HPS Accelerated Spectral Solvers for Time Dependent Problems: Part I, Algorithms

Tracy Babb, Per-Gunnar Martinsson, and Daniel Appelö

1 Introduction

In this chapter, part two in a two part series, describes a sequence of numerical experiments demonstrating the performance of a highly computationally efficient solver for equations of the form

$$\kappa \frac{\partial u}{\partial t} = \mathscr{L}u(\boldsymbol{x}, t) + g(u, \boldsymbol{x}, t), \quad \boldsymbol{x} \in \Omega, t > 0, \tag{1}$$

with initial data $u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x})$. Here $\mathscr{L}$ is an elliptic operator acting on a fixed domain Ω and f is lower order, possibly nonlinear terms. We take κ to be real or imaginary, allowing for parabolic and Schrödinger type equations.

The "Hierarchial Poincaré–Steklov (HPS)" solver has already been demonstrated to be a highly competitive spectrally accurate solver for elliptic problems [1, 4, 7] and has also been used together with a class of exponential integrators [5], to evolve solutions to hyperbolic differential equations. As just mentioned, the focus here is on differential equations in the form (1) whose discretization leads to stiff system of ODE that can beneficially be advanced in time using Explicit, Singly Diagonally Implicit Runge–Kutta (ESDIRK) methods. ESDIRK methods offer the advantages of stiff accuracy and L-stability and are well suited for the HPS algorithm as they only require a single matrix factorization. They are also easily combined with

T. Babb · D. Appelö (✉)
University of Colorado, Boulder, CO, USA
e-mail: tracy.babb@colorado.edu; daniel.appelo@colorado.edu

P.-G. Martinsson
University of Texas, Austin, TX, USA
e-mail: pgm@ices.utexas.edu

S. J. Sherwin et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*, Lecture Notes in Computational Science and Engineering 134, https://doi.org/10.1007/978-3-030-39647-3_11

explicit Runge–Kutta method leading to so called Additive Runge–Kutta (ARK) methods [6].

To this end we investigate the stability and accuracy that is obtained when combining high-order time-stepping schemes with the HPS method for solving elliptic equations. We restrict attention to relatively simple geometries (rectangles) but note that the method can without difficulty be generalized to domains that can be expressed as a union of rectangles, possibly mapped via curvilinear smooth parameter maps.

The rest of this chapter is organized as follows. In Sect. 2 we present results illustrating that the order reduction phenomena for DIRK methods observed in [8] can be circumvented when formulating the time stepping in terms of slopes (with boundary conditions differentiated in time) rather than formulating it in terms of stage solutions. In Sect. 3 we present numerical results for Schrödingers equation in two dimensions and in Sect. 4 we present numerical results for a nonlinear problem, viscous Burgers' equation in two dimensions. Finally, in Sect. 5 we summarize and conclude. For a longer description of the method we refer to thee first part of this paper and to [2].

2 Time Dependent Boundary Conditions

This section discusses time-dependent boundary conditions within the two different Runge–Kutta formulations. In particular, we investigate the order reduction that has been documented in [8] for implicit Runge–Kutta methods and earlier in [3] for explicit Runge–Kutta methods.

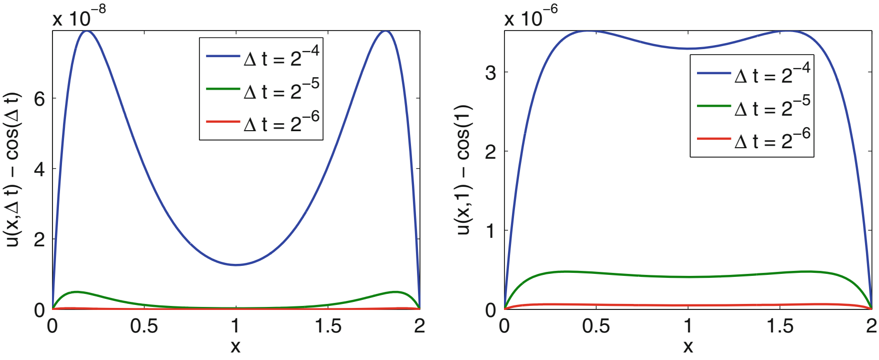
In this first experiment, introduced in [8], we solve the heat equation in one dimension

$$u_t = u_{xx} + f(t), \qquad x \in [0, 2], \quad t > 0. \tag{2}$$

We set the initial data, Dirichlet boundary conditions and the forcing $f(t)$ so that exact solution is $u(x, t) = \cos(t)$. This example is designed to eliminate the effect of the spatial discretization, with the solution being constant in space and allows for the study of possible order reduction near the boundaries.

