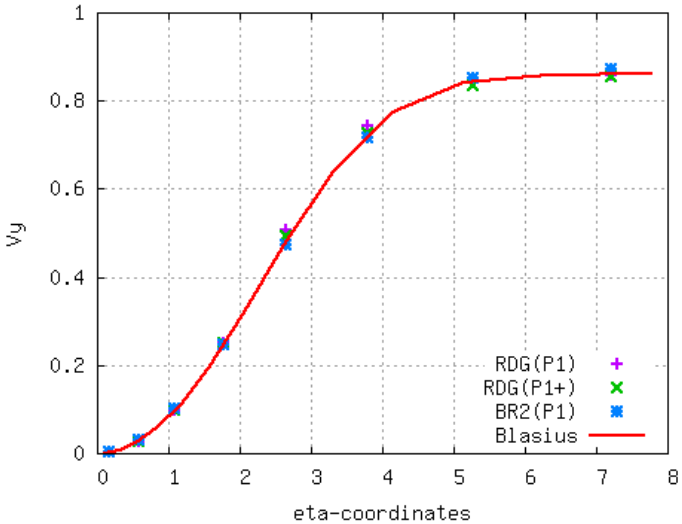
Eta=1.2

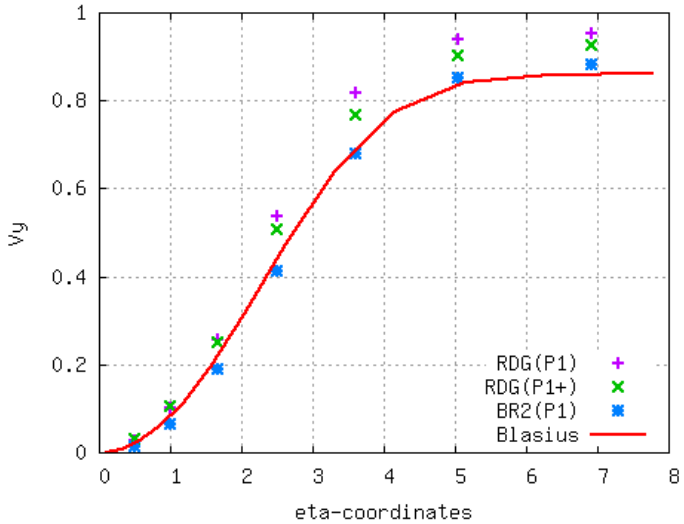




Eta=1.3

Hybrid

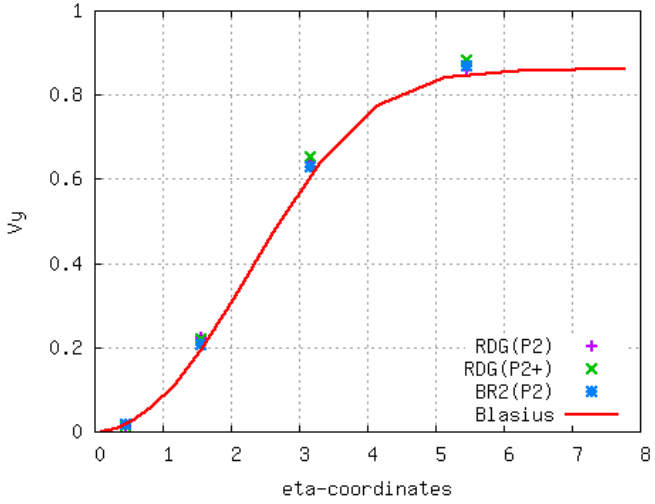


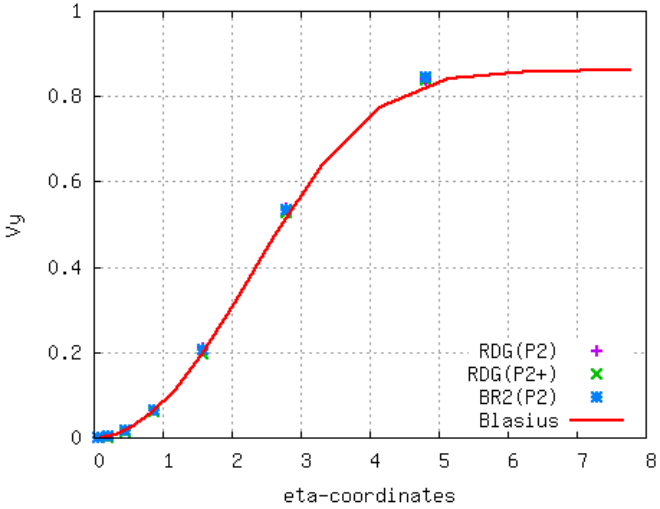


Triangular

Computed Y-velocity profiles at x=0.2(RDG(P2))

