Adaptive Exponential Time Integration of the Navier-Stokes Equations

Conference Paper · January 2020

DOI: 10.2514/6.2020-2033

3 authors, including:

Shu-Jie Li
Beijing Computational Science Research Center
18 PUBLICATIONS **73** CITATIONS

SEE PROFILE

AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-2033

Adaptive Exponential Time Integration of the Navier-Stokes Equations

Shu-Jie Li*
Beijing Computational Science Research Center, Beijing, 100193, China
Lili Ju†
University of South Carolina, Columbia, SC 29208, USA
Hang Si‡
Weierstrass Institute for Applied Analysis and Stochastics, Berlin,10117,Germany

In this paper, exponential time integration methods are assessed in a solution adaptation frame with a model discontinuous Galerkin (DG) method for high-order simulations of realistic, three-dimensional viscous flows. The adaptive cell order of accuracy is determined dynamically by a local cell error indicator. The adaptive algorithm starts up with the first order of accuracy globally and then is locally refined up to its desirable highest order of accuracy in a fully multigrid like procedure. Compared with the uniformly high-order exponential DG method, the adaptive framework requires much less memory and leads to nearly three-fold times speedups for the cases tested while maintaining the same level of accuracy.

I. Introduction

The development of an efficient high-order computational fluid dynamics (CFD) solver is one of the most concerned problems towards industrial applications. While various high-order methods have been gaining popularity for their accuracy and flexibility on unstructured grids, the computational efficiency of any high-order method could be enhanced by using a fast time marching method. In this context, a class of exponential time integration methods has been developed for fast time stepping of three-dimensional, high-order flow simulations governed by the compressible Navier-Stokes equations. Our previous work shows that the predictor-corrector exponential (PCEXP) time scheme[3] implemented in a high-order DG framework HA3D[1,3] can eliminate the Courant-Friedrichs-Lewy (CFL) restriction while maintains low absolutely temporal errors. The PCEXP scheme has been successfully applied to inviscid flows,[2–4] viscous steady flows,[5] and viscous unsteady flows,[6] showing a particular capability of achieving fast time stepping for both steady and unsteady flows.

The basic idea of exponential time discretizations is fundamentally different from traditional explicit and implicit methods, which stems from the matrix exponential expression of the exact solution of ordinary partial differential equations. And such type of schemes owns an inherent feature of having low absolutely temporal errors thanks to their essential connections to the exact matrix-form integral solutions of linear or nonlinear equations.[3] Like fully implicit methods, the exponential methods are of global coupling nature and the flow transport information can propagate through the whole computational domain with large time steps. The price to pay for efficiency, accuracy, and flexibility offered by the exponential methods is their relatively high computational cost per step and storage requirement.

In this work, we exploit the possibility of reducing the computational cost and storage requirement of the exponential methods in a high-order solution adaptation frame. In which cell order of accuracy is

*Research Assistant Professor, Mechanics Division, #10 West Dongbeiwang Road, Haidian District, Beijing 100193, China, E-mail: shujie@csrc.ac.cn. Member AIAA .
†Professor, Department of Mathematics, 1523 Greene Street, Columbia, SC 29208, USA, E-mail: ju@math.sc.edu.
‡Staff member, Mohrenstr. 39, 10117 Berlin, Germany, E-mail: si@wias-berlin.de

dynamically determined with a local cell error indicator in a modal discontinuous Galerkin method, resulting in a cell-wise adaptive distribution of cell orders and reduced computational cost and storage requirement.

The remaining parts of this abstract are organized as follows. Section III presents numerical results obtained by the adaptive exponential DG method, including detailed numerical studies with analytical and experimental results. Finally, the Caradonna-Tung rotor in hover[22] is simulated with the adaptive method for demonstrating its applicability to real-world problems.

II. Predictor-Corrector EXPonential time integrator scheme (PCEXP)

Consider a semi-discretized ordinary differential equations (ODEs) obtained from a high-order spatial discretization such as the discontinuous Galerkin methods

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{R}(\boldsymbol{u}) \tag{1}$$

where $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t), \boldsymbol{x} \in \Omega \subset \mathbb{R}^3$ denotes the solution vector, and $\boldsymbol{R}(\boldsymbol{u})$ the right-hand side term of the spatial discretization used. Without loss of generality, we focus on a single time step $t \in [t_n, t_{n+1}]$. Splitting the $\boldsymbol{R}(\boldsymbol{u})$ into the following form

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{J}_n\boldsymbol{u} + \boldsymbol{N}(\boldsymbol{u}), \tag{2}$$

where $\boldsymbol{J}_n = \mathrm{D}_u\boldsymbol{R}(\boldsymbol{u}_n)$ denotes the Jacobian matrix of $\boldsymbol{R}$, $\boldsymbol{N}(\boldsymbol{u}) = \boldsymbol{R}(\boldsymbol{u}) - \boldsymbol{J}_n\boldsymbol{u}$ is the remainder. The PCEXP scheme is used to solve the equation (2), which is written as

$$\boldsymbol{u}^* = \boldsymbol{u}_n + \Delta t\, \boldsymbol{\Phi}_1(\Delta t \boldsymbol{J}_n)\boldsymbol{R}(\boldsymbol{u}_n). \tag{3}$$

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}^* + \frac{1}{2}\Delta t\, \boldsymbol{\Phi}_1(\Delta t \boldsymbol{J}_n)\left[(\boldsymbol{N}(\boldsymbol{u}^*) - \boldsymbol{N}(\boldsymbol{u}_n)\right]. \tag{4}$$

Where a new function $\phi_k(\Delta t \boldsymbol{J}_n)$ is defined as

$$\boldsymbol{\Phi}_1(\Delta t \boldsymbol{J}_n) := \frac{\boldsymbol{J}_n^{-1}}{\Delta t}\left[\exp\left(\Delta t \boldsymbol{J}_n\right) - \boldsymbol{I}\right], \tag{5}$$

In this scheme, only the current solutions at the t_n is required so that it is a one-step method. The t_n solution is advanced by the first-order formula (3) to obtain the a predicted solution $\boldsymbol{u}^*$, and then the solution $\boldsymbol{u}_{n+1}$ can be obtained with the predicted $\boldsymbol{u}^*$ plus a nonlinear correction term using the frozen Jacobian $\boldsymbol{J}_n$. The scheme only use the one-time Jacobian evaluation which leads to efficient implementations.

