AIAA SciTech
4-8 January 2016, San Diego, California, USA
54th AIAA Aerospace Sciences Meeting

AIAA 2016-0066

The 54th AIAA Aerospace Sciences Meeting, 4-8 January 2016, San Diego, California

Towards Efficient Parallel-in-Time Simulation of Periodic Flows

J. Leffell[1], J. Sitaraman[2], V. Lakshminarayan[1] and A. Wissink[3]

[1] *Science & Technology Corporation, NASA Ames Research Center, Moffett Field, CA*
[2] *Parallel Geometric Algorithms, LLC, Sunnyvale, CA*
[3] *U.S. Army Aviation Development Directorate (AMRDEC), Moffett Field, CA*

In response to the stagnation of computer microprocessor speeds over the past decade, the design emphasis of novel supercomputing architectures has focused primarily on increasing the overall number of available cores and reducing communication bottlenecks. Typically, flow solvers have been able to achieve parallel efficiency using domain decomposition, but this approach has the natural limitation that saturation will manifest itself on a finite number of cores at which point parallel speedup stalls and eventually deteriorates. In order to improve parallel scalability we seek to leverage the existing knowledge base on spatial decomposition, while attempting to exploit additional parallelism in the temporal dimension. Specifically, we explore the case of time periodic flows using Parallel-in-Time (PinT) variants of the Time-Spectral (TS) method. A framework employing a Python-based infrastructure is described including a standalone library that can be coupled to existing flow solvers in order to facilitate PinT calculations. A model problem of a periodic density pulse is used to examine the different discretization options. Implications for application to wind turbines and rotors are addressed.

I. Introduction

Advances in computing technology have facilitated the solution of problems in computational fluid dynamics (CFD) that would have seemed unfathomable mere decades ago. The advent of parallel computing in combination with seemingly ever-increasing processor speeds suggested that problems of extreme size and complexity could eventually be solved using traditional algorithms. A sustained and consistent increase in transistor density since the middle of the twentieth century led to exponential growth of processor clock speeds, but a natural limit has been reached. In response to the stagnation of clock speeds, the design emphasis of novel supercomputing architectures has focused primarily on increasing the overall number of available cores and reducing communication bottlenecks. Typically, CFD flow solvers have been able to achieve parallel efficiency using domain decomposition, but this approach has the natural limitation that saturation will manifest itself on a finite number of cores at which point parallel speedup stalls and eventually deteriorates; much effort has focused on optimizing algorithms for minimum operation count, which exacerbates the issue by increasing the communication quotient of the overall budget. Exascale computing is rapidly approaching, so we are tasked with improving scalability on the next generation of hardware. In order to accomplish this objective we seek to leverage the existing knowledge base on spatial decomposition, while attempting to exploit additional parallelism in the temporal dimension to improve speedup where parallelism in space alone is no longer viable.

While it is often assumed that integration of time-dependent PDEs must necessarily be executed sequentially, there is a rather long and rich history of techniques focused on parallel time integration going back to Nievergelt [1] in 1964. Other work in this field includes [2–22]. Gander [23] offers a comprehensive survey of such schemes and the reader is directed to the cited reference for additional background. Recent advancements in Parallel-in-Time (PinT) methods have demonstrated great scalability on an exceptionally large number of cores [24], but it remains to be seen how well these approaches translate to aerodynamic flows. The initial objective of this work involves developing a framework to facilitate highly scalable PinT

---

calculations and determine what types of problems are best suited for such treatment. Ultimately, we seek to incorporate parallel time integration within high-fidelity production solvers and leverage the vast computational resources expected with the impending arrival of Exascale hardware to dramatically reduce compute times for realistic problems involving rotors and turbines.

Time-periodic flows are particularly amenable to PinT time integration as they are invariant in the frequency domain and can therefore be treated as a steady-state problem in space-time. Fourier pseudospectral methods were developed to exploit this fact by capitalizing on the spectral resolution properties of the Fourier series to provide accurate solutions using a limited number of temporal degrees of freedom. In such calculations, the instantaneous solution at N time samples over the period are solved simultaneously by approximating the temporal derivative with an infinitely-supported differentiation operator derived from the discrete Fourier transform. Mavriplis and Yang [25] distributed the time instances to different processors providing temporal parallelism in addition to spatial parallelism. The infinite support of the differentiation operator requires significant communication between the solution at the various time instances which can stifle the efficiency of the PinT Fourier-based methods. Therefore, alternative PinT schemes for periodic flows are being explored with the goal of providing solutions in a fraction of the time taken by tradition sequential time-advancement schemes.

Aerodynamic calculations often involve a number of complexities such as relative motion between components, turbulence modeling and irregular meshes. We have elected to forgo such complications and initiate this work on fixed grids using a lightweight Cartesian flow solver that was developed as a testbed for novel numerical schemes. We will explore Parallel-in-Time time integration for the case of time periodic flows using variants of the Time-Spectral method. We first apply the PinT method to a wind turbine problem where the rotor blades are modeled using an actuator line module. Based on the flow signature, we analyze an even simpler problem consisting of a periodic density pulse to characterize accuracy and scalability of various periodic PinT approaches.

Following an introduction of the compressible Navier-Stokes equations in §II, the paper continues with a description of the spatial (§III) and temporal (§IV) discretization options available in the solver framework. The various modules of the framework are outlined in §V including its Python-based infrastructure, the Cartesian flow solver, the standalone PinT library, the domain connectivity module and the actuator line model that provides relevant test cases without the need to handle moving geometry with dynamic hole cutting. The aforementioned numerical experiments are presented in §VI.

