Time Integration of Reacting Flows and Energy-Stable Hybrid Spatial Discretizations

Cory Mikida
Department of Aerospace Engineering
University of Illinois at Urbana-Champaign

January 13, 2021

Contents

1 Time Integration of Reacting Flows

1.1 Motivation & Background

1.1.1 Goals, Bounds, and Impact

The primary goal of this section of the thesis is demonstration of a novel time integration scheme driving a canonical or especially demonstrative reacting flow problem (ideally one that points directly to real-world impact) with improved efficacy identified by either superior accuracy and stability or faster performance. Also critical in this effort will be maintaining a solution that lies on the ideal gas constraint manifold (i.e. a physical solution) - this has been identified in early work as a potential sticking point when fully differential approaches to solving the governing equations (as opposed to differential-algebraic approaches that directly satisfy the constraints) are used. While the currently working method of solving the equations uses a chemical kinetic Jacobian modified to include the algebraic constraints explicitly, fully differential approaches that use attenuation to force the solution towards the constraint manifold may also be employed to greater effect.

Perhaps the next most prominent goal is a thorough investigation of the design space presented by the multi-rate framework we will use for time integration (see *The Plan*). The design choices inherent to this scheme include:

- Evaluation order ("fastest-first" vs. "slowest-first")
- Re-extrapolation
- Inclusion of additional history beyond order requirements (shown in [13] to provide real-axis stability improvement)
- Which solution components to include in error estimation for adaptive timestep control
- Whether error control is accomplished through timestep control or step ratio control

Another critical component would be comparison of our scheme with CVODE as a proxy for the state of the art in time integration of chemically reacting flows, ideally achieving superiority in terms of observed temporal accuracy of the entire system, or in terms of performance (most accurately measurable by a reduction in the number of chemistry right-hand-side evaluations required to reach a given solution time).

Meanwhile, an aspect of the algorithms being implemented that has been identified early as beyond our bounds is the presence of multi-variable solves in a number of potential adaptive implicit integration scenarios, namely:

- Coupled implicit right-hand-sides
- Use of adaptive timestep controllers where the local error estimate of all solution components makes use of implicit and explicit state estimates.

With these goals and bounds in mind, the proposed thesis has the potential to have a significant impact on the simulation of reacting flows, of which there are countless real-world applications including (but not limited to) subsonic and supersonic combustion for propulsion (gas turbines, ramjets/scramjets, rockets), furnaces and residential heating, and solution of problems relating to atmospheric sciences.

The proposed work would also serve to augment the capabilities of Leap and Dagrt, two Python packages for code generation of time integrators, by implementing the proposed integrators. While this work would focus the attention of those new integrators on reacting flows, additional useful application of implicit-explicit multi-rate with error-based step control amongst the software's user base is possible, if not probable.

Finally, the proposed work would also formally introduce new capabilities to another existing combustion software package, PyJac-V2. This software is responsible for code generation of callable source term and Jacobian functions for chemical kinetics, and our work (given its application to problems that often have large temperature ranges, not to mention a focus on high accuracy) would add a NASA9 polynomial representation of thermodynamic quantities to its codebase.

1.2 Governing Equations

The problem at hand involves the simulation of chemically reacting flows, the governing equations of which amount to the Navier-Stokes equations with (in our form):

- A source term added to the energy equation in the form of heat flux
- The addition of a governing equation for the rate of change of the mass fractions of each species in the chemical mechanism simulated
- The addition of governing equations specifying the rate of change of temperature and pressure, derived from a constant internal energy assumption (for the former) and the ideal gas law (for the latter).