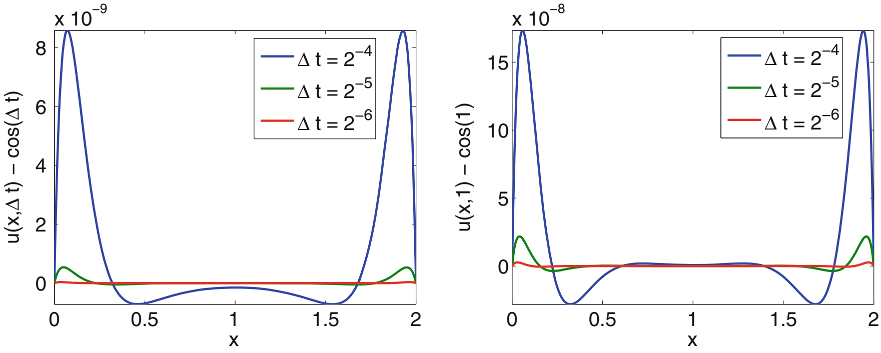
We use the HPS scheme in space and use 32 leafs with $p = 32$ Chebyshev nodes per leaf. We apply the third, fourth, and fifth order ESDIRK methods from [6]. We consider solving for the intermediate solutions, or as we refer to it below "the u_i formulation" with the boundary condition enforced as $u_i^n = \cos(t_n + c_i \Delta t)$. We also consider solving for the stages, which we refer to as "the k_i formulation" with boundary conditions imposed as $k_i^n = -\sin(x, t_n + c_i \Delta t)$.

Error reduction for time dependent boundary conditions has been studied both in the context of explicit Runge–Kutta methods in e.g. [3] and more recently for implicit Runge–Kutta methods in [8]. In [8] the authors report observed orders of accuracy equal to two (for the solution u) for DIRK methods of order 2, 3, and 4 for



(a) (b)

Fig. 1 The error in solving (2). Results are for a third order ESDIRK. (**a**) Displays the single step error which converges with fourth order of accuracy. (**b**) Displays the global error at $t = 1$ converging at third order. Both errors converge at one order higher than what is expected from the analysis in [8]



(a) (b)

Fig. 2 The error in solving (2). Results are for a fifth order ESDIRK. (**a**) Displays the single step error which converges with fourth order of accuracy. (**b**) Displays the global error at $t = 1$ converging at third order. Both errors converge at one order higher than what is expected from the analysis in [8] but still lower than expected

the problem (2) discretized with a finite difference method on a fine grid (the spatial errors are zero) using the u_i formulation.

Figures 1 and 2 show the error for the third and fifth order ESDIRK methods, respectively, as a function of x for a single step and at the final time $t = 1$. Figure 3 shows the maximum error for the third, fourth, and fifth order methods as a function of time step Δt after a single step and at the final time $t = 1$.

In general, for a method of order p we expect that the single step error decreases as Δt^{p+1} while the global error decreases as Δt^{p}. However, with time dependent

