

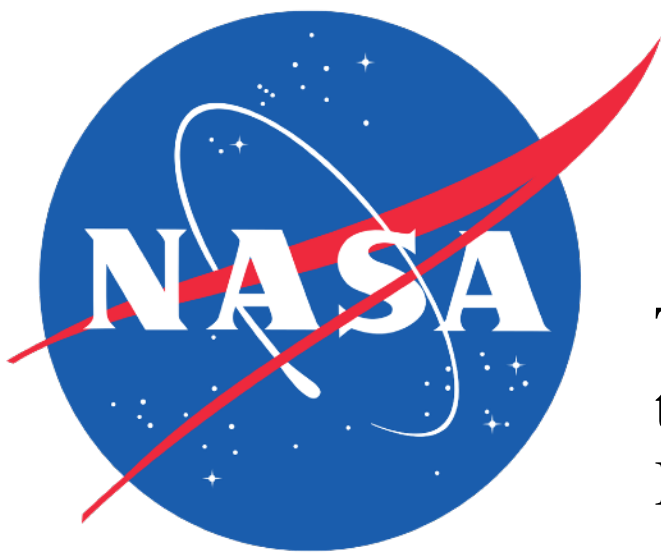


Toward a Third-Order Accurate, Second Derivative-Free, Shock-Capturing Finite-Volume Method for Hypersonic Flows on Tetrahedral Grids

Hiroaki Nishikawa, *National Institute of Aerospace*

Jeffery White, *NASA Langley Research Center*

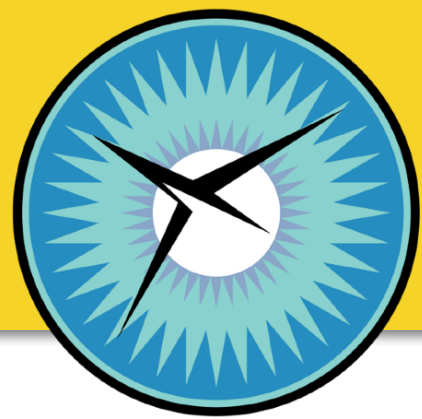
Matt O'Connell, *NASA Langley Research Center*



This work was supported by the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch of the NASA Langley Research Center, under Contract No. 80LARC17C0004

AIAA Aviation, June 12, 2023

Copyright 2023 United States Government as represented by the Administrator of the National Aeronautics and Space Administration and the National Institute of Aerospace. All Rights Reserved. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.



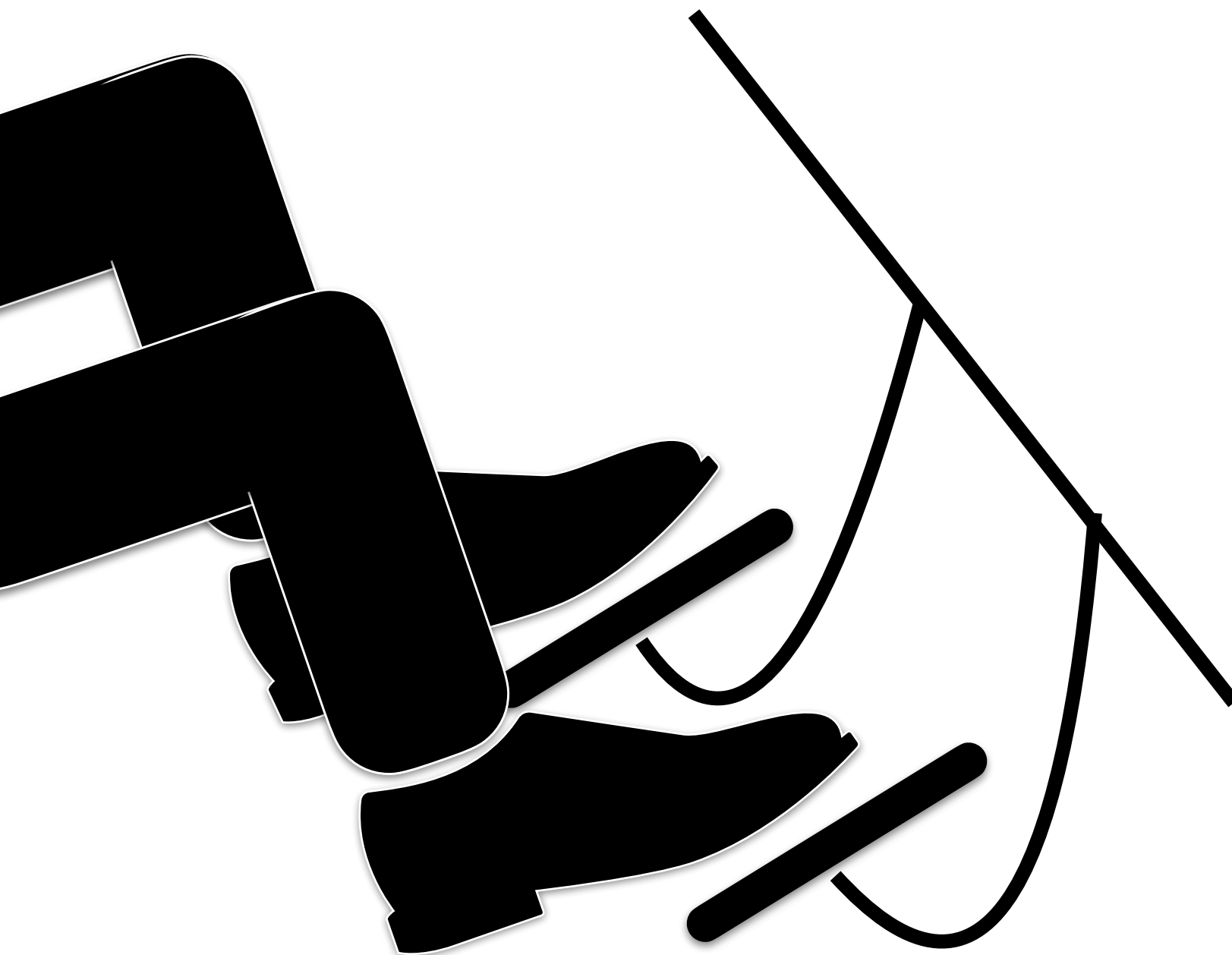
Our world is highly unstructured, but things are designed often with a clear structure: e.g.,

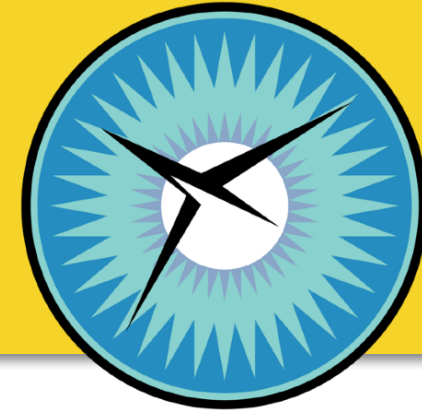
Cars have a similar structure with the left pedal to brake and the right to accelerate.

So, you can drive in the same way any car anywhere.

The same is true in CFD.

Structure within an unstructured grid is useful for constructing a simple and efficient algorithm.





Baseline 2nd-order CCFV in the NASA VULCAN-CFD Code:

<https://vulcan-cfd.larc.nasa.gov>

$$V_j \frac{\partial \mathbf{u}_j}{\partial t} + \sum_{k \in \{k_j\}} \Phi_{jk}(\mathbf{w}_L, \mathbf{w}_R) |\mathbf{n}_T| - \mathbf{s}_j = \mathbf{0} \quad \text{Inviscid discretization}$$

Conventional CCFV [JANNAF 2017 2019, AIAA2019]:

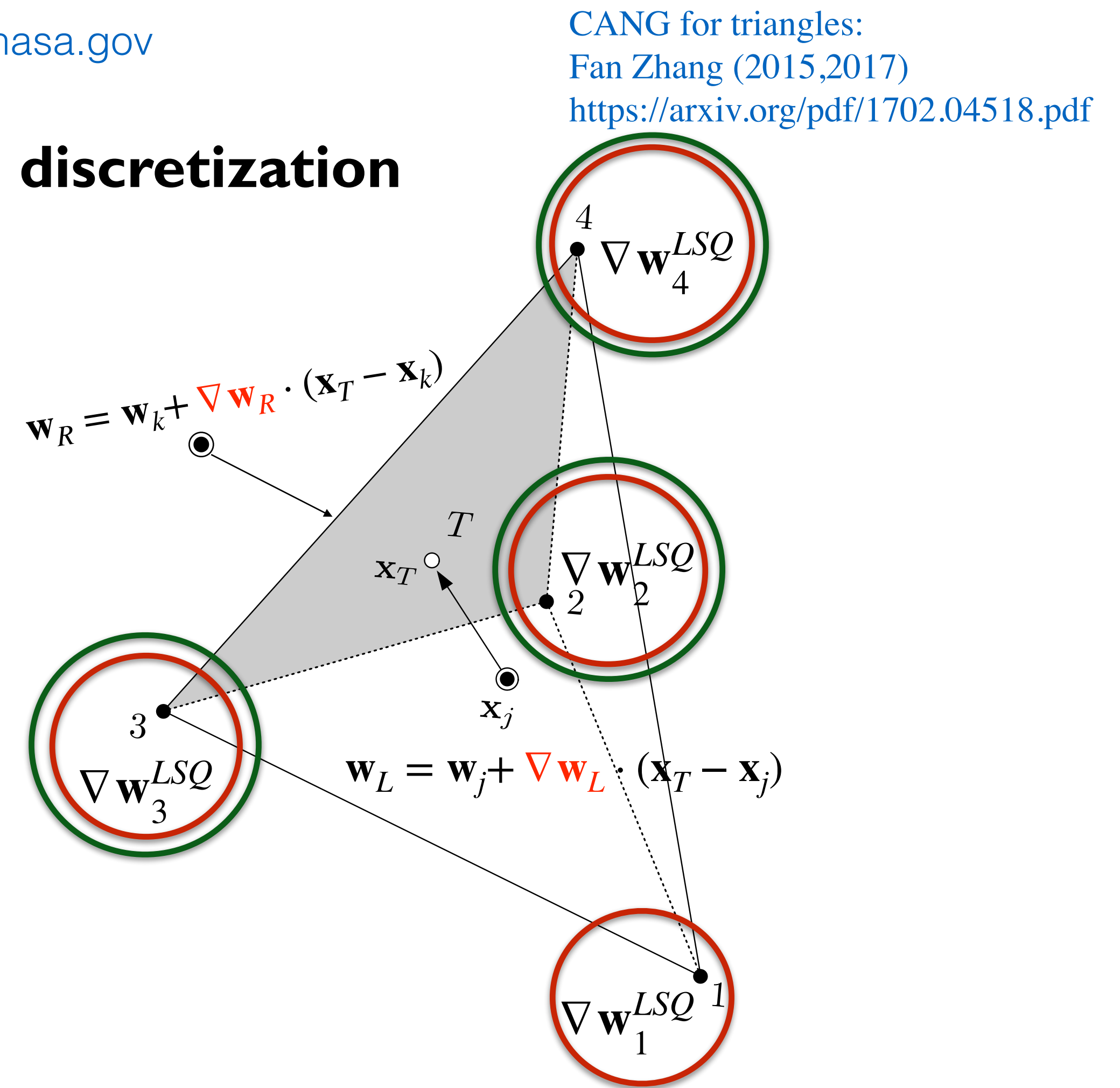
$$\nabla \mathbf{w}_L = \nabla \mathbf{w}_j^{LSQ}, \quad \nabla \mathbf{w}_R = \nabla \mathbf{w}_k^{LSQ}$$

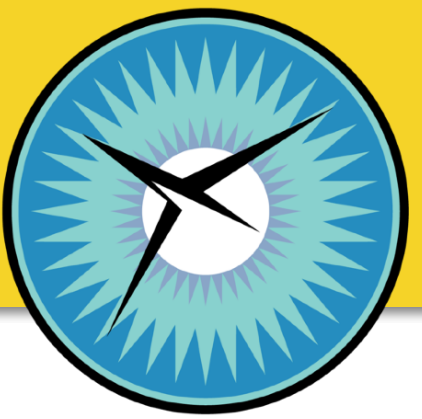
Cell-Averaged Nodal Gradient (CANG) [AIAA2020~]:

$$\nabla \mathbf{w}_L = \frac{\nabla \mathbf{w}_1^{LSQ} + \nabla \mathbf{w}_2^{LSQ} + \nabla \mathbf{w}_3^{LSQ} + \nabla \mathbf{w}_4^{LSQ}}{4}$$