These equations are given below, along with a brief explanation of the notation/nomenclature used.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j + P\delta_{ij} - \tau_{ij}) = 0 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}((\rho E + P)u_j + q_j - u_i\tau_{ij}) = 0 \tag{3}$$

$$\frac{\partial}{\partial t}(\rho Y_k) + \frac{\partial}{\partial x_j}(\rho Y_k u_j - \varphi_{ki}) = W_k\dot{\omega}_k \tag{4}$$

$$\frac{\partial T}{\partial t} + \frac{\sum_{k=1}^{N_{sp}} U_k(T)\frac{\partial \rho Y_k}{\partial t}\frac{1}{W_k}}{\sum_{k=1}^{N_{sp}}[C]_k C_{v,k}(T)} = 0 \tag{5}$$

$$\frac{\partial P}{\partial t} - \frac{R}{V}(T\frac{\partial n}{\partial t} + \frac{\partial T}{\partial t}n) = 0 \tag{6}$$

In these equations, ρ is the fluid density, u_i is the velocity in the i-th direction, and E is the total energy. δ_{ij} is the Kronecker delta, P is the pressure, T is the temperature, and Y_k is the mass fraction of species k. $[C]_k$ is the concentration of species k in the gas mixture, $C_{v,k}(T)$ is the specific heat at constant volume of species k at temperature T, and $U_k(T)$ is the internal energy of species k at temperature T. These quantities are defined using the NASA 9-coefficient polynomial parameterization [11], such that

$$\frac{C_{v,k}(T)}{R} = a_0 + a_1 T + a_2 T^2 + a_3 T^3 + a_4 T^4 - 1 \tag{7}$$

$$\frac{U_k(T)}{RT} = a_0 + \frac{a_1}{2} T + \frac{a_2}{3} T^2 + \frac{a_3}{4} T^3 + \frac{a_4}{5} T^4 + \frac{a_5}{T} - 1, \tag{8}$$

where the a coefficients are defined per-species for an arbitrary number of temperature regions.

Returning to the Navier-Stokes equations (1) - (6), N_{sp} is the number of species in the chemical mechanism, W_k is the molecular weight of species k, n is the number of moles of the gas mixture, V is the volume of the gas, and $\dot{\omega}_k$ is the net production rate of species k. τ_{ij} is the viscous stress tensor, φ_{ki} are the diffusion fluxes, and q_j are the heat fluxes, defined by

$$q_j = -\frac{\partial(\lambda T)}{\partial x_j} + h_k \varphi_{kj}. \tag{9}$$

In this expression, λ is the thermal conductivity, and h_k is the enthalpy of species k. As for the diffusion fluxes, these are defined using a mixture-averaged approach - that is, φ_{ki} is given by

$$\varphi_{ki} = \varphi^*_{ki} + \varphi^c_{ki}, \tag{10}$$

where φ^*_{ki} is the mixture-averaged approximation, and φ^c_{ki} is a correction term to ensure mass conservation. The mixture-averaged approximation is defined by

$$\varphi^*_{ki} = -\rho D_{k,m} \frac{W_k}{W} \frac{\partial X_k}{\partial x_i} \tag{11}$$

where $D_{k,m}$ is the mixture-averaged diffusivity of species k, W is the mean molecular weight, and X_k is the mole fraction of species k. Finally, the correction term is given by

$$\varphi^c_{ki} = -Y_k \sum_{n=1}^{N_{sp}} \varphi^*_{ni} \tag{12}$$

In this set of governing equations (1) - (6), the mass fraction source term on the right side of equation (4) is typically observed to be stiff relative to the surrounding equations governing the motion of the gas mixture and the change in its physical properties. This presents a time integration problem in that this term alone typically dictates either an oppressively low timestep or the need for an implicit approach, often using tools such as

CVODE. In practice, fluid solvers also often decouple an explicit treatment of the Navier-Stokes equations (1) - (3) from the implicit treatment of the chemical kinetics, resulting in a splitting approach that is at best first order in time.

The topic of this section of the proposal, then, is to answer the following research question: **can we derive performance and/or temporal accuracy benefits from an application of multi-rate Adams integrators to this set of governing equations?**

