Recovery Discontinuous Galerkin Jacobian-Free Newton-Krylov Method for All-Speed Flows

Fifth International Conference on Computational Fluid Dynamics (ICCFD5)

HyeongKae Park
Robert Nourgaliev
Vincent Mousseau
Dana Knoll

July 2008

Recovery Discontinuous Galerkin Jacobian-free Newton-Krylov Method for All-Speed Flows

HyeongKae Park, Robert Nourgaliev, Vincent Mousseau and Dana Knoll

Idaho National Laboratory Multiphysics Methods Group,
Idaho Falls, ID, 83415-3840, USA `Ryosuke.Park@inl.gov`
`Robert.Nourgaliev@inl.gov Vincent.Mousseau@inl.gov Dana.Knoll@inl.gov`

1 Introduction

There is an increasing interest to develop the next generation simulation tools for the advanced nuclear energy systems. These tools will utilize the state-of-art numerical algorithms and computer science technology in order to maximize the predictive capability, support advanced reactor designs, reduce uncertainty and increase safety margins. In analyzing nuclear energy systems, we are interested in compressible low-Mach number, high heat flux flows with a wide range of Re, Ra, and Pr numbers. Under these conditions, the focus is placed on turbulent heat transfer, in contrast to other industries whose main interest is in capturing turbulent mixing. Our objective is to develop single-point turbulence closure models for large-scale engineering CFD code, using Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES) tools, requireing very accurate and efficient numerical algorithms.

The focus of this work is placed on fully-implicit, high-order spatiotemporal discretization based on the discontinuous Galerkin method solving the conservative form of the compressible Navier-Stokes equations. The method utilizes a local reconstruction procedure derived from weak formulation of the problem, which is inspired by the recovery diffusion flux algorithm of van Leer and Nomura [VN05] and by the piecewise parabolic reconstruction [CW84] in the finite volume method. The developed methodology is integrated into the Jacobian-free Newton-Krylov framework [KK04] to allow a fully-implicit solution of the problem.

2 Jacobian-free Newton-Krylov Framework

The Jacobian-free Newton-Krylov (JFNK) [KK04] framework enables a fully-coupled solution of the nonlinear system. JFNK is a combination of quadratically convergent Newton's method and a Krylov subspace method [Saa03]. In each Newton iteration, the following linear system of equations is solved:

$$J^k \delta \mathbf{U}^k = -Res(\mathbf{U}^k) \tag{1a}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \delta \mathbf{U}^k \tag{1b}$$

where, the superscript k denotes the kth Newton iterate. $J \equiv \frac{\partial Res}{\partial U}$ is the Jacobian matrix and $Res(\mathbf{U}^k)$ denotes the nonlinear residual functions. Computation of the Jacobian matrix can be prohibitively expensive. The JFNK method eliminates the direct computation of the Jacobian matrix by taking advantage of the fact that Krylov methods require only matrix-vector product to perform the iterations. To this end, the Jacobian-vector product is approximated by the forward finite difference of the form:

$$J\mathbf{v} \approx \frac{Res(\mathbf{U} + \varepsilon\mathbf{v}) - Res(\mathbf{U})}{\varepsilon} \tag{2}$$

With this formulation, the linear system of equations can be solved by providing only subroutines for the nonlinear residual function evaluation. Thus, the JFNK framework provides a powerful platform to include a variety of nonlinear phenomena, especially useful for multiphysics simulations. This "Jacobian-free" version of Krylov methods can also be applied to the preconditioned system, which is the key to efficient numerical algorithm[1]. The Jacobian-vector product of the (right-) preconditioned system can be expressed as:

$$JM^{-1}\mathbf{v} \approx \frac{Res(\mathbf{U} + \varepsilon M^{-1}\mathbf{v}) - Res(\mathbf{U})}{\varepsilon} \tag{3}$$