A. Adaptation control strategy

In this adaption framework, the solution starts up with the first order of accuracy globally and then is locally refined to a higher order of accuracy in a cell-wise manner, similar to a fully multigrid procedure. The adaptive strategy defines variable DG polynomial order p locally so that higher-order approximations can be placed in key flow regions such as those of near body and wake flows. To identify these regions, a so-called spectral delay error (SDE) indicator[21] is used which was originally designed for locating shock waves. The SDE indicator can also measure how well-resolved the approximation is, thus it can also be used for p-adaptive refinement. For a given solution variable u, a modal DG expansion reads

$$u = \sum_{j=1}^{N(p)} \hat{u}_j \psi_j(\boldsymbol{x}) \tag{6}$$

where the dimension of the polynomial space $N(p)$ is computed as $(p+1)(p+2)(p+3)/6$ for 3D problems. For a given cell polynomial degree p in the $\mathrm{P_p}$ polynomial space, the truncated expansion for a lower degree can be obtained as

$$\bar{u} = \sum_{j=1}^{N(p-1)} \hat{u}_j \psi_j(\boldsymbol{x}) \tag{7}$$

The truncated expansion applying the total energy variable $\rho E = \rho e + \frac{1}{2}\rho v^2$ to the indicator, namely

$$CE = \frac{\int (u - \bar{u})^2 \, \mathrm{d}\boldsymbol{x}}{\int u^2 \, \mathrm{d}\boldsymbol{x}} = \frac{\int (\rho E - \overline{\rho E})^2 \, \mathrm{d}\boldsymbol{x}}{\int (\rho E)^2 \, \mathrm{d}\boldsymbol{x}} \tag{8}$$

The cell error indicator CE (8) is computed in each cell at the end of each level of adaptation. The adaptation process starts with the P_0 approximation globally and then refines hierarchically until the maximum order $\mathrm{P}_{\max}$. We remark that $\mathrm{P}_{\max}$ can be estimated automatically by checking the CE error such that the solution is considered well resolved when the error decreases to a given error tolerance, *e.g.*, 10^{-9}, alternatively it can be supplied as a user-defined parameter. The cell polynomial order p is updated individually with the following criterion

$$p = \begin{cases} 0, & CE \in [10^{-1}, +\infty) \\ p+1, & CE \in [10^{-9}, 10^{-1}) \\ p, & CE \in [0, 10^{-9}) \end{cases} \tag{9}$$

In each order level, the above criterion is applied only when the variation of global maximal Mach number is within 10^{-5} to prevent occurring premature adaptive solutions that are still in the process of dynamic evaluations. As such, the solution process is similar to a cell-wise, fully multigrid evaluation which is also observed in the test cases.

B. Definition of time step

The time step Δt of the adaptive exponential DG is dynamically determined via a residual monitoring strategy (10). In which, the convection time step Δt_{c} and the diffusion time step Δt_{d} are given as

$$\begin{aligned} \Delta t_{\mathrm{c}} &= \frac{\mathrm{CFL}\, h_{3\mathrm{D}}}{(2p+1)\left(\|\boldsymbol{v}\| + c\right)}; \\ \Delta t_{\mathrm{d}} &= \frac{\mathrm{CFL}\, h_{3\mathrm{D}}^2}{(p+1)^2 \left(\frac{2\mu M_\infty}{\rho Re} \max(\frac{4}{3}, \frac{\gamma}{Pr})\right)}. \end{aligned} \tag{10}$$

where $h_{3\mathrm{D}}$ represents a characteristic size of a cell in 3D defined by the ratio of its volume and surface area. To support 2-D computations via the 3-D solver HA3D, a quasi-3D mesh is obtained by extruding the 2-D mesh by one layer of cells, and $h_{2\mathrm{d}}$ has to be computed by eliminating the effect of z dimension for obtaining a truly 2-D time step. Given the cell size Δz in the z direction, $h_{2\mathrm{D}}$ is determined by

$$\frac{2}{h_{2\mathrm{D}}} = \frac{3}{h_{3\mathrm{D}}} - \frac{1}{\Delta z}. \tag{11}$$

The cell-wise time step is defined as $\Delta t = \min\{\Delta t_c, \Delta t_d\}$ and the globally minimal Δt is used instead as the final time step for maintaining better stability.

C. Variable order residual Jacobians

The global residual Jacobian $\boldsymbol{J} = \partial R / \partial u$ of the PCEXP scheme is composed of cell-wise, variable order elemental Jacobians in the current adaptation framework. When different order exists in two adjacent cells, no special interface order coupling[26] is required, since the modal based DG method is compatible with variable accuracy implementation without the need of introducing new interfacial nodes.