1.3 Numerical Methods

1.3.1 Adams Methods

To fix notation for new schemes developed later, we here give a brief derivation of a standard Adams-Bashforth (AB) integrator, as described in [1]. We start with a model IVP given by

$$\frac{\mathrm{d}y}{\mathrm{d}t} = F(t, y), \quad y(0) = y_0.$$

This is the form that results from a method of lines (MOL) approach to solving time-dependent partial differential equations like those considered in this study. We approximate the time dependency of the right-hand side function with a polynomial with coefficients $\boldsymbol{\alpha}$ (formed by interpolating past values of $F(t, y)$), extrapolate with that polynomial approximation, and integrate the extrapolant. We use a Vandermonde matrix to construct a linear system to obtain the coefficients $\boldsymbol{\alpha}$ to be used in extrapolation from history values:

$$V^T \cdot \boldsymbol{\alpha} = \int_0^{\Delta t} \tau^i d\tau = \frac{(\Delta t)^i}{i+1}, \quad i = 1, 2..., n, \quad V = \begin{bmatrix} 1 & t_1 & \dots & t_1^{n-1} \\ 1 & t_2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \dots & t_n^{n-1} \end{bmatrix}, \tag{13}$$

where $\int_0^{\Delta t} \tau^i d\tau$ is a vector evaluating the integral of the interpolation polynomial, and V is the Vandermonde matrix with monomial basis and nodes $t_1, t_2, \dots t_n$, corresponding to past time values. In (13), n is equal to the order of the integrator, and t_i are the time history values, with $0 \leq t_1 < t_2 \dots < t_n$. The coefficients $\boldsymbol{\alpha}$ are used to extrapolate to the next state via

$$\begin{aligned} y(t_{i+1}) = y(t_i) &+ \alpha_1 F(t_{i-n}, y_{i-n}) + \alpha_2 F(t_{i-n-1}, y_{i-n-1}) \\ &+ \dots + \alpha_n F(t_i, y_i). \end{aligned} \tag{14}$$

Clearly, the length of the past history needed to calculate a step (and, thus, the memory required) influences the order of accuracy attained.

An alternative time integration method is required for the first few time steps (the exact number of which is dependent on the number of history values needed) in order to establish right-hand side history and "bootstrap" the method. We use a third-order Runge-Kutta (RK3) integrator [6] to bootstrap the third-order AB methods, whereas a fourth-order Runge-Kutta (RK4) integrator [9] is used to bootstrap the fourth-order AB methods.

1.3.2 Multi-rate Adams Methods

We now describe a multi-rate generalization of the scheme, making use of the algorithm introduced in [5]. We consider the following model system with "fast" and "slow" solution components:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} f(t) \\ s(t) \end{pmatrix} = \begin{pmatrix} a_f(f,s) \\ a_s(f,s) \end{pmatrix}. \tag{15}$$

With this in mind, we can set a slow (larger) time step H for a_s such that we maintain stability in the integration of the slow component. We also set a fast time step h for a_f such that H is an integer multiple of h, and define the ratio between the two, $\mathrm{SR} = H/h$, as the step ratio of the MRAB scheme. While the results presented here make use of only two separate state components, each with its own right-hand side function and independent rate, the theory is readily extensible to any number of rates.

Within this two-component scheme, a few design choices are available:

- The order in which we evaluate and advance the solution components. Namely, two primary options are advancing the fast-evolving solution component through all of its micro-timesteps h and waiting to perform the single macro-timestep H required for the slow component until the end (a "fastest-first" scheme, per the nomenclature of [5]), or pursuing an algorithm in which the slow component is instead advanced first.
- For slowest-first evaluation schemes, the choice of whether or not to re-extrapolate the slow state after additional state and right-hand side information is gathered at the micro-timestep level.

Empirical observations on the effects of these choices are made in [7]. It is useful to step through a brief example of a multi-rate Adams-Bashforth integrator, using a system with a fast component requiring twice as many timesteps as the slow component to remain well-resolved ($\mathrm{SR} = 2$). We lay out the steps of a third-order fastest-first MRAB scheme with no re-extrapolation, assuming that a_s evolves at the slow rate (macro-timestep $H = 2h$) and a_f evolves at the fast rate (micro-timestep h). $\hat{a}$ denotes extrapolants of the right-hand side functions as polynomial functions of both time t and the set of history values $\vec{a}_{\text{hist}}$: $\hat{a} = P(t, \vec{a}_{\text{hist}})$. These polynomials approximating the evolution of a_f and a_s in time are what we will integrate to march a_f and a_s, and will be updated to replace older history values with new right-hand side evaluations during the course of integration through a macro-timestep H. We assume availability of right-hand side histories to start the AB method.