3 Recovery Discontinuous Galerkin Method

The "recovery" discontinuous Galerkin (rDG) method was first introduced by van Leer and Nomura [VN05] for a diffusion operator. Their idea was to approximate a diffusion flux at cell edges by a higher-order reconstruction of the solution. As we found out, the concept of "recovery" can be extended to the advection and reaction operators. The rDG for the advection operator may be considered as the generalization of the Piecewise Parabolic finite volume method (PPM) [CW84]. In rDG, the higher-order solution within a cell is reconstructed by utilizing the solutions at the immediate neighboring cells. To do so, we first define the "recovered" function in cell i as:

$$\tilde{\mathbb{U}}_i(x) = \sum_{n=0}^{R} \tilde{\mathbb{U}}_i^n \mathfrak{L}_n(x) \tag{4}$$

where $\tilde{\mathbb{U}}_i^n$ is the coefficient of the nth-order recovered function $\tilde{\mathbb{U}}_i(x)$. The recovered function is the Rth-order polynomial, where $R = 3p + 2$, and p

---

[1] For recent developments of the physics-based preconditioning in nuclear reactor applications, see [PNM08]

is the order of the original DG method. The coefficients are computed by enforcing the following weak formulations:

$$\int_{\mathcal{I}_m} \mathfrak{L}_n(x)\, \tilde{\mathbb{U}}_m(x)\, dx = \int_{\mathcal{I}_m} \mathfrak{L}_n(x)\, U_m(x)\, dx$$
$$m = i, i\pm1 \quad \text{and} \quad n = 0,\dots,p \tag{5}$$

where U_m is the original DG solution in cell m. Eq. (5) produces the $(R+1)\times(R+1)$ system of equations, which can be solved analytically. Since the weak statements are used in reconstruction, the procedure is conservative and the first p-moments of the recovered coefficients are equal to those of the original DG solution. The advantage of the rDG is that the higher-order corrections are used only to compute the numerical fluxes at cell edges or integral reaction terms which appears upon DG discretization. As a result, it does not increase the number of unknowns of the solved nonlinear system.

Two-dimensional extension of rDG The recovery procedure discussed above can be extended to two-dimensional problems. The lowest order discretization $rDG_0(=\text{PPM})$ uses the direction-by-direction recovery procedure; therefore it is only 1st order in the cross derivatives. The recovered in-cell solution of the high-order rDG, on the other hand, includes high-order cross-derivative terms, as well as normal derivatives. The key to this procedure is the choice of degrees of freedom (DoF) that avoids a null space. Fig. 1 shows our choice of DoF from neighboring cells in rDG_1, and Eq. (6) expresses the resulting recovered function.

$$U_{i,j}(x,y) = U_{ij}^{00} + U_{ij}^{10} + U_{ij}^{01}$$

Fig. 1. The choice of the in-cell solution reconstruction for rDG_1

$$
\begin{aligned}
\tilde{\mathbb{U}} = \ & \tilde{\mathbb{U}}^{00}\mathfrak{L}_0(x)\mathfrak{L}_0(y) \\
& + \tilde{\mathbb{U}}^{10}\mathfrak{L}_1(x)\mathfrak{L}_0(y) + \tilde{\mathbb{U}}^{01}\mathfrak{L}_0(x)\mathfrak{L}_1(y) \\
& + \tilde{\mathbb{U}}^{20}\mathfrak{L}_2(x)\mathfrak{L}_0(y) + \tilde{\mathbb{U}}^{02}\mathfrak{L}_0(x)\mathfrak{L}_2(y) + \tilde{\mathbb{U}}^{11}\mathfrak{L}_1(x)\mathfrak{L}_1(y) \\
& + \tilde{\mathbb{U}}^{30}\mathfrak{L}_3(x)\mathfrak{L}_0(y) + \tilde{\mathbb{U}}^{03}\mathfrak{L}_0(x)\mathfrak{L}_3(y) + \tilde{\mathbb{U}}^{12}\mathfrak{L}_1(x)\mathfrak{L}_2(y) + \tilde{\mathbb{U}}^{21}\mathfrak{L}_2(x)\mathfrak{L}_1(y) \\
& + \tilde{\mathbb{U}}^{40}\mathfrak{L}_4(x)\mathfrak{L}_0(y) + \tilde{\mathbb{U}}^{04}\mathfrak{L}_0(x)\mathfrak{L}_4(y) + \tilde{\mathbb{U}}^{22}\mathfrak{L}_2(x)\mathfrak{L}_2(y) \\
& + \tilde{\mathbb{U}}^{50}\mathfrak{L}_5(x)\mathfrak{L}_0(y) + \tilde{\mathbb{U}}^{05}\mathfrak{L}_0(x)\mathfrak{L}_5(y) \\
& + O(\Delta x^6, \Delta y^6, \Delta x^3 \Delta y, \Delta x \Delta y^3)
\end{aligned}
\tag{6}
$$