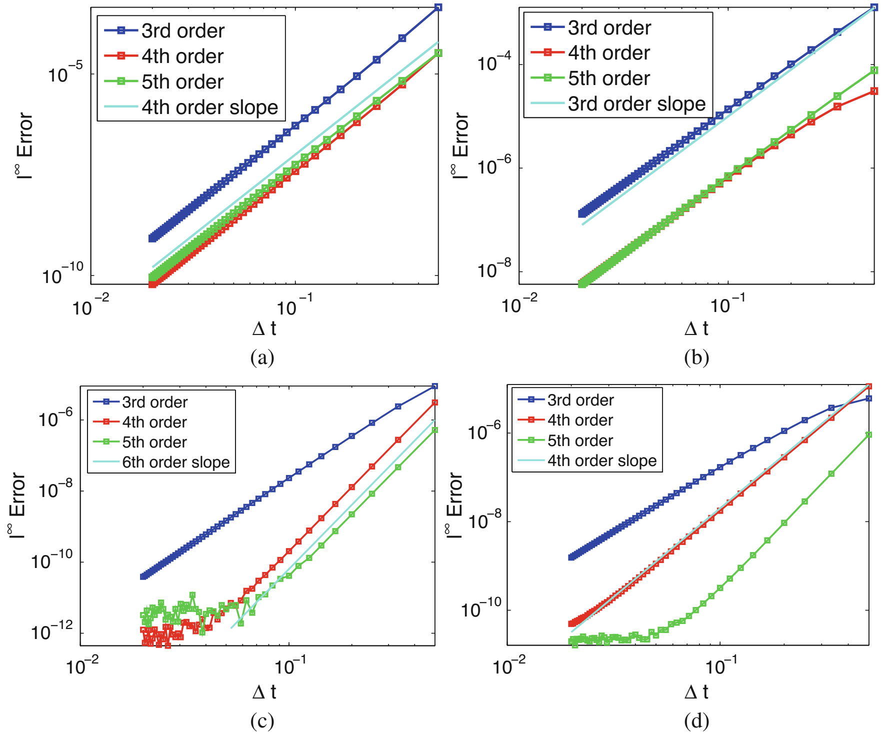


Fig. 3 The maximum error (here denoted l^∞) in solving (2) for the third, fourth, and fifth order ESDIRK methods for a sequence of decreasing time steps. (**a**, **c**) are errors after one time step and (**b**, **d**) are the errors at time $t = 1$. The top row are for the u_i formulation and the bottom row is for the k_i formulation. Note that the k_i formulation is free of order reduction

boundary conditions implemented as $u_i^n = \cos(t_n + c_i \Delta t)$ the results in [8] indicate that the rate of convergence will not exceed two for the single step or global error.

The results for the third order method ($p = 3$) displayed in Fig. 1 show that the single step error decreases as Δt^{p+1} while the global error decreases as Δt^p, which is better than the results documented in [8]. However, we still see that a boundary layer appears to be forming, but it is of the same order as the error away from the boundary. The results for the fifth order method ($p = 5$) displayed in Fig. 2 show that the single step error decreases as Δt^4 while the global error decreases as Δt^3, which is still better than the results documented in [8]. However, the boundary layer is giving order reduction from Δt^{p+1} for the single step error and Δt^p for the global error. We note that our observations differ from those in [8] but that this possibly can be attributed to the use of a ESDIRK method rather than a DIRK method.

We repeat the experiment but now we use the k_i formulation for Runge–Kutta methods and for the boundary condition we enforce $k_i^n = -\sin(t_n + c_i \Delta t)$. The intuition here is that k_i^n is an approximation to u_t at time $t_n + c_i \Delta t$ and we use the value of u_t for the boundary condition of k_i^n. Intuitively we expect that the fact that we reduce the index of the system of differential algebraic equation in the u_i formulation by differentiating the boundary conditions can restore the design order of accuracy.

In the previous examples the Runge–Kutta method introduced an error on the interior while the solution on the boundary was exact. If the error on the boundary is on the same order of magnitude as the error on the interior then the error in u_{xx} is of the correct order, but when the value of u is exact on the boundary it introduces a larger error in u_{xx}. In the k_i formulation, for each intermediate stage we find $u_{xx} = 0$ and then $k_i^n = -\sin(t_n + c_i \Delta t)$ on the interior and on the boundary. So at a fixed time the solution is constant in x and a boundary layer does not form. Additionally, the error is constant in x at any fixed time and for a method of order p we obtain the expected behavior where the single step error decreases as Δt^{p+1} and the global error decreases as Δt^p.

Figure 3 shows the maximum error for the third, fourth, and fifth order methods as a function of time step Δt after a single step and at the final time $t = 1$. The results show that the methods behave exactly as we expect. The single step error behaves as Δt^{p+1} for the third and fifth order methods and Δt^{p+2} for the fourth order method. The fourth order method gives sixth order error in a single step because the exact solution is $u(x, t) = \cos(t)$, which has every other derivative equal to zero at $t = 0$ and for a single step we start at $t = 0$. The global error behaves as Δt^p for each method.

