Implementation Of Mimetic Relaxation Runge Kutta Methods

Anand Srinivasan, Johnny Corbino, and Jose E. Castillo

January 3, 2022

Publication Number: CSRCR2022-01

Computational Science & Engineering Faculty and Students Research Articles

Database Powered by the Computational Science Research Center Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

Implementation Of Mimetic Relaxation Runge Kutta Methods

Anand Srinivasan, Johnny Corbino, José E. Castillo

CSRC Report, 3-Jan-2022

1 Introduction

This report summarizes some of the key aspects behind the implementation of the Mimetic Relaxation Runge Kutta (RRK) methods for hyperbolic partial differential equations (PDEs). The implementation has been performed in Matlab using the Mimetic Operators Library Enhanced (MOLE). The integration of time-dependant PDEs requires a spatial and temporal discretization in order to obtain a numerical solution. The spatial discretization has been performed using the Mimetic discretization methods, and the temporal discretization using the energy conserving Relaxation Runge Kutta methods. The codes referenced in this report can be accessed via `https://github.com/asrinivasan0709/ICOSAHOM2021`.

Mimetic methods [1] work on a staggered grid, and the discrete approximations for vector calculus quantities satisfy a global conservation law. The Castillo-Grone [2] Mimetic methods possess high order approximations for the vector calculus quantities of divergence and gradient by satisfying the extended Gauss' divergence theorem. These approximations for DIV and GRAD are of uniform order of accuracy at the interiors and the boundaries, and are achieved without the use of ghost nodes. More recently, the discretizations introduced by the Corbino-Castillo [3] Mimetic methods possess optimal bandwidth for the DIV and GRAD matrices. These DIV and GRAD Corbino-Castillo matrices are implemented as part of the Mimetic Operators Library Enhanced (MOLE) open-source code. The library can be accessed viahttps://github.com/csrc-sdsu/mole.

This report is organized as follows: we present an overview of the interpolation matrices that are required to map data from staggered to nodal grids (and vice-versa). This is followed by a description of the traditional Runge Kutta (RK) methods and a review of energy stability for PDE's. Finally, we introduce the RRK methods, and conclude with a description of the numerical examples implemented in Matlab.

2 Interpolant Matrices

The Mimetic DIV and GRAD operators work on a staggered grid. On a 1D domain, the DIV operator is defined at the cell centers, and maps the functional values from the nodes to cell centers. The requirement for an interpolant matrix is thus evident, and can be underscored as follows: Consider, for example, the 1D advection equation $u_t + \nabla \cdot u = 0$. This can be discretized using the divergence operator $\hat{\mathbf{D}} \in \mathbb{R}^{(N+2)\times(N+1)}$ acting on U to obtain $U_t + \hat{\mathbf{D}}U = 0$. However, U is defined on a staggered grid, with $U \in \mathbb{R}^{(N+2)\times 1}$. An interpolant is therefore necessary to map the functional values from a staggered to a nodal grid at each step of the numerical integration.

Interpolants can be obtained by constructing Vandermonde matrices for the polynomial basis $\{1, x, x^2, \dots\}$. This approach does present the drawback of numerical oscillations (i.e., Runge phenomenon) for interpolants of orders 9 or higher. Our focus here is for interpolants of even orders 2, 4 and 6, and we therefore do not expect numerical oscillations to be a concern with the interpolant weights. The Vandermonde approach offers the advantage of a matrix-representation for the interpolant weights, making it readily usable in the Mimetic framework offered by the MOLE library.

At the interiors of a 1D grid, the value of the function at nodal location x_i can be obtained by passing a polynomial through the adjacent points $\{\dots, x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, x_{i+1/2}, \dots\}$. As an example, the fourth order interpolant can be obtained by centering it at $x_i = 0$, and passing the polynomial through the points $\{-3/2, -1/2, 1/2, 3/2\}$. A vandermonde matrix V is then constructed for these interpolating points. The weights for the interpolating polynomial are obtained by solving the system of linear equations $Vw = I$, where I is the identity matrix. The first row of the matrix w then becomes the coefficients for the interpolating divergence matrix at the interiors of the grid. A similar procedure can be used to obtain the coefficient weights for the boundaries by implementing a one-sided polynomial interpolation. In the case of the fourth order interpolant, the points on the left boundary of the 1D grid are $\{-1, -1/2, 1/2, 3/2, 5/2\}$.