Face-Averaged Nodal Gradient (FANG) [JCP2020~]:

$$\nabla \mathbf{w}_L = \nabla \mathbf{w}_R = \frac{\nabla \mathbf{w}_2^{LSQ} + \nabla \mathbf{w}_3^{LSQ} + \nabla \mathbf{w}_4^{LSQ}}{3}$$





3rd-order CCFV for tetrahedral grids can be written as

$$V_j \frac{\partial \mathbf{u}_j}{\partial t} + \sum_{k \in \{k_j\}} \Phi_{jk}(\mathbf{w}_L, \mathbf{w}_R) |\mathbf{n}_T| - \mathbf{s}_j V_j = \delta \mathbf{f}_j + \delta \mathbf{s}'_j \quad \mathbf{s}' = \mathbf{s}_j - \frac{\partial \mathbf{u}}{\partial t}$$

which is the baseline 2nd-order + high-order corrections:

Quadratic
interpolation:

$$\mathbf{w}_L = \mathbf{w}_j + \nabla \mathbf{w}_L (\mathbf{x}_T - \mathbf{x}_j) + \frac{1}{2} (\mathbf{x}_T - \mathbf{x}_j)^T \nabla^2 \mathbf{w}_j (\mathbf{x}_T - \mathbf{x}_j)$$

Flux correction:

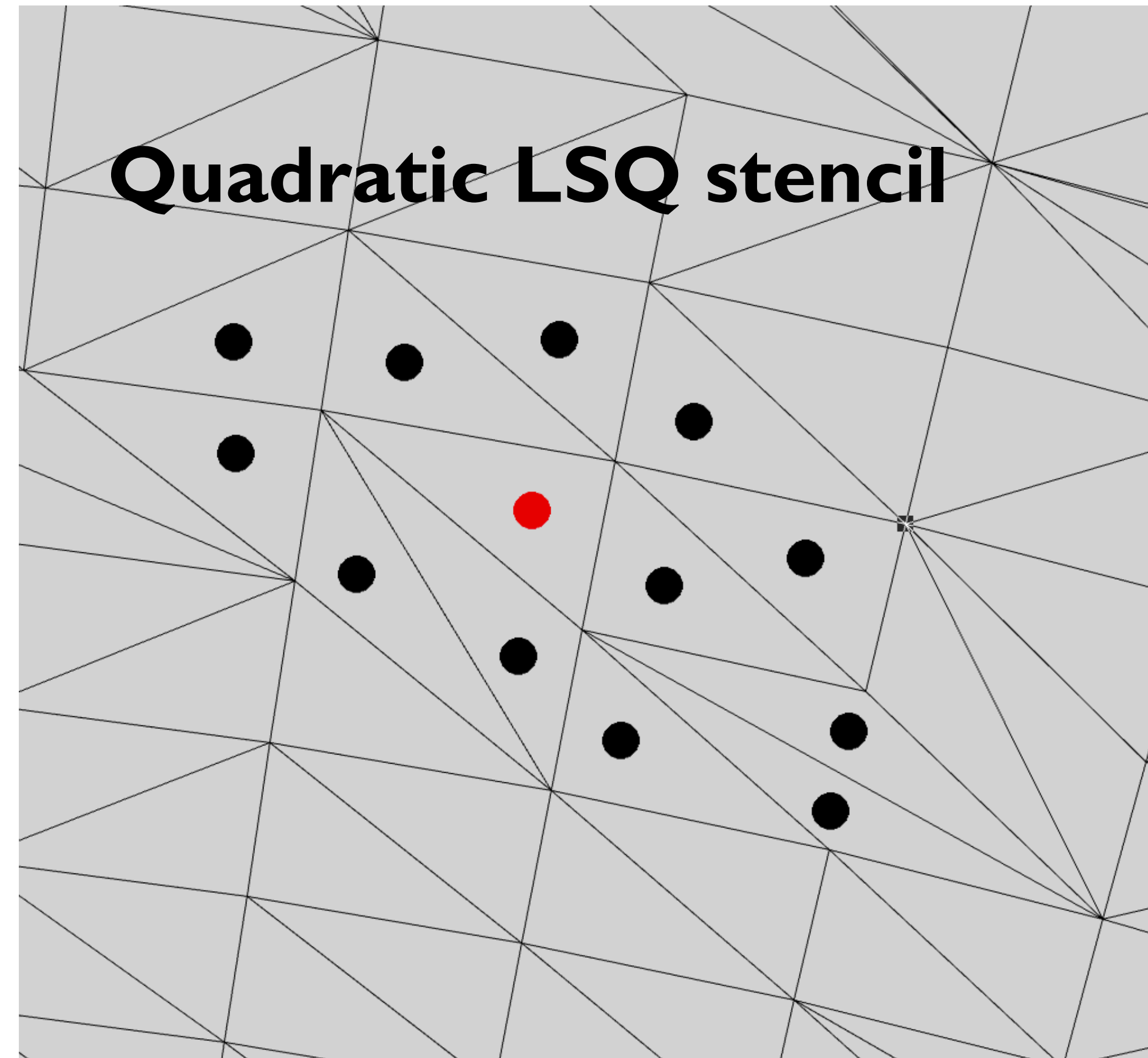
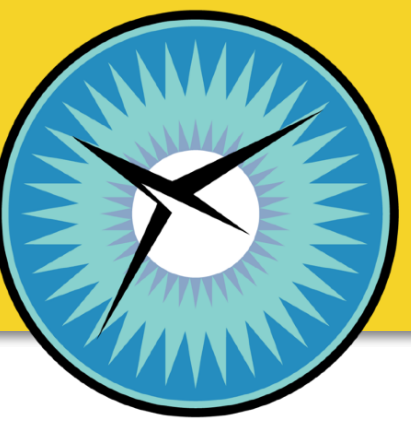
$$\delta \mathbf{f}_j = \frac{1}{24} \sum_{k \in \{k_j\}} \sum_{i=1}^3 (\mathbf{x}_i - \mathbf{x}_T)^t \nabla^2 \mathbf{f}_T (\mathbf{x}_i - \mathbf{x}_T) |\mathbf{n}_T|$$

Source correction:

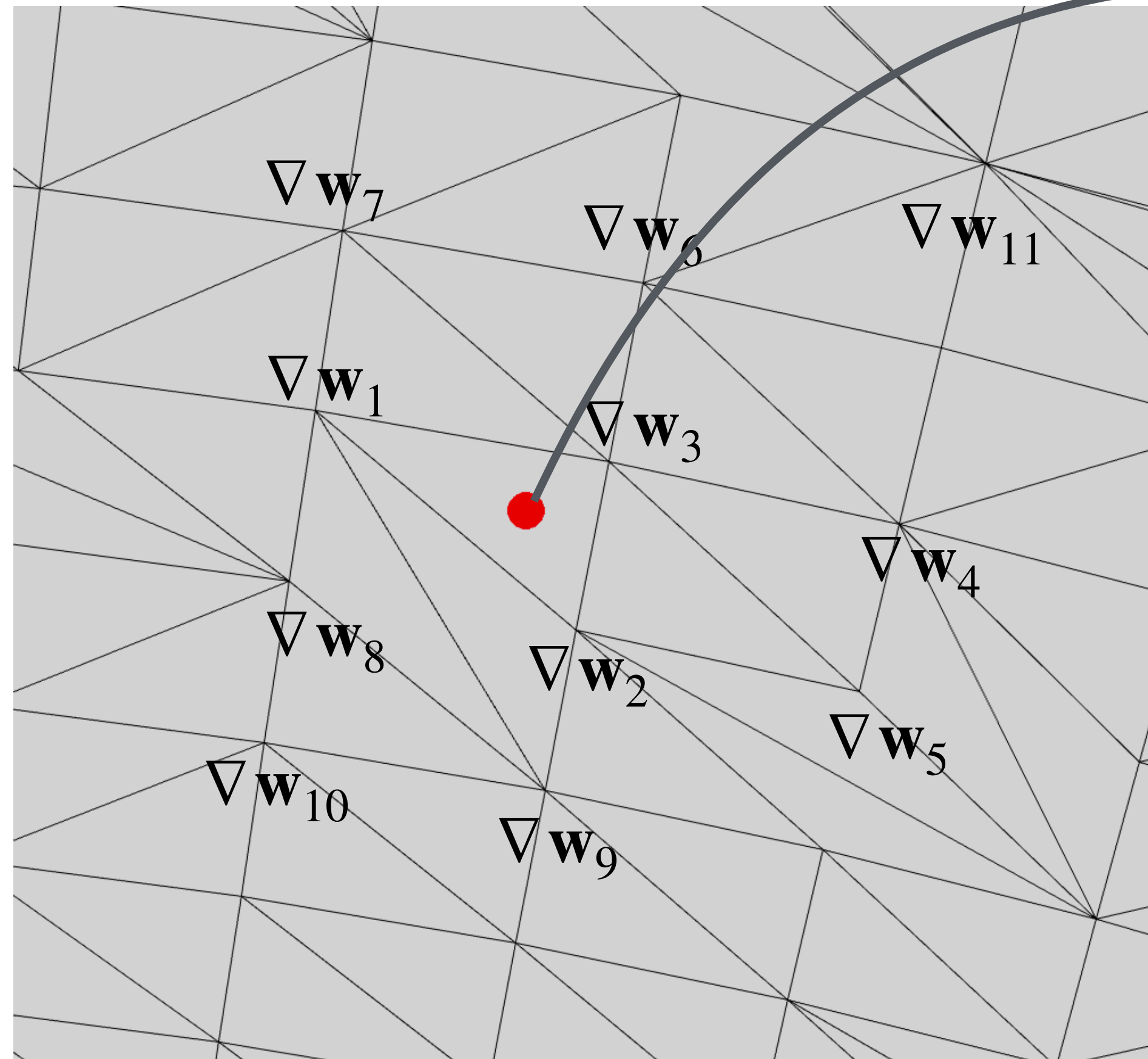
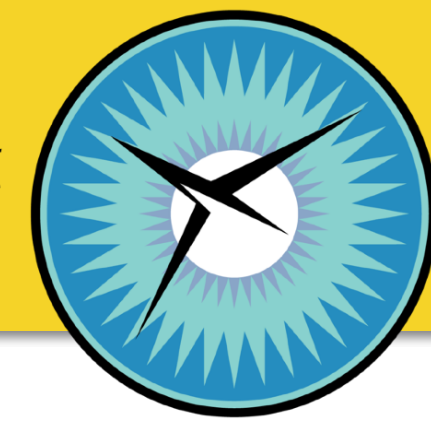
$$\delta \mathbf{s}'_j = \frac{1}{40} \sum_{i=1}^4 (\mathbf{x}_i - \mathbf{x}_j)^t \nabla^2 \mathbf{s}'_j (\mathbf{x}_i - \mathbf{x}_j) V_j$$

Relatively simple to implement, but **second derivatives** are required...