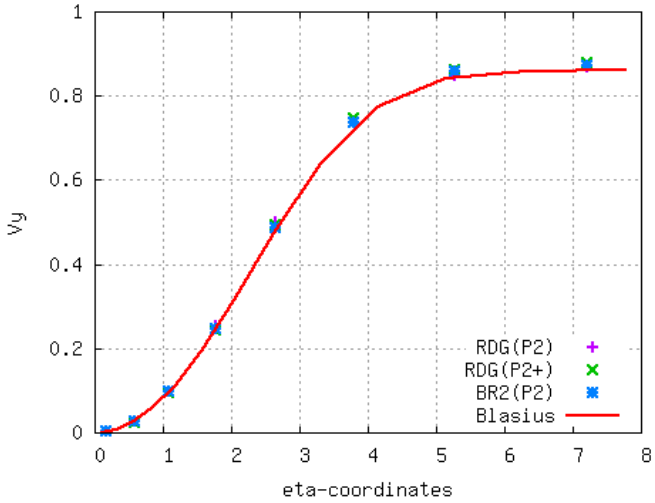
Eta=1.2

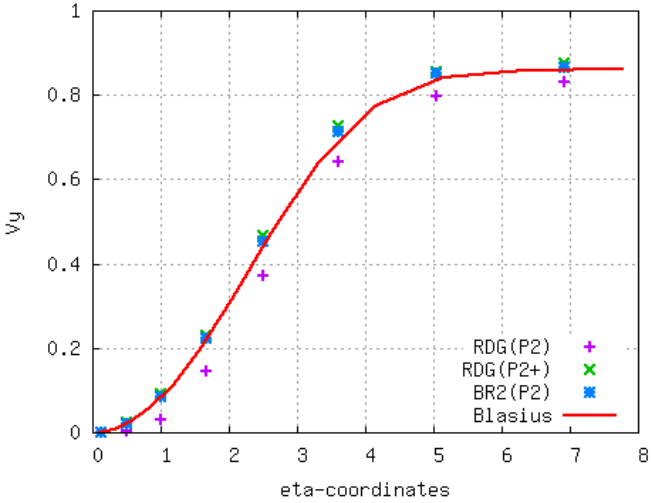




Eta=1.3

Hybrid



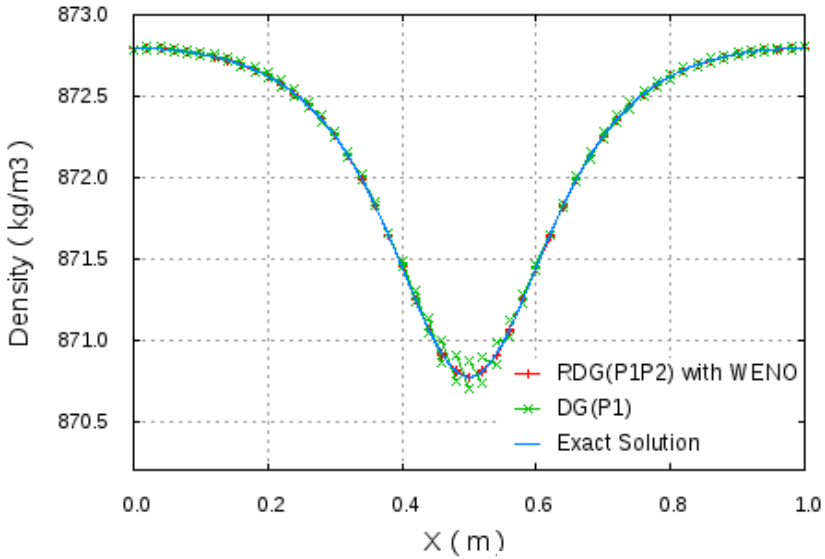


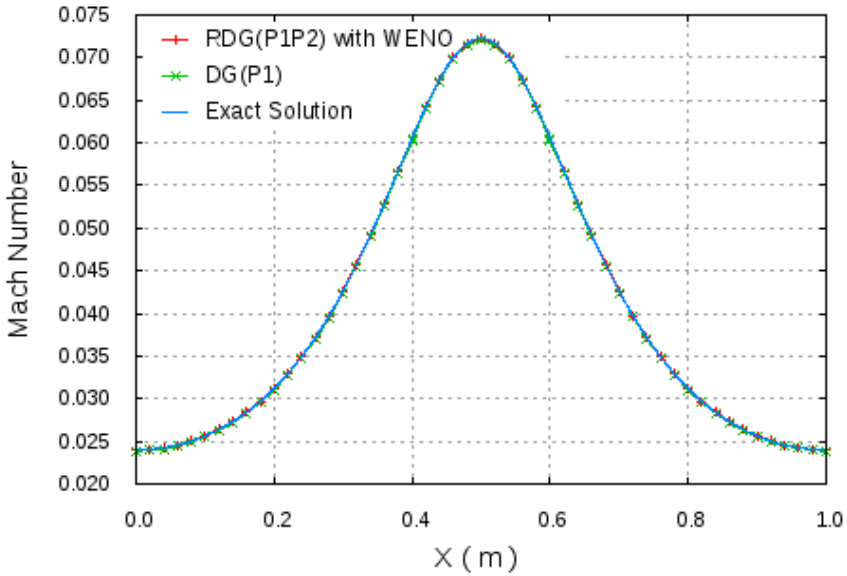
Triangular

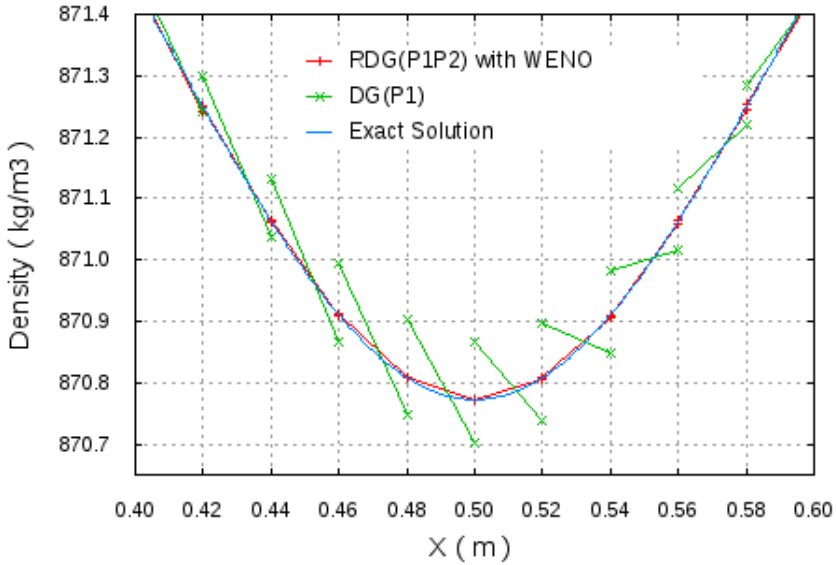
Strengths of the RDG method

Robustness Demonstration

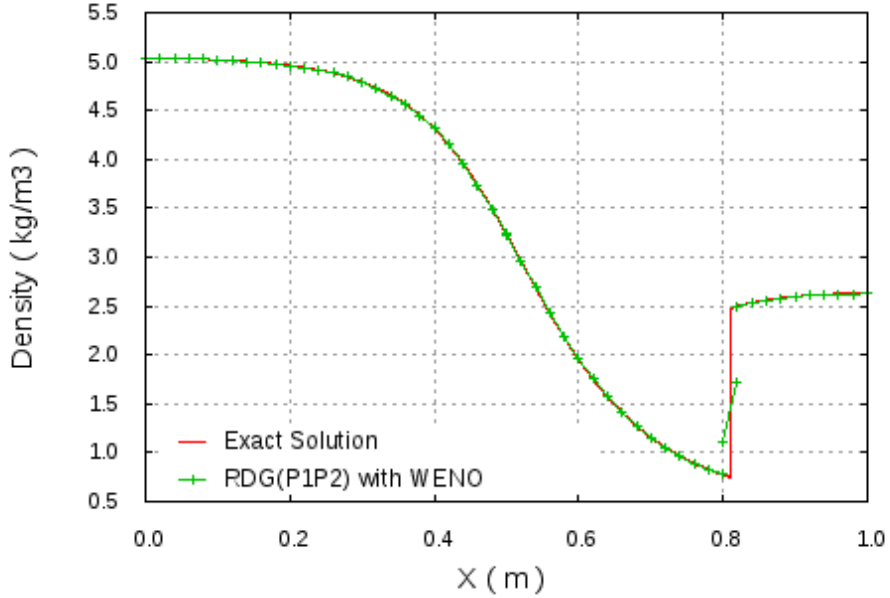
Water flow in a convergent-divergent nozzle

