Edge-Based Viscous Method for Node-Centered Formulations

Yi Liu and Boris Diskin
National Institute of Aerospace

Kyle Anderson, Eric Nielsen, and Li Wang
NASA Langley Research Center

The Virtual 2021 AIAA AVIATION Forum

August 3, 2021

Motivation

- **Edge-based finite-volume methods are widely used**
  - offer advantages of efficiency and generality
  - inviscid and viscous fluxes computed in a loop over edges
  - typically, require a discretization stencil extended beyond nearest neighbors
- **In FUN3D finite-volume (FUN3D-FV) discretization, viscous fluxes employ Green-Gauss theorem to compute gradients at grid cells**
  - cell-based viscous (CBV) method equivalent to Galerkin method on tetrahedra
  - compact nearest-neighbor stencil
  - viscous fluxes computed in a loop over cells
  - extensively verified and validated
- **Edge-based viscous (EBV) method on tetrahedra**
  - accuracy is similar to CBV method
  - improves efficiency of CBV computations (residuals and Jacobian)
  - maintains the same compact nearest-neighbor stencil
  - additional memory is required to store a few coefficients per edge
  - EBV coefficients represent grid metrics, do not depend on solution, can be precomputed

FUN3D-FV Solutions

- **Formulation**
  - Reynolds-averaged Navier-Stokes (RANS) equations with SA-neg turbulence model
- **Discretization scheme**
  - node-centered scheme on general unstructured grids, dual volumes around grid points
  - inviscid fluxes:
    - edge based
    - 2nd-order UMUSCL reconstruction to edge median
    - Roe's approximate Riemann solver
  - CBV fluxes
  - Green-Gauss gradients on dual volumes for SA-neg source term
- **Iterative solvers**
  - baseline iterative solver
    - point-implicit Gauss-Seidel sweeps, approximate Jacobian, multicolor ordering, prescribed CFL
    - Jacobian updates are scheduled according to residual convergence
  - hierarchical adaptive nonlinear iteration method (HANIM)[1]
    - strong nonlinear solver with improved efficiency, robustness, and solution-adaptive CFL

[1]Wang, L., Diskin, B., Nielsen, E., and Liu, Y. "Improvements in Iterative Convergence of FUN3D Solutions," AIAA 2021-0857

CBV Formulation on Tetrahedron

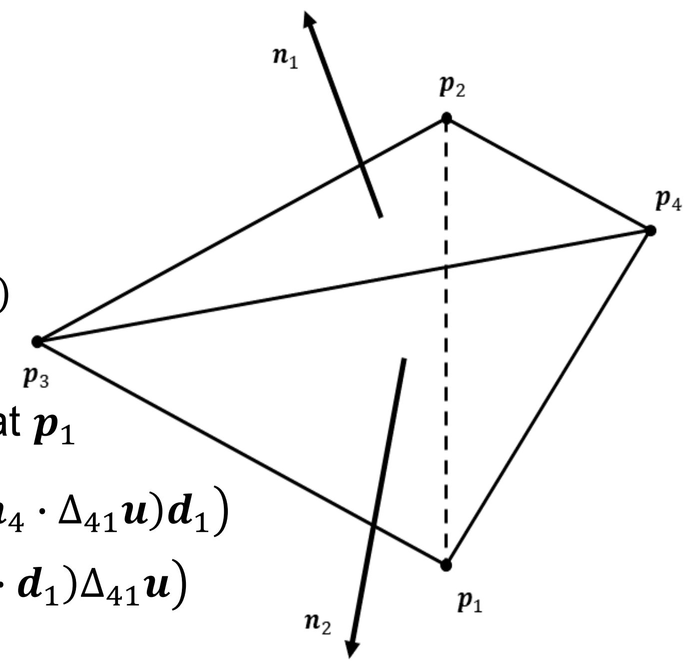
NATIONAL INSTITUTE OF AEROSPACE

Green-Gauss gradient evaluation on a tetrahedron:

$$\boldsymbol{\nabla}^h\varphi = \frac{(\varphi_2+\varphi_3+\varphi_4)\boldsymbol{n}_1 + (\varphi_1+\varphi_3+\varphi_4)\boldsymbol{n}_2 + (\varphi_1+\varphi_2+\varphi_4)\boldsymbol{n}_3 + (\varphi_1+\varphi_2+\varphi_3)\boldsymbol{n}_4}{3\,Vol}$$