where the subscripts denote the order of polynomial in x- and y-direction, respectively. From Eq. (6), it is clear that the reconstructed function is the 6th order accurate in x- and y-direction and the 4th order accurate in cross-derivative terms.

4 Numerical Examples

In this section, we demonstrate high-order convergence of the rDG by solving the one- and two-dimensional Navier-Stokes equations. As the first example, we use a one-dimensional manufactured solution [NTP08]. Fig. 2 (left) shows the density error vs. total number of DoFs, used for rDG_{0-3}. All spatial discretizations exhibit nearly theoretical convergence rates. Fig. 2 (right) shows the convergence of five fully implicit time integration schemes, which also demonstrates the expected convergence rates.
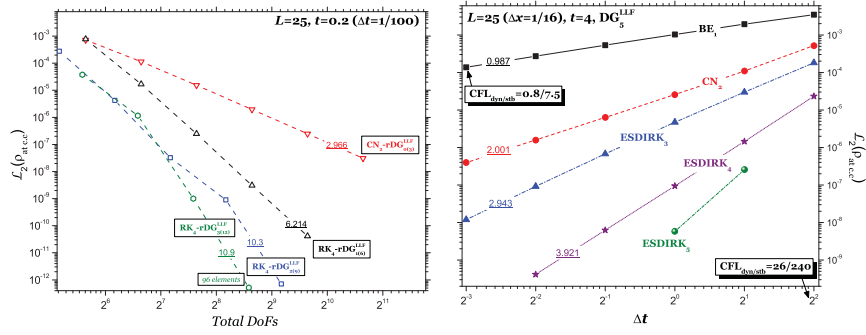