Compute and Store $\nabla^2 \mathbf{w}_j, \nabla^2 \mathbf{f}_T, \nabla^2 \mathbf{s}'_j \dots\dots\dots$



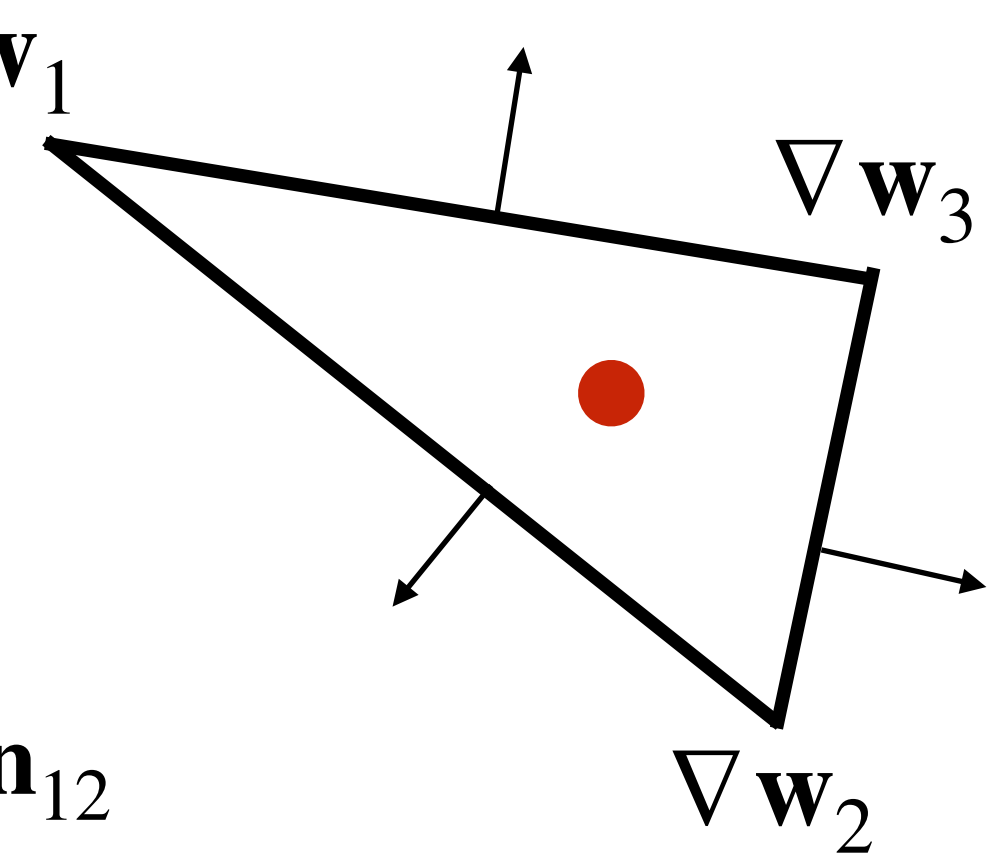
Requires the storage of additional cell-centered LSQ coefficients and second derivatives for solutions, source terms, fluxes... Can we avoid all these?



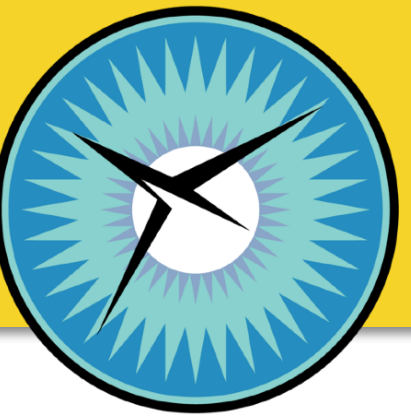
A nice structured grid !

Green-Gauss Gradient:

$$\nabla^2 \mathbf{w}_j^{GG} = \frac{1}{V_j} \left[\frac{\nabla \mathbf{w}_1 + \nabla \mathbf{w}_2}{2} \mathbf{n}_{12} + \frac{\nabla \mathbf{w}_2 + \nabla \mathbf{w}_3}{2} \mathbf{n}_{23} + \frac{\nabla \mathbf{w}_3 + \nabla \mathbf{w}_1}{2} \mathbf{n}_{31} \right]$$



Second derivatives can be computed in the same form in any triangle,
expressed only using the gradients at nodes.



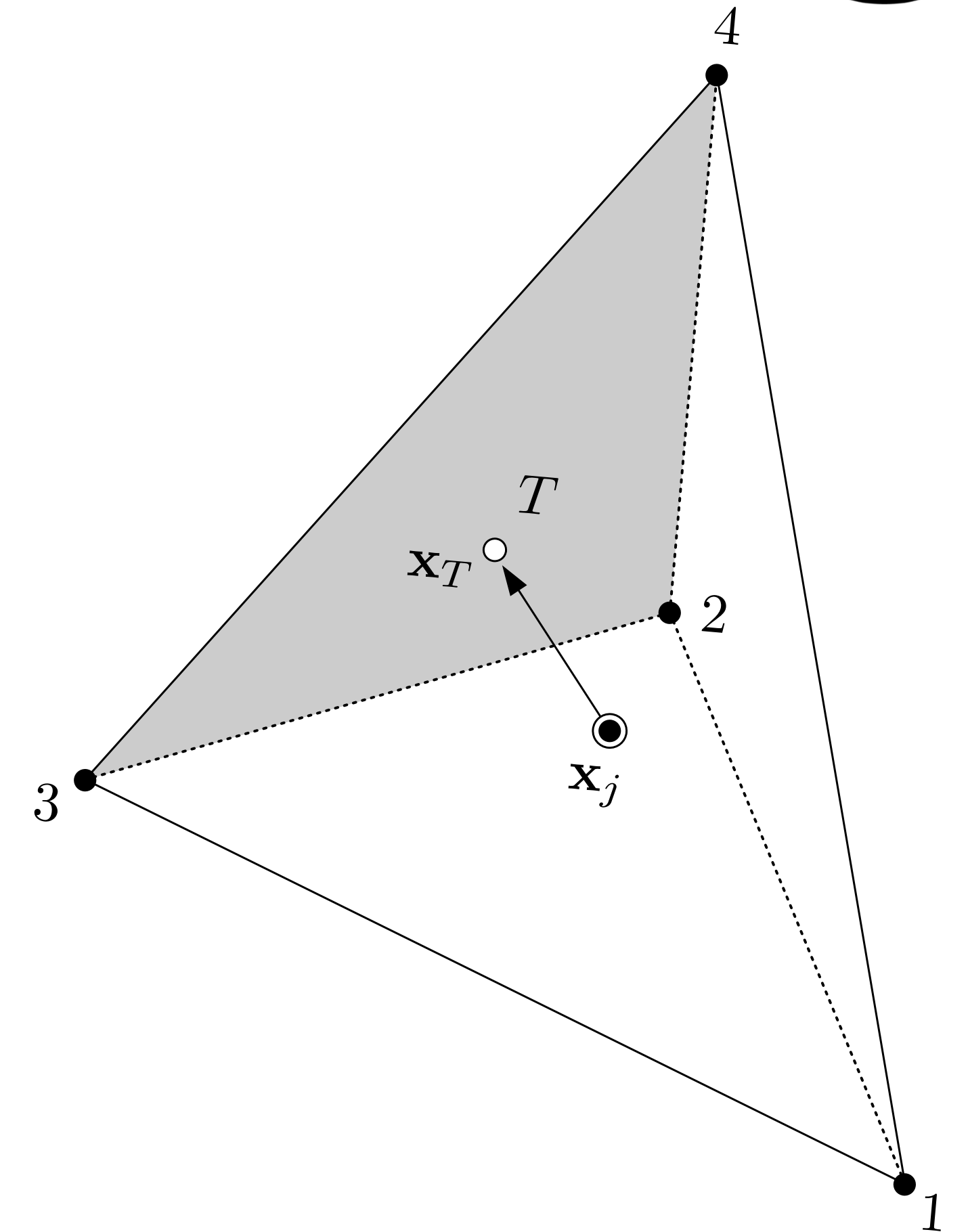
Projected derivative with Green-Gauss gradient:

$$\nabla \mathbf{w}_j^{GG} \cdot (\mathbf{x}_T - \mathbf{x}_j) = \frac{1}{4} \left(\frac{\mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4}{3} - \mathbf{w}_1 \right)$$

N.T. Frink AIAAJ 1998 (NASA USM3D)

$$= \frac{\mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4}{3} - \frac{\mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3 + \mathbf{w}_4}{4}$$

$$= \overline{\mathbf{w}}_T - \overline{\mathbf{w}}_j \quad (\text{Nishikawa\&White 2023}).$$



We apply this to second derivatives:

$$\nabla^2 \mathbf{w}_j^{GG} \cdot (\mathbf{x}_T - \mathbf{x}_j) = \overline{\nabla \mathbf{w}}_T - \overline{\nabla \mathbf{w}}_j \quad \text{Only need nodal gradients!}$$

Efficient nodal-gradient quadratic interpolation (NGQI):

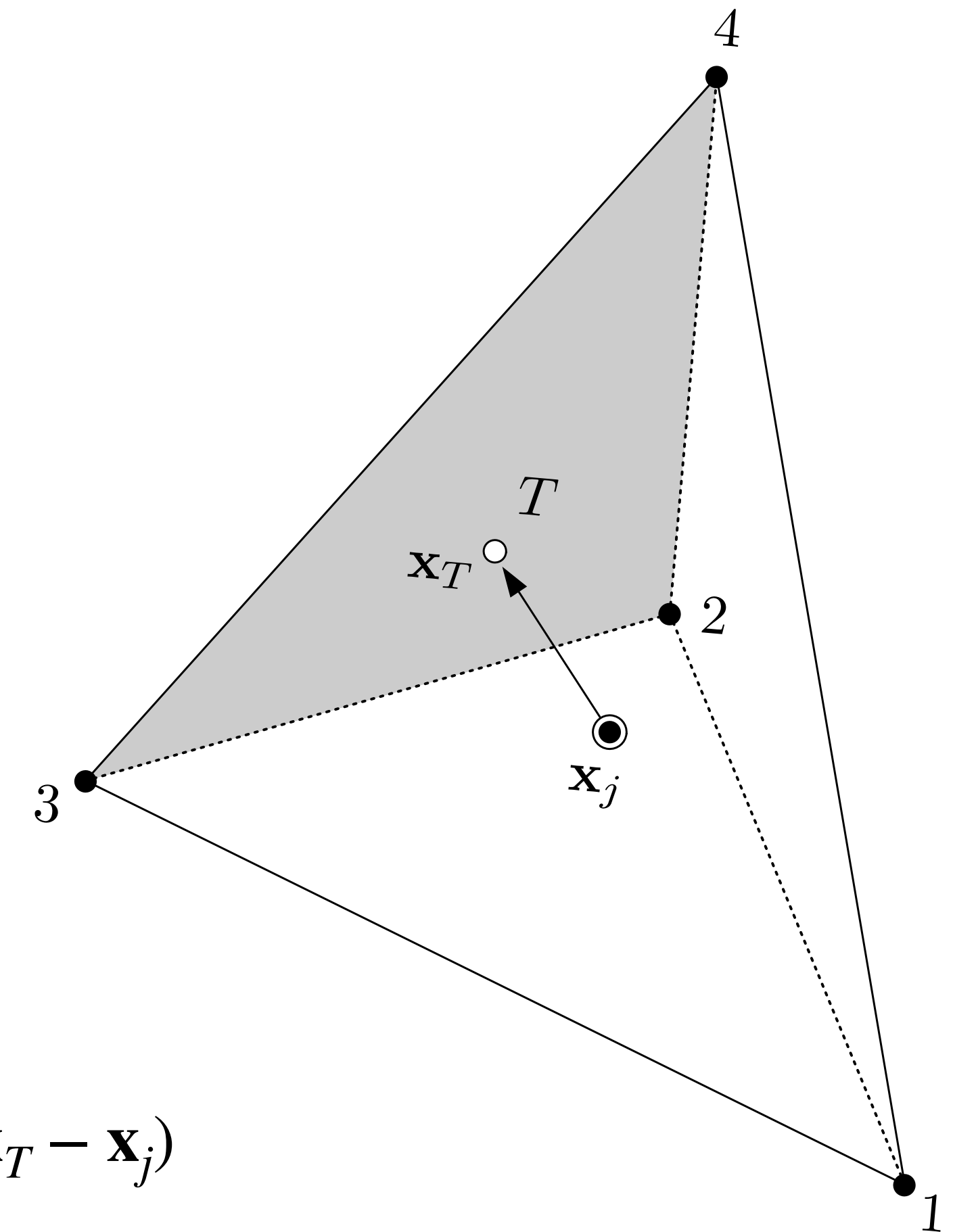
$$\begin{aligned}
 \mathbf{w}_L &= \mathbf{w}_j + \nabla \mathbf{w}_L(\mathbf{x}_T - \mathbf{x}_j) + \frac{1}{2}(\mathbf{x}_T - \mathbf{x}_j)^T \nabla^2 \mathbf{w}_j(\mathbf{x}_T - \mathbf{x}_j) \\
 &= \mathbf{w}_j + \overline{\nabla \mathbf{w}_j}(\mathbf{x}_T - \mathbf{x}_j) + \frac{1}{2}(\overline{\nabla \mathbf{w}_T} - \overline{\nabla \mathbf{w}_j})(\mathbf{x}_T - \mathbf{x}_j) \\
 &= \mathbf{w}_j + \frac{1}{2}(\overline{\nabla \mathbf{w}_T} + \overline{\nabla \mathbf{w}_j})(\mathbf{x}_T - \mathbf{x}_j)
 \end{aligned}$$

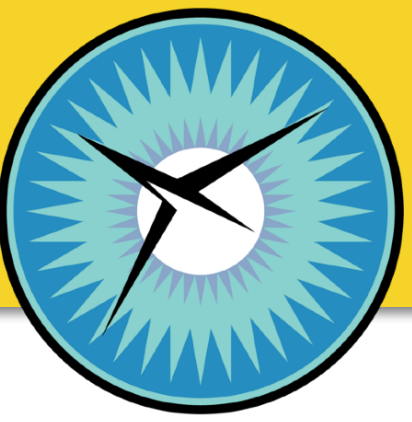
This is a second-derivative-free quadratic interpolation formula.