$\boldsymbol{n}_i$ is outward directed area vector of face opposite to $\boldsymbol{p}_i$

$$\boldsymbol{n}_1 = -(\boldsymbol{n}_2+\boldsymbol{n}_3+\boldsymbol{n}_4)$$



Edge-based difference operator: $\Delta_{ij}\varphi \equiv \varphi_i - \varphi_j$

Edge-based gradient: $\boldsymbol{\nabla}^h\varphi = \frac{-1}{3\,Vol}(\Delta_{21}\varphi\boldsymbol{n}_2 + \Delta_{31}\varphi\boldsymbol{n}_3 + \Delta_{41}\varphi\boldsymbol{n}_4)$

Viscous-flux edge-based contributions to momentum residual at $\boldsymbol{p}_1$

$$\boldsymbol{R}_{m1} = \boldsymbol{R}_{m1} + \frac{\mu+\mu_t}{3\,Vol}\Big[-\frac{2}{3}\big((\boldsymbol{n}_2\cdot\Delta_{21}\boldsymbol{u})\boldsymbol{d}_1 + (\boldsymbol{n}_3\cdot\Delta_{31}\boldsymbol{u})\boldsymbol{d}_1 + (\boldsymbol{n}_4\cdot\Delta_{41}\boldsymbol{u})\boldsymbol{d}_1\big)$$
$$+\big((\boldsymbol{n}_2\cdot\boldsymbol{d}_1)\Delta_{21}\boldsymbol{u} + (\boldsymbol{n}_3\cdot\boldsymbol{d}_1)\Delta_{31}\boldsymbol{u} + (\boldsymbol{n}_4\cdot\boldsymbol{d}_1)\Delta_{41}\boldsymbol{u}\big)$$

$\boldsymbol{u}$ is velocity vector

$\boldsymbol{d}_i = \frac{1}{3}\boldsymbol{n}_i$ is directed area vector of dual-volume surface within tetrahedron associated with $\boldsymbol{p}_i$

CBV formulation is not edge-based because viscosity is evaluated at cell centroid

EBV Formulation on Tetrahedron

Viscosity evaluated at edge

$$(\mu + \mu_t)_{ji} = \frac{(\mu+\mu_t)_j+(\mu+\mu_t)_i}{2}$$

Viscous-flux contributions to momentum residual at $\boldsymbol{p}_1$ from edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$

$$\boldsymbol{R}_{m1} = \boldsymbol{R}_{m1} + \frac{(\mu + \mu_t)_{21}}{9\,Vol}\left[(\boldsymbol{n}_2 \cdot \boldsymbol{n}_1)\boldsymbol{I} - \frac{2}{3}\boldsymbol{n}_1\boldsymbol{n}_2^{\mathrm{T}} + \boldsymbol{n}_2\boldsymbol{n}_1^{\mathrm{T}}\right]\Delta_{21}\boldsymbol{u}$$

3×3 matrix represents nine EBV coefficients collected over all tetrahedra ($i = 1, 2, 3, \ldots$) that share edge $[\boldsymbol{p}_1, \boldsymbol{p}_2]$

$$\boldsymbol{C}_{2,1} = \sum_i \frac{1}{9\,Vol_i}\left[(\boldsymbol{n}_{i2} \cdot \boldsymbol{n}_{i1})\boldsymbol{I} - \frac{2}{3}\boldsymbol{n}_{i1}\boldsymbol{n}_{i2}^{\mathrm{T}} + \boldsymbol{n}_{i2}\boldsymbol{n}_{i1}^{\mathrm{T}}\right]$$