Step 1: Form the polynomial extrapolants we will integrate, per the AB methods described in the previous subsection:

$$\hat{a}_f(t) = P\Big(t, [a_f(f(t_{i-2}), s(t_{i-2})), a_f(f(t_{i-1}), s(t_{i-1})), a_f(f(t_i), s(t_i))]\Big)$$
$$\hat{a}_s(t) = P\Big(t, [a_s(f(t_{i-4}), s(t_{i-4})), a_s(f(t_{i-2}), s(t_{i-2})), a_s(f(t_i), s(t_i))]\Big)$$

The right-hand side history values of a_f (used to form $\hat{a}_f$) have been obtained at time points $t_{i-2} = t - 2h$, $t_{i-1} = t - h$, and current time $t_i = t$, whereas the right-hand side history values of a_s (used to form $\hat{a}_s$) have been obtained at time points $t_{i-4} = t - 2H$, $t_{i-2} = t - H$, and $t_i = t$.

Step 2: March both f and s to time t_{i+1} by integrating the polynomial extrapolants $\hat{a}_s(t)$ and $\hat{a}_f(t)$ formed in Step 1:

$$f(t_{i+1}) = f(t_i) + \int_{t_i}^{t_{i+1}} \hat{a}_f(\tau)\,d\tau,$$
$$s(t_{i+1}) = s(t_i) + \int_{t_i}^{t_{i+1}} \hat{a}_s(\tau)\,d\tau.$$

This results in a set of intermediate values $f(t_{i+1})$ and $s(t_{i+1})$.

Step 3: Evaluate the fast right-hand side $a_f(f(t_{i+1}), s(t_{i+1}))$.

Step 4: Update the set of right-hand side history values for a_f to include these new values, and construct a new extrapolant $\hat{a}_f$:

$$\hat{a}_f(t) = P\left(t, [a_f(f(t_{i-1}), s(t_{i-1})), a_f(f(t_i), s(t_i)), a_f(f(t_{i+1}), s(t_{i+1}))]\right).$$

Step 5: March s to time t_{i+2} by integrating the extrapolant formed in Step 1:

$$s(t_{i+2}) = s(t_i) + \int_{t_i}^{t_{i+2}} \hat{a}_s(\tau)\,d\tau.$$

Step 6: March f to time t_{i+2} by integrating the extrapolant formed in Step 3:

$$f(t_{i+2}) = f(t_{i+1}) + \int_{t_{i+1}}^{t_{i+2}} \hat{a}_f(\tau)\,d\tau.$$

Step 7: Evaluate $a_f(f(t_{i+2}), s(t_{i+2}))$ and $a_s(f(t_{i+2}), s(t_{i+2}))$ and update extrapolants:

$$\hat{a}_f(t) = P\Big(t, [a_f(f(t_i), s(t_i)), a_f(f(t_{i+1}), s(t_{i+1})), a_f(f(t_{i+2}), s(t_{i+2}))]\Big),$$
$$\hat{a}_s(t) = P\Big(t, [a_s(f(t_{i-2}), s(t_{i-2})), a_s(f(t_i), s(t_i)), a_s(f(t_{i+2}), s(t_{i+2}))]\Big).$$

The scheme evaluates the fast-evolving right-hand side a_f twice per macro-timestep H, whereas the slowly-evolving right-hand side a_s is only evaluated once. For the results shown later, this is the scheme we will use, albeit generalized to different step ratios $\mathrm{SR} = H/h$.

1.3.3 Timestep Control Algorithm

Early implicit-explicit runs of small-scale reacting flow problems quickly identified a need to adapt the timestep of the integrator based on relative and/or absolute local error demands.