Yes, this is the average of FANG and CANG linear interpolations:

$$\mathbf{w}_L = \frac{1}{2} \left(\underbrace{\mathbf{w}_j + \overline{\nabla \mathbf{w}_T}(\mathbf{x}_T - \mathbf{x}_j)}_{\text{FANG}} + \underbrace{\mathbf{w}_j + \overline{\nabla \mathbf{w}_j}(\mathbf{x}_T - \mathbf{x}_j)}_{\text{CANG}} \right) = \mathbf{w}_j + \frac{1}{2}(\overline{\nabla \mathbf{w}_T} + \overline{\nabla \mathbf{w}_j})(\mathbf{x}_T - \mathbf{x}_j)$$

Very simple to implement because we already have both FANG and CANG.





Economical 3rd-order FV discretization:

$$V_j \frac{\partial \mathbf{u}_j}{\partial t} + \sum_{k \in \{k_j\}} \Phi_{jk}(\mathbf{w}_L, \mathbf{w}_R) |\mathbf{n}_T| - \mathbf{s}_j V_j = \delta \mathbf{f}_j + \delta \mathbf{s}'_j \quad \mathbf{s}' = \mathbf{s}_j - \frac{\partial \mathbf{u}}{\partial t}$$

where

Quadratic
interpolation:

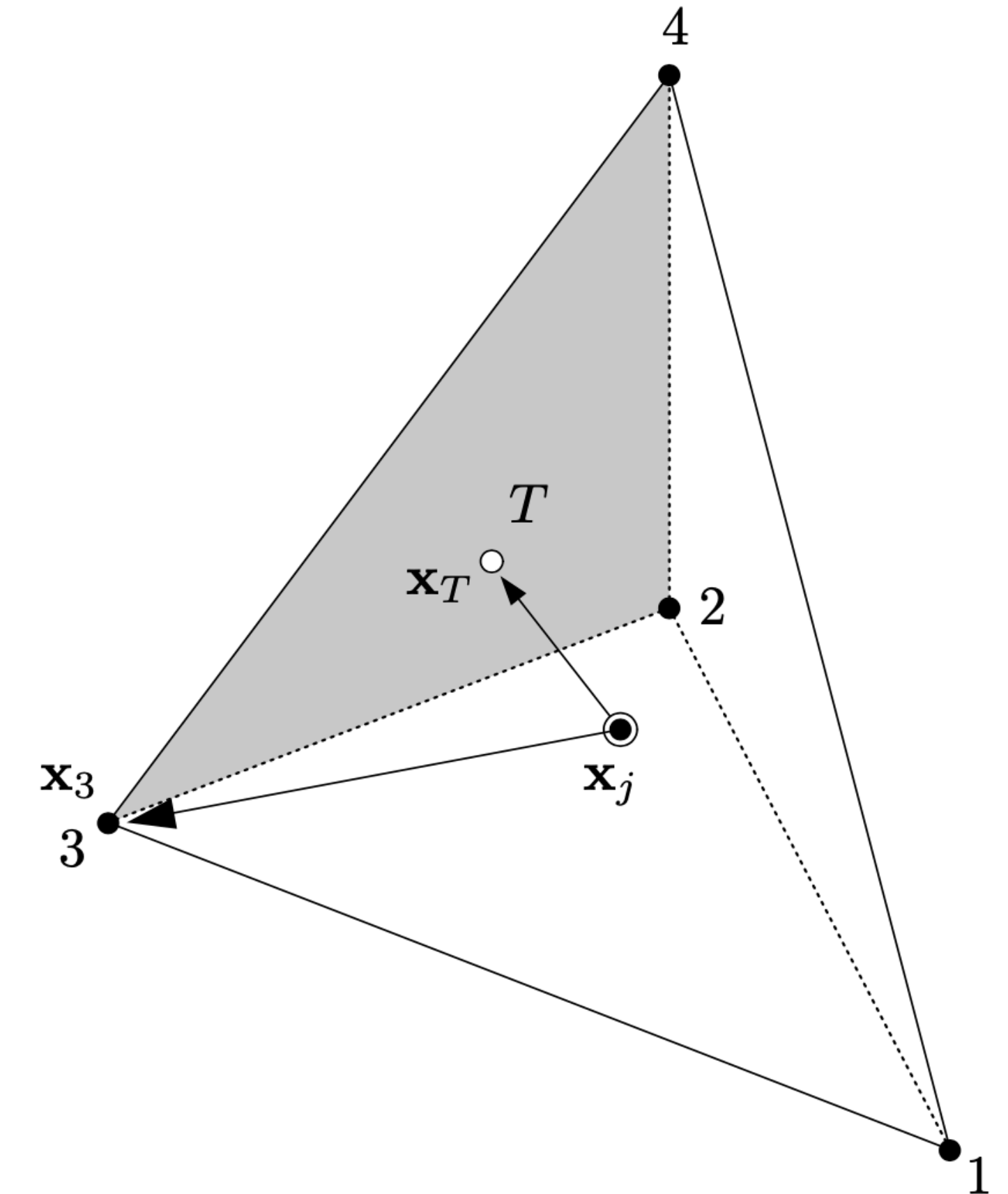
$$\mathbf{w}_L = \mathbf{w}_j + \frac{1}{2} (\overline{\nabla \mathbf{w}_T} + \overline{\nabla \mathbf{w}_j}) (\mathbf{x}_T - \mathbf{x}_j)$$

Flux correction:

$$\delta \mathbf{f}_j = \frac{1}{24} \sum_{k \in \{k_j\}} \sum_{i=1}^3 \left(\frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right) \nabla \mathbf{w}_i (\mathbf{x}_i - \mathbf{x}_T) |\mathbf{n}_T|$$

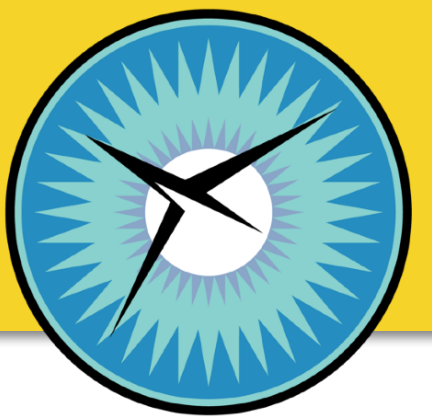
Source correction:

$$\delta \mathbf{s}'_j = \frac{1}{40} \sum_{i=1}^4 \nabla \mathbf{s}'_i (\mathbf{x}_i - \mathbf{x}_j) V_j$$



We then just need to upgrade the nodal linear LSQ to quadratic LSQ.

Yielding a second-derivative-free economical 3rd-order discretization!



Need a high-order accuracy preserving limiter.

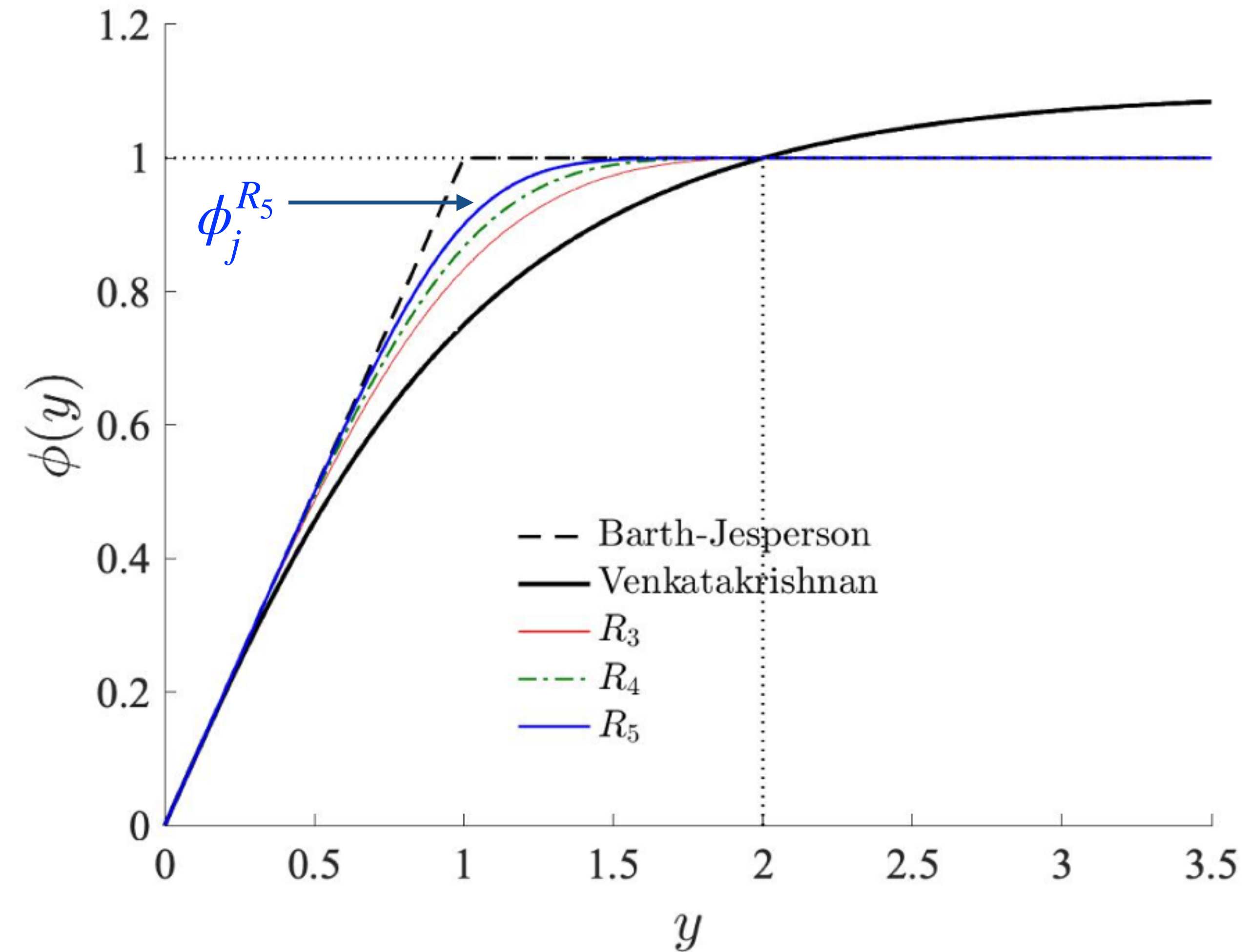
R_5 limiter preserves up to 5th-order accuracy