3 Schrödinger Equation

Next we consider the Schrödinger equation for $u = u(x, y, t)$

$$\begin{aligned} i\hbar u_t &= -\frac{\hbar^2}{2M}\Delta u + V(x, y)u, \quad t > 0, \quad (x, y) \in [x_l, x_r] \times [y_b, y_t], \\ u(x, y, 0) &= u_0(x, y). \end{aligned} \tag{3}$$

Here we nondimensionalize in a way equivalent to setting $M = 1, \hbar = 1$ in the above equation. We choose the potential to be the harmonic potential

$$V(x, y) = \frac{1}{2}\left(x^2 + y^2\right).$$

This leads to an exact solution

$$u(x, y, t) = Ae^{-it}e^{-\frac{(x^2+y^2)}{2}}, \tag{4}$$

where we set $A = 1/\sqrt{\sqrt{\pi}}$ and solve until $t = 2\pi$ on the domain $(x, y) \in [-8, 8]^2$.

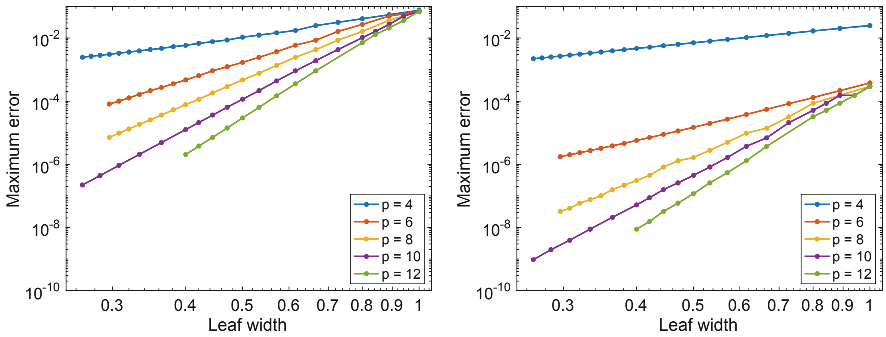


Fig. 4 Error in the Schrödinger equation as a function of leaf size. The exact solution is given in Eq. (4)

Table 1 Estimated rates of convergence for different Runge–Kutta methods and different orders of approximation

p	4	6	8	10	12
ESDIRK3	2.59	5.73	7.72	9.69	11.47
ESDIRK4	1.89	6.47	7.82	9.76	11.69
ESDIRK5	1.84	4.42	7.69	9.71	11.48

The computational domain is subdivided into $n_x \times n_y$ panels with $p \times p$ points on each panel. To begin, we study the order of accuracy with respect to leaf size. To eliminate the effect of time-stepping errors we scale $\Delta t = h^{p/q_{\text{RK}}}$, where q_{RK} is the order of the Runge–Kutta method. In Fig. 4 we display the errors as a function of the leaf size for $p = 4, 6, 8, 10, 12, 16$ and for the third and fifth order Runge–Kutta methods ($q_{RK} = 3, 5$). The rates of convergence are found for all three Runge–Kutta methods and summarized in Table 1. As can be seen from the table, $p = 4$ appears to converge at second order, while for higher p we generally observe a rate of convergence approaching to p.

In this problem the efficiency of the method is limited by the order of the Runge–Kutta methods. However, as our methods are unconditionally stable we may enhance the efficiency by using Richardson extrapolation to achieve a highly accurate solution in time. We solve the same problem, but now we fix $p = 12$ and take $5 \cdot 2^n$ time steps, with $n = 0, 1, \ldots, 5$. For the third order ESDIRK method we use 60×60 leaf boxes. For the fourth order ESDIRK method we use 90×90 leaf boxes. For the fifth order ESDIRK method we use 120×120 leaf boxes. Table 2 shows that we can easily achieve much higher accuracy by using Richardson extrapolation.