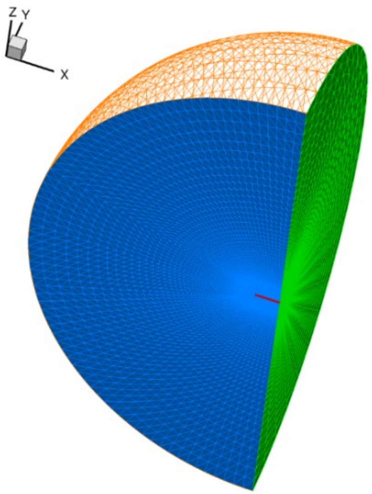
Additional EBV coefficient is needed for heat flux / SA-neg diffusion term

In reported EBV computations, separate sets of coefficients stored for each endpoint of an edge – total 20 EBV coefficients per edge

As shown in paper, EBV memory footprint can be dramatically reduced

- 7 EBV coefficients per interior edge
- 10 EBV coefficients per boundary edge

EBV and CBV Solutions on Tetrahedral Grids (Baseline Iterative Solver, T2 Grid)

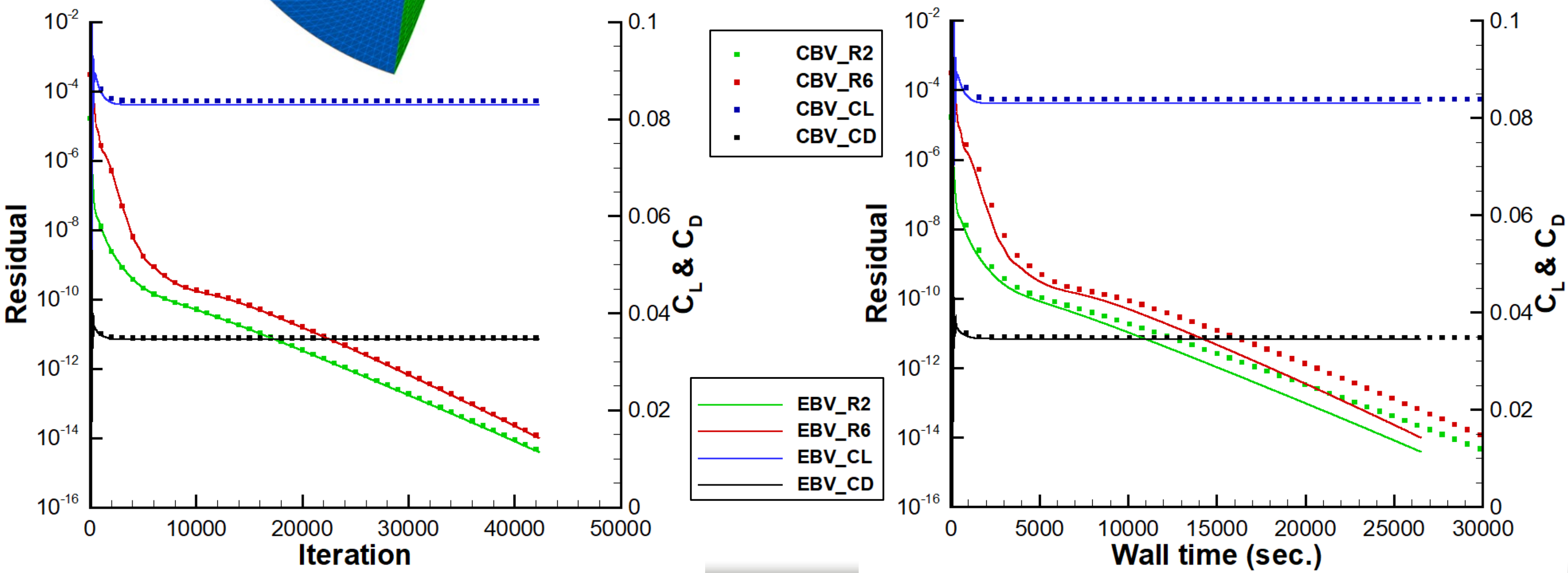


Subsonic separated flow around hemisphere cylinder

$M_\infty = 0.6$, $\alpha = 19°$, $Re_L = 3.5 \times 10^5$

Family of tetrahedral grids for hemisphere cylinder