[Nishikawa, AIAA2022-1374.](#)

$$\rho_L = \rho_j + \phi_j^{R_5} \left(\rho_L^{Quadratic} - \rho_j \right)$$

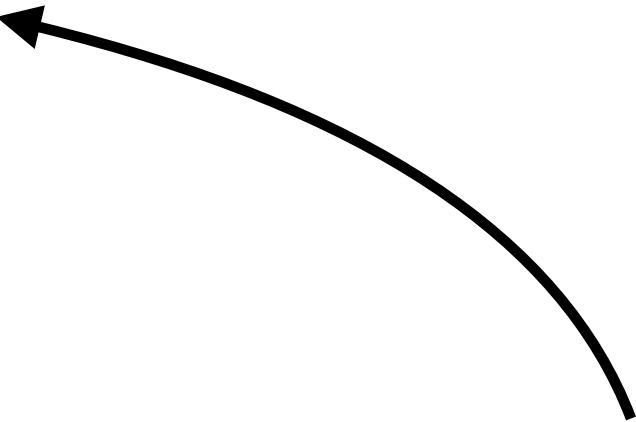
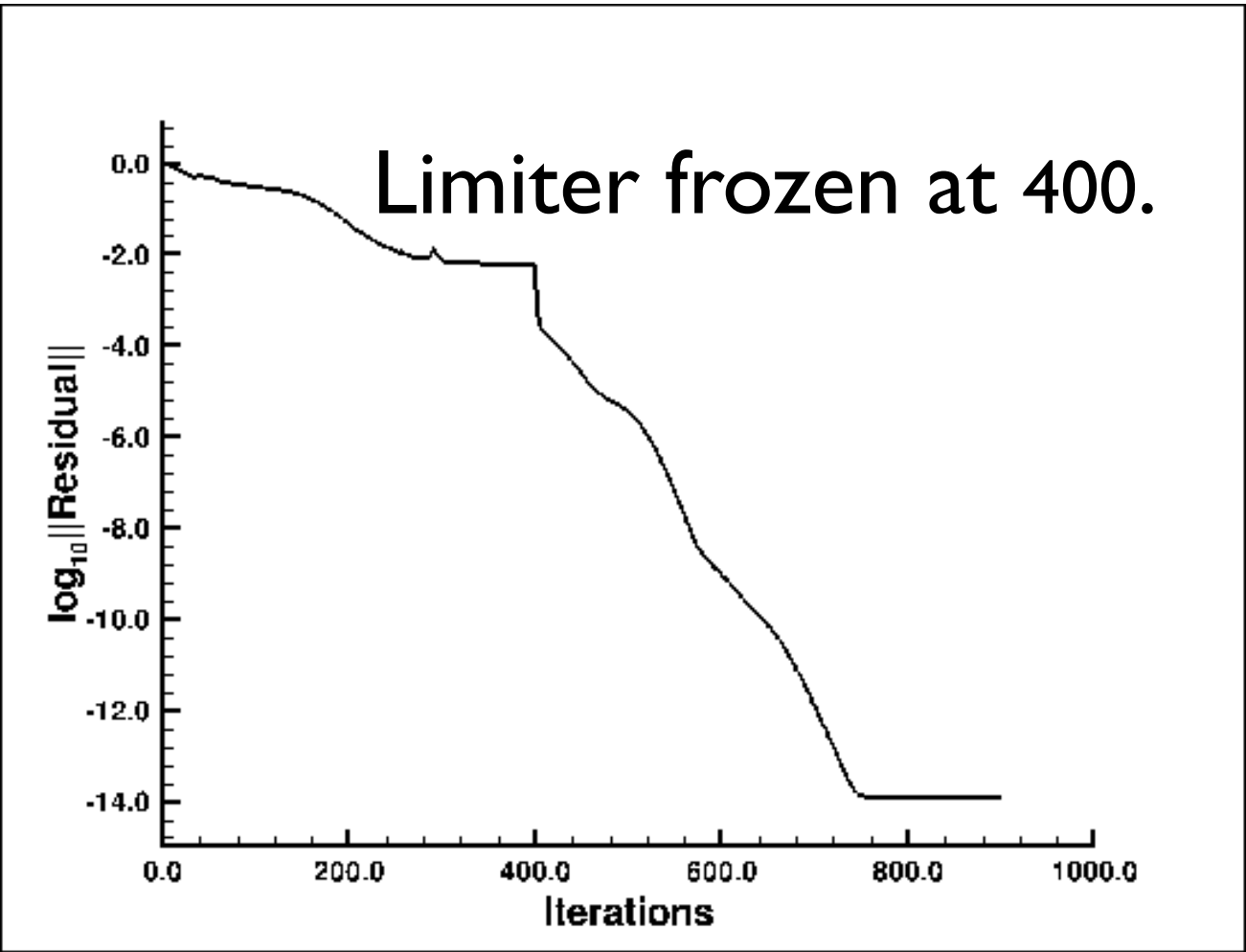
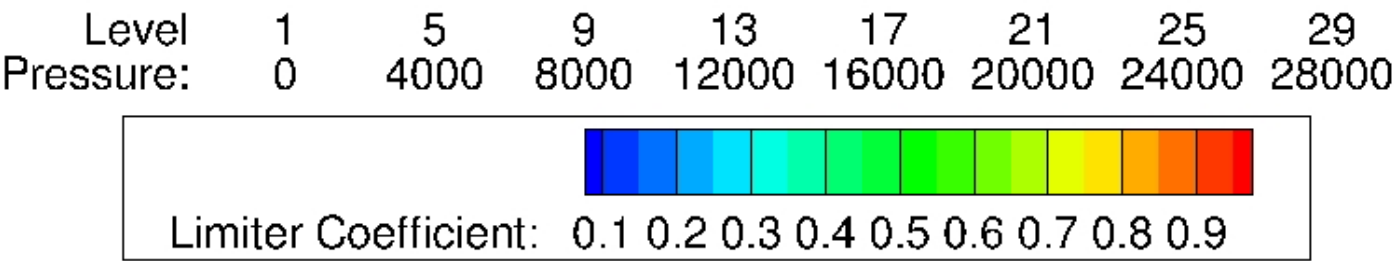
Combined with Gnoffo's pressure limiter
(AIAA2010-1271) for robustness,

Details can be found in the paper.

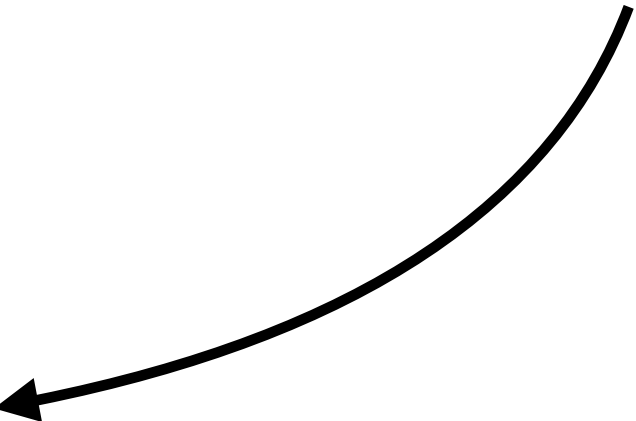
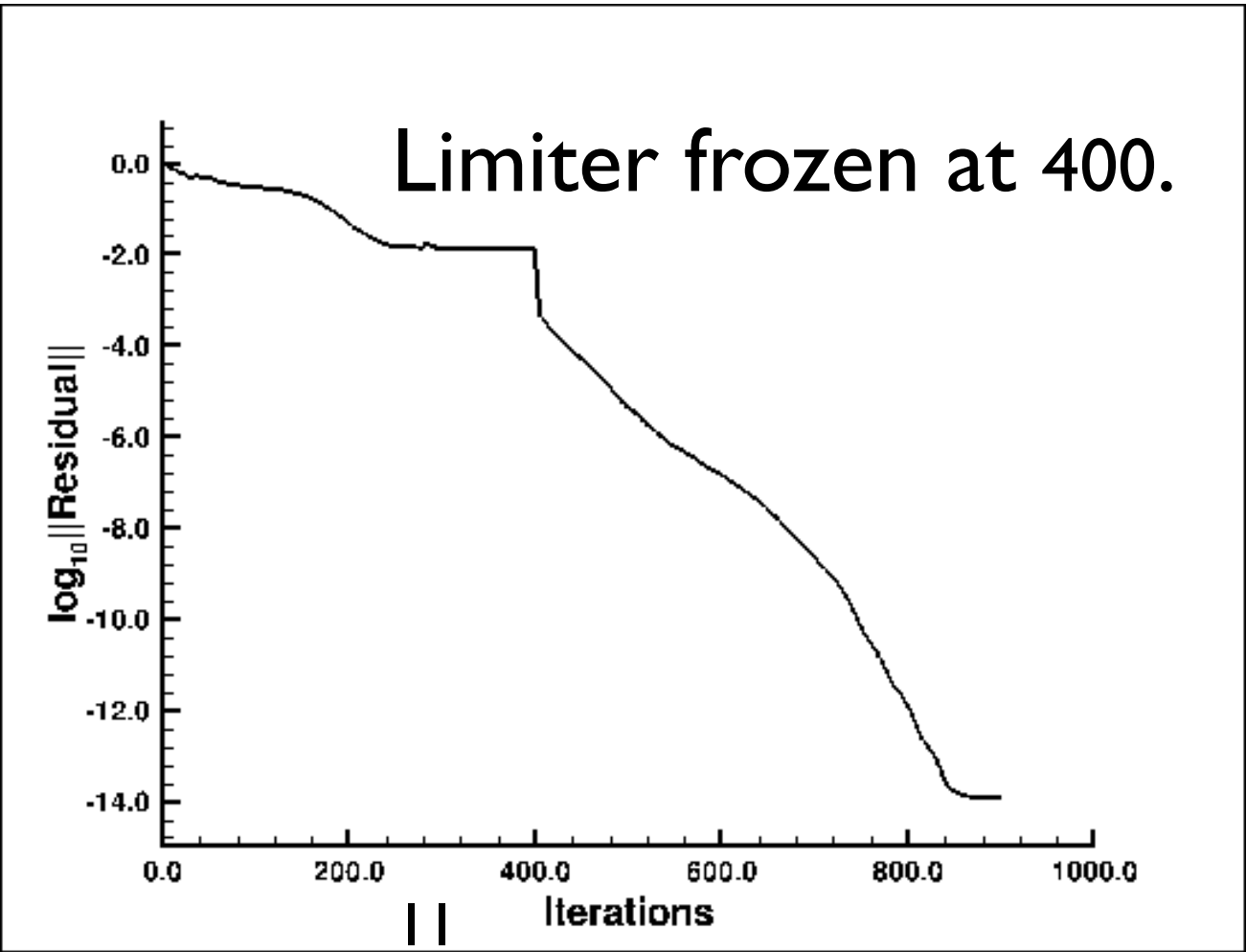
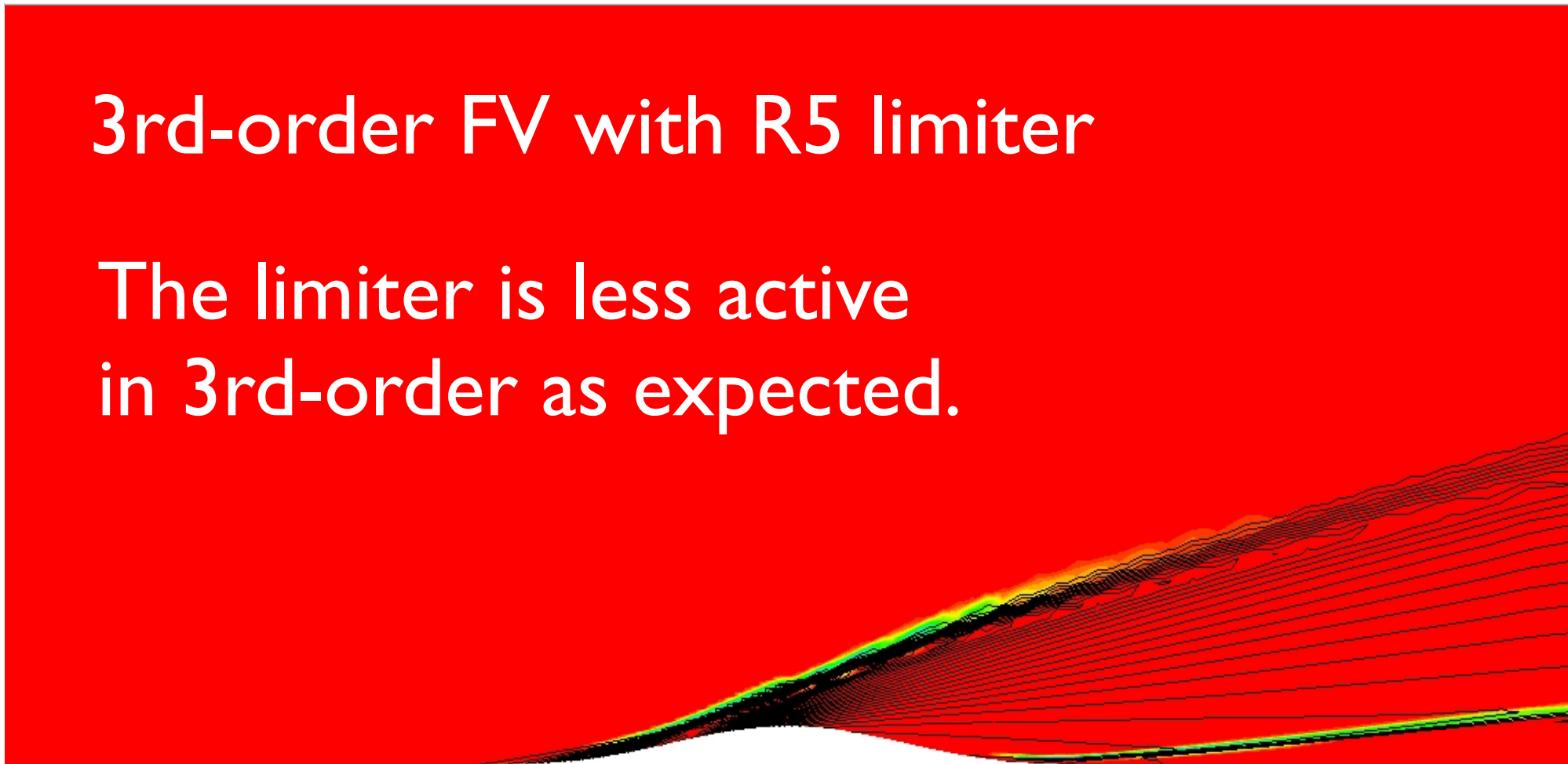


This is work in progress, and we are currently trying to improve it further.