II. Governing Equations

The Navier-Stokes equations are statements of the conservation of mass, momentum and energy that can be expressed in Cartesian coordinates in strong conservation-law form.

$$\frac{\partial q}{\partial t} + \nabla \cdot \mathbf{F}\left(q\right) = 0 \tag{1}$$

The state vector, $q = \left\{\rho, \rho\mathbf{u}, e\right\}^T$, consists of the conserved quantities of mass, momentum, and energy in three spatial dimensions. The flux vector, $\mathbf{F}$, is decomposed into its convective and viscous constituents, $\mathbf{F} = \mathbf{F}^c - \mathbf{F}^v$.

$$q = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ e \end{bmatrix}, \qquad F_i^c = \begin{bmatrix} \rho u_i \\ \rho u_1 u_i + \delta_{i1} p \\ \rho u_2 u_i + \delta_{i2} p \\ \rho u_3 u_i + \delta_{i3} p \\ \left(e + p\right) u_i \end{bmatrix}, \qquad F_i^v = \begin{bmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ u_j \tau_{ij} + k_i \end{bmatrix}$$

The viscous stress tensor, $\boldsymbol{\tau}$, and heat flux, $\mathbf{k}$, are defined as

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \lambda \delta_{ij} \frac{\partial u_k}{\partial x_k}$$
$$k_i = \kappa \frac{\partial T}{\partial x_i} = \frac{\gamma}{\gamma - 1} \frac{\mu}{Pr} \frac{\partial}{\partial x_i} \left(\frac{p}{\rho}\right)$$

where the absolute viscosity, μ, is defined as a function of temperature, T, according to Sutherland's law. The equation of state for an ideal gas closes the system.

$$p = (\gamma - 1)\left[e - \frac{1}{2}\rho\,(u_i u_i)\right]$$

The ratio of specific heats, γ, is taken to be 1.4, the Prandtl number, Pr, is fixed at 0.71 and the speed of sound is defined as $a = \sqrt{\gamma p/\rho}$.

III. Spatial Discretization

A high-order Cartesian flow solver is used as the main testbed for all of the numerical methods investigated as part of this work. The Cartesian formulation provides an efficient solution process and the ability to easily implement high-order schemes. The solver is equipped with up to sixth-order accurate finite differences of the inviscid flux, $\mathbf{F}^c = \{E, F, G\}$, in combination with first-, third- or fifth-order accurate artificial dissipation terms. Additionally, second- and fourth-order accurate viscous flux discretizations have been implemented. Insight gained applying Parallel-in-Time methods to the Cartesian solver should directly translate to other types of discretizations and mesh topologies.

The inviscid finite-difference spatial discretizations are expressed in pseudo-finite-volume form.

$$\frac{\partial E}{\partial x} \approx \frac{\hat{E}_{i+1/2} - \hat{E}_{i-1/2}}{\Delta x} \tag{2}$$

The total inviscid flux evaluated at the cell-face, $\hat{E}$, is a combination of the the physical flux term, $\tilde{E}$, and the artificial dissipation term, $\tilde{D}$.

$$\hat{E}_{i+1/2} = \tilde{E}_{i+1/2} - \tilde{D}_{i+1/2} \tag{3}$$

The physical fluxes corresponding to central differences of second-, fourth- and sixth-order accuracy are defined as:

$$\begin{aligned}
\tilde{E}^{II}_{i+1/2} &= \frac{1}{2}\left(E_{i+1} + E_i\right) && (4)\\
\tilde{E}^{IV}_{i+1/2} &= \tilde{E}^{II}_{i+1/2} + \frac{1}{12}\left(-E_{i+2} + E_{i+1} + E_i - E_{i-1}\right) && (5)\\
\tilde{E}^{VI}_{i+1/2} &= \tilde{E}^{IV}_{i+1/2} + \frac{1}{60}\left(E_{i+3} - 3E_{i+2} + 2E_{i+1} + 2E_i - 3E_{i-1} + E_{i-2}\right) && (6)
\end{aligned}$$

The second-, fourth- and sixth-difference dissipation terms, which are first-, third- and fifth-order accurate, respectively, are defined as:

$$\begin{aligned}
\tilde{D}^{II}_{i+1/2} &= \frac{|\sigma|_{i+1/2}\Delta x}{2}\left.\frac{\partial q}{\partial x}\right|_{i+1/2} && (7)\\
\tilde{D}^{IV}_{i+1/2} &= -\frac{|\sigma|_{i+1/2}\Delta x^3}{12}\left.\frac{\partial^3 q}{\partial x^3}\right|_{i+1/2} && (8)\\
\tilde{D}^{VI}_{i+1/2} &= \frac{|\sigma|_{i+1/2}\Delta x^5}{60}\left.\frac{\partial^5 q}{\partial x^5}\right|_{i+1/2} && (9)
\end{aligned}$$

where σ is the spectral radius of the inviscid flux Jacobian scaled by the dissipation coefficient. The discrete form of these derivatives are defined as:

$$\begin{aligned}
\tilde{D}^{II}_{i+1/2} &= \frac{|\sigma|_{i+1/2}}{2}\left(q_{i+1} - q_i\right) && (10)\\
\tilde{D}^{IV}_{i+1/2} &= \tilde{D}^{II}_{i+1/2} - \frac{|\sigma|_{i+1/2}}{12}\left(q_{i+2} + 3q_{i+1} - 3q_i - q_{i-1}\right) && (11)\\
\tilde{D}^{VI}_{i+1/2} &= \tilde{D}^{IV}_{i+1/2} + \frac{|\sigma|_{i+1/2}}{60}\left(q_{i+3} - 5q_{i+1} + 5q_i - q_{i-2}\right) && (12)
\end{aligned}$$

Spatial discretization reduces the governing Navier-Stokes equations to an ODE form denoted by

$$\frac{d}{dt}q + \mathbf{R}\left(q\right) = 0 \tag{13}$$

where $\mathbf{R}\left(q\right)$ is the discrete spatial residual.