Grid	Points	Cells	Edges
T1	8,995,153	53,084,160	62,373,200
T2	1,143,081	6,635,520	7,852,072
T3	147,637	829,440	995,444



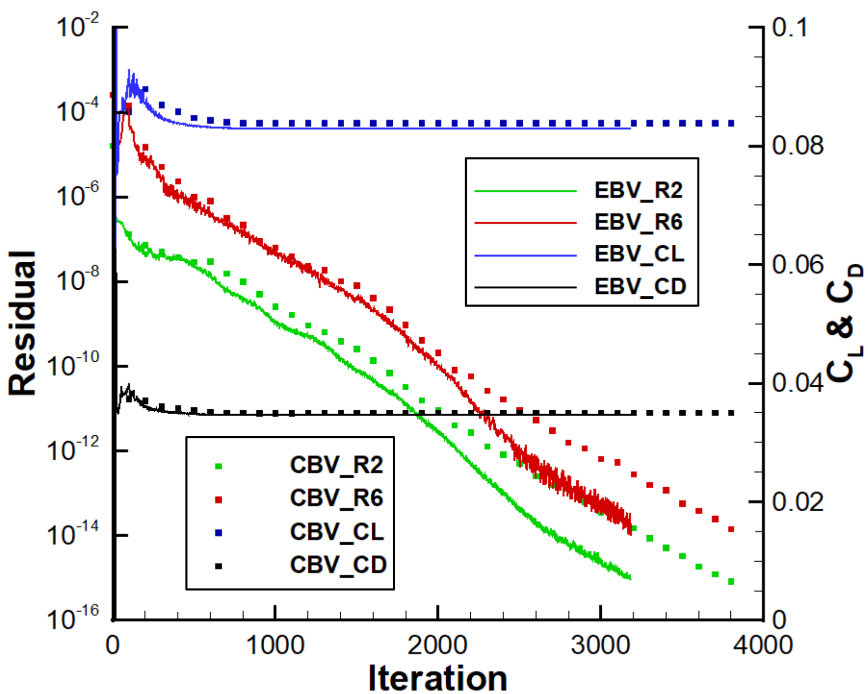
EBV Speedup on Tetrahedral Grids (Baseline Iterative Solver, T2 Grid)

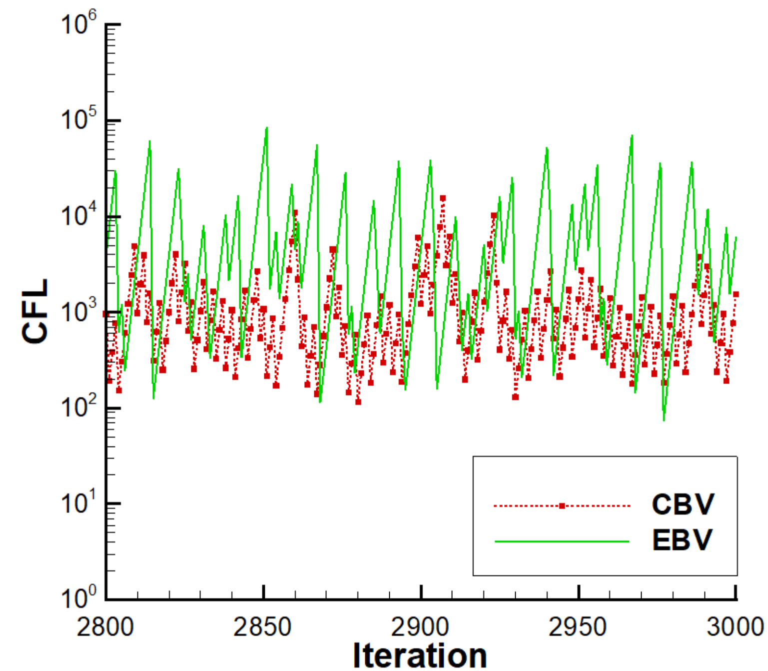
Computations	CBV time (sec.)	EBV time (sec.)	EBV speedup
Meanflow viscous fluxes	0.020	0.012	37%
SA-neg diffusion	0.061	0.009	85%
Meanflow viscous-flux Jacobian	0.131	0.076	42%
SA-neg diffusion Jacobian	0.158	0.012	92%
Viscous fluxes / diffusion (total)	0.370	0.109	70%
Iteration with Jacobian update	1.115	0.851	24%
Iteration without Jacobian update	0.656	0.602	8%
Converged solution	30,174	26,502	12%