We therefore employ a timestep control algorithm which, in its current form, is that of ODE45, wherein a timestep multiplier r is calculated based on a local error estimate constructed from state estimates of two (differing) explicit orders:

$$r = \frac{\|s_{q+1} - s_q\|_2}{\text{ATOL} + \text{RTOL} \cdot \max(\|s_q\|_2, \|s_{q+1}\|_2)} \tag{16}$$

Here, s_q is the state estimate obtained using an order-q scheme, and ATOL and RTOL are absolute and relative error tolerances specified by the user. Once r is calculated, the timestep is *decreased* if $r >= 1$ via

$$\Delta t = 0.9\Delta t(r)^{-1/q} \tag{17}$$

and *increased* if $r < 1$ via

$$\Delta t = 0.9\Delta t(r)^{-1/(q+1)} \tag{18}$$

This added capability alone presents a number of questions in terms of how best to accomplish this error control. In particular, step ratio adjustment in situ, rather than timestep adjustment, to meet error needs provides an implementation challenge in terms of code generation of these integrators, whereas the choice of which solution component(s) are involved in error estimation/timestep control also presents a design decision that could have notable effect on numerical performance.

1.4 Validation

With the plan for the numerical methods to be implemented in place, the dominant planning question then becomes: what is the test problem to which these new methods will be applied to establish their viability? The current plan is to use a reacting mixing layer problem with a hyperbolic tangent velocity profile for the baseflow (as used by [12] and [2]), with the San-Diego 9-species chemical mechanism employed for the chemical kinetics. By perturbing this baseflow with the mode having the highest growth rate to instability (based on the inviscid analyses of [12] and [2]), we can establish a fast-evolving solution component (the chemical kinetics/mass fractions of the species) and a slow-evolving solution component (the fluid), both possessing accessible means of rate modification (for the chemistry, we can accomplish this by scaling the pre-exponential factors in the Arrhenius reaction rates, and for the fluid, we can scale the magnitude of the initial crossflow).

In doing so, what should result is a problem with ample testbed capabilities for our new multi-rate integrators, and also a problem firmly rooted in physical utility in that it provides the simplest model for scramjet/ramjet combustion.

1.5 Results

(Insert text here)

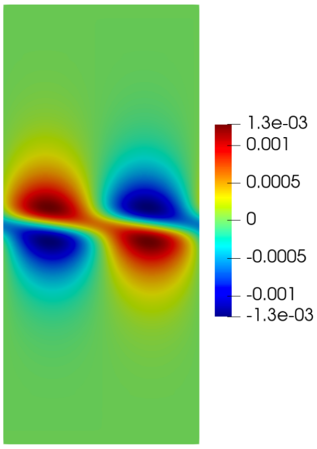

Figure 1: Perturbed vertical momentum of the hyperbolic tangent example - multispecies cold flow.

1.6 Outlook

1.6.1 Current Status

At present, implicit-explicit time integration of reacting flows using Runge-Kutta based methods in conjunction with code generation of source terms and Jacobians for the chemical mechanisms is implemented and undergoing testing with the fluid solver application. Implicit Adams methods (with single-rate Adams-Moulton being the baseline) for the purposes of chemistry integration are also being implemented, along with an application of the error-informed timestep control algorithm to both single-rate and multi-rate Adams methods. Construction of implicit-explicit multi-rate Adams methods with error-informed adaptivity are well underway, with testing of these new methods on small-scale Cantera reactor system problems in progress.

As for the reacting crossflow validation problem, an initial setup employing cold flow (no autoignition) with the perturbed baseflow applied to a nine-species (San Diego) gas mixture is complete, with validation via the growth rate of the unstable mode ongoing. Also implemented in the reacting flow solver is a one-dimensional laminar free flame problem, which may also be used as a first-pass validation (via comparison to the Cantera-estimated steady-state flame speed) for new integration methods for reacting flows as they become available.

1.6.2 Risk Mitigation

An important question to ask amidst all of these promises is: textbfwhat if this doesn't work out? Assuming demonstration of improved performance or accuracy of the *multi-rate* integrators over the state-of-the-art is for reasons unforseen impossible, a "fallback" takeaway would be demonstration of improvement over the CVODE-fluid first-order splitting approach via Leap implementation of an existing IMEX Runge-Kutta based scheme that treats the chemistry implicitly via code-generated analytical Jacobians. Given the use of a coupled implicit-explicit scheme (for which high order has already been proven in other circumstances), as well as a chemical Jacobian that is more accurately obtained than via finite differences (the method of CVODE), this outcome should be attainable at minimum.