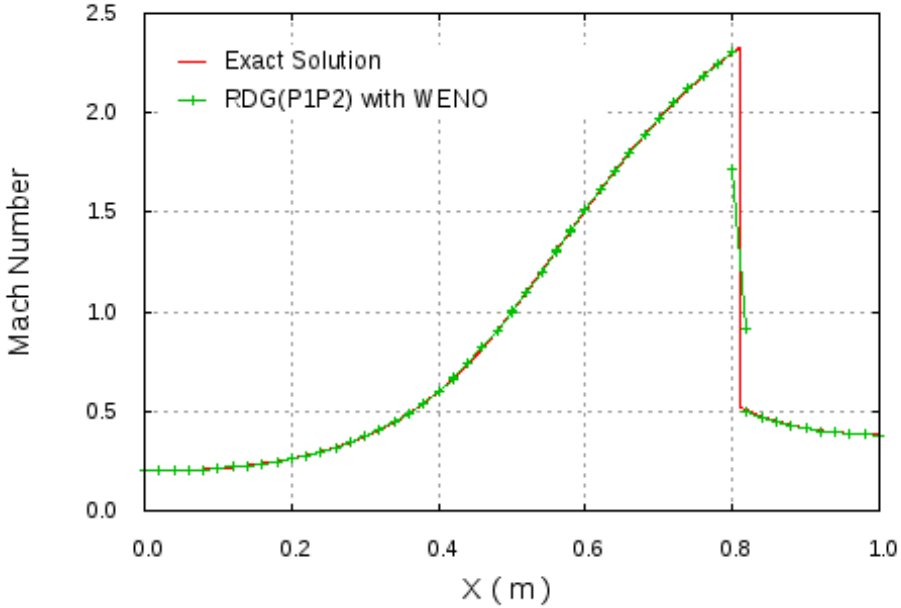






Vapor flow in a convergent-divergent nozzle





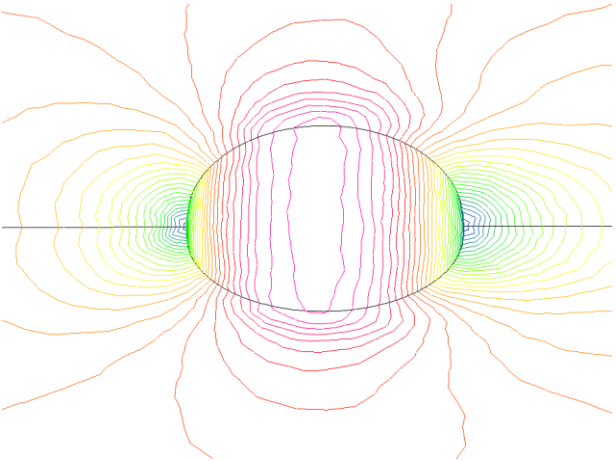
Stiffened EOS

No single parameter is changed !!!

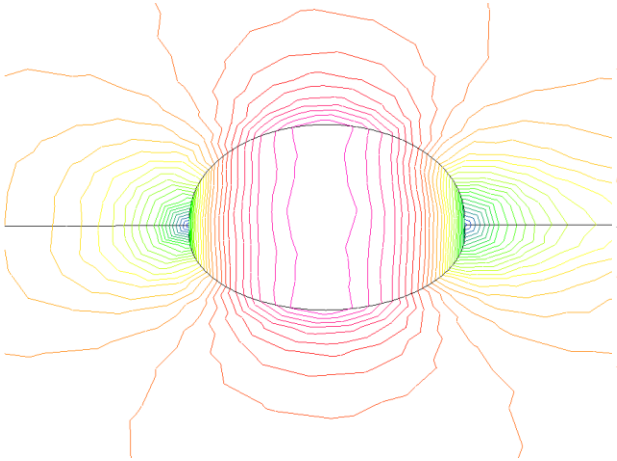
No time-derivative preconditioner is required !!!

Low Mach Number Flow past a Sphere

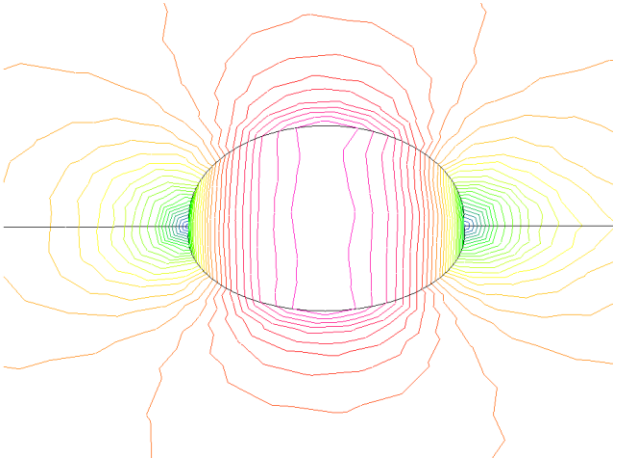
- Access the accuracy for solving low Mach number flow problems.