Finally, we solve a problem without an analytic solution. In this problem the initial data

$$u(x, y, t) = 3\sin(x)\sin(y)e^{-(x^2+y^2)},$$

Table 2 Estimated errors at the final time after Richardson extrapolation

q_{RK}/ extrapolations	0	1	2	3	4	5	6
3	1.32 (−1)	1.01 (−2)	1.27 (−4)	1.17 (−5)	6.98 (−8)	8.62 (−10)	7.40 (−6)
4	2.70 (−4)	6.46 (−6)	1.23 (−7)	2.95 (−10)	1.59 (−11)	3.70 (−14)	1.20 (−11)
5	1.28 (−3)	9.67 (−6)	6.30 (−8)	1.86 (−10)	4.11 (−13)	9.27 (−14)	5.08 (−11)

The notation $d(-p)$ means $d \cdot 10^{-p}$

Table 3 Errors computed against a p and h refined solution

p/panels	2	4	8	16	32
8	1.11 (0)	1.39 (−1)	8.74 (−3)	1.50 (−4)	2.45 (−6)
Rate	*	3.00	3.99	5.87	5.92
10	5.87 (−1)	3.16 (−2)	4.62 (−4)	6.17 (−6)	5.21 (−8)
Rate	*	4.21	6.10	6.22	6.89

The errors are maximum errors at the final time $t = 4$. The notation $d(-p)$ means $d \cdot 10^{-p}$

interacts with the weak and slightly non-symmetric potential

$$V(x, y) = 1 - e^{-(x+0.9y)^4},$$

allowing the solution to reach the boundary where we impose homogenous Dirichlet conditions.

We evolve the solution until time $t = 4$ using $p = 8$ and 10 and 2, 4, 8, 16 and 32 leaf boxes in each direction of a domain of size 12×12. The errors computed against a reference solution with $p = 12$ and with 32 leaf boxes can be found in Table 3.

In Fig. 5 we display snapshots of the magnitude of the solution at the initial time $t = 0$, the intermediate times $t \approx 1.07$, $t \approx 1.68$ and at the final time $t = 4.0$.

4 Burgers' Equation in Two Dimensions

As a first step towards a full blown flow solver we solve Burgers' equation in two dimensions using the additive Runge–Kutta methods described in the first part of this paper. Precisely, we solve the system

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = \varepsilon \Delta \mathbf{u}, \quad \mathbf{x} \in [-\pi, \pi]^2, \;\; t > 0, \tag{5}$$

where $\mathbf{u} = [u(x, y, t), v(x, y, t)]^T$ is the vector containing the velocities in the x and y directions.

The first problem we solve uses the initial condition $\mathbf{u} = 5[-y, x]^T \exp(-3r^2)$ and the boundary conditions are taken to be no-slip boundary conditions on all sides.