Standard practice involves combining fourth-difference dissipation with either second- or fourth-order accurate inviscid flux differences and sixth-difference dissipation with sixth-order accurate inviscid flux differencing resulting in schemes that are overall second-, third- and fifth-order accurate, respectively. Two fringe layers are required for each block using second- and third-order accurate (i.e. second- and fourth-order central differencing plus fourth-difference dissipation) schemes and a third fringe layer is required for the fifth-order accurate (sixth-order central differencing plus sixth-difference dissipation) scheme.

All of the numerical results presented in §VI employ second-order accurate inviscid flux differencing with third-order accurate artificial dissipation and second-order accurate viscous flux differencing (where applicable).

IV. Temporal Discretization

The conventional approach for evolving the solution of time-dependent discretized PDEs involves what will herein be referred to as Sequential- or Serial-in-Time (SinT) time integration. In this approach, the solution is advanced forward in time by a single physical time step in an iterative manner until the desired final time is reached. The most common method for integration of unsteady PDEs has been a single step scheme such as the first-order explicit forward Euler or the implicit backward Euler methods. Higher accuracy can be achieved using additional time samples, as is the case for linear multistep methods (the most popular choice being the second-order BDF2 scheme) or multi-stage methods such as explicit or implicit Runge-Kutta methods.

Other solution methods are available for the case of periodic flows that take advantage of the steady nature of such flows in the frequency domain. Fourier pseudospectral methods have demonstrated marked success in reducing the computational costs associated with simulating periodic flows [26]. The trigonometric representation of periodic phenomenon provides spectral convergence as the number of time-samples, and correspondingly, the number of resolvable harmonics increases. The spectral convergence rate of Fourier methods is superior to the algebraic convergence rates associated with traditional time-marching schemes for unsteady calculations, implying that a given level of accuracy can be achieved with significantly fewer degrees of freedom [27]. Using this approach, the solutions at N equispaced time instances are coupled through the temporal derivative term and solved simultaneously. Mavriplis and Yang [25] elected to distribute the N time instances to different processors to achieve fully parallel (space and time) calculations. We also follow a parallel space-time approach in the same vein. Additionally, to mitigate the large communication requirements for the parallel Time-Spectral scheme, an algebraically-accurate analog using a circulant central difference operator is introduced with goal of achieving improved scalability.

IV.A. Explicit Runge-Kutta Method

Runge-Kutta methods provide high-order accurate explicit and implicit time-integration. The default explicit time-marching scheme is the low-storage three-stage Runge-Kutta (RK3) scheme proposed by Wray [28]. The scheme is given as follows:

$$\begin{aligned}
q^{p} &= q^{n} + \frac{\alpha_1}{\Delta t}\mathbf{R}(q^{n}) \tag{14}\\
q^{*} &= q^{n} + \frac{\alpha_2}{\Delta t}\mathbf{R}(q^{n}) \tag{15}\\
q^{**} &= q^{p} + \frac{\alpha_3}{\Delta t}\mathbf{R}(q^{*}) \tag{16}\\
q^{n+1} &= q^{p} + \frac{\alpha_4}{\Delta t}\mathbf{R}(q^{**}) \tag{17}
\end{aligned}$$

The coefficients $[\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ are equal to $[\frac{1}{4}, \frac{8}{15}, \frac{5}{12}, \frac{3}{4}]$ respectively. This scheme requires only two storage locations for the field variables: one that can be reused for q^n, q^p and q^{n+1} and another that can be reused for q^* and q^{**}.

IV.B. Backward Difference Formula (BDF)

Implicit schemes are often desirable for stiff systems that would impose a prohibitively small time step restriction on explicit discretizations. One of the most popular implicit time-marching methods is the family of Backward Difference Formula (BDF) schemes. The first-order backward difference scheme (BDF1) only depends on the solution at the current time step.

$$\frac{q^{n+1}-q^{n}}{\Delta t}+\mathbf{R}\left(q^{n+1}\right)=0 \tag{18}$$

The second-order backward difference scheme (BDF2) relies additionally on the solution at the previous time step (BDF1 is often used as a startup procedure for BDF2).

$$\frac{3q^{n+1}-4q^{n}+q^{n-1}}{2\Delta t}+\mathbf{R}\left(q^{n+1}\right)=0 \tag{19}$$

Typically we employ a dual time-stepping procedure for BDF schemes [29] to ensure sufficient reduction of the unsteady residual at each physical time step. This is accomplished by adding a pseudotime continuation term and advancing the solution such that the subiterate $s+1$ converges sufficiently within each physical time step.