The dimension of global residual Jacobian is dynamically composed of elemental residual Jacobians, and the elemental Jacobian has a dimension of $N(p) = (p+1)(p+2)(p+3)/6$. The global Jacobian is obtained analytically for performance consideration.[3,5] We first manually derive the Jacobians as possible as we can until to the hard-bone parts such as the evaluation of the Riemann flux Jacobians and the Bassi-Rebay 2 (BR2) viscous discretization,[25] wherein we use auto-differentiation (AD) tools are used instead to generate error-free Jacobians for accurately incorporating the information of viscous flow propagation. To minimize memory usage, the dynamic array of FORTRAN language is used which permits larger scale computations than those of nonadaptive computations.

III. Numerical results

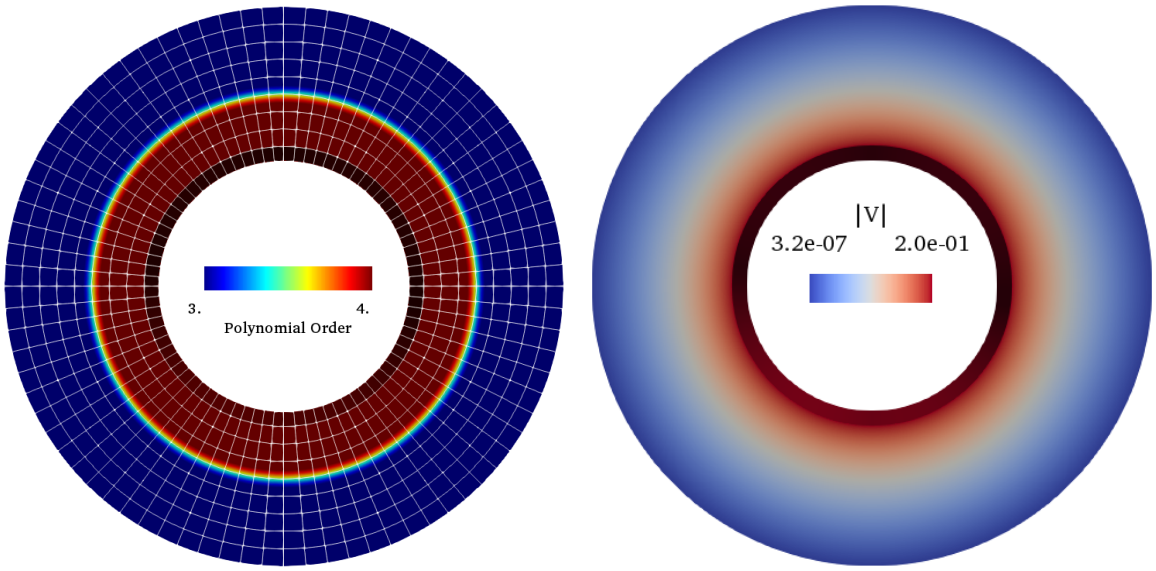
A. Rotating flow between two concentric cylinders

The implementation of adaptive exponential discontinuous Galerkin discretization for the moving boundary Navier-Stokes equations is firstly verified on the rotating flow between two concentric cylinders or Taylor Couette flow. The fluid is driven by two concentric cylinders which are rotating with constant angular velocities of ω_0 and ω_1. A low Reynolds number $Re = 10$ for maintaining a laminar state is used which is defined by the tangential velocity and the radius of the inner cylinder. Constant viscosity is used in this case for allowing comparisons with the analytical solution shown below

$$u_\theta = r_0\omega_0\frac{r_1/r - r/r_1}{r_1/r_0 - r_0/r_1} + r_1\omega_1\frac{r/r_0 - r_0/r}{r_1/r_0 - r_0/r_1} \tag{12}$$

where u_θ is the tangential velocity, $r_0 = 1$ and $r_1 = 2$ are the inner radius and the outer radius. The isothermal boundary condition is set on the inner cylinder with angular velocity ω_0 at Mach number $M_0 = 0.2$. The outer cylinder is stationary with $\omega_1 = 0$ and uses the adiabatic wall boundary condition. The front and the back faces of z-direction use the symmetric boundary condition for conducting quasi-2D computations. Quadratic curved elements are used on the curved boundary surfaces. The convergence order of the viscous flow is computed on a sequence of three meshes, using first- to fifth-order (P_0 – P_4) DG schemes with the HLLC flux for the convection term. The L_2 norms of velocity errors are detailed in Tab. 1. The expected order of convergence is observed for all the order levels, thereby verifying the high-order implementation of DG viscous discretization and also the use of curved elements.

We then use the adaptive ($P_{0\to4}$) accuracy to simulate this case again. The computed results are shown in Fig. 1, where the polynomial order distribution on the mesh ($80\times$ 8 $\times$ 1) is shown in Fig. 1(a) and the velocity magnitude is shown in Fig. 1(b), where the boundary layer is automatically adapted to the highest P_4 accuracy. The L_2 error of the $P_{0\to4}$ solution is also given in the Tab. 1, which is very close to the uniform P_4 solution with only 38.5% of the total 640 elements solved in P_4 accuracy. The results show good accuracies when employing the uniform and adaptive exponential DG methods.



(a) (b)

Figure 1. Rotating flow between two concentric cylinders: (a) adaptive distribution of polynomial order with mesh 80×8. (b) velocity magnitude contour of the $P_{0\to4}$ adaptive solution.

Table 1. Rotating flow between two concentric cylinders: uniform and adaptive order results of the L_2 error expressed in the $\log_{10}$ scale. #P_4(%) denotes the percentage of the number of P_4 cells of all the cells obtained with the adaptive method using cellwise polynomial refinement from P_0 to P_4 order ($P_{0\to4}$).