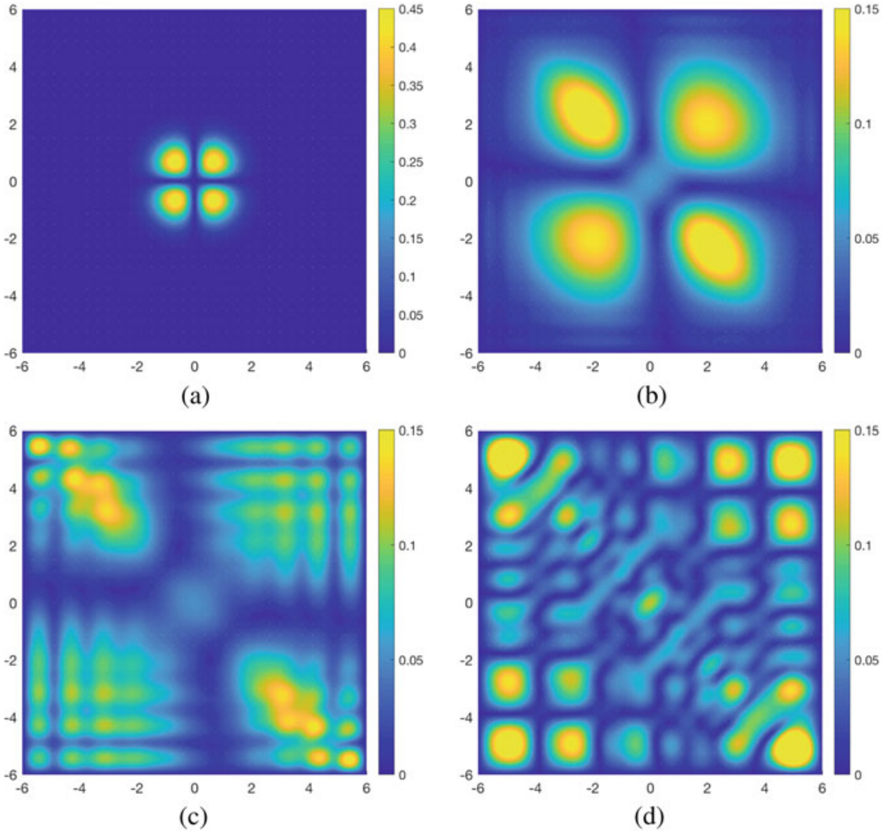


Fig. 5 Snapshots of the magnitude of the solution at the initial time (**a**) $t = 0$, the intermediate times (**b**) $t \approx 1.07$, (**c**) $t \approx 1.68$ and at the final time (**d**) $t = 4.0$

We solve the problem using 24×24 leafs, $p = 24$, $\varepsilon = 0.005$, and the fifth order ARK method found in [6]. We use a time step of $k = 1/80$ and solve until time $t_{\max} = 5$. The low viscosity combined with the initial condition produces a rotating flow resembling a vortex that steepens up over time.

In Fig. 6 we can see the velocities at times $t = 0.5$ and $t = 1$. The fluid rotates and expands out and eventually forms a shock like transition. This creates a sharp flow region with large gradients resulting in a flow that may be difficult to resolve with a low order accurate method. These sharp gradients can be seen in the two vorticity plots in Fig. 6 along with the speed and vorticity plots in Fig. 7.

In our second experiment we consider a cross stream of orthogonal flows. We use an initial condition of

$$\mathbf{u} = [8y\,e^{-36\left(\frac{y}{2}\right)^8}, -8xe^{-36\left(\frac{x}{2}\right)^8}]^T, \tag{6}$$

and time independent boundary conditions that are compatible with the initial data.