$$\frac{q^{s+1}-q^{s}}{\Delta\tau}+\frac{3q^{s+1}-4q^{n}+q^{n-1}}{2\Delta t}+\mathbf{R}\left(q^{s+1}\right)=0 \tag{20}$$

In order to solve this nonlinear system of equations, we linearize the residual about the state vector at subiteration s and manipulate the equations into *delta*-form,

$$\left[I+\Delta\tilde{\tau}\left.\frac{\partial\mathbf{R}}{\partial q}\right|_{s}\right]\Delta q=-\Delta\tilde{\tau}\left[\frac{3q^{s}-4q^{n}+q^{n-1}}{2\Delta t}+\mathbf{R}\left(q^{s}\right)\right] \tag{21}$$

where $\Delta q=q^{s+1}-q^{s}$ and $\Delta\tilde{\tau}=\Delta\tau/\left(1+\frac{3\Delta\tau}{2\Delta t}\right)$.

Starting from an initial condition, the solution is updated over a number of subiterations using one of the implicit operators described in §V.A. The process continues until a specified number of subiterations, S, has been computed or until a specified reduction in the unsteady residual is achieved (typically two or more orders of magnitude). The solution is then assigned to q^{n+1} and the algorithm proceeds using the solution from the preceding time step as the initial condition for the subsequent time step. BDF2 is the default sequential time-advancement scheme.

IV.C. Time-Spectral Method

Forced periodic flows arise in a broad range of aerodynamic applications such as rotorcraft, turbomachinery, and flapping-wing configurations. The standard procedure for simulating such flows involves advancing the unsteady governing equations forward in time long enough for the initial transient to exit the computational domain and for a statistically stationary flow to be achieved. It is often necessary to simulate several periods of motion to accomplish this task, making unsteady design optimization prohibitively expensive for many realistic problems. An effort to reduce the computational cost of these calculations led to the development of the Harmonic Balance (HB) method by Hall et al. [30, 31], which capitalizes on the periodic nature of the solution, while maintaining the ability to resolve nonlinearities inherent in the underlying physics. This approach exploits the fact that time-periodic flow, while varying in the time domain, is invariant in the frequency domain. Expanding the temporal variation at each spatial node as a truncated Fourier series transforms the unsteady governing equations into a coupled set of steady equations in integer harmonics that can be tackled with the acceleration techniques afforded to steady-state flow solvers. Other similar Fourier pseudospectral approaches, such as the Non Linear Frequency Domain (NLFD) [26,32,33], Reduced Frequency [34], and Time-Spectral (TS) [35–37] methods, were also developed. These methods can be applied both to flows where the fundamental frequency is known a priori (e.g. rotorcraft, turbomachinery, etc.) or not (e.g. vortex shedding off of a bluff body)[1] and a recent extension by Mavriplis and Yang and Yang et al.

---

[1] McMullen [38] developed a gradient-based approach for Fourier pseudospectral schemes to determine the response frequency if unknown or not assumed a priori.

[25, 39] provides treatment for quasi-periodic flows characterized by strong periodic content in addition to a slowly varying mean (e.g. rotorcraft undergoing a maneuver). The same work introduced the utilization of an additional dimension of parallelization by instantiating the various time samples on different processors, resulting in a PinT Time-Spectral approach that we follow here. There have also been efforts to relax the infinite support of these methods by employing finite elements in time [40]. Here, the solution is expanded within time slabs (no longer restricted to periodic or quasi periodic problems) and coupled to the previous time slabs and solved sequentially. Temporal parallelism is still achieved between the degrees of freedom within each time slab.

We are investigating a more basic approach to improve the parallel scalability of the traditional Time-Spectral method that forfeits the spectrally-accurate, yet dense differentiation operator, in favor of an algebraically accurate operator with a smaller bandwidth that is independent of N; this approach is currently restricted to periodic flows.

Standard Time-Spectral Method

The standard Time-Spectral method is derived as a Fourier collocation scheme [25,41] for a time-periodic solution, q, starting from the semi-discretized governing equations.

$$\frac{d}{dt}q + \mathbf{R}\left(q\right) = 0 \tag{22}$$

Since the solution is assumed a priori to be periodic in time, such that $q\left(t+T\right) = q\left(t\right)$, it can be expressed as a Fourier series whose basis functions, $\phi_k\left(t\right) = e^{i\omega k t}$, are the complex exponentials in integer harmonics of the fundamental frequency, $\omega = 2\pi/T$.

$$q\left(t\right) = \sum_{k=-\infty}^{\infty} \tilde{q}_k \phi_k\left(t\right) \tag{23}$$

A discrete approximation is required, and therefore the Fourier series is truncated to N terms which retains K harmonics of the fundamental frequency,

$$q\left(t\right) \approx \sum_{k\in\mathcal{K}} \tilde{q}_k \phi_k\left(t\right) \tag{24}$$

where

$$K = \begin{cases} (N-1)/2, & N \text{ odd} \\ N/2 - 1, & N \text{ even} \end{cases} \tag{25}$$

with the following set of integer harmonics, $\mathcal{K}$.

$$\mathcal{K} = \begin{cases} \{-K, \ldots, K\}, & N \text{ odd} \\ \{-1-K, \ldots, K\}, & N \text{ even} \end{cases} \tag{26}$$

The method of weighted residuals is applied to approximate the solution to PDEs by minimizing a discrete residual, R_N, of Eq. 22.

$$R_N\left(t\right) = \frac{\partial}{\partial t}q\left(t\right) + \mathbf{R}\left(q\left(t\right)\right) \tag{27}$$