Ncell	20×2	40×4	80×8	order
P_0	-7.962E-01	-1.109E+00	-1.435E+00	**1.07**
P_1	-1.828E+00	-2.378E+00	-2.880E+00	**1.75**
P_2	-2.394E+00	-3.459E+00	-4.418E+00	**3.36**
P_3	-3.292E+00	-4.542E+00	-5.752E+00	**4.09**
P_4	-4.112E+00	-5.559E+00	-6.941E+00	**4.70**
$P_{0\to4}$	-4.112E+00	-5.559E+00	-6.066E+00	n/a
#P_4(%)	100%	100%	38.5%	n/a

B. Lid-driven cavity flow

The strategy of the adaptive exponential framework is evaluated on the lid-driven cavity flow. Quasi-3D adaptive solutions are assessed with the baseline results of Ghia[20] at a high Reynolds number $Re = 10^4$. In this computational condition, secondary vortices show up in the corners of the cavity so that high-resolution simulations are required. Instead of using a fine mesh to increase resolution, a very coarse mesh (20×20) is used along with the high-order adaptive framework.

We start by studying the impacts of discrete accuracy on the flow structures. The top-up order of accuracy from P_1 to P_4 is used and the streamline plots are given in Fig. 2. From this, we can see that higher-order adaptations help capture secondary vortices in the corners. The $P_{0\to1}$ adaptation is too diffusive to capture the corner vortices, even if the adaptive procedure of $P_{0\to1}$ leads to a 100% P_1 cell fill-in. The situation is improved when employing uniform P_4 approximation which recovers all the corner vortices and shows an increased resolution of flow structures, as shown in Fig. 3 (left). Interestingly, when employing adaptation, the polynomial refinement occurs primarily in the boundary layer and strong shear regions, where the energy cascade transfers energy from large scale to small scale flow structures, as shown in Fig. 3 (right). Notice that the $P_{0\to4}$ adaptive solution (Fig. 2 bottom right) is essentially similar to the uniform P_4 solution (Fig. 3 right). Convergence histories are given in Fig. 4 in which the adaptive P_{0-4} converges even faster than the uniform P_4. To compare the results, Fig. 5 gives horizontal (u) and vertical (v) velocity profiles along the central lines of x and y axis. The $P_{0\to4}$ and P_4 solutions are in good agreements with the reference solutions of Ghia,[20] while the low-order, $P_{0\to1}$ solution fails to match the velocity profiles. Tab. 2 lists the computational cost and memory requirement of the cases. From which, we find that the adaptation procedures offer significant cost reduction in terms of the total degree of freedoms (DOFs) and CPU time. The $P_{0\to4}$ solution offers a two-fold reduction in DOFs and a three-fold speedup compared with the uniform P_4 solution. The memory usage of the global Jacobian matrix is also given, the $P_{0\to4}$ saves half of the memory of P_4. This case shows that the adaptive exponential frame is effective and accurate for viscous flows.

C. Caradonna-Tung rotor in hover

A real-world problem is considered in this case corresponding to the experimental model hover test conditions of Caradonna and Tung.[22] The experimental model consists of a two-bladed rigid rotor with rectangular planform blades with no twist or taper. The blades are made of NACA0012 airfoil sections with an aspect ratio of 6. The computational condition uses the case of tip Mach number $M_{tip} = 0.4395$, collective pitch $\theta = 8°$, and the Reynolds number is based on the blade tip speed and chord, $Re = 1.92 \times 10^6$. Other parameters are given in Tab. 3. The $P_{0\to2}$ adaptive solution is computed on a coarse mesh of 96,222 curved elements. Fig. 6 (left) shows the polynomial order distribution. Notice that the higher-order cells are all placed in key flow regions such as the ones of near body, vortex trajectory, and wakes. The computed vortex contour is also displayed in the right picture of Fig. 6. Detailed comparisons to experimental data are given in Fig. 7, where sectional pressure coefficients and vortex position are all in fair agreements with the experimental data, demonstrating the feasibility of applications to practical problems.