Obtained by the RDG(P_0P_1) on the finest grid

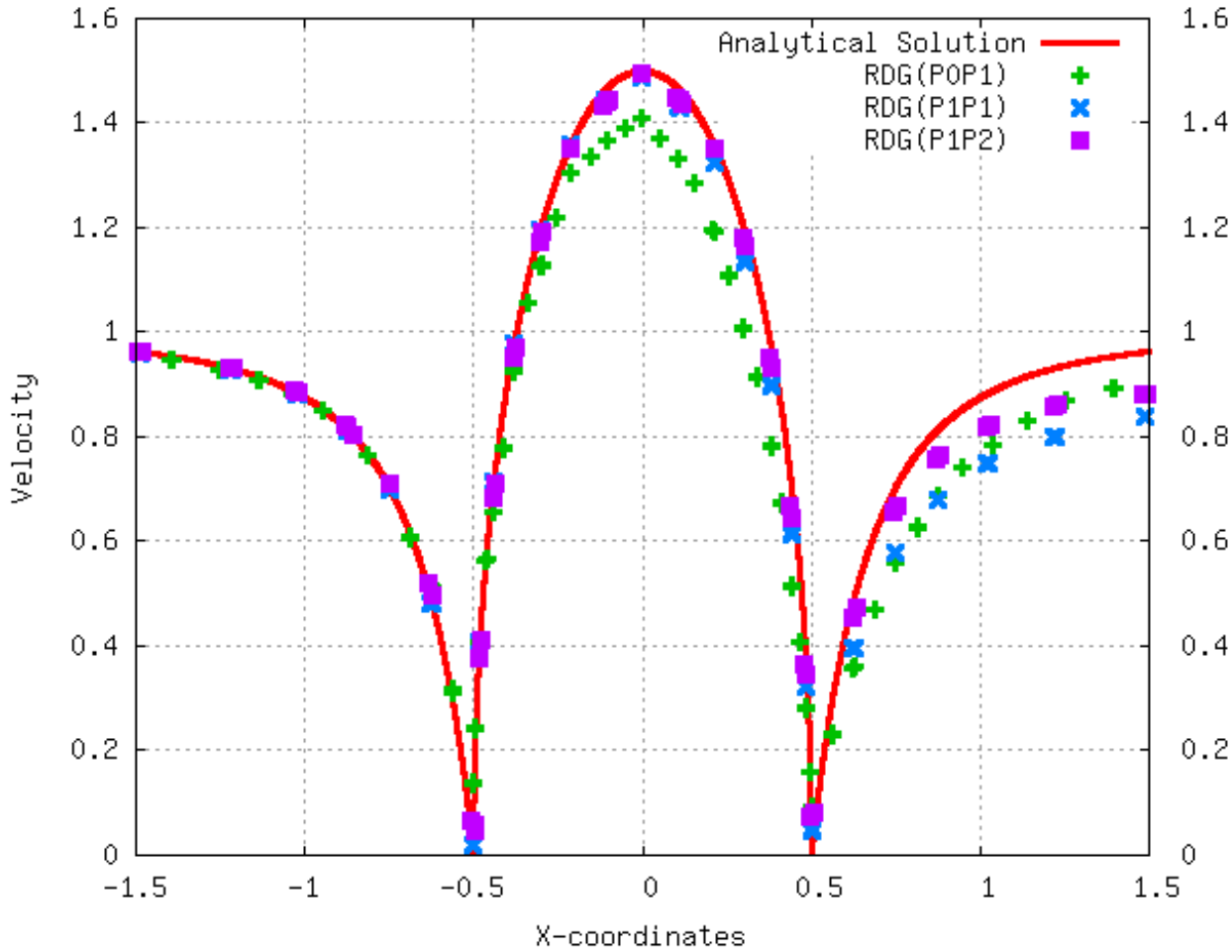


Obtained by the RDG(P_1P_1) on the fine grid



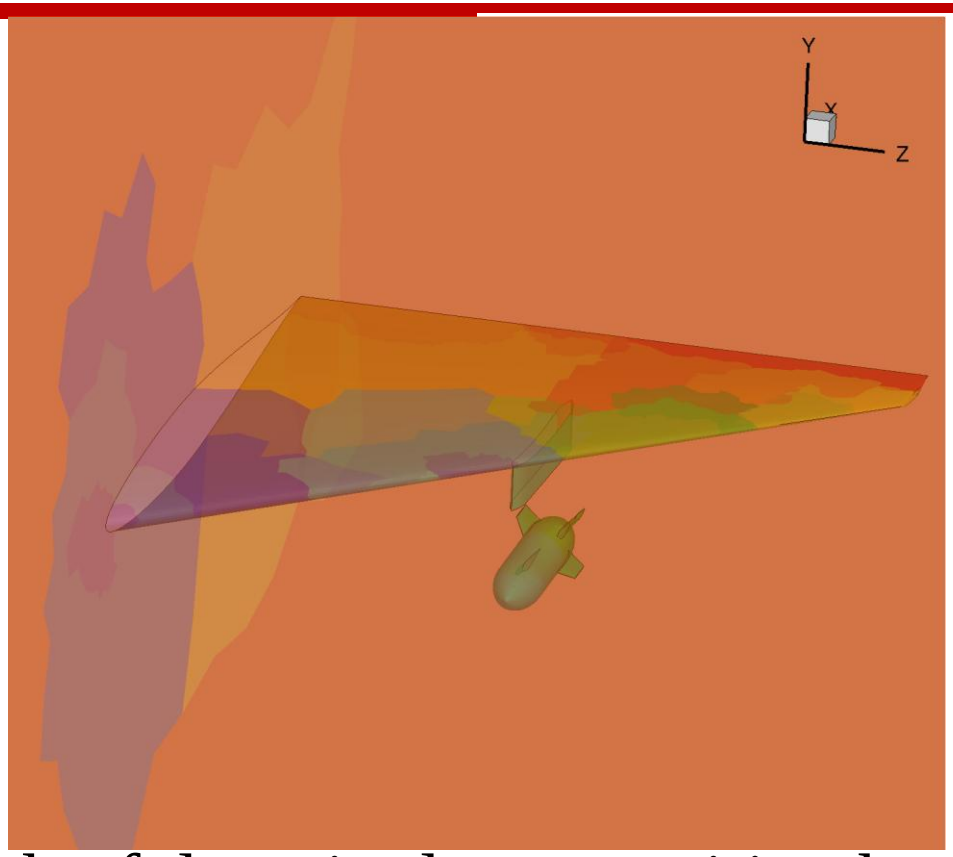
Obtained by the RDG(P_1P_2) on the fine grid