It requires that R_N integrates to zero against an appropriate set of test functions, ψ, over the period [42].

$$\left(R_N, \psi_n\right)_w = \int_0^T R_N\, \psi_n\, w\, dt = 0, \quad \forall n \in \{0, \ldots, N-1\} \tag{28}$$

The choice of weight and test functions, $w = 1$ and $\psi_n = \delta\left(t - t_n\right)$, respectively, defines a collocation scheme as it eliminates the discrete residual at each of the collocation points, t_n, where $q^n = q\left(t_n\right)$.

$$\frac{d}{dt}q^n + \mathbf{R}\left(q^n\right) = 0, \quad \forall n \in \{0, \ldots, N-1\}, \tag{29}$$

Because the temporal basis functions are the complex exponentials, the collocation points are distributed uniformly over the period, $t_n = nT/N$.

The temporal derivative in Eq. 29 is evaluated by analytically differentiating the truncated Fourier series in Eq. 24

$$\frac{d}{dt} q\left(t\right) = \sum_{k \in \mathcal{K}} i \omega k \tilde{q}_k e^{i \omega k t} \tag{30}$$

where $i = \sqrt{-1}$. The Fourier coefficients, $\tilde{q}_k$, are determined from the solution values at the N collocation points by applying the *discrete* Fourier transform.

$$\tilde{q}_k = \frac{1}{N} \sum_{n=0}^{N-1} q^n e^{-i \omega k t_n} \tag{31}$$

While this results in a pseudospectral method, it enables spatial operators from existing flow solvers to be leveraged directly because the spatial residual operator, $\mathbf{R}\left(q\right)$, is evaluated in the time domain (in contrast to *pure* spectral methods).

Substituting the definition of $\tilde{q}_k$ from Eq. 31 into Eq. 30 defines the temporal derivative at a particular time sample, t_n, as a weighted sum of the solution at every other time sample ($d_n^n = 0$).

$$\begin{aligned} \frac{d}{dt} q^n &= \sum_{k \in \mathcal{K}} i \omega k \, \tilde{q}_k e^{i \omega k t_n} && (32) \\ &= \sum_{j=0}^{N-1} \left(\frac{1}{N} \sum_{k \in \mathcal{K}} i \omega k e^{i \omega k \left(t_n - t_j\right)} \right) q^j && (33) \\ &= \sum_{j=0}^{N-1} d_j^n q^j && (34) \end{aligned}$$

Defining the time derivative at each collocation point and collecting the expressions into matrix form arrives at the definition of the *Fourier interpolation* derivative operator [43]; the time-domain temporal differentiation operator, D_N, acts on the time-history of the solution at a particular point in space, $\mathbf{x}$, $\mathbf{q}\left(\mathbf{x}\right) = \left\{ q\left(\mathbf{x}, t_0\right), \ldots, q\left(\mathbf{x}, t_{N-1}\right) \right\}^T$.

$$\frac{d}{dt} \mathbf{q}\left(\mathbf{x}\right) = D_N \mathbf{q}\left(\mathbf{x}\right) \tag{35}$$

The elements, d_j^n, of the temporal differentiation operator, D_N, are defined as follows [36]:

$$d_j^n = \left\{ \begin{array}{rl} \frac{2\pi}{T} \frac{1}{2} \left(-1\right)^{n-j} \Phi\left(\frac{\pi(n-j)}{N}\right), & n \neq j \\ 0, & n = j \end{array} \right. \tag{36}$$

where $\Phi\left(\theta\right) = \csc\left(\theta\right)$ for N odd and $\Phi\left(\theta\right) = \cot\left(\theta\right)$ for N even. Replacing the analytic temporal derivative in Eq. 29 with the discrete approximation in Eq. 34 results in the global system of equations.

$$\sum_{j=0}^{N-1} d_j^n q^j + \mathbf{R}\left(q^n\right) = 0, \quad \forall n \in \left\{0, \ldots, N-1\right\} \tag{37}$$

We can apply pseudotime continuation to Eq. 37 to drive the system of equations towards a converged discrete solution.

$$\frac{d}{d\tau} q^n + \sum_{j=0}^{N-1} d_j^n q^j + \mathbf{R}\left(q^n\right) = 0, \quad \forall n \in \left\{0, \ldots, N-1\right\} \tag{38}$$

Equation 38 is iterated in pseudotime until satisfactory convergence is achieved. The solution can then be reconstructed at any point in continuous time by applying Eq. 24 to the converged Fourier coefficients determined by the DFT (Equation 31).

As mentioned above, Mavriplis and Yang [25] and Yang et al. [39] extended the scope of the Time-Spectral method to include quasi-periodic flows by resolving a slowly-varying mean flow in addition to the strong periodic content. The method is derived via polynomial subtraction and includes a relatively low-order polynomial representation of the solution as it moves from one period to the next; the spectrally-accurate representation of the original TS approach is used to capture the dominant periodic content. The period of oscillation is treated as a large physical time step which is used for the BDF approximation and N sub-steps are used within each physical time step to resolve the periodic content. We intend to incorporate this hybrid BDF/TS scheme in the future.