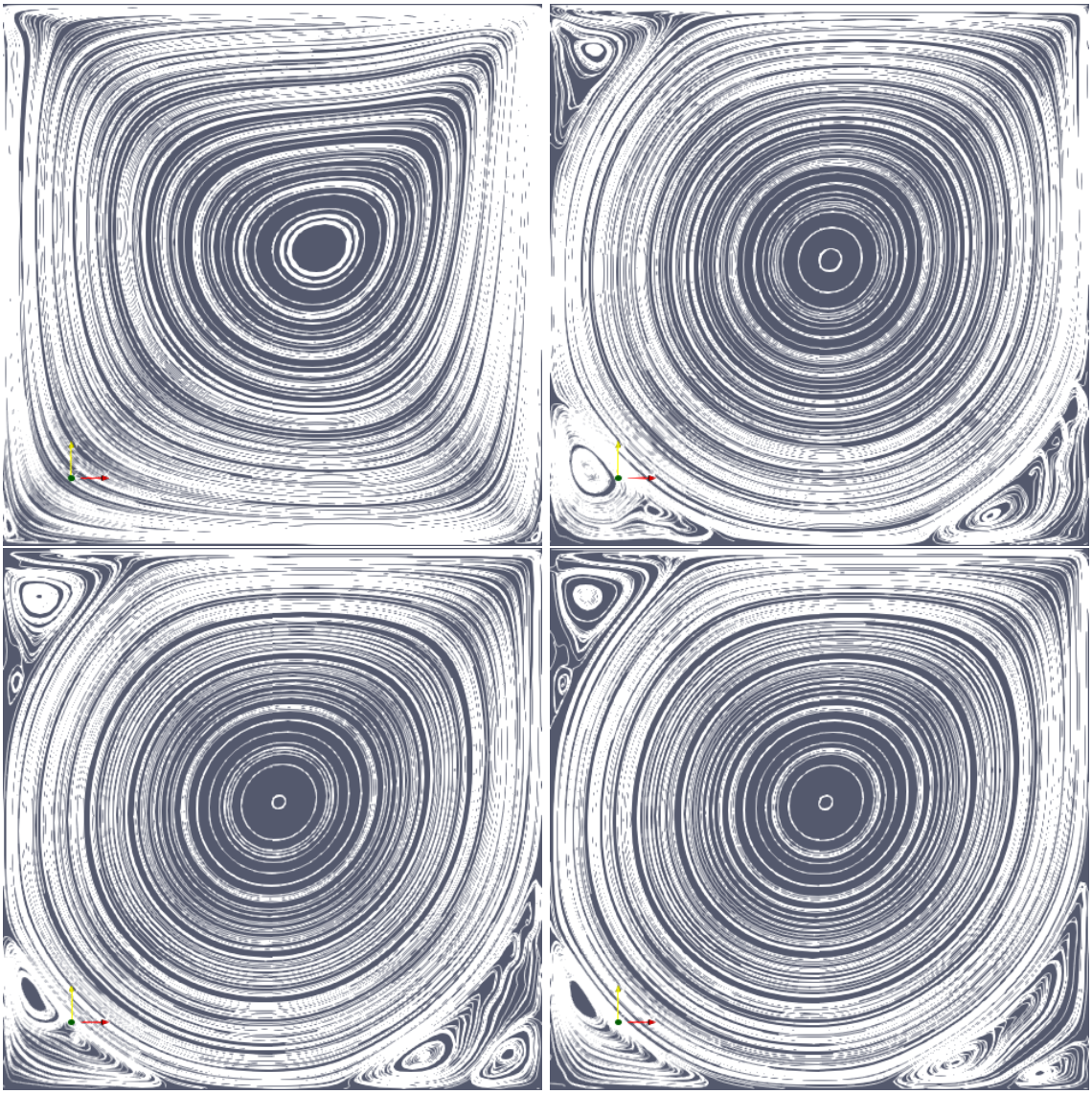


Figure 2. Lid-driven cavity flow at $Re = 10^4$: streamline plots with $P_{0\to1}$ to $P_{0\to4}$ adaptations (from left to right, from top to bottom). The secondary vortices in the corners gradually become apparent and smooth.

Table 2. Results statistics of the lid-driven cavity flow at $Re = 10^4$. The CPU time is normalized by the one of $P_{0\to4}$. $\#P_k$ denotes the number of the k-order cells in the percentage of total cells. Nearly three-fold speedups are gained when comparing to the $P_{0\to4}$ and P_4 results. The memory usage of the global Jacobian is expressed in Megabytes (M).

	$\#P_0$	$\#P_1$	$\#P_2$	$\#P_3$	$\#P_4$	DOFs	CPU time	Storage (M)
$P_{0\to4}$	0	0	56.25%	27.50%	16.25%	6,581	1.00	0.52
$P_{0\to3}$	0	0	63.00%	37.00%	-	5,480	0.46	0.45
$P_{0\to2}$	0	2.75%	97.25%	-	-	3,934	0.17	0.36
$P_{0\to1}$	0	100%	-	-	-	1,600	0.02	0.05
P_4 uniform	-	-	-	-	100%	14,000	2.70	0.92

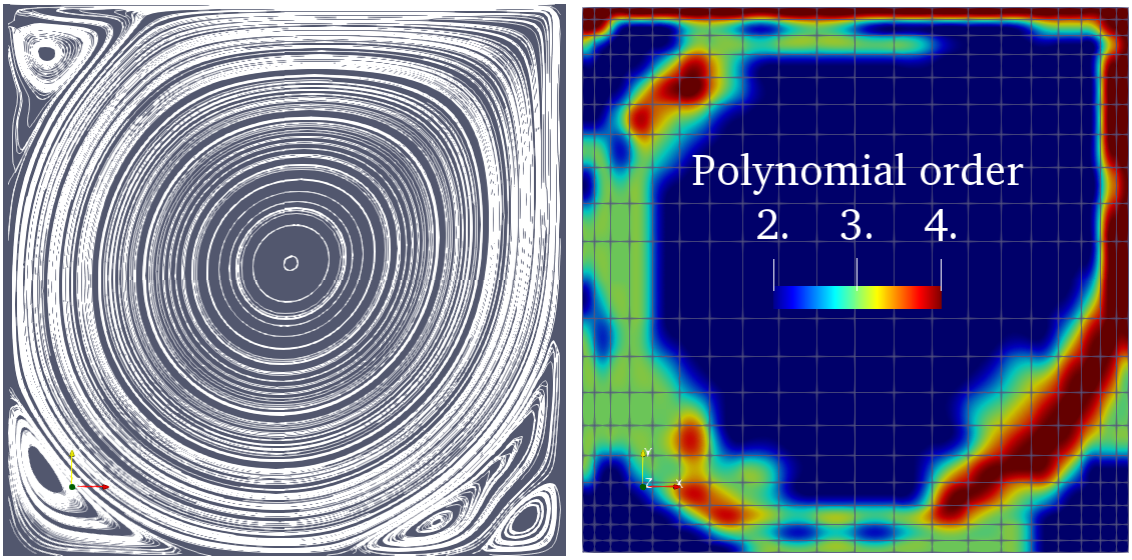


Figure 3. Lid-driven cavity flow at $Re = 10^4$: reference uniform P_4 solution (left); order distribution, $\mathrm{P}_{0\to4}$ (right). Notice that only 65 cells marked with red color are solved in P_4 accuracy