The resulting second, fourth and sixth order interpolant matrices that map the functional data from a staggered to a nodal grid are as noted below:

$$I_D^2 = \begin{bmatrix} 1 & 0 & \dots & & & & \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & \dots & & \\ & & & & \ddots & & \\ & & & \dots & \frac{1}{2} & \frac{1}{2} & 0 \\ & & & & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(N+1)\times(N+2)}$$

$$I_D^4 = \begin{bmatrix} 1 & 0 & \dots & & & & & \\ -\frac{1}{7} & \frac{5}{8} & \frac{5}{8} & -\frac{1}{8} & \frac{1}{56} & 0 & \dots & \\ 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & \dots & \\ & & & \ddots & & \ddots & & \\ & & \dots & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 \\ & & \dots & \frac{1}{56} & -\frac{1}{8} & \frac{5}{8} & \frac{5}{8} & -\frac{1}{7} \\ & & & & & \dots & 0 & 1 \end{bmatrix}$$

$$I_D^6 = \begin{bmatrix} 1 & 0 & \dots & & & & & & & \\ -\frac{1}{11} & \frac{63}{128} & \frac{105}{128} & -\frac{21}{64} & \frac{9}{64} & -\frac{5}{128} & \frac{1}{201} & 0 & \dots & \\ \frac{1}{33} & -\frac{7}{64} & \frac{35}{64} & \frac{21}{32} & -\frac{5}{32} & \frac{7}{192} & -\frac{1}{235} & 0 & \dots & \\ 0 & \frac{3}{256} & -\frac{25}{256} & \frac{75}{128} & \frac{75}{128} & -\frac{25}{256} & \frac{3}{256} & 0 & \dots & \\ & & & & \ddots & & \ddots & & & \\ & \dots & 0 & \frac{3}{256} & -\frac{25}{256} & \frac{75}{128} & \frac{75}{128} & -\frac{25}{256} & \frac{3}{256} & 0 \\ & \dots & 0 & -\frac{1}{235} & \frac{7}{192} & -\frac{5}{32} & \frac{21}{32} & \frac{35}{64} & -\frac{7}{64} & \frac{1}{33} \\ & \dots & 0 & \frac{1}{201} & -\frac{5}{128} & \frac{9}{64} & -\frac{21}{64} & \frac{105}{128} & \frac{63}{128} & -\frac{1}{11} \\ & & & & & & & \dots & 0 & 1 \end{bmatrix}$$

With the divergence interpolators defined as above, the advection equation can be discretized as $U_t + \hat{\mathbf{D}}\left(I_D^k\, U\right)$, where k is the same order of accuracy as that of $\hat{\mathbf{D}}$.

A similar procedure can be implemented to obtain the interpolant matrices corresponding to the GRAD operator, where the interpolant maps functional data from the cell centers to nodes. The interpolating matrices corresponding to the GRAD operator for orders 2,4 and 6 are as noted below:

$$I_G^2 = \begin{bmatrix} 1 & 0 & \dots & & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \dots & & \\ & & & \ddots & & \\ & & \dots & 0 & \frac{1}{2} & \frac{1}{2} \\ & & & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{(N+2)\times(N+1)}$$

$$I_G^4 = \begin{bmatrix} 1 & 0 & \dots & & & & & \\ \frac{35}{128} & \frac{35}{32} & -\frac{35}{64} & \frac{7}{32} & -\frac{5}{128} & 0 & \dots & \\ -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} & 0 & 0 & \dots & \\ & & & \ddots & & \ddots & & \\ & & \dots & 0 & -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ & & \dots & -\frac{5}{128} & \frac{7}{32} & -\frac{35}{64} & \frac{35}{32} & \frac{35}{128} \\ & & & & & \dots & 0 & 1 \end{bmatrix}$$

$$I_G^6 = \begin{bmatrix} 1 & 0 & \dots & & & & & & & \\ \frac{30}{133} & \frac{134}{99} & -\frac{97}{86} & \frac{231}{256} & -\frac{131}{271} & \frac{20}{133} & -\frac{4}{195} & 0 & \dots & \\ \frac{-4}{195} & \frac{55}{149} & \frac{311}{337} & \frac{-105}{256} & \frac{67}{363} & \frac{-27}{512} & \frac{1}{146} & 0 & \dots & \\ \frac{3}{256} & -\frac{25}{256} & \frac{75}{128} & \frac{75}{128} & -\frac{25}{256} & \frac{3}{256} & 0 & 0 & \dots & \\ & & & & \ddots & & \ddots & & & \\ & \dots & 0 & 0 & \frac{3}{256} & -\frac{25}{256} & \frac{75}{128} & \frac{75}{128} & -\frac{25}{256} & \frac{3}{256} \\ & \dots & 0 & \frac{1}{146} & \frac{-27}{512} & \frac{67}{363} & \frac{-105}{256} & \frac{311}{337} & \frac{55}{149} & \frac{-4}{195} \\ & \dots & 0 & \frac{-4}{195} & \frac{20}{133} & \frac{-131}{271} & \frac{231}{256} & \frac{-97}{86} & \frac{134}{99} & \frac{30}{133} \\ & & & & & & & \dots & 0 & 1 \end{bmatrix}$$

The advection equation can be written using the gradient operator as $u_t + \nabla u = 0$, and the Mimetic discretized version becomes $U_t + \left(I_G^k \, \mathbf{G}\right) \; U = 0$. The interpolation matrices can thus be used in conjunction with the Mimetic DIV and GRAD matrices that are available as part of the MOLE library. Extensions to higher dimensions can be performed by utilizing the kronecker products. At the time of writing this document, the

interpolant matrices have been included as subroutines `interpDmat.m` and `interpGmat.m` in the Matlab implementation of MOLE.

3 Runge Kutta Methods

An s-stage Runge Kutta method with real coefficients $a_{s1}, a_{s2}, \ldots, b_1, \ldots, b_s, c_2, \ldots, c_s$ of the Butcher tableau [4]

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}$$

with n denoting the spatial and time step discretizations, is given by

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{i=1}^{s} b_i \, \mathbf{k}_i^n, \quad \text{where} \tag{1}$$

$$\mathbf{k}_s^n = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} f(t^n + c_j \, \Delta t, a_{ij} \mathbf{k}_j^n), \quad i = 1, 2, \ldots, s \tag{2}$$

The Butcher tableaus for the three schemes (Heun's third order, classical fourth order and Verner's sixth order [5]) that have been implemented in this work are shown in fig. (1).

4 Energy Stability

Kreiss & Scherer [6] established the existence of SBP-operators, which is the discrete equivalent of integration by parts represented as

$$\int_a^b u dv = uv|_a^b - \int_a^b v du \quad \Rightarrow \quad \langle U, \mathbb{D}V \rangle_H = U_b V_b - U_a V_a - \langle V, \mathbb{D}U \rangle_H$$

Strand [7] showed that an operator $\mathbb{D}$ is an SBP-operator for the first derivative if it satisfies $Q = H\mathbb{D}$, where $Q^T + Q = diag(-1, 0, \ldots, 1)$. H is the weight matrix that is invertible. The choice of diagonal weight matrices with positive weights results in the first derivative approximation that is of even order (or order one-less) at the interiors and boundaries.

Hicken & Zingg [8] had showed that this property of SBP-operators leads to energy-preserving solutions for partial differential equations, as long as the discrete equivalent is indeed an accurate representation of the continuous counter-part. They also showed that the structure of the quadrature weights is an accurate representation of the integral if and only if the coefficients satisfy the Euler-Maclaurin series sum.

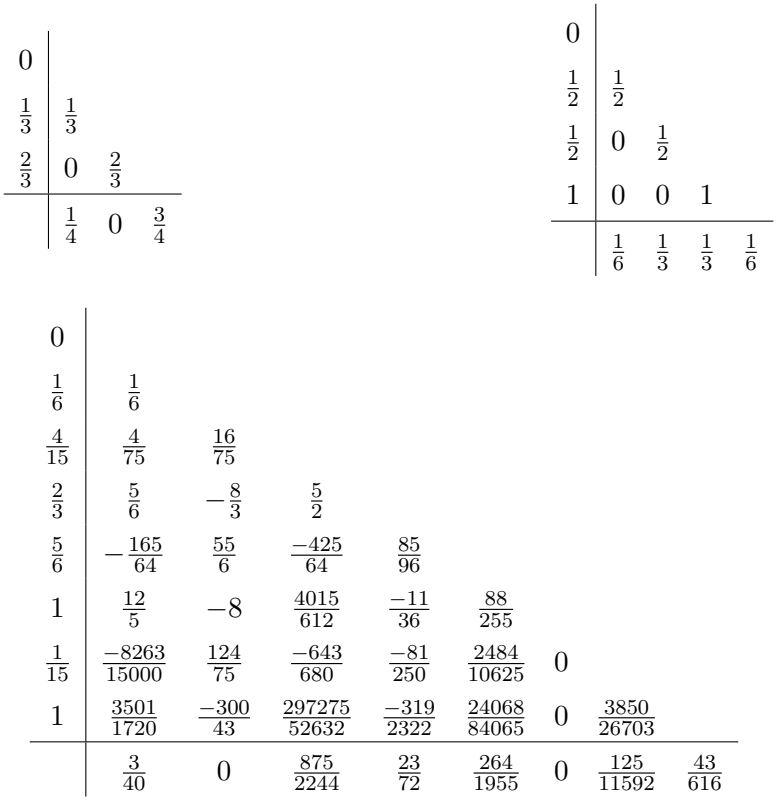

Figure 1: Butcher tableaus for the RK schemes; top left - Heun's third order, top right - fourth order RK scheme, bottom - Verner's sixth order RK scheme.

In the Mimetic framework, the discrete equivalent of the extended Gauss' divergence theorem is given by

$$\int_{\Omega} (\nabla \cdot \vec{v})\, f dV + \int_{\Omega} \vec{v}\ \cdot (\nabla f)\, dV = \int_{\partial\Omega} (f\vec{v}) \cdot \vec{n}\ dS \tag{3}$$

$$\Rightarrow \left\langle \hat{\mathbf{D}} v, \hat{f} \right\rangle_Q + \left\langle \mathbf{G}^T v, \hat{f} \right\rangle_P = \left\langle \mathbf{B} v, \hat{f} \right\rangle \tag{4}$$

Here, the equivalent of the SBP-operators satisfies $Q\hat{\mathbf{D}} + \mathbf{G}^T P = \hat{\mathbf{B}}$. The quadratures P and Q are positive diagonal weight matrices for even orders up to 6. They satisfy the coefficients of the Euler-Maclaurin series sum. Preliminary investigations as outlined in [9] showed the applicability of Mimetic quadratures for numerical integration. The discretizations obtained from using the Mimetic methods are therefore structure preserving, and ensure positivity of solutions.

We present the following remarks in regards to the Mimetic discretization of the advection equation $u_t + \nabla \cdot u = 0$. The norm is defined as $||u||^2 = \langle u, u \rangle = \int_{\Omega} u^2 dV$.

Lemma 1. A numerical integration scheme needs to discretely mimic the conservation law. That is, the change in quantities (eg., mass and energy in the case of the Navier Stokes' equation) within the system has to equal the flux at the boundaries.

Remark 1. By construction, the Castillo-Grone Mimetic coefficients satisfy the discrete version of the extended Gauss' divergence theorem. The extended Gauss' theorem can be re-written as

$$\int_{\Omega} (\nabla \cdot \vec{u})\, u\ dV + \int \vec{u} \cdot (\nabla u)\, dV = \int_{\Omega} \nabla \cdot (u^2) dV = \int_{\partial\Omega} u^2 \cdot \vec{n}\ dS \tag{5}$$

The Castillo-Grone and the Corbino-Castillo Mimetic coefficients satisfy the conservation principle, and are therefore structure-preserving.

Lemma 2. A partial differential equation is said to be well-posed in the sense of energy-stability [10] if the numerical solution is continually dependent on the initial and boundary conditions.

Remark 2. The energy method can be implemented for the advection equation by multiplying it with the solution u and integrating over the domain, to obtain

$$\int_{\Omega} u\ u_t\ dV + \int_{\Omega} u\ \nabla \cdot u\ dV = 0 \tag{6}$$

The first term in eq.(6) becomes $\frac{1}{2}\frac{d}{dt}\int_{\Omega} u^2 dV = \frac{1}{2}\frac{d}{dt}||u||^2$. The second term can be

represented in terms of the extended Gauss' divergence theorem as $\int_\Omega u \ \nabla \cdot u \ dV = \frac{1}{2}\int_\Omega \nabla \cdot (u^2) dV = \frac{1}{2}\int_{\partial\Omega} u^2 dS$.

Thus, we obtain for 1D, $\frac{d}{dt}||u||^2 = -u^2|_0^1$. This is an initial value problem ordinary differential equation and the solution depends continually on the initial and boundary conditions. The Mimetic discretization of the advection equation therefore leads to a well-posed and energy-stable numerical solution of the PDE. Moreover, the Castillo-Grone and Corbino-Castillo Mimetic discretizations for DIV and GRAD are accurate to high order. The resulting quadrature matrices Q and P ensure that the functional estimates for the integral are accurate for the domain.

5 Relaxation RK Methods

The previous section highlighted the fact that the Castillo-Grone & Corbino-Castillo Mimetic methods are energy-preserving spatial discretizations. The traditional RK methods do not conserve energy at each time step of numerical integration. The Relaxation RK methods [11, 12] ensure energy conservation by introducing the relaxation parameter γ. The step update formula therefore becomes

$$\mathbf{u}_\gamma^{n+1} = \mathbf{u}^n + \gamma^n \Delta t \sum_{i=1}^{s} b_i \ \mathbf{k}_i^n \tag{7}$$

The relaxation parameter is calculated such that the energy norm is satisfied at each time step. Thus, $||\mathbf{u}_\gamma^{n+1}|| \leq ||\mathbf{u}^n||$. In the case of the advection equation, the energy norm at time $t = 0$ is given by $E_0 = ||u||^2 = \langle u, u \rangle$. Taking the norm on both sides of eq. (7) and using $\mathbf{d}^n = \sum_{i=1}^{s} b_i \ \mathbf{k}_i^n$, we obtain

$$\begin{aligned} ||\mathbf{u}_\gamma^1|| &= ||\mathbf{u}^0 + \gamma \Delta t \mathbf{d}^0|| &(8)\\ &= E_0 + \gamma^2 \Delta t^2 \left\langle \mathbf{d}^0, \mathbf{d}^0 \right\rangle + 2\gamma \Delta t \left\langle \mathbf{u}^0, \mathbf{d}^0 \right\rangle &(9) \end{aligned}$$

Since the choice of γ ensures energy conservation, the term $||\mathbf{u}_\gamma^1|| - E_0$ equals zero, from which we obtain an express1986numericalion for γ as

$$\gamma^n \Delta t = -\frac{2\left\langle \mathbf{u}^n, \mathbf{d}^n \right\rangle}{\left\langle \mathbf{d}^n, \mathbf{d}^n \right\rangle} \tag{10}$$

Ranocha et al [12] note that a root finding algorithm is necessary in order to calculate γ in lieu of using a closed form solution as noted above. The implementation of the RRK methods in this present work does not include a root finding algorithm, and is intended to be added in the future.

The combination of the structure-preserving Mimetic spatial discretization method with the energy-preserving RRK temporal scheme is well suited for problems such as hyperbolic PDE's, where numerical energy conservation is necessary.

6 Implementation in Matlab

The advection equation was solved on a one dimensional grid $x \in [-5, 5]$ and $a = 1$, with a Gaussian initial condition

$$u(x,0) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{(x-\mu)^2}{2\sigma}\right), \quad \mu = 0,\ \sigma = 0.15 \tag{11}$$

Table (1) shows the convergence rates calculated at time $t = 1s$, with the maximum error norm calculated as the difference between the exact and numerical solutions. The fourth and sixth order Mimetic schemes were compared with those of Wicker & Skamarock's [13] flux discretization schemes. Both the Mimetic and the finite difference methods achieve the desired order of accuracy, with the Mimetic discretization achieving a slightly lower error norm.

The time integration was performed using three different schemes:

- fourth order Mimetic & third order Runge Kutta (legends MIM4-RK3 & MIM4-RRK3),
- fourth order Mimetic & Runge Kutta (legends MIM4-RK4 & MIM4-RRK4),
- sixth order Mimetic & Runge Kutta (legends MIM6-RK6 & MIM6-RRK6).

RK3 refers to the third order scheme of Heun, RK4 refers to the traditional RK method, and RK6 refers to the sixth order Verner's method. Figure (2) shows the energy evolution for the standard versus the Relaxation RK schemes. The Mimetic-RRK schemes preserve numerical energy to numeric precision.

Additional examples illustrating the Mimetic-RRK schemes for the 1D wave equation, and the non-linear inviscid Burgers' equation are also available in `https://github.com/asrinivasan0709/ICOSAHOM2021`.

Table 1: Example 1, Order of accuracy

Type	N	Δh	RK - error	RRK - error	RK - order	RRK - order
WIC4	100	0.1000	0.0026	0.0026		
	200	0.0500	1.7536e-04	1.7443e-04	3.898	3.889
	400	0.0250	1.1074e-05	1.1043e-05	3.985	3.981
	800	0.0125	6.9471e-07	6.9374e-07	3.995	3.993
	1600	0.0063	4.3451e-08	4.3421e-08	3.999	3.998
MIM4	100	0.1000	0.0024	0.0023		
	200	0.0500	1.5358e-04	1.5257e-04	3.941	3.934
	400	0.0250	9.7021e-06	9.6707e-06	3.985	3.98
	800	0.0125	6.0791e-07	6.0696e-07	3.996	3.994
	1600	0.0063	3.8021e-08	3.7991e-08	3.999	3.998
WIC6	100	0.1000	1.7976e-04	1.7975e-04		
	200	0.0500	3.0320e-06	3.0319e-06	5.89	5.89
	400	0.0250	4.8326e-08	4.8326e-08	5.971	5.971
	800	0.0125	7.6017e-10	7.6016e-10	5.99	5.99
	1600	0.0063	1.1891e-11	1.1894e-11	5.998	5.998
MIM6	100	0.1000	1.3871e-04	1.3870e-04		
	200	0.0500	2.3713e-06	2.3712e-06	5.87	5.87
	400	0.0250	3.7856e-08	3.7855e-08	5.969	5.969
	800	0.0125	5.9390e-10	5.9393e-10	5.994	5.994
	1600	0.0063	9.3284e-12	9.4157e-12	5.992	5.979

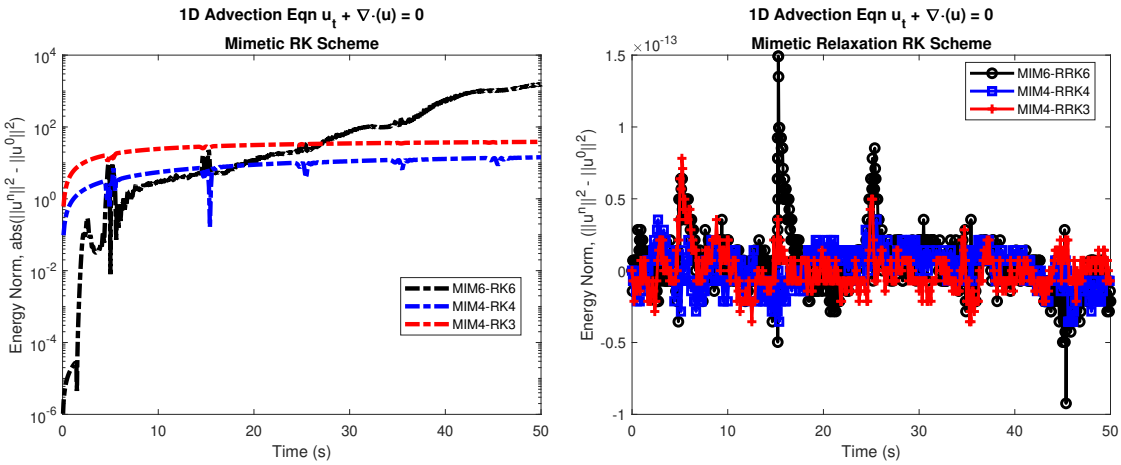

Figure 2: Energy evolution using the standard RK methods (left) and the Relaxation RK methods (right). The Mimetic-RRK methods preserve energy to machine numeric precision.

7 Conclusion

This report has summarized the implementation of the Mimetic Relaxation Runge Kutta methods for hyperbolic PDE's. The Matlab codes that accompany this report have been posted on github. The Mimetic RRK schemes ensure conservation of numerical energy at each step of integration. Suggestions for future work can be outlined as follows:

- The current work has only considered explicit RK schemes. The performance of the Mimetic schemes with implicit RK methods will need to be investigated. This will provide a comparison of the computational cost between an explicit method that requires calculations for γ, versus that of an implicit scheme.
- The comparison of the Mimetic RRK schemes with those of area-preserving symplectic methods require a more in-depth investigation.
- The Rosenbrock temporal schemes [14] that exist in the literature aim to avoid implicit calculations by introducing the Jacobian calculations at each time step. It will be worthwhile to investigate the possibility of Mimetic-Rosenbrock methods in an attempt to understand the energy-preserving nature of such discretization schemes.
- The extension of the Mimetic-RRK methods for parabolic equations and other nonlinear problems also remains to be investigated. Once again, the aim here will be to understand the computational cost versus benefits of these schemes.

References

[1] José E Castillo and Guillermo F Miranda. *Mimetic discretization methods*. CRC Press, 2013.

[2] Jose E Castillo and RD Grone. A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law. *SIAM Journal on Matrix Analysis and Applications*, 25(1):128–142, 2003.

[3] Johnny Corbino and Jose E Castillo. High-order mimetic finite-difference operators satisfying the extended gauss divergence theorem. *Journal of Computational and Applied Mathematics*, 364:112326, 2020.

[4] John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[5] Ernst Hairer, Syvert P Nørsett, and Gerhard Wanner. *Solving ordinary differential equations. 1, Nonstiff problems*. Springer-Vlg, 1993.

[6] H-O Kreiss and Godela Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. In *Mathematical aspects of finite elements in partial differential equations*, pages 195–212. Elsevier, 1974.

[7] Bo Strand. Summation by parts for finite difference approximations for d/dx. *Journal of Computational Physics*, 110(1):47–67, 1994.

[8] Jason E Hicken and David W Zingg. Summation-by-parts operators and high-order quadrature. *Journal of Computational and Applied Mathematics*, 237(1):111–125, 2013.

[9] Anand Srinivasan and Jose E. Castillo. Implementation of the newton-cotes and mimetic quadrature coefficients for numerical integration. *CSRC Report*, 2016.

[10] Lucas Friedrich, DC Fernández, Andrew R Winters, Gregor J Gassner, David W Zingg, and Jason Hicken. Conservative and stable degree preserving sbp finite difference operators for non-conforming meshes. *arXiv preprint arXiv:1611.00979*, 2016.

[11] David I Ketcheson. Relaxation runge–kutta methods: Conservation and stability for inner-product norms. *SIAM Journal on Numerical Analysis*, 57(6):2850–2870, 2019.

[12] Hendrik Ranocha, Mohammed Sayyari, Lisandro Dalcin, Matteo Parsani, and David I Ketcheson. Relaxation runge–kutta methods: Fully discrete explicit entropy-stable schemes for the compressible euler and navier–stokes equations. *SIAM Journal on Scientific Computing*, 42(2):A612–A638, 2020.

[13] Louis J Wicker and William C Skamarock. Time-splitting methods for elastic models using forward time schemes. *Monthly weather review*, 130(8):2088–2097, 2002.

[14] William H Press, William T Vetterling, Saul A Teukolsky, and Brian P Flannery. *Numerical recipes*, volume 818. Cambridge university press Cambridge, 1986.