Mach 6 Inviscid Flow over a Smooth Bump

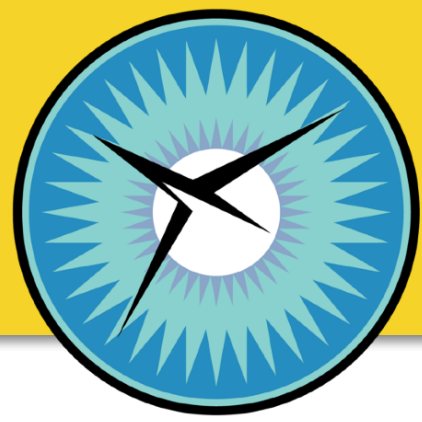


Used the same existing implicit solver:
First-order HLLC Jacobian + GS relax.

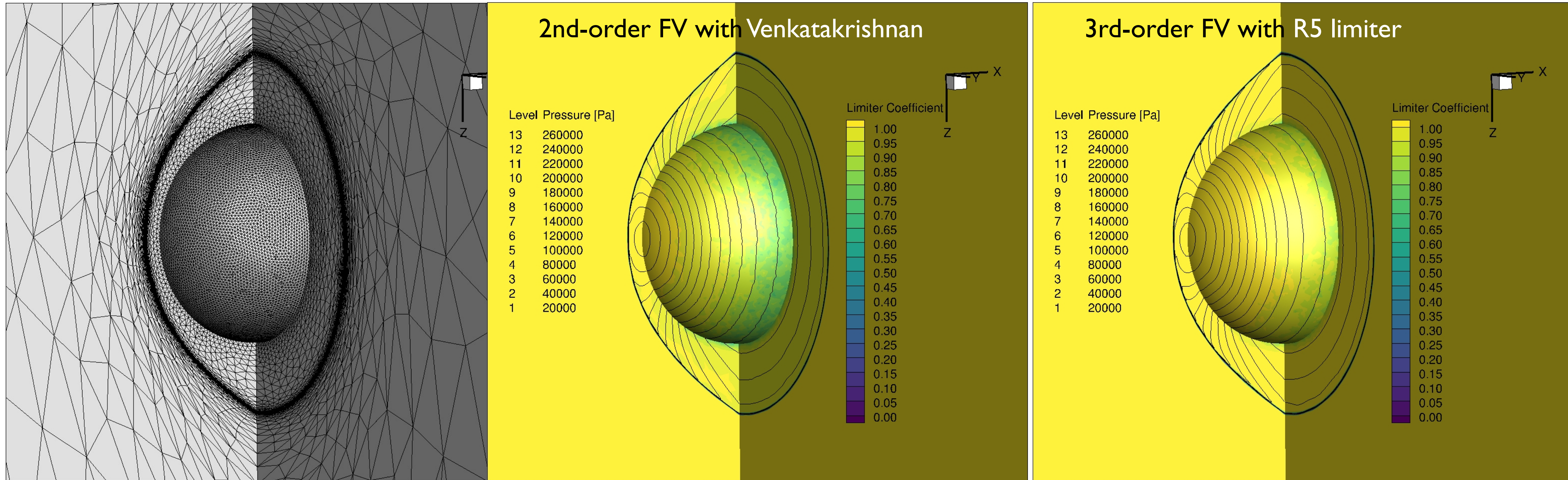


The solver works equally well for the
3rd-order discretization.

Mach 6 Inviscid Flow over a Sphere with Adapted Mesh

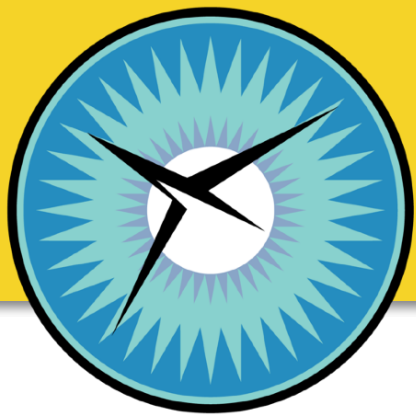


A tetrahedral mesh adapted by the refine package developed at NASA Langley.



Residuals reduction by at least 2 orders of magnitude (limiters not frozen).

3rd-order solution : (1) The same solver works robustly,
(2) Limiter is less active,
(3) Pressure contours are slightly smoother.



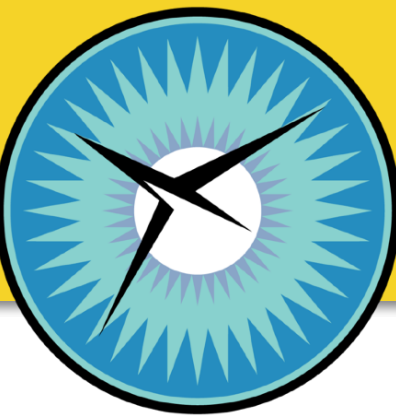
Conclusions:

- Developed a 3rd-order method with no 2nd derivatives at all.
- Implemented it in the NASA VULCAN-CFD code with a high-order limiter.
- Demonstrated an existing implicit solver works robustly for the 3rd-order scheme. for hypersonic flows with tetrahedral grids, including an adapted grid.

Future work:

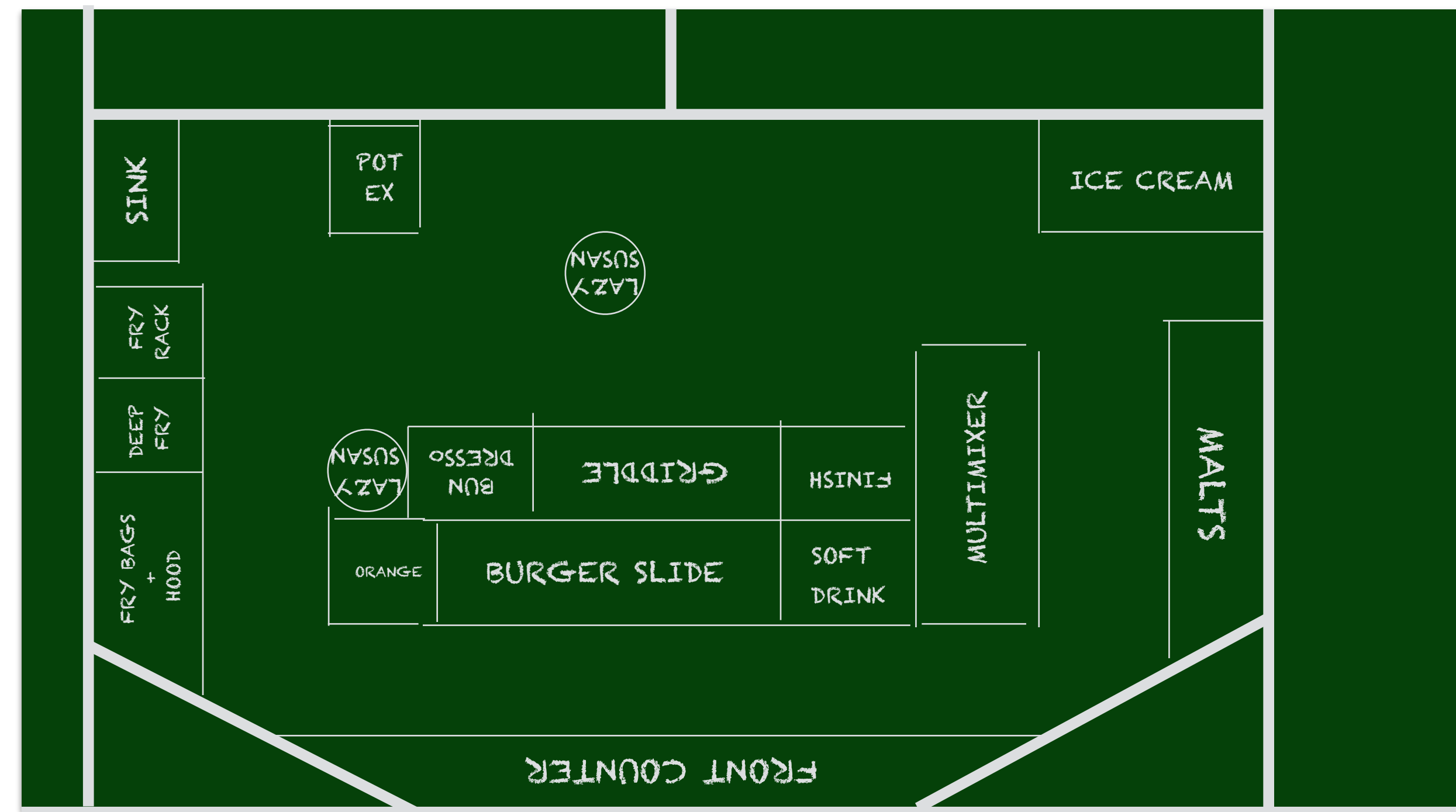
- Further investigations and enhancements to the limiter.
- Improve the viscous discretization.
- Extensions to other types of cells (hex, prism, pyramid).

More structure within unstructured



More to discover by seeking structure within unstructured data.