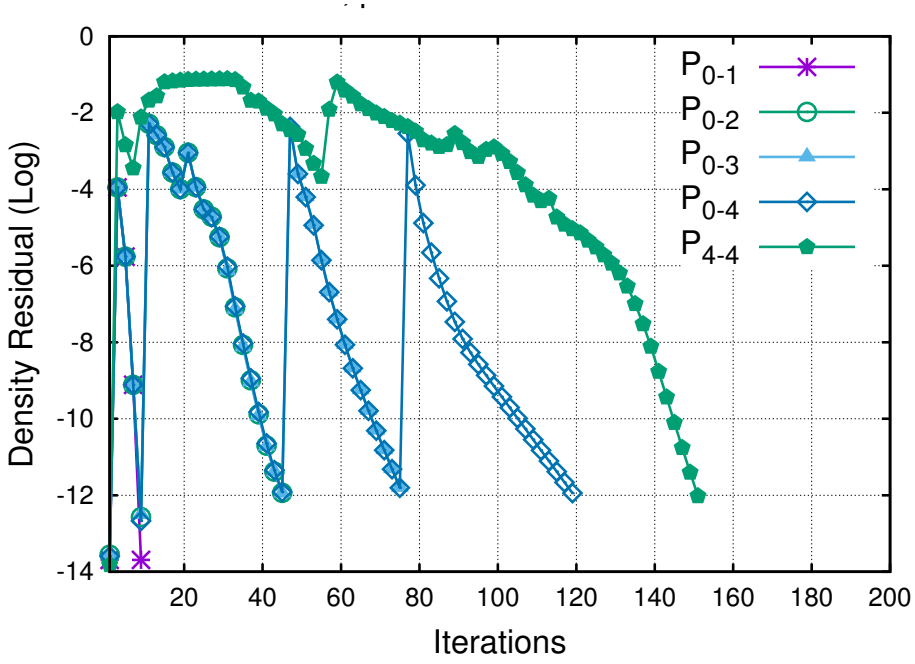


Figure 4. Lid-driven cavity flow at $Re = 10^4$: convergence comparison. Notice that the convergences are similar to full multigrid methods due to the hierarchically adaptive refinement of the cell order. The P_{0-4} solution converges faster than the uniform P_4 solution.

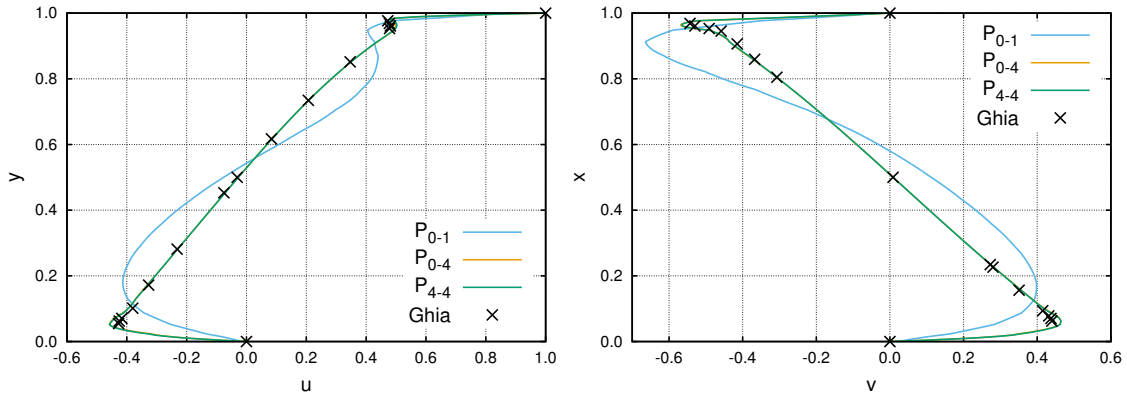


Figure 5. Lid-driven cavity flow at $Re = 10^4$: comparison of the velocity profiles. u-velocity profile along the line $x = 0.5$ (left); v-velocity profile along the line $y = 0.5$ (right).

Table 3. Computational conditions of the Caradonna-Tung rotor in hover.

Parameter	Value
Rotor diameter D	2.286 m
Angular velocity, Ω	1250 rpm
Tip Mach number, M_{tip}	0.439
Reynolds number, Re	1.92×10^6
Blade chord length, c	0.1905
Blade aspect ration, AR	6.0
Blade twist, θ_t	0°
Blade collective pitch, θ_c	8°
Blade profile	NACA0012