2 Energy-Stable Hybrid Spatial Discretizations

2.1 Motivation & Background

2.1.1 Goals, Bounds, and Impact

This section of the proposal targets a lack of high-order accurate and provably stable interface conditions between structured and unstructured meshes for computational fluid dynamic simulations. In particular, our aim is to create a provably stable and accurate interface between summation-by-parts finite difference discretizations (structured) and discontinuous Galerkin methods (unstructured). Achieving this would provide an option to create localized areas of unstructured meshing (with superior flexibility for discretizing complex boundaries/geometries) in existing simulations, providing an alternative to overset curvilinear meshes.

2.2 Governing Equations

The initial problem for which our new hybrid method will be targeted is the acoustic wave equation in two dimensions, in first-order form:

$$\rho \frac{\partial v_i}{\partial t} + \frac{\partial p}{\partial x_i} = 0 \;\; (i = 1, 2), \tag{19}$$

$$\frac{\partial p}{\partial t} + \lambda \left(\frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} \right) = 0 \tag{20}$$

As in [8], our aim will be to prove the semi-discrete hybrid discretization stable by proving that

$$\frac{d\mathcal{E}}{dt} \leq 0 \tag{21}$$

where the total energy of the semi-discrete system $\mathcal{E}$ is obtained by summing the energy E of each discretization block:

$$\mathcal{E} = \sum_{blocks} E \tag{22}$$

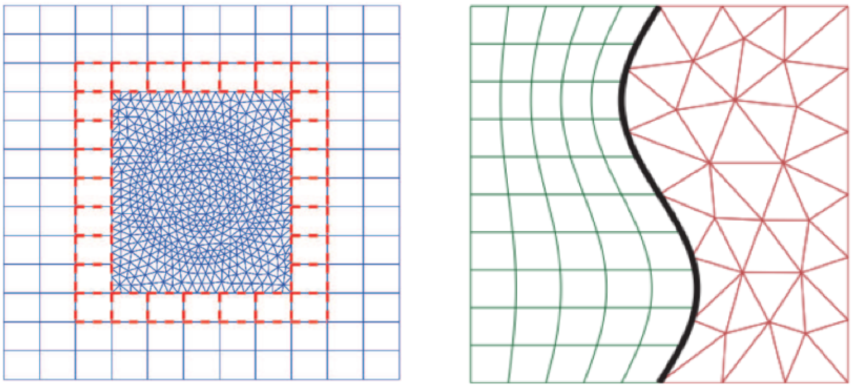

Figure 2: Two examples demonstrating use of nonconforming interfaces. Left: Zhu, Chen, Zhong, and Liu [17]. Right: Kozdon and Wilcox [8]

2.3 Numerical Methods

2.3.1 Summation-by-Parts Operators

For the structured discretization that makes up one component of the proposed hybrid, we make use of several finite difference operators that possess the summation-by-parts (SBP) property. Taking two matrices P, Q, we here state that these two matrices are SBP matrices of order p provided

- $P^{-1}Q\mathbf{v}$ is an order h^p approximation to $\partial/\partial x$, where h is the spatial step size in one dimension.
- P is a symmetric positive-definite matrix.
- $Q + Q^T = \text{diag}(-1, 0, 0, ..., 0, 1)$.

These conditions together ensure that the discrete version of the integration by parts property holds; that is,

$$\langle P^{-1}Q\mathbf{x}, \mathbf{y}\rangle_P = \mathbf{x}_N\mathbf{y}_N - \mathbf{x}_1\mathbf{y}_1 - \langle \mathbf{x}, P^{-1}Q\mathbf{y}\rangle_P.$$

The resulting operators can be either explicit (in this case, P is purely diagonal) or implicit. This theory, originally presented in [14], can also be extended to higher dimensions using Kronecker products. For example, in two dimensions, the matrices $H^{-1}G_x$ and $H^{-1}G_y$ define the x- and y- derivatives on a two-dimensional grid:

$$\begin{aligned} H &= P_x \otimes P_y \\ G_x &= Q_x \otimes P_y \\ G_y &= P_x \otimes Q_y \end{aligned}$$