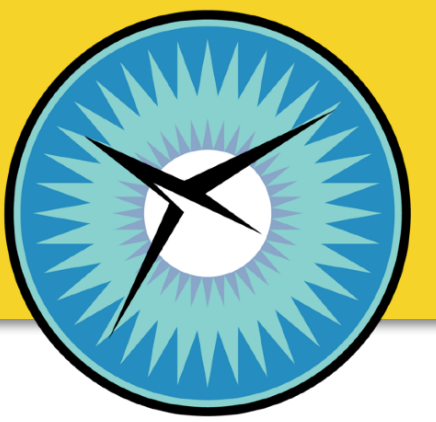
*Or create a better structure to make things easier and more efficient,
like the structured kitchen in some famous fast-food chain:*



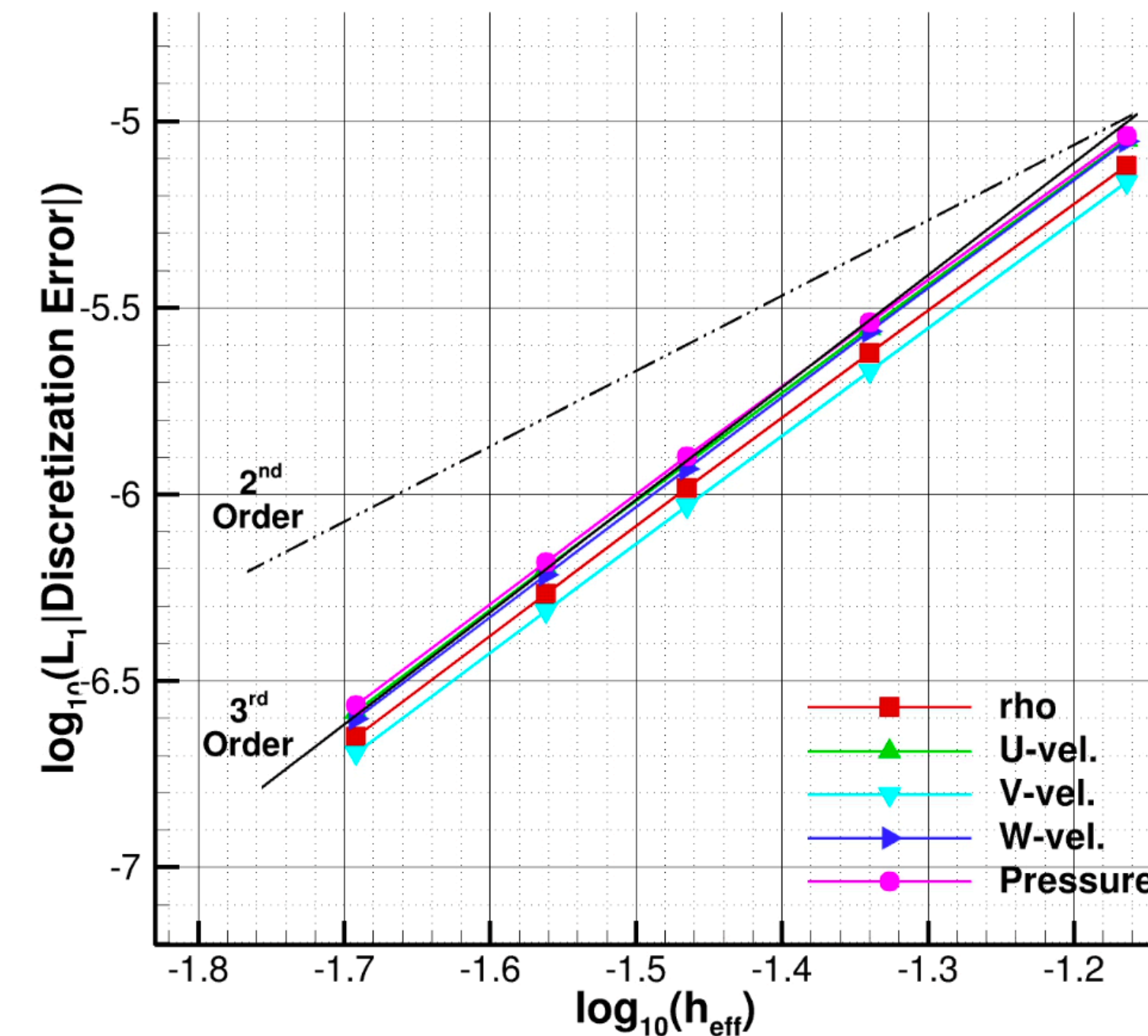
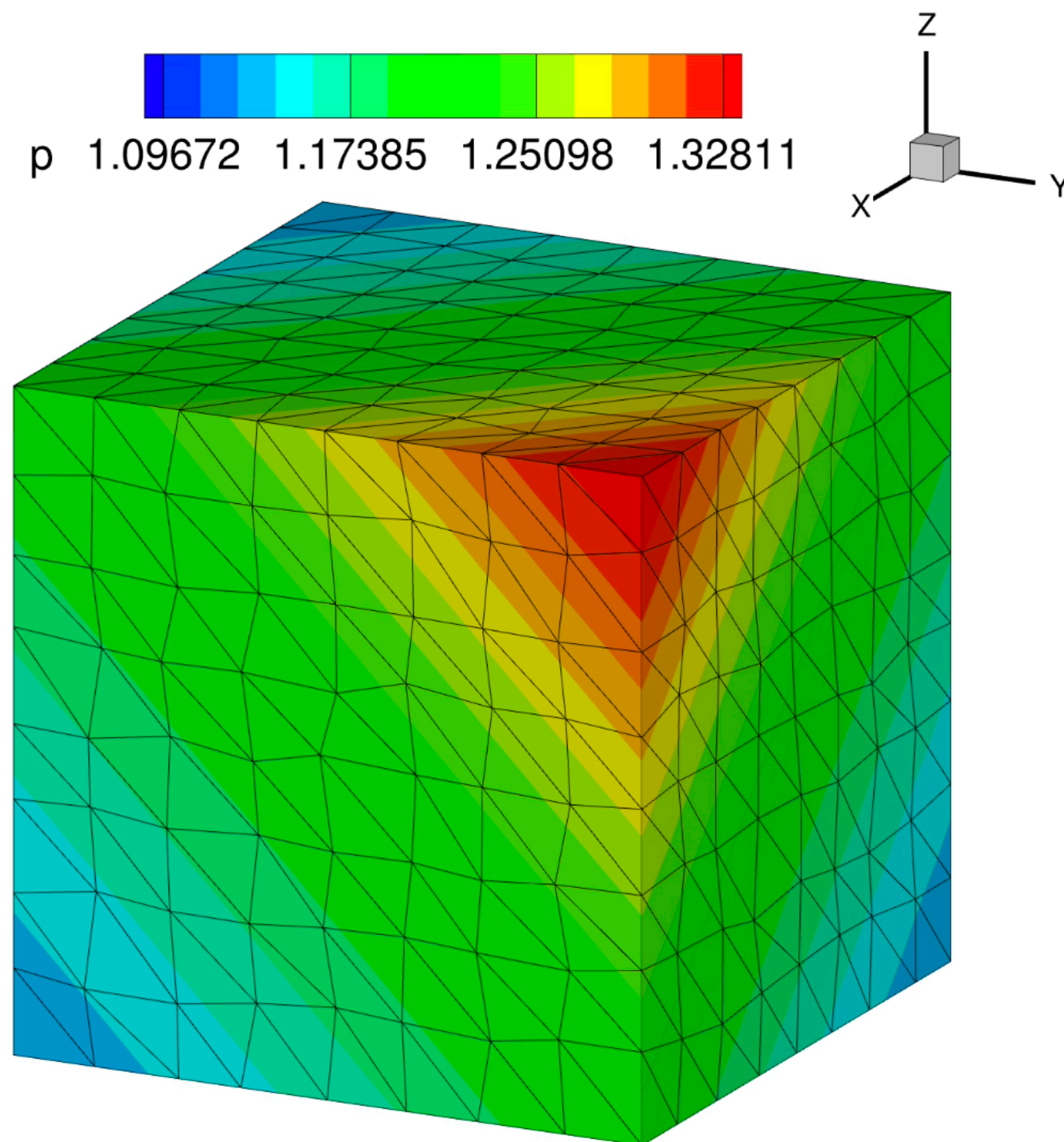
created to get “Orders ready in 30 seconds, not 30 minutes”, a founder.

Extra slides

Verified the Implementation in VULCAN-CFD

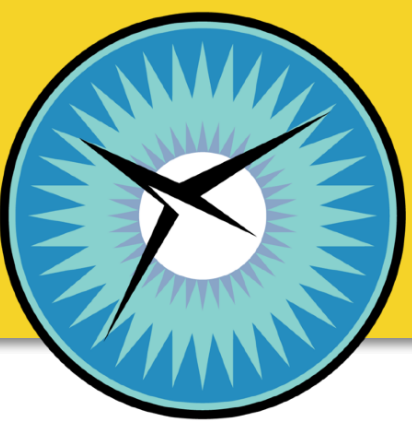


Verified 3rd-order accuracy using the method of manufactured solutions:

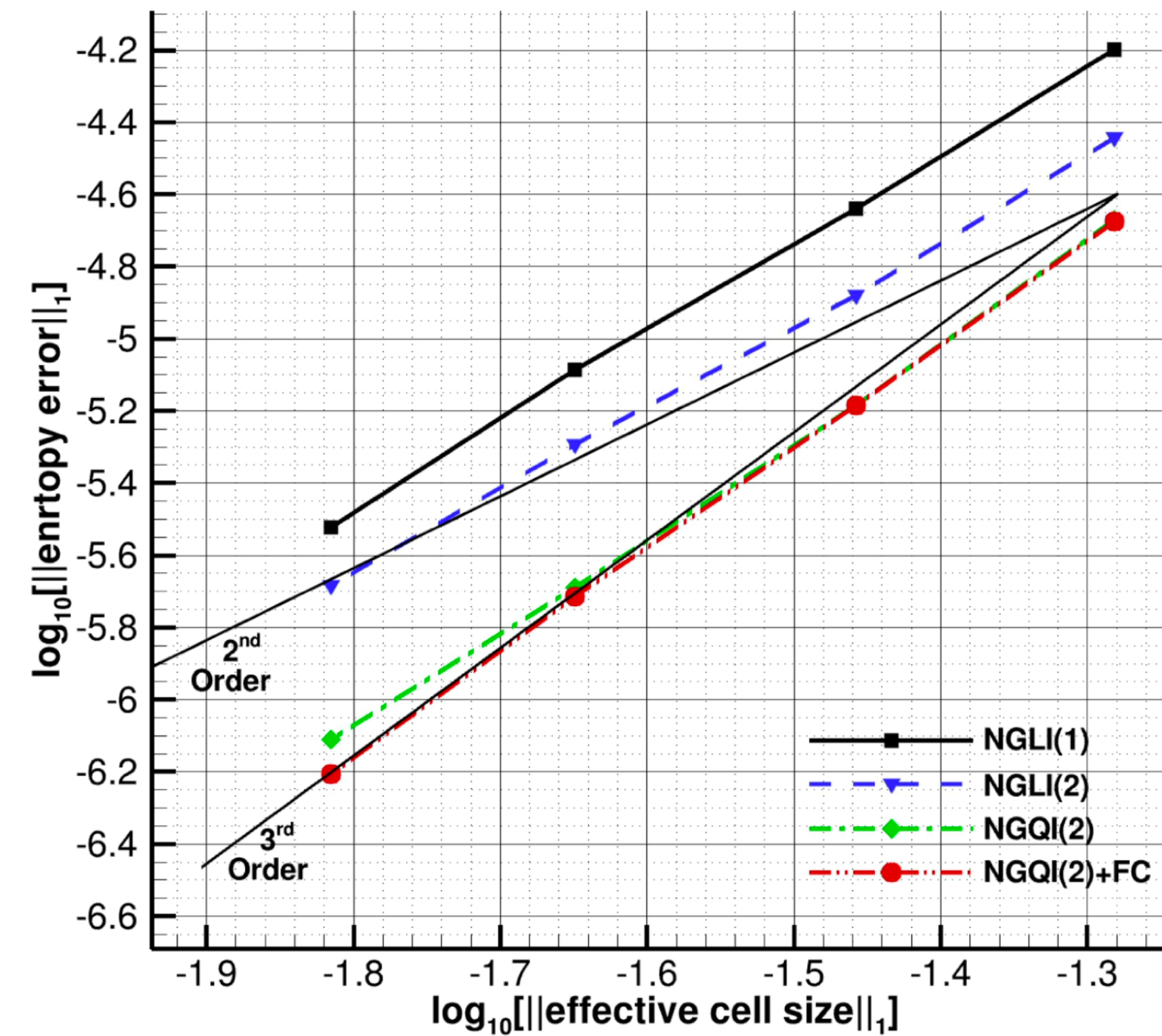
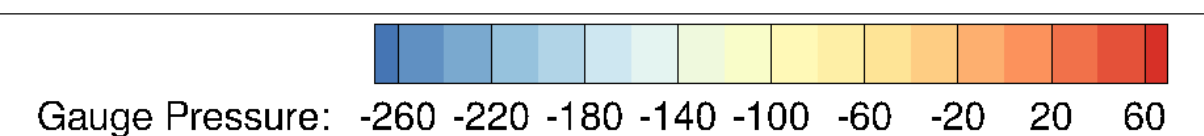
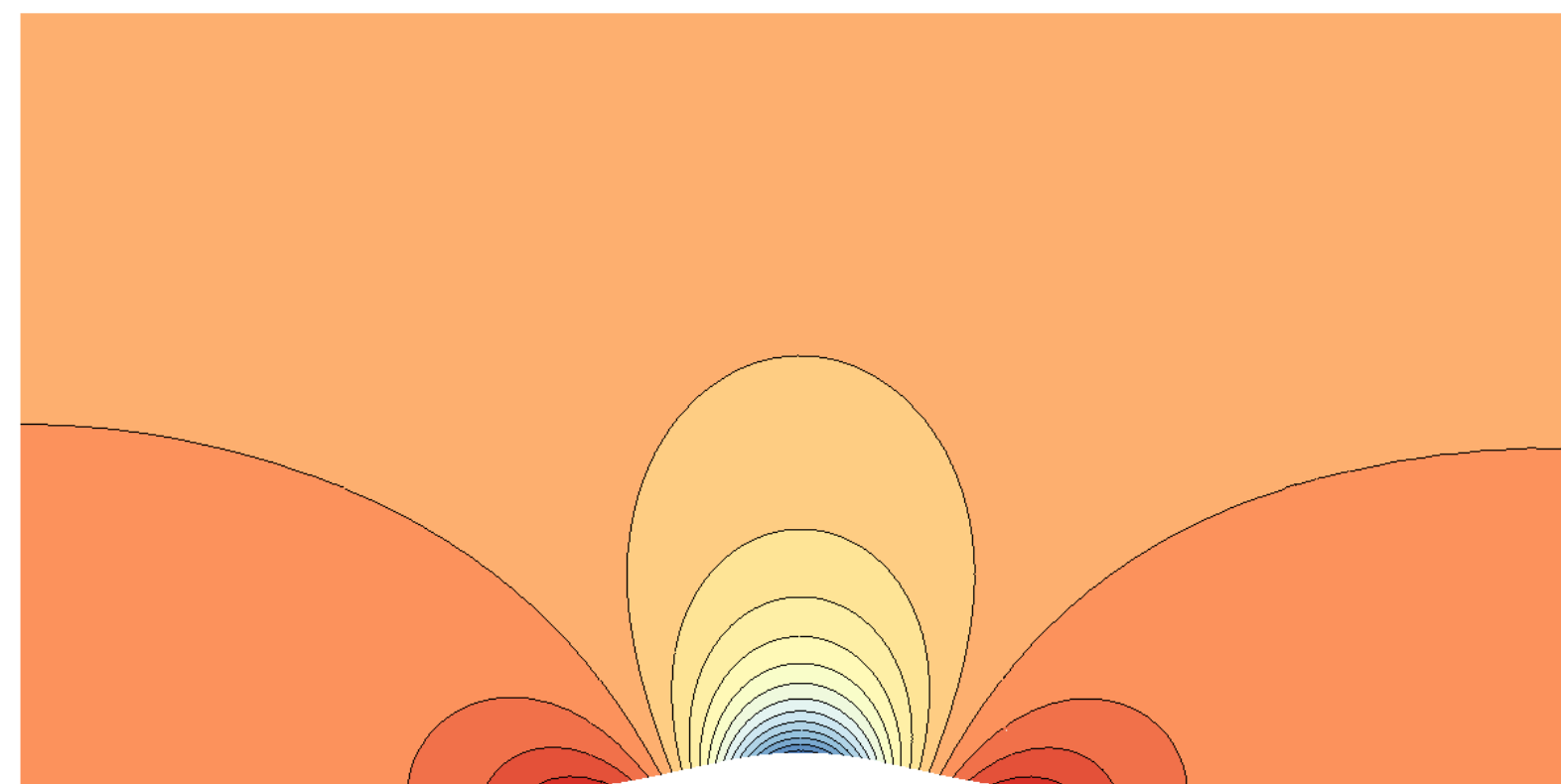
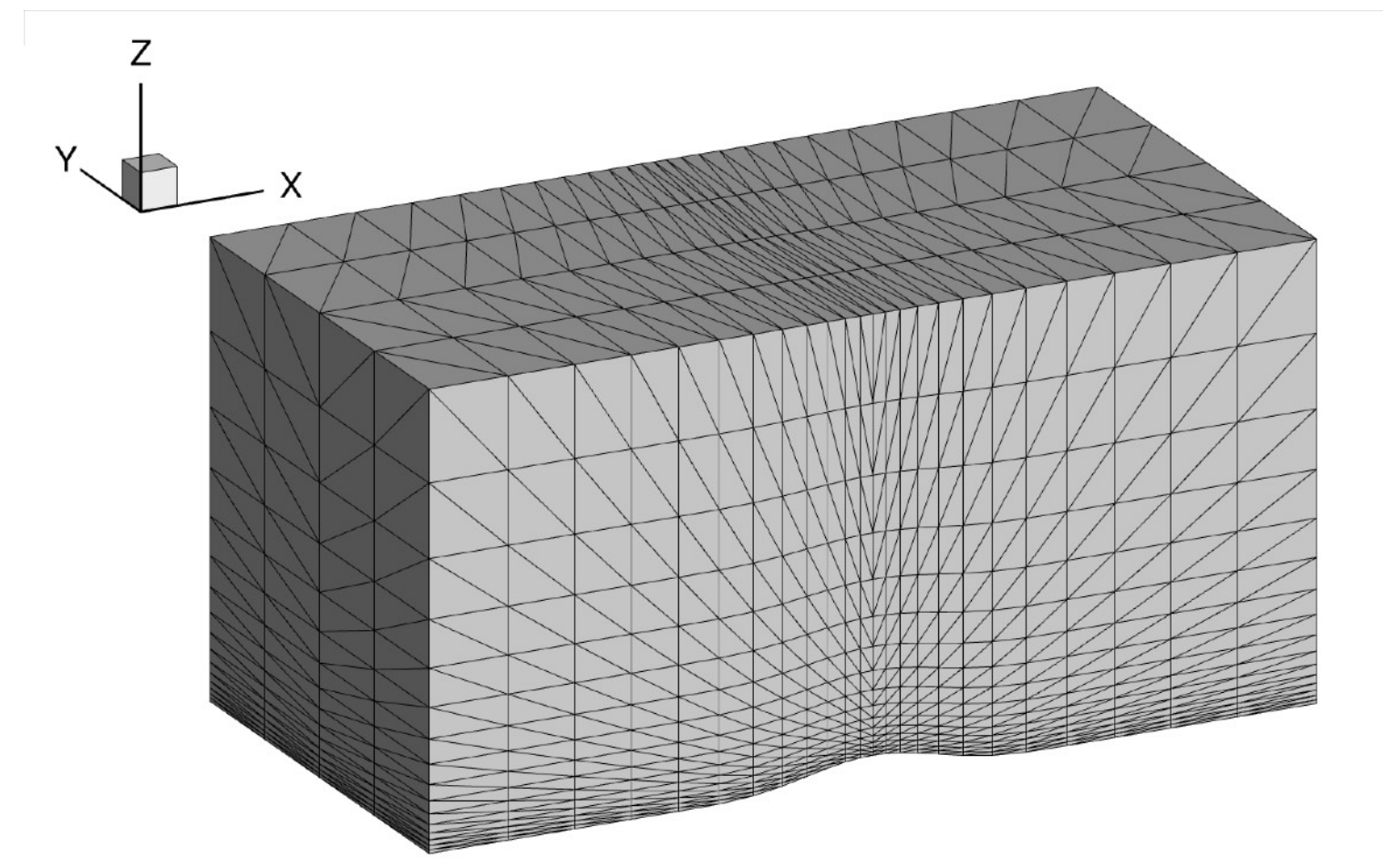


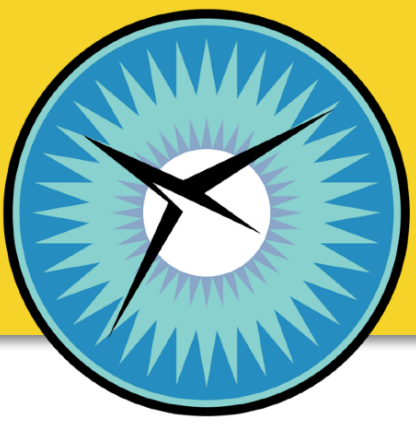
Quadratic interpolation, flux and source corrections have been implemented and verified.

Verified the Implementation in VULCAN-CFD



Verified 3rd-order accuracy in the entropy error for a subsonic flow over a smooth bump.





Need a high-order accuracy preserving limiter.

R_5 limiter preserves up to 5th-order accuracy

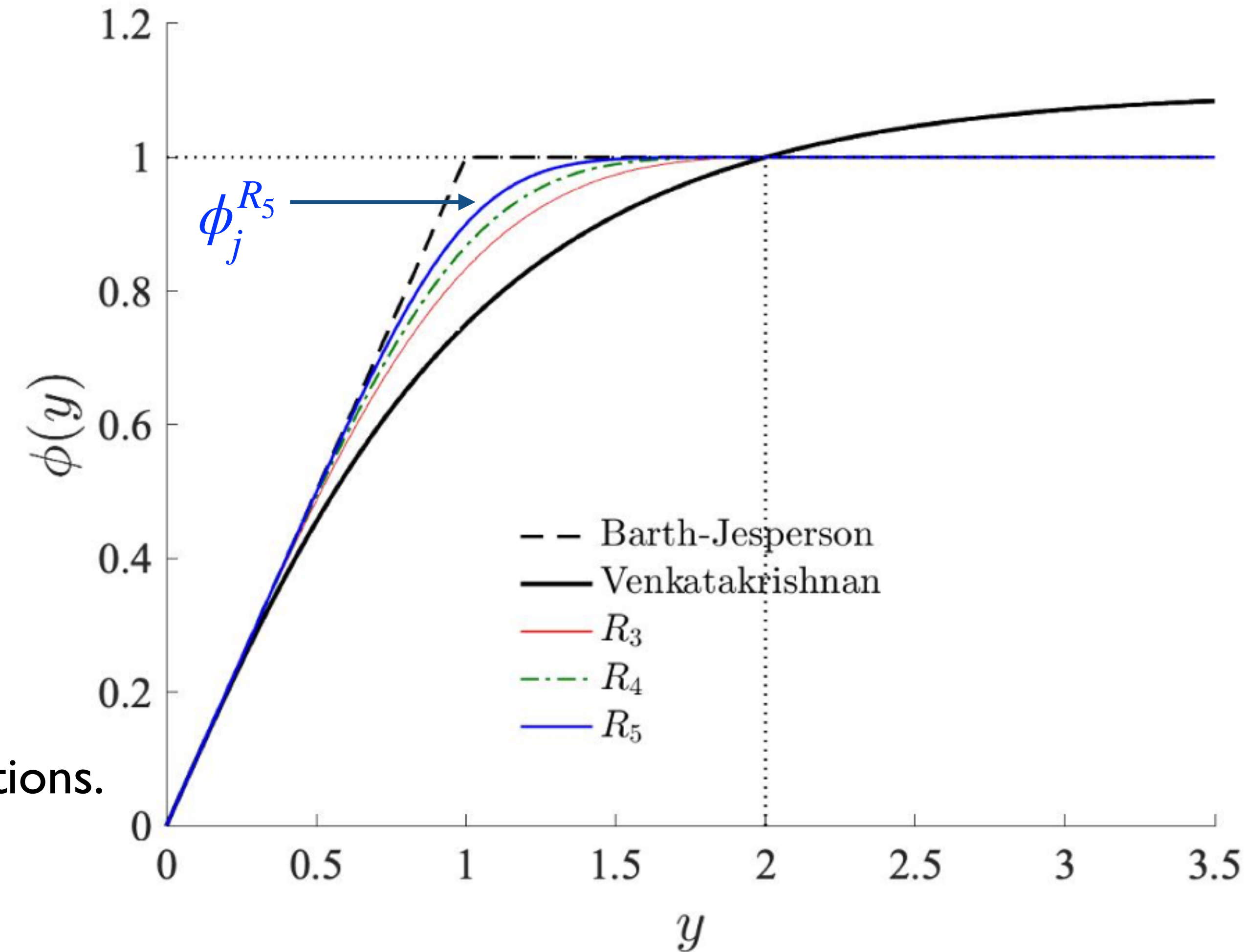
[AIAA2022-1374](#).

$$\rho_L = \rho_j + \phi_j^{R_5} \left(\rho_L^{Quadratic} - \rho_j \right)$$

ϕ_j^{press} : Gnoffo's pressure limiter (AIAA2010-1271)
= 0 at shocks or 1.

- (1) ρ_{min} and ρ_{max} are found from face neighbors.
- (2) Enforce maximum principle at face centers.
- (3) Limiter is applied also to the flux and source corrections.
- (4) Limiter is applied to the mass matrix for unsteady.

Further details in the paper.



This is work in progress, and we are currently trying to improve it further.