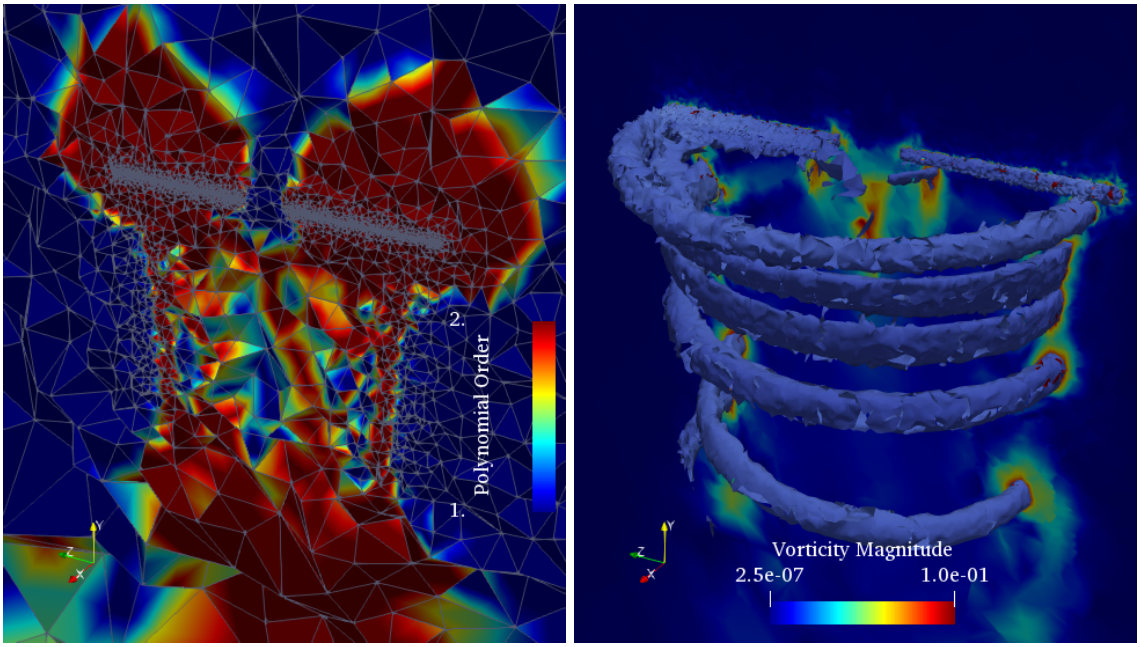


Figure 6. Caradonna-Tung rotor in hover: adaptive polynomial order (left); vorticity contour (right).

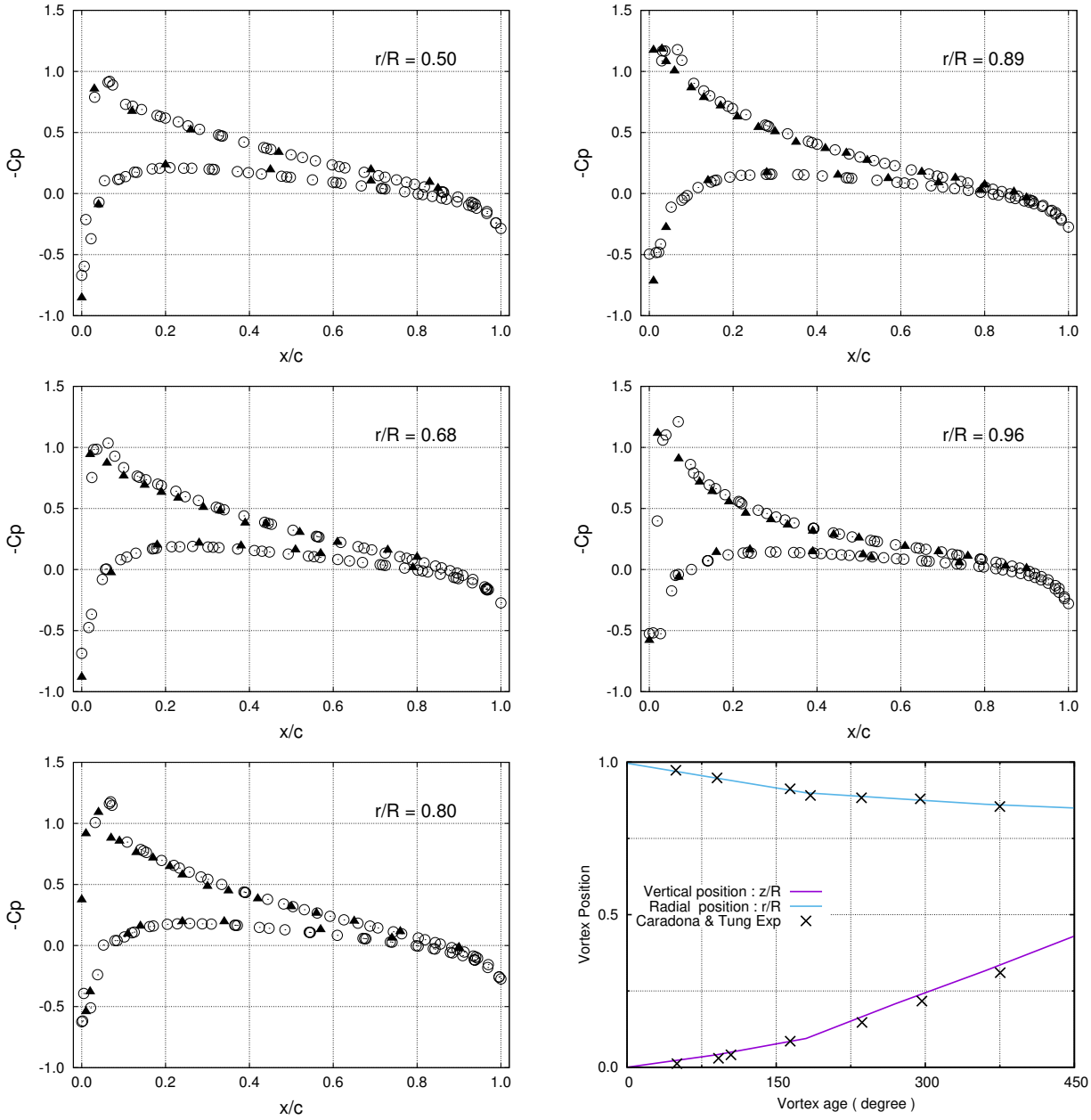


Figure 7. Caradonna-Tung rotor in hover: pressure coefficients computed at $M_{tip} = 0.4395$, $\theta = 8^{\circ}$ and $Re = 1.92 \times 10^6$ ($\odot$ 96,222 cells) compared with the experimental data (▲) at different sectional locations r/R. The bottom right figure shows the trajectory of vortex core compared with the experimental data.