Fig. 2. Spatial (left), and temporal (right) convergence tests for rDG_{0-3}

The next example is the 2-D traveling wave problem, defined as

$$
u(x,y,t) = 1 + 2\cos(2\pi(x-t))\sin(2\pi(y-t)) \tag{7a}
$$

$$
v(x,y,t) = 1 - 2\sin(2\pi(x-t))\cos(2\pi(y-t)) \tag{7b}
$$

$$
P(x,y,t) = P_0 - (\cos(4\pi(x-t)) + \cos(4\pi(y-t))) \tag{7c}
$$

Fig. 3 summarizes the convergence studies, comparing the DG_1 with rDG_1, for total energy and x-momentum variables. Samples of the velocity and vorticity fields at t=0.0 and t=0.15 are shown at the bottom. As can be seen from Fig. 3, the convergence rates are the 2nd and the 6th order for DG_1 and rDG_1, respectively. A significant increase in accuracy is gained. Fig. 4 shows the efficacy (accuracy vs. CPU time), comparing the DG_1 with the rDG_1. Since the average number of the GMRES and Newton iterations does not increase in this problem, the CPU time solely depends on the number of nonlinear function evaluations. As one can see from Fig. 4, up to five orders of magnitude more accurate result can be obtained with same CPU time, On the other hand, to obtain the same accuracy, thr rDG_1 is ≈ 100 times faster than the DG_1.
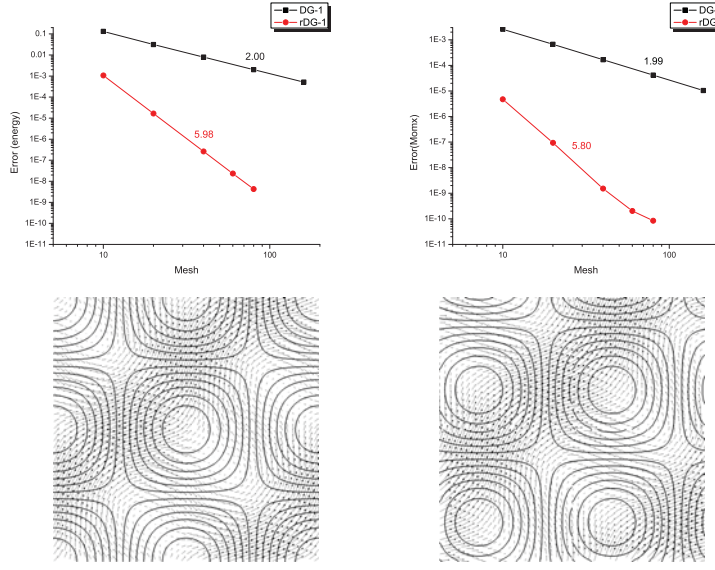

Fig. 3. Grid convergence of rDG for total energy (upper left), and x-momentum (upper right). Vorticity and velocity fields are shown at t=0.0 (bottom left), t=0.15 (bottom right)

5 Conclusion

In this work, we have developed very high-order spatial discretization scheme for hyperbolic and reaction operators based on the recovery of DG method. Numerical examples demonstrated nearly theoretical convergence for the smooth-flow tests. A significant saving in the computational time has been
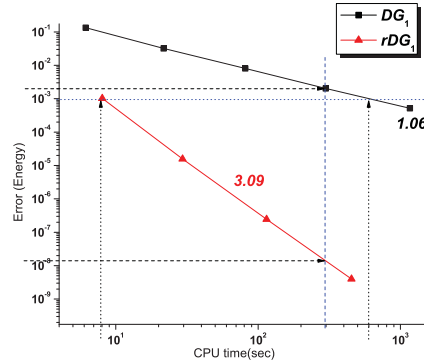
Fig. 4. Error vs. CPU time for the traveling-wave problem

achieved. The recovery procedure uses only the immediate neighbors to estimate higher order correction terms in the solution, which can be utilized for *a posteriori* error indicator and subsequent mesh adaptivity. The next step is to extend the procedure to unstructured grids as well as to the three-dimensional problems.

References

[KK04]   Knoll, D. A., Keyes, D.: Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications Journal of Computational Physics, **193**, 357 (2004)

[CW84]   Colella, P., Woodward, P. R.: The Piecewise Paraboilic Method (PPM) for Gas-Dynamical Simulations Journal of Computational Physics **54**, 174 (1984)

[VN05]   van Leer, B., Nomura, S.: Discontinuous Galerkin for Diffusion, 17th AIAA Computational Fluid Dynamics Conference Tronto, Ontario, Canada (2005)

[NTP08] Nourgaliev, R. R., et al.: Direct Numerical Simulation of Interfacial Flows: Implicit Sharp-Intrface Method (I-SIM), 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, (2008)

[Saa03] Saad, Y.: Iterative Methods for Sparse Linear Systems, SIAM, philadelphia, 2nd edition (2003)

[PNM08] Park, H., et al.: Jacobian-free Newton-Krylov discontinuous Galerkin method and physics-based preconditioning for nuclear reactor simulations, International Conference on the Physics of Reactors (PHYSOR2008), Interlaken, Switzerland (2008)