Finite Difference Method in Time (FDMT)

The Fourier interpolation differentiation operator described in the previous section has the property of being an optimally-accurate central difference operator with a bandwidth equivalent to one half the number of time samples, i.e. a dense matrix. This provides spectrally-accurate differentiation, but requires data from every time sample due to the infinite support of the complex exponential basis functions from which it is derived. With a goal of using time-parallelism to reduce the wall time by a factor of a hundred or a thousand, we expect to use between 100 to 1000 time samples or more; using a spectrally-accurate operator is perhaps overkill in this scenario and may be prohibitively expensive. Thus, alternative approaches that abandon infinite support of the Fourier representation are being investigated.

The most straightforward approach in this effort retains central differencing while reducing the bandwidth of the temporal differentiation operator. This is accomplished by employing circulant 2nd-, 4th-,..., pth-order accurate central difference operators in time. For the standard Time-Spectral method, the temporal differentiation operator, D_N, is dense but the temporal differentiation operator for the finite-difference approach, D_N^p has a bandwidth of only $M = p/2$. Thus, the temporal derivative at a particular time sample is expressed as a weighted sum of the solution of the M nearest neighbors.

$$\frac{d}{dt}q^n = \sum_{j=1}^{M} d_j^p \left(q^{n+j} - q^{n-j}\right) \tag{39}$$

The coefficients for the second-, fourth- and eighth-order accurate Finite-Difference Method in Time (FDMT-p) schemes are defined as follows:

$$\begin{aligned} d^{II} &= \frac{\omega N}{2\pi}\left\{\frac{1}{2}\right\} \\ d^{IV} &= \frac{\omega N}{2\pi}\left\{\frac{2}{3}, -\frac{1}{12}\right\} \\ d^{VIII} &= \frac{\omega N}{2\pi}\left\{\frac{4}{5}, -\frac{1}{5}, \frac{4}{105}, -\frac{1}{280}\right\} \end{aligned}$$

where $2\pi/\omega N = T/N = \Delta t$.

In this approach, the temporal communication is limited to the nearest M neighbors for an operator of accuracy $p = 2M$. Thus, as the number of time-steps per period, N, grows, we expect the problem to scale weakly on N processors as the communication footprint is fixed, unlike in the Time-Spectral approach where the bandwidth, and correspondingly the communication burden, grows linearly with N. Figure 1 demonstrates the sparsity, or lack thereof, of the temporal differentiation operators for the Time-Spectral method and a second-order periodic central difference approach (FDMT-2). Naturally, the accuracy per degree of freedom is reduced in the case of finite differences, but the extreme reduction of temporal coupling has powerful advantages in the context of temporal parallelism; a weakly-scaling compactly-supported approach, provided with the ability to leverage a large numbers of cores, will hopefully enable PinT calculations of periodic flows with the required resolution[2] in significantly reduced wall time with respect to serial time integration.

The TS operator is purely imaginary so we have focused initially on undamped central difference operators; it has been shown that temporal dissipation has been successful for TS applications [44–46] so finite-difference-based temporal artificial dissipation will be explored as part of future work in addition to one-sided

---

[2] State-of-the-art rotorcraft calculations typically employ second- to fourth-order accurate temporal discretizations using 0.25° time steps ($\Delta t = T/1440$).

circulant differentiation operators (e.g. BDF2). Consequently, application of a compactly-supported differentiation operator also alleviates the difficulties associated with applying the TS method within the overset grid approach, where complete time-histories of the solution may not be universally defined [45, 47].

V. Solver Framework

Ultimately we are interested in applying temporal parallelism to full-scale, high-fidelity solvers such as the CREATE-AV$^{\text{TM}}$ Helios package which is the state-of-the-art rotorcraft simulation code. However, in the interest of rapid development and testing, we have elected to forgo the complexities associated with such codes and focus instead on a simplified analog employing a Python-based infrastructure similar to the production version. Worthwhile advances realized with the research code will be incorporated into Helios in subsequent releases.

There are several elements that comprise the solver framework including the base Cartesian flow solver, a standalone PinT library for periodic flows, an actuator line module enabling rotor simulations without requiring moving geometry, a domain connectivity module to connect Cartesian blocks of different refinement levels and the high-level Python-based infrastructure to facilitate data transfer between the different modules and control the advancement of the overall solution procedure.

V.A. Cartesian Flow Solver

The sole flow solver currently available in the framework is a lightweight, efficient and scalable high-order Navier-Stokes solver (See Figure 2) with both explicit and implicit time-marching schemes. The grid system consists of a hierarchy of Cartesian meshes using fixed refinement between levels. Each level is partitioned into an appropriate number of blocks; communication between blocks is handled internally but communication between grid levels in handled by a separate domain connectivity module (See §V.D). Several available linear solvers include Lower-Upper Symmetric Gauss-Seidel (LU-SGS), Alternating Direction Implicit (ADI), Diagonally-Dominant ADI (DDADI), Gauss-Seidel Line Relaxation (GSLR) schemes and a point-wise Gauss-Seidel (GSP) scheme with concurrent data communication between blocks at the linear iteration level. A GMRES solver has also been implemented that applies the aforementioned approximate-factorization schemes to precondition the resulting linear system.