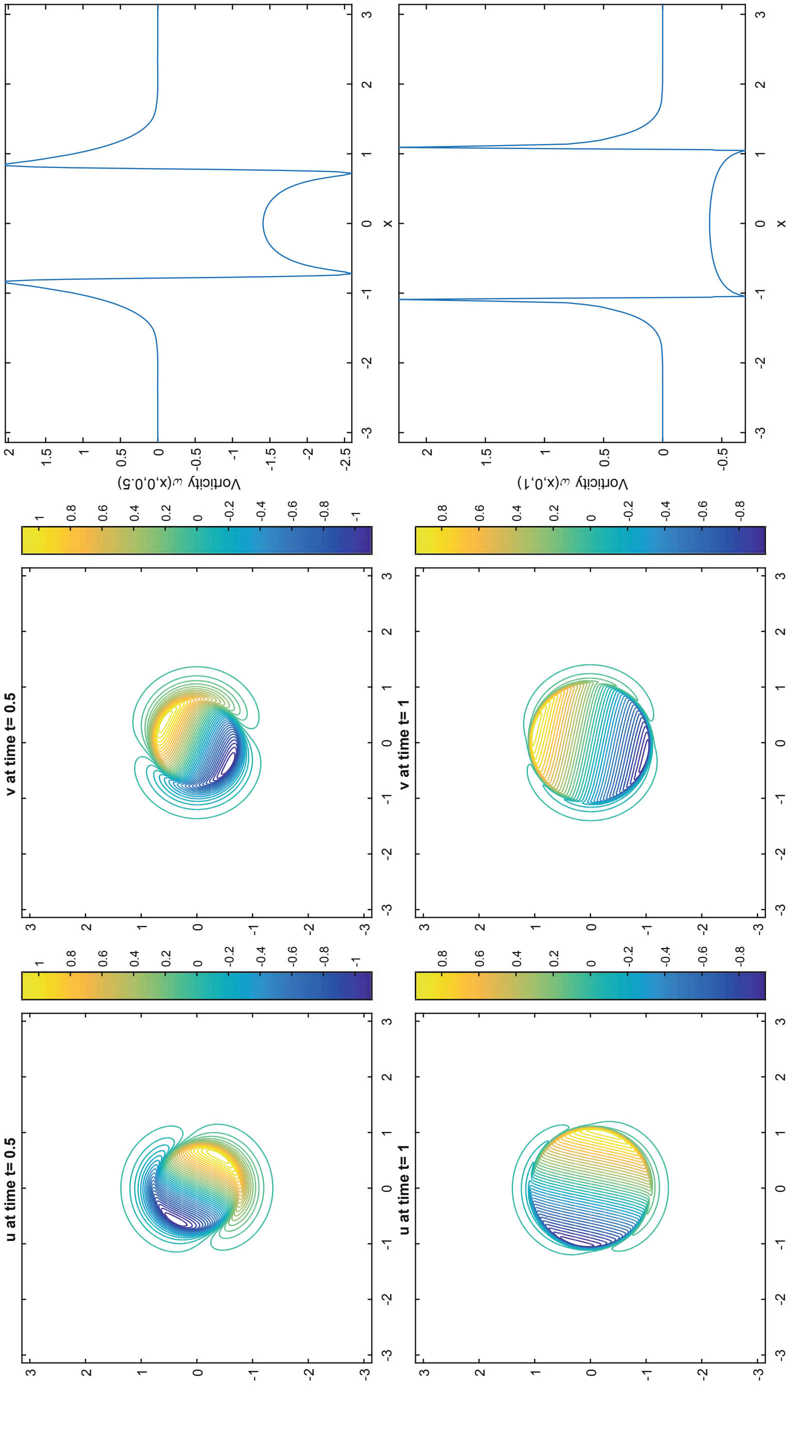


Fig. 6 The four plots on the left show the velocities in the x and y directions at times $t = 0.5$ and $t = 1$. We can see the fluid rotating and expanding. The two plots in the third column show the vorticity at these times. We see sharp gradients near the edge of the rotating fluid