Comparison of the Computed Velocity Distributions on the Surface of the Sphere

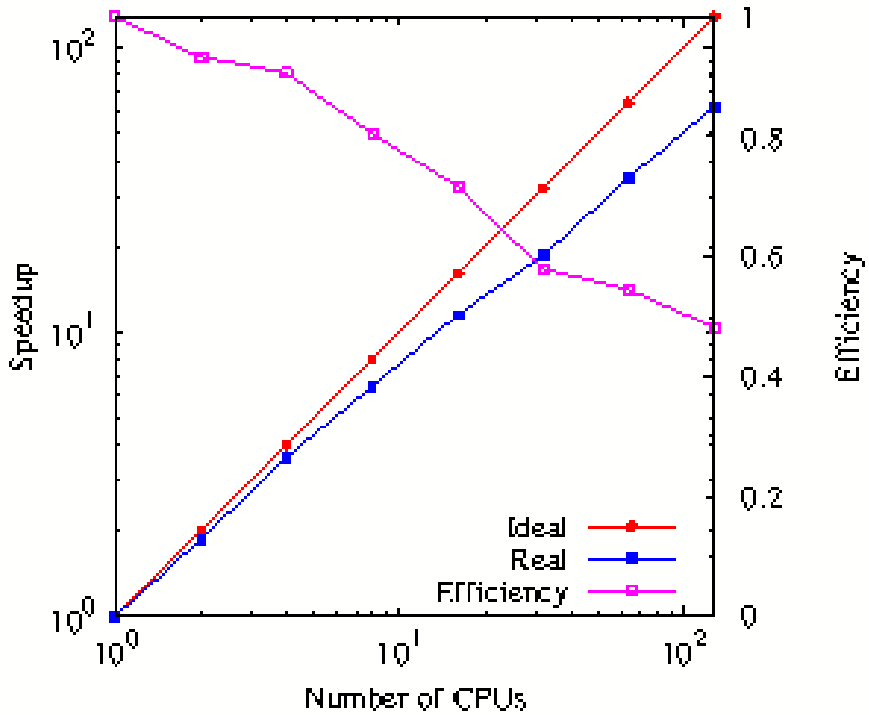


124,706 elements

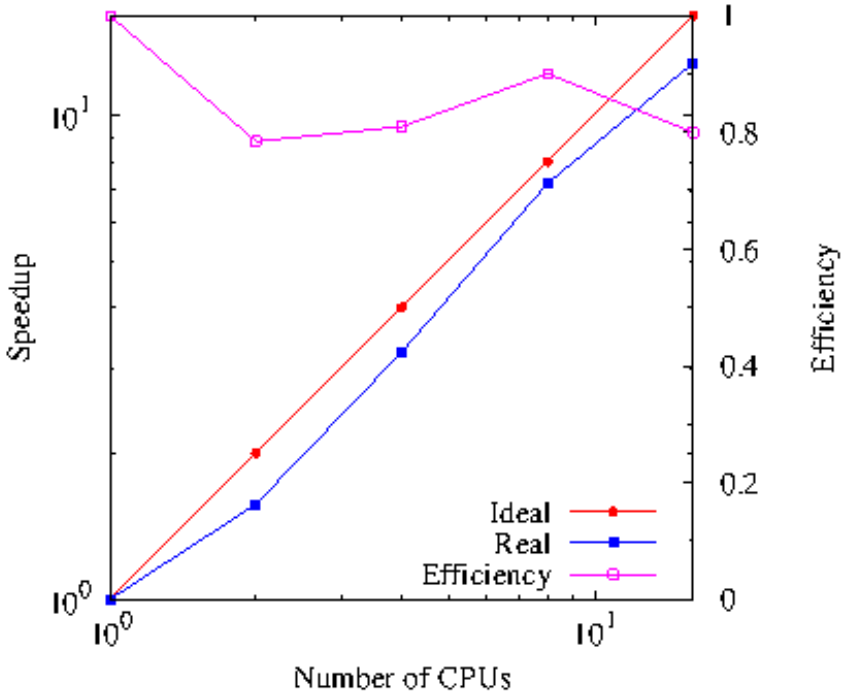
Parallel Performance



Graph of domain decomposition by METIS
128 partitions and 124,706 elements

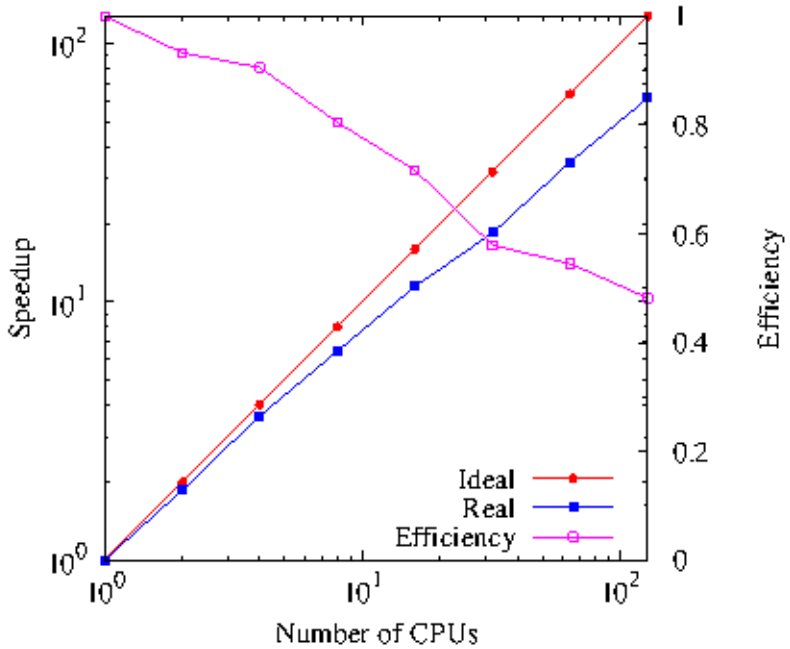


Explicit Method

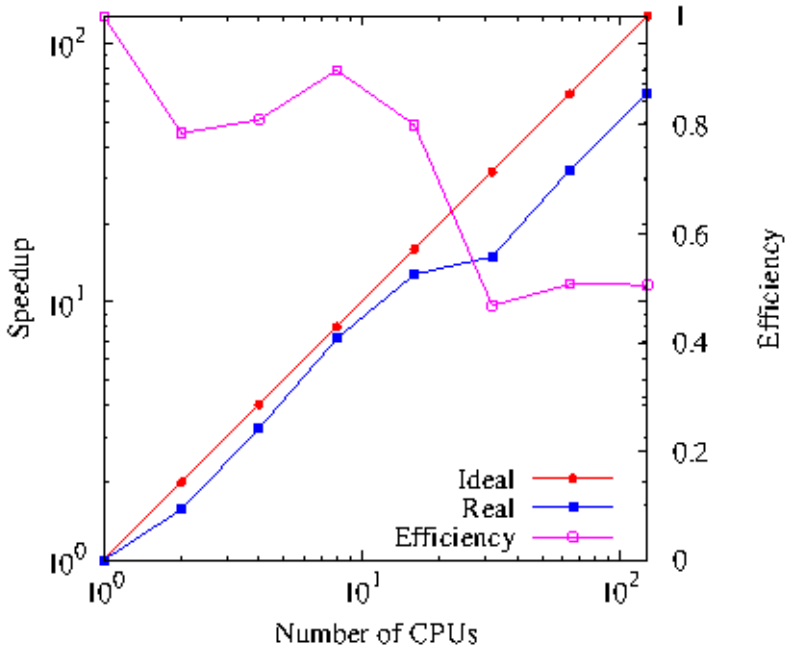


Implicit Method

Parallel speedup and efficiency on a single node (up to 16 CPUs)