Implicit Operators

Newton-linearization of the discrete form of the governing PDE, $\mathbf{R}(q) = 0$ (e.g. the right-hand side of Eq. 21), can be written as

$$\left[\frac{I}{\Delta\tau} + \frac{\partial \mathbf{R}}{\partial q}\right]\Delta q = \left[\frac{I}{\Delta\tau} + \frac{\partial R^X}{\partial q} + \frac{\partial R^Y}{\partial q} + \frac{\partial R^Z}{\partial q}\right]\Delta q = -\mathbf{R}(q^s) \tag{40}$$

For efficiency, the implicit operator (LHS of Eq. 40) is formulated using a lower order formulation of the interface fluxes and their associated Jacobians. The second order interface flux formulation is given by

$$\hat{F}_{x+\frac{1}{2}} = \frac{1}{2}\left(F_x + F_{x+1} - D_{x+\frac{1}{2}}(q_{x+1} - q_x)\right), \tag{41}$$

where F represents physical fluxes and D represents the dissipation. On linearization, this flux function yields operators that can be written as:

$$\begin{aligned}
\frac{\partial R^x}{\partial q}\Delta q_{jkl} &= [\mathbf{A}_x]\Delta q_{x-1} + [\mathbf{B}_x]\Delta q_{jkl} + [\mathbf{C}_x]\Delta q_{x+1} \qquad (42)\\
\mathbf{A}_x &= -\frac{1}{2}\left(\frac{\partial \mathbf{F}_{x-1}}{\partial q_{x-1}} + D_{x-\frac{1}{2}}\right)\\
\mathbf{B}_x &= \frac{1}{2}(D_{x-\frac{1}{2}} + D_{x+\frac{1}{2}})\\
\mathbf{C}_x &= \frac{1}{2}\left(\frac{\partial \mathbf{F}_{x+1}}{\partial q_{x+1}} - D_{x+\frac{1}{2}}\right)
\end{aligned}$$

where x denotes each of the three spatial directions, X, Y and Z, and their indices j, k, and l, respectively.

In the present work, three line-based techniques are used, namely the approximately-factored alternating direction implicit (ADI) and diagonally-dominant ADI (DDADI) schemes and a Gauss-Seidel Line Relaxation (GSLR) scheme. (GSLR). For the ADI scheme, the implicit operator in Eq. 40 is multiplied by the pseudo time step $\Delta\tau$ and factorized in each of the coordinate directions and expressed as:

$$\left[I + \Delta\tau\frac{\partial R^X}{\partial q}\right]\left[I + \Delta\tau\frac{\partial R^Y}{\partial q}\right]\left[I + \Delta\tau\frac{\partial R^Z}{\partial q}\right]\Delta q = -\Delta\tau\mathbf{R}(q^s) \tag{43}$$

Each of the above operators have a block tridiagonal structure, which are inverted directly using a variant of the popular Thomas algorithm. In the case of DDADI, the factorized system is written such that there is a more diagonally dominant term that aids in solution convergence and adds to the numerical stability of the scheme when compared to the ADI scheme. The DDADI factored scheme is

$$(\mathbf{D} + \mathbf{O}_X)\mathbf{D}^{-1}(\mathbf{D} + \mathbf{O}_Y)\mathbf{D}^{-1}(\mathbf{D} + \mathbf{O}_Z)\Delta q = -\Delta\tau\mathbf{R}(q^s) \tag{44}$$

where the diagonal term $\mathbf{D}$ and the $\mathbf{O}$ matrices are given by

$$\begin{aligned}
\mathbf{D} &= [I + \Delta\tau\left([\mathbf{B}_j] + [\mathbf{B}_k] + [\mathbf{B}_l]\right)] \\
\mathbf{O}_X &= \Delta\tau([\mathbf{A}_j],\ 0,\ [\mathbf{C}_j]) \\
\mathbf{O}_Y &= \Delta\tau([\mathbf{A}_k],\ 0,\ [\mathbf{C}_k]) \\
\mathbf{O}_Z &= \Delta\tau([\mathbf{A}_l],\ 0,\ [\mathbf{C}_l])
\end{aligned}$$

For the GSLR scheme a Gauss-Seidel sweep is performed on a line basis, i.e. implicit inversion is performed for each line and the changes in the conservative variables (Δq) are updated to the right-hand-side of subsequent lines as soon as a new update is available, i.e.

$$\begin{aligned}
[\mathbf{A}_j]\Delta q_{j-1} + [\mathbf{D}]\Delta q_{jkl} + [\mathbf{C}_j]\Delta q_{j+1} = &-\mathbf{R}(q^s) \\
&- [\mathbf{A}_k]\Delta q^*_{k-1} - [\mathbf{C}_k]\Delta q^*_{k+1} \\
&- [\mathbf{A}_l]\Delta q^*_{l-1} - [\mathbf{C}_l]\Delta q^*_{l+1}
\end{aligned} \tag{45}$$

with the diagonal matrix $\mathbf{D}$ given by Eq. 45. The values with the asterisk ($*$) indicate intermediate values obtained during the Gauss-Seidel sweep. To eliminate the sweep bias, symmetry is obtained by sweeping once in a prescribed direction and then sweeping again in the reverse direction. A detailed discussion on the applicability of each of these methods is described by Buelow [48]. LU-SGS and point-based Gauss-Seidel schemes are also available.

GMRES Solver

The Generalized Minimum Residual (GMRES) [49] solver module from the new mStrand code [50] for semi-structured multi-strand grids has been modified for Cartesian grids and implemented within the framework to accelerate convergence. The process is described in the provided reference. Each of the linear solvers outlined above can be used to precondition the GMRES linear system.