IV. Conclusion

An adaptive high-order DG framework has been developed with the PCEXP exponential time marching scheme. By using the adaptive residual Jacobian, the performance of the PCEXP scheme is shown to be enhanced in terms of computational cost and memory usage. The correctness of the variable order implementation of PCEXP is firstly validated in the case of rotating flow between two concentric cylinders. While uniform order solver can deliver formal convergence rates, the adaptive P_{0-4} solution can even give a comparable error to the uniform P_4 solution. In the cavity flow case, it is found that the use of spectral decay error indicator is effective for capturing key flow structures such that the secondary corner vortices can be captured. Performance statistic shows that a three-fold reduction in CPU time and a two-fold reduction in memory usage are gained. Finally, the adaptive solver is applied to the Caradonna-Tung rotor in hover, where 3D wake flows are adaptively captured with the coarse mesh. The results are in good agreement to the experimental data[22] which demonstrates its applicability to practical 3D complex flows.

Acknowledgments

SJL acknowledge the financial supports from the National Natural Science Foundation of China (NSFC) under the Grant U1930402 and Beijing Computational Science Research Center. LLJ acknowledge the support from the US National Science Foundation under the Grant DMS-1818438 and the US Department of Energy under the Grant DE-SC0020270.

References

[1]Li, S.-J., A Parallel Discontinuous Galerkin method with Physical Orthogonal Basis on Curved Elements, Procedia Engineering, 61(2013):144–151.

[2]Li, S.-J., Wang, Z.J., Ju, L., and Luo, L.-S., Explicit Large Time Stepping with A Second-Order Exponential Time Integrator Scheme for Unsteady and Steady Flows, AIAA paper, 2017–0753.

[3]Li, S.-J., Luo, L.-S., Wang, Z.J., and Ju, L, An exponential time-integrator scheme for steady and unsteady inviscid flows, J. Comput. Phys., 365 (2018): 206–225.

[4]Li, S.-J, Efficient p-multigrid method based on an exponential time discretization for compressible steady flows, arXiv:1807.0115.

[5]Li, S.-J., Wang, Z.J., Ju, L., and Luo, L.-S., Fast time integration of Navier-Stokes equations with an exponential-integrator scheme, AIAA paper, 2018–0369.

[6]Li, S.-J., Ju, L., Exponential time-marching method for the unsteady Navier-Stokes equations, AIAA paper 20190907.

[7]Cox, S.M., Matthews, P.C., "Exponential time differencing for stiff systems", J. Comput. Phys., 176 (2002): 430–455.

[8]Ju, L., Zhu, L.Y., Zhang, J., Du, Q., "Fast explicit integration factor methods for semilinear parabolic equations", J. Sci. Comput., 62 (2015): 431–455.

[9]Hochbruck, M. Lubich, C., Selhofer, H., "Exponential integrators for large systems of differential equations", SIAM J. Sci. Comput., 19 (1998): 1552–1574.

[10]Hochbruck, M. Ostermann, A., Schwitzer, J., "Exponential Rosenbrock-type methods", SIAM J. Numer. Anal., 47 (2009): 786–803.

[11]Hochbruck, M., Lubich, C., "On Krylov subspace approximations to the matrix exponential operator", SIAM J. Numer. Anal., 34 (1997) :1911–1925.

[12]Ostermann, A., Thalhammer, M., Wright, W.M., "A class of explicit exponential general linear methods", BIT Numer. Math., 46 (2) (2006): 409–431.

[13]Caliari, M., Ostermann, A., "Implementation of exponential Rosenbrock-type integrators", Appl. Numer. Math., 59 (2009): 568–582.

[14]Loffeld, J., Tokman, M., "Comparative performance of exponential, implicit, and explicit integrators for stiff systems of ODEs", J. Comput. Appl. Math., 241 (2013): 45–67.

[15]Tokman, M., "Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods", J. Comput. Phys., 213 (2006): 748–776.

[16]Tokman, M., "A new class of exponential propagation iterative methods of Runge-Kutta type (EPIRK)", J. Comput. Phys., 230 (24) (2011): 8762–8778.

[17]Tokman, M., Loffeld, J., "Efficient design of exponential-Krylov integrators for large scale computing", Procedia Comput. Sci., 1 (2010): 229–237.

[18]Saad, Y., "Analysis of some Krylov subspace approximations to the matrix exponential operator", SIAM J. Numer., Anal., 29 (1992): 209–228.

[19]Moler, C.B., van Loan, C.F., "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later", SIAM Rev., 45 (2003): 3–49.

[20]Ghia, U., Ghia, K. N., Shin, C. T., "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method", J. Comput. Phys., 48 (1982) 387–411.

[21]Persson, P.-O. , Peraire, J., Sub-Cell shock capturing for discontinuous Galerkin methods. AIAA paper, 2006–112.

[22]Caradonna, F. X. and Tung, C. Experimental and analytical studies of a model helicopter rotor in hover. NASA Technical Memorandum 1981-81232, NASA, Ames Research Center, Moffett Field, California, September 1981.

[23]Taneda S., "Experimental investigation of the wake behind a sphere at low Reynolds numbers". Journal of the Physical Society of Japan, 11(10)(1956): 1104–1108.

[24]Roe. P.L.,"Approximate Riemann solvers, parameter vectors, and difference schemes",J. Comput. Phys., 43(2) (1981) 357–372.

[25]Bassi, F., Rebay, S., "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations",. J. Comput. Phys., 131(2) (1997) 267–279.

[26]Cagnone, J.S., et al. A p-adaptive LCP formulation for the compressible NavierStokes equations. J. Comput. Phys., 233(2013) 324–338.