$$\text{EBV speedup} = \frac{(\text{CBV time}) - (\text{EBV time})}{(\text{CBV time})} \times 100\%$$

The EBV speedup of a nonlinear iteration depends on the fraction of time that is spent on operations related to viscous terms. This fraction is significantly higher when Jacobians are evaluated.

EBV and CBV Solutions on Tetrahedral Grids (HANIM, T2 Grid)





	Baseline Solver		HANIM		HANIM speedup
	Iterations	**Wall time (sec.)**	**Iterations**	**Wall time (sec.)**	
CBV	42,528	30,174	3,875	6,304	79%
EBV	42,308	26,502	3,182	4,844	82%
EBV speedup		12%		23%	

Additional Contributions

- **Additional paper content**
  - formulations of the EBV method for the energy equation and the SA-neg turbulence model diffusion
  - simulations and profile data for all hemisphere-cylinder tetrahedral grids in the family
  - solutions on mixed-element grids using hybrid EBV/CBV method for a flow around a NASA juncture-flow model
  - demonstration of conservation property of CBV and EBV methods
- **Further performance improvements for EBV and CBV methods**
  - significant reduction of EBV memory footprint
  - optimization of meanflow Jacobian implementation
  - optimization of SA-neg diffusion implementation on tetrahedra (especially for CBV)
  - single edge-loop implementation benefits EBV method

Concluding Remarks

- **EBV method derived for tetrahedra and implemented in NASA's unstructured-grid, node-centered, finite-volume RANS solver FUN3D-FV**
- **EBV method verified and assessed using a benchmark subsonic separated flow around a hemisphere-cylinder configuration**
  - convergence per iteration of the baseline solver and the converged solutions computed with the EBV and CBV methods are almost identical
  - EBV speedup for the meanflow viscous fluxes and Jacobian is 37%-42%
  - EBV speedup for the SA-neg diffusion and Jacobian is 85%-92%
  - EBV speedup translates into 8%-24% reduction in the time to solution
  - opportunities for further efficiency improvements for both the EBV and CBV methods on tetrahedral grids have been identified
- **EBV method seamlessly combined with CBV method for simulations on mixed-element grids**

Acknowledgements

NATIONAL INSTITUTE OF AEROSPACE

Following NASA projects supported this work:

- Transformative Tools and Technologies (TTT) project of Transformative Aeronautics Concepts Program (TACP)
- Revolutionary Vertical Lift Technology (RVLT) project of Advanced Air Vehicles Program (AAVP)
- The first two authors are supported by the NASA Langley Research Center under the cooperative agreement 80LARC17C0004 with the National Institute of Aerospace

Most of computations were performed on NASA LaRC K-Cluster and on NASA Advanced Supercomputing facility's Pleiades supercomputer

Authors would like to thank Dr. Hiro Nishikawa and Dr. Mike Park for reviewing the paper and providing helpful comments

Thank You!