Explicit Method



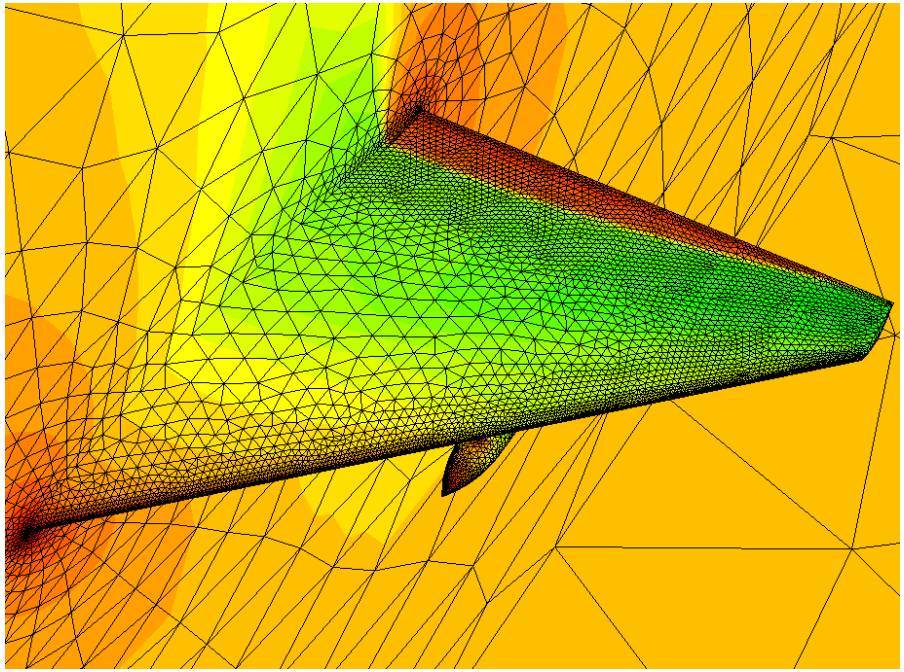
Implicit Method

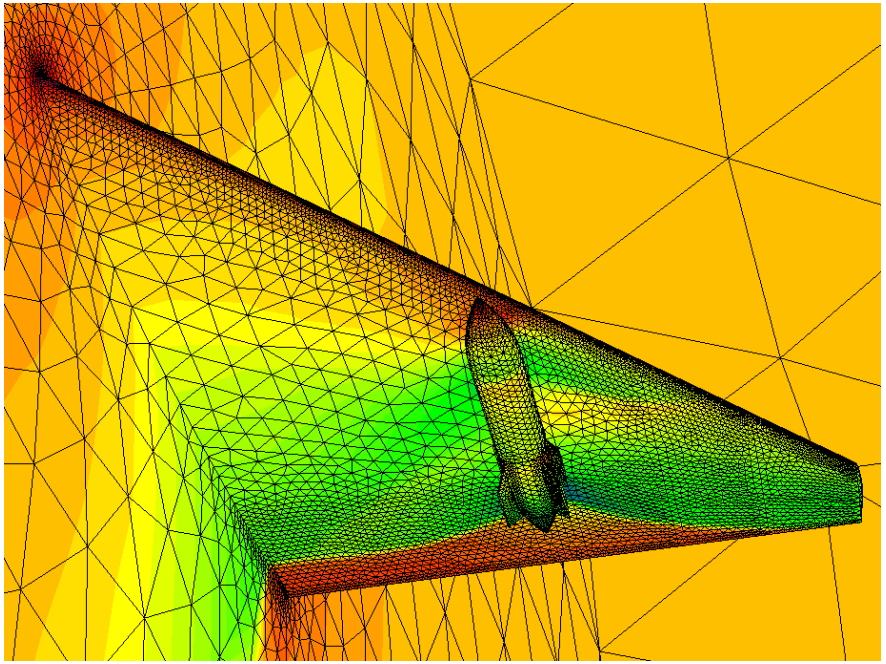
Parallel speedup and efficiency on a multiple nodes (up to 8 nodes with16 CPUs per node)

Applications

Transonic Flow past past a Wing/Pylon/Finned-Store Configuration (M_∞=0.95, α=0°)

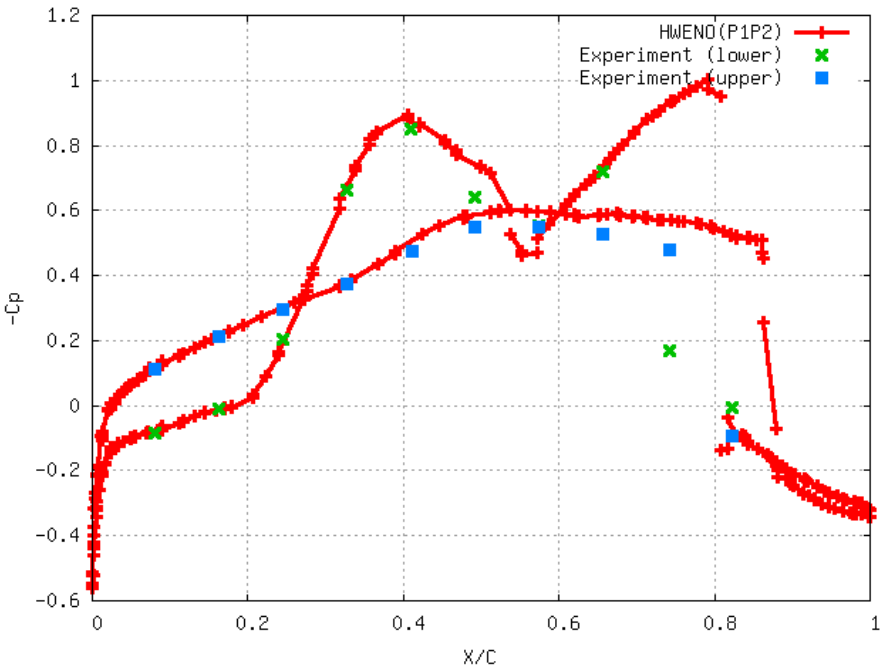
- Access the accuracy and non-oscillatory property of the HWENO(P_1P_2) method for flows with strong discontinuities.



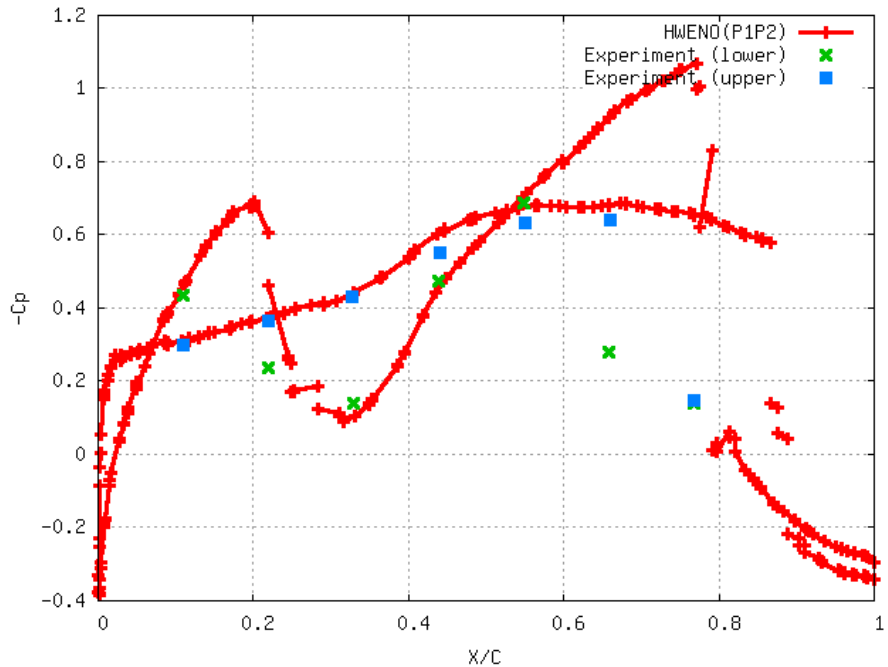


Computed Pressure Contours
(nelem=319,134, npoin=61,075, nboun=14,373)