Note that this formulation assumes that we have P_x, Q_x, a pair of $n_x \times n_x$ SBP matrices of approximation order p, and P_y, Q_y, a pair of $n_y \times n_y$ SBP matrices of approximation order q. Finally, we also note that these SBP operators do not guarantee strict stability for an initial boundary value problem. We must also apply the boundary conditions using a formulation that permits an energy estimate. More information on these boundary conditions, called simultaneous approximation term (SAT) boundary conditions, can be found at [15], [16], and [3].

More information on SBP operators of various order, including the coefficients themselves, can be found in [14], [4], [10]. For this work, we will employ diagonal-norm SBP operators for the finite difference discretizations.

2.3.2 Discontinuous Galerkin Method

2.3.3 Interface Method

2.4 Validation

The test problem for validation will match that of [8], modeling the two-dimensional acoustic wave equation in first-order form with the following initial condition:

$$p(x_1, x_2, 0) = \cos(k_1 x_1)\cos(k_1 x_2) + \sin(k_2 x_1)\sin(k_2 x_2), \tag{23}$$

$$v_i(x_1, x_2, 0) = 0, \qquad i = 1, 2, \tag{24}$$

where $k_1 = \pi/2$ and $k_2 = \pi$. All exterior boundary conditions are zero pressure (free-surface) conditions. The exact solution of this problem is

$$p(x_1, x_2, t) = \cos(\omega_1 t)\cos(k_1 x_1)\cos(k_1 x_2) + \cos(\omega_2 t)\sin(k_2 x_1)\sin(k_2 x_2), \tag{25}$$

$$v_1(x_1, x_2, t) = \frac{k_1}{\omega_1}\sin(\omega_1 t)\sin(k_1 x_1)\cos(k_1 x_2) - \frac{k_2}{\omega_2}\sin(\omega_2 t)\cos(k_2 x_1)\sin(k_2 x_2), \tag{26}$$

$$v_2(x_1, x_2, t) = \frac{k_1}{\omega_1}\sin(\omega_1 t)\cos(k_1 x_1)\sin(k_1 x_2) - \frac{k_2}{\omega_2}\sin(\omega_2 t)\sin(k_2 x_1)\cos(k_2 x_2), \tag{27}$$

where $\omega_j = k_j\sqrt{2}$ for $j = 1, 2$.

As for the spatial discretization, we will employ a square spatial domain with the left half using diagonal-norm SBP operators for finite differencing, and the right half using a discontinuous Galerkin method. The interior order of the SBP method, as well as the order of the projection across the interface, is specified to match the order of the discontinous Galerkin method.

Our success metrics for this test problem include

- Demonstrating accurate recreation of the total energy of the system, as observed in the numerical simulation of the problem, via a total energy relation derived from the spatial and interfacial schema.
- Demonstrating that this energy does not increase (condition for semi-discrete stability).
- Demonstrating the prescribed order of accuracy using L2 norms.