V.B. Python-based Infrastructure

The production framework Helios relies on a multi-solver paradigm [51] to simulate complex rotorcraft flows with realistic geometry. We have modeled the current framework based on the production version using a similar Python-based infrastructure that handles calling all of the disparate modules that are required to work in concert. While there are less modules available in the current work, the simplified framework provides an opportunity for rapid development of novel schemes and solution procedures with the ability to easily migrate technology to the production version. The Helios framework [52] includes multiple near-body solvers (structured, unstructured and semi-structured), an off-body Cartesian solver capable of highly-scalable adaptive mesh refinement, structural coupling, highly-scalable domain connectivity [53] providing efficient handling of moving bodies and well-defined APIs encouraging continued collaboration with industry, academia and other governmental entities. In contrast, the simplified framework consists of a single flow solver using Cartesian grids with fixed refinement, an open-source domain connectivity module [54] and an actuator line model used to introduce rotor wakes without the need to handle moving bodies. Additionally, the current framework includes a standalone library to facilitate PinT calculation.

It is crucial that the Parallel-in-Time capability be agnostic to the flow solver and require only minimal modifications of the solver(s) in order to enable time parallelism with existing codebases. In the current work, this involves wrapping the spatial residual evaluation routines (`solver.rhs()`) and implicit inversion operation (`solver.lhs()`) that can be called from the high-level Python controller. The space-time approximate factorization approach [45–47, 55–57] effectively decouples the implicit solutions of the spatial and temporal systems providing a straightforward implementation of the standalone PinT module; the spatial residual is passed from the flow solver(s) to the PinT module, which performs the necessary temporal coupling (e.g. `ts.af_update()`) and feeds the result back to the flow solver which resumes the spatial inversion process. Currently, the space-time GMRES solver is implemented entirely within the flow solver, but a more involved high-level Python controller will be implemented to maintain the decoupling between the flow solver(s) and the standalone PinT module for GMRES calculations.

V.C. Parallel-in-Time Module

There are several libraries to facilitate PinT time integration for unsteady PDEs such as libPFASST [58] based on the Parallel Full Approximation Storage in Space and Time (PFASST) library [20, 59] and the Xbraid code based on the Multigrid Reduction in Time (MGRIT) method [60]. However, we are interested in the sub-class of unsteady problems that include periodic and quasi-periodic flows resulting from rotors and turbines. Such flows can be simulated more efficiently using methods specifically designed for periodic solutions.

To achieve temporal parallelism, the solution for each time sample is instantiated and advanced on separate processors. The solutions at disparate time samples are coupled through evaluation of the temporal derivative source term, $D_N \mathbf{q}$, and the direct inversion of the temporal operator in the space-time approximate factorization scheme, if used. Evaluation of the temporal derivative source term is achieved by a sequence of $2b$ Message-Passing Interface (MPI) calls, where b is the bandwidth of the operator. For the case of the Time-Spectral method, $b = N/2$ or $(N-1)/2$ and the process executes in a ring topology [61] where each process multiples its solution by the appropriate element of the D_N matrix and passes it to the subsequent process (time sample) to be aggregated using `MPI_sendrecv_replace`. For the case of the FDMT-p approach, only p `MPI_sendrecv` calls are required to the $M = p/2$ neighboring processes. For the case of the space-time approximate factorization, parallel DFT/IDFT matrix-vector products are performed using N `MPI_reduce` calls each to transform the system into the frequency domain, perform the intermediate update using scalar complex arithmetic, and transform it back into the time domain. After the temporal coupling stage, the solution process continues execution in parallel.

As the bandwidth of the TS operator scales with N, the method is not expected to scale weakly; the wall time per iteration is not expect to stay constant if N and the number of processors are doubled because twice as many sequential temporal coupling steps are required. In contrast, the FDMT approach only requires p sequential coupling steps and therefore expect the wall time per iteration to remain constant if the problem size and the number of processors are doubled. However, the total wall time for a computation, t_{exec}, is a product of both the wall time per iteration, t_{iter}, and the overall number of iterations required, N_{iter}.

$$t_{\text{exec}} = t_{\text{iter}} \times N_{\text{iter}} \tag{46}$$

While the FDMT approach provides weak scaling for the first component, the current lack of a direct or GMRES solver prevents weak scaling with respect to the overall execution time; using larger values of N with the FDMT subiterative scheme imposes a more restrictive time step, and therefore requires more iterations to reach a given level of convergence. As it stands, the most pressing issue included in future work concerns development of a more robust FDMT solver that does not directly invert the implicit temporal operator. The inverse of implicit operator associated with the circulant differentiation matrix is dense and solving such a system directly forfeits the advantage of the reduced-bandwidth approach.

In addition to performing the necessary temporal coupling, the standalone PinT module is responsible for defining the necessary MPI communicators. The processes allocated to each grid level, p, are divided into N time instances, resulting in a spatial partition of $p_x = p/N$ processes per grid level at a given time level. The PinT module creates spatial and temporal MPI communicators to facilitate fringe communication in space and temporal coupling, respectively. Other communicators are created to facilitate domain connectivity between the different grid levels at a given time instance in addition to providing the desired i/o functionality.

The following sections outline the solution procedures of the the PinT time integration schemes described in §IV; implementation details of the standard SinT time marching schemes like Runge-Kutta and implicit BDF1 and BDF2 defined in sections IV.A and IV.B, respectively, and no further detail is provided.

Starting with the original expression for the system of equations for the Time-Spectral discretization in Eq. 38, we apply first-order backward Euler approximation to the pseudotime derivative, where $q^{n,s}$ refers to the iteration