Computed Pressure Coefficient Distributions at different spanwise locations



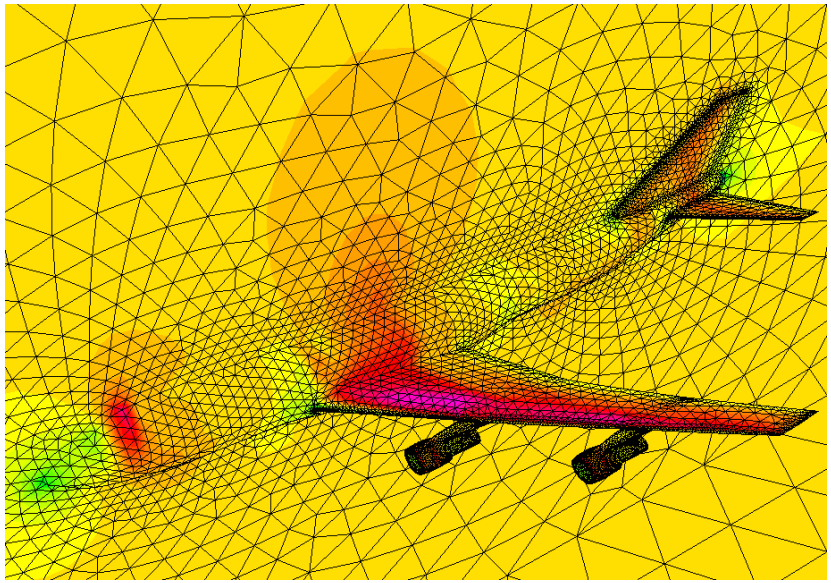
η=0.4077



η=0.51

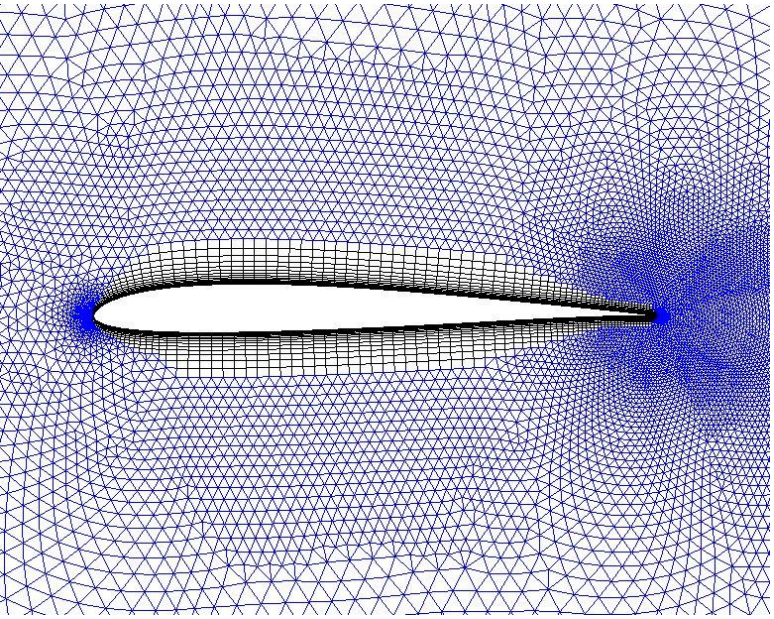
Transonic flow past a B747 configuration

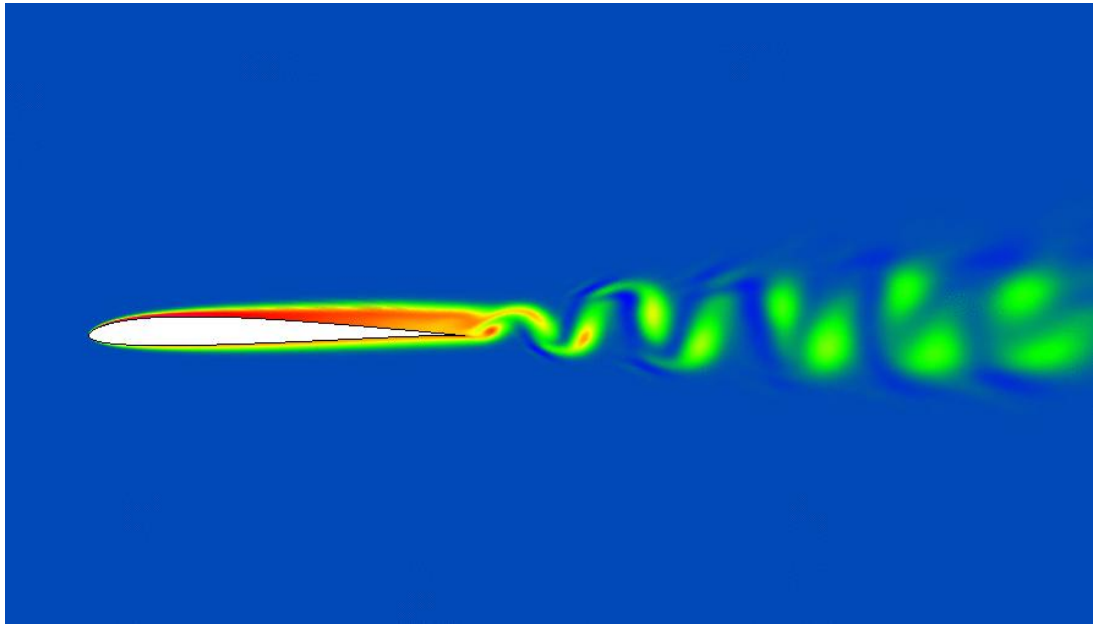
- Demonstrate that the HWENO(P_1P_2) method can be used for computing complicated flows of practical interest.
- Flow condition: M_∞=0.85, α=2°



(nelem = 253,577, npoin = 48,851, nboun = 11,802)
Computed Mach Number Contours

Subsonic flow past a SD7003 airfoil (M=0.1, Re=10,000)

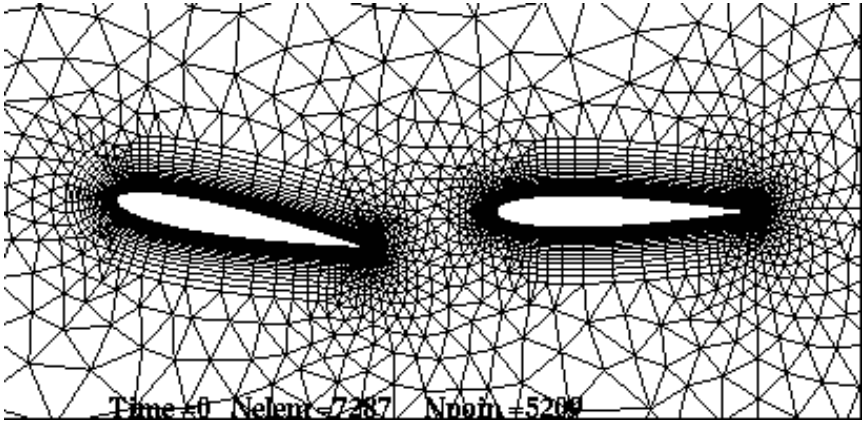


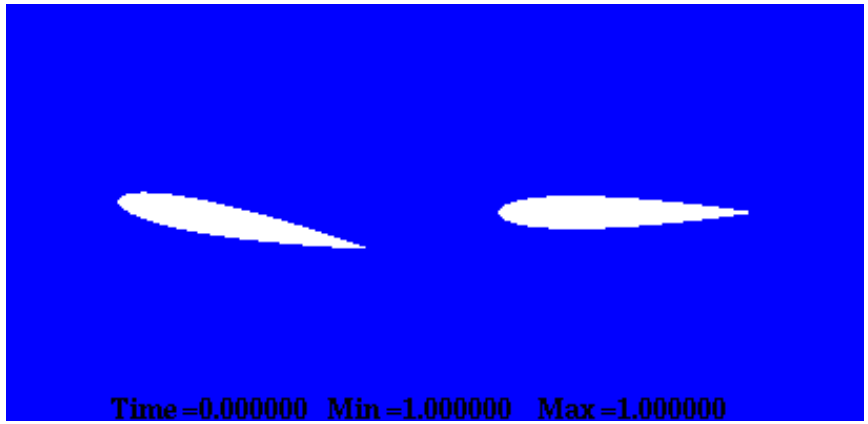


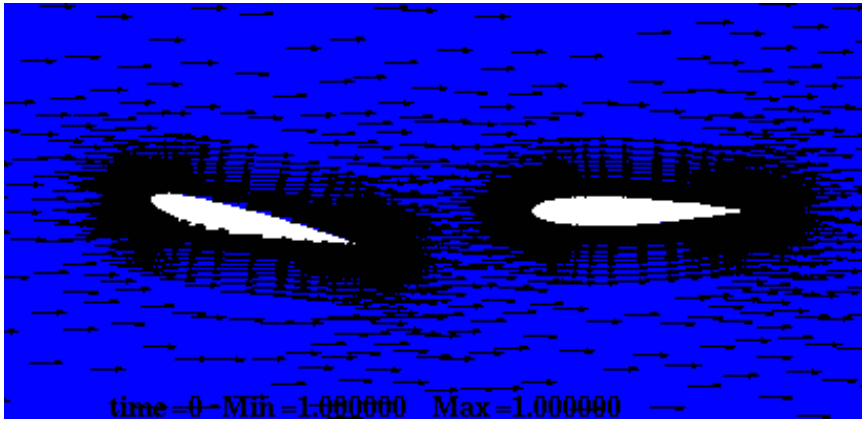
23,172 prismatic elements,
2,225 hexhedral elements
25,397 grid

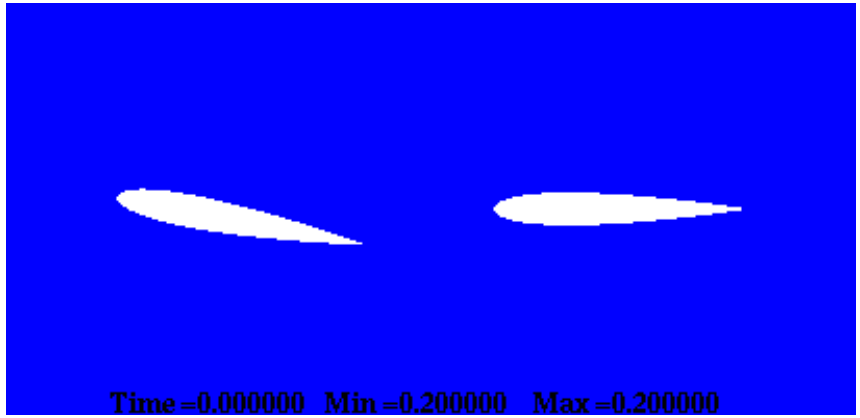
Computed vorticity contours

Unsteady Viscous Flow over Tandem Airfoils
M=0.2, Re=10,000, α=0





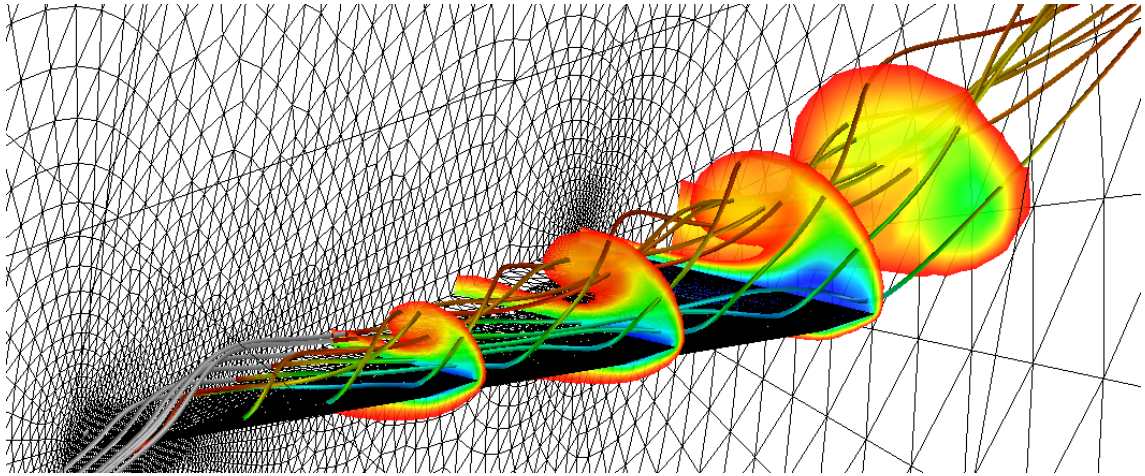




2,902 hexahedral elements, 4,385 prisms, 10,418 grid points,

Subsonic Flow past a Delta Wing

- Demonstrate that the WENO(P_1P_2) method can be used for computing vortex flows of practical interest.
- Flow condition: M_∞=0.3, Re=4,000



(Tetrahedral grid: Nelem = 674,260, Npoin = 120,531, Nboun = 12,991)

Computed Mach Number Contours and streamlines

Concluding Remarks

- A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction, HWENO(P_1P_2) has been developed for compressible flows at all speeds on hybrid grids.

- The HWENO(P_1P_2) method is able to provide sharp resolution of shock waves essentially without over- and under-shoots for discontinuities and achieve the designed third-order of accuracy for smooth flows.

- RDG methods have the potential to provide a superior alternative to the traditional FV methods, and to become a main choice for the next generation of CFD codes.

- A higher-order RDG-based CFD code will ultimately deliver a more accurate, efficient, robust, and reliable simulation tool with confidence that will enable us to solve flow problems at resolutions never before possible by the current state-of-the-art CFD technology.

Current Work

- Extension of the RDG method for turbulent flows
- Implementation of *hp*-adaptation on hybrid grids
- Port of the RDGFLO code on hybrid CPU/GPU architectures

Thank you