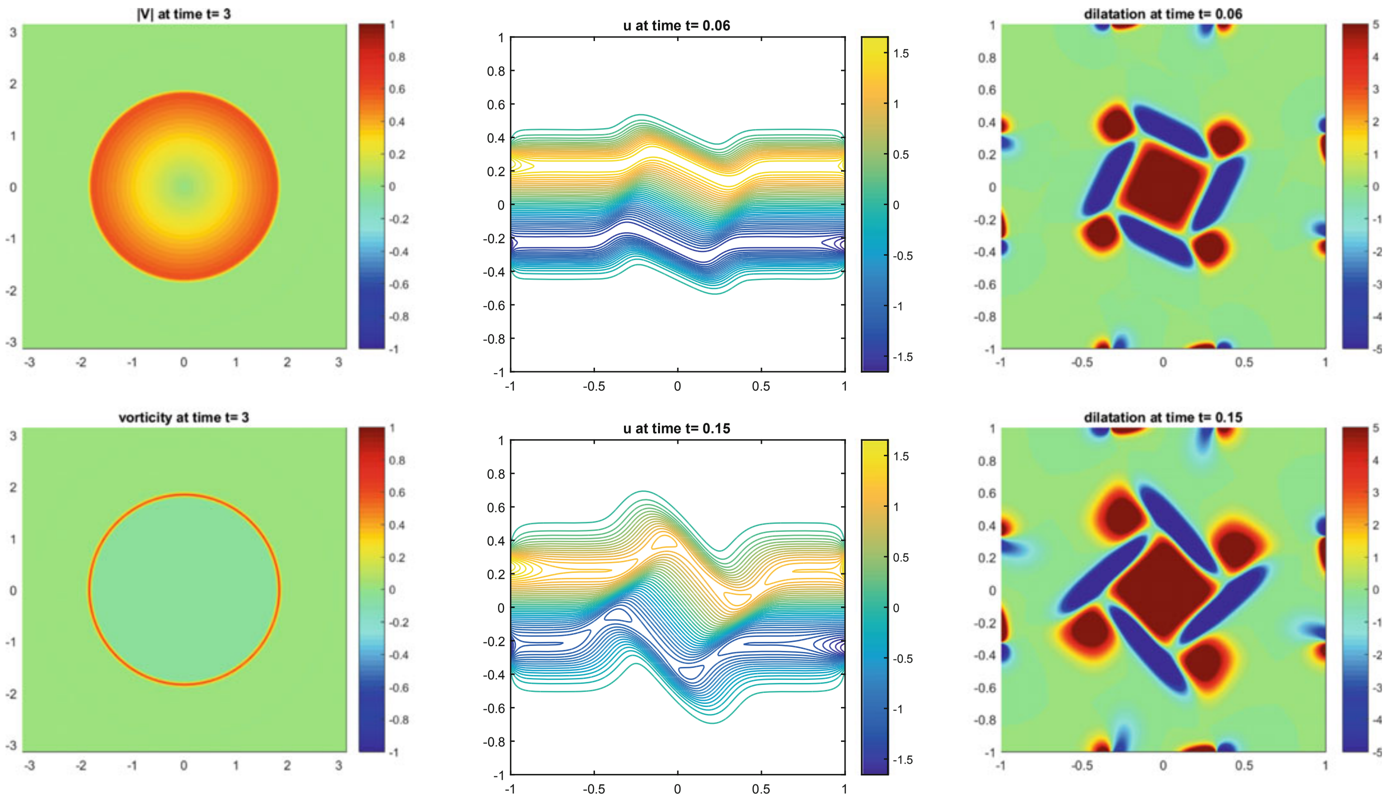


Fig. 7 The plots in the first column correspond to the rotating shock problem. The second and third columns show the velocity in the x direction and the dilatation at times $t = 0.06$ and $t = 0.15$ for the cross flow problem

This initial horizontal velocity drops to zero quickly as we approach $|y| = 0.5$. For $|y| < 0.5$ the exponential term approaches $\exp(0)$ and the velocity behaves like $u = 8y$. The flow has changed slightly by $t = 0.06$, but we can see in Fig. 7 the flow is moving to the right for $y > 0$ and the flow is moving the left for $y < 0$ and all significant behavior is in $|y| < 0.5$. A plot of the velocity v would show similar behavior. We also use 24×24 leafs, $p = 24$, $\epsilon = 0.025$, $k = 1/200$, and $t_{\max} = 0.75$. We show plots of the horizontal velocity u and the dilatation at time $t = 0.06$ and $t = 0.15$. We only show plots before time $t = 0.15$ when the fluid is hardest to resolve and we observe that after $t = 0.15$ the cross streams begin to dissipate. This problem contains sharp interfaces inside $\mathbf{x} \in [-0.5, 0.5]^2$.

5 Conclusion

In this two part series we have demonstrated that the spectrally accurate Hierarchial Poincaré–Steklov solver can be easily extended to handle time dependent PDE problems with a parabolic principal part by using ESDIRK methods. We have outlined the advantages of the two possible ways to formulate implicit Runge–Kutta methods within the HPS scheme and demonstrated the capabilities on both linear and non-linear examples.

There are many avenues for future work, for example:

- Extension of the solvers to compressible and incompressible flows.
- Application of the current solvers to inverse and optimal design problems, in particular for problems where changes in parameters do not require new factorizations.

References

1. Babb, T., Gillman, A., Hao, S., Martinsson, P.: An accelerated Poisson solver based on multidomain spectral discretization. BIT Numer. Math. **58**, 851–879 (2018)
2. Babb, T., Martinsson, P.-G., Appelö, D.: HPS accelerated spectral solvers for time dependent problems (2018). arXiv:1811.04555
3. Carpenter, M., Gottlieb, D., Abarbanel, S., Don, W.-S.: The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error. SIAM J. Sci. Comput. **16**, 1241–1252 (1995)
4. Gillman, A., Martinsson, P.: A direct solver with $\mathscr{O}(N)$ complexity for variable coefficient elliptic PDEs discretized via a high-order composite spectral collocation method. SIAM J. Sci. Comput. **36**, A2023–A2046 (2014). arXiv:1307.2665
5. Haut, T., Babb, T., Martinsson, P., Wingate, B.: A high-order scheme for solving wave propagation problems via the direct construction of an approximate time-evolution operator. IMA J. Numer. Anal. **36**, 688–716 (2016)
6. Kennedy, C., Carpenter, M.: Additive Runge–Kutta schemes for convection-diffusion-reaction equations. Appl. Numer. Math. **44**, 139–181 (2003)

7. Martinsson, P.: A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method. J. Comp. Phys. **242**, 460–479 (2013)
8. Rosales, R., Seibold, B., Shirokoff, D., Zhou, D.: Order reduction in high-order Runge–Kutta methods for initial boundary value problems (2017). arXiv:1712.00897