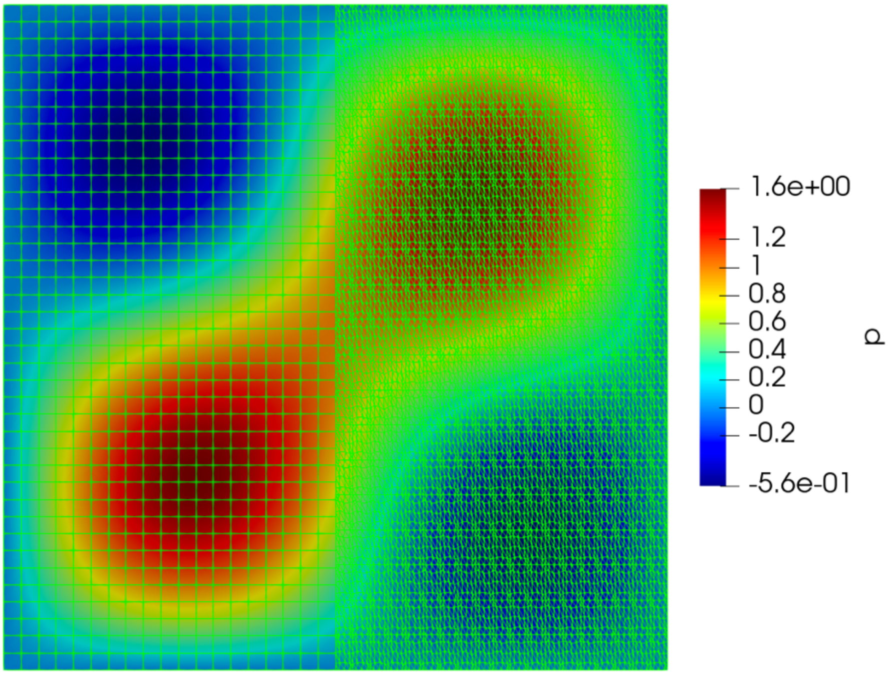

Figure 3: Pressure initial condition for test problem, also showing split-domain discretization.

Upon fulfillment of these goals, extension to three dimensions will be considered.

2.5 Results

(Insert text here)

2.6 Outlook

2.6.1 Current Status

2.6.2 Risk Mitigation

References

[1] Francis Bashforth and John Couch Adams. *An attempt to test the theories of capillary action.* University Press, 1883.

[2] William Blumen. Shear layer instability of an inviscid compressible fluid. *Journal of Fluid Mechanics*, 40(4):769–781, 1970.

[3] Daniel J Bodony. Accuracy of the simultaneous-approximation-term boundary con-

dition for time-dependent problems. *Journal of Scientific Computing*, 43(1):118–133, 2010.

[4] Mark H Carpenter, David Gottlieb, and Saul Abarbanel. Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. Technical report, NASA, 1993.

[5] Charles William Gear and DR Wells. Multirate linear multistep methods. *BIT Numerical Mathematics*, 24(4):484–502, 1984.

[6] Karl Heun. Neue methoden zur approximativen integration der differentialgleichungen einer unabhängigen veränderlichen. *Z. Math. Phys*, 45:23–38, 1900.

[7] Andreas Klöckner. *High-performance high-order simulation of wave and plasma phenomena*. PhD thesis, Brown University, 2010.

[8] Jeremy E Kozdon and Lucas C Wilcox. Stable coupling of nonconforming, high-order finite difference methods. *SIAM Journal on Scientific Computing*, 38(2):A923–A952, 2016.

[9] Wilhelm Kutta. Beitrag zur näherungweisen integration totaler differentialgleichungen. 1901.

[10] Ken Mattsson, Magnus Svärd, and Jan Nordström. Stable and accurate artificial dissipation. *Journal of Scientific Computing*, 21(1):57–79, 2004.

[11] Bonnie J McBride. *NASA Glenn coefficients for calculating thermodynamic properties of individual species*. National Aeronautics and Space Administration, John H. Glenn Research Center ..., 2002.

[12] Alfons Michalke. On the inviscid instability of the hyperbolictangent velocity profile. *Journal of Fluid Mechanics*, 19(4):543–556, 1964.

[13] Cory Mikida, Andreas Klöckner, and Daniel Bodony. Multi-rate time integration on overset meshes. *Journal of Computational Physics*, 396:325–346, 2019.

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

[15] Magnus Svärd, Mark H Carpenter, and Jan Nordström. A stable high-order finite difference scheme for the compressible Navier–Stokes equations, far-field boundary conditions. *Journal of Computational Physics*, 225(1):1020–1038, 2007.

[16] Magnus Svärd and Jan Nordström. A stable high-order finite difference scheme for the compressible Navier–Stokes equations: no-slip wall boundary conditions. *Journal of Computational Physics*, 227(10):4805–4824, 2008.

[17] Bao Zhu, Jiefu Chen, Wanxie Zhong, and Qing Huo Liu. A hybrid fetd-fdtd method with nonconforming meshes. *Communications in Computational Physics*, 9(3):828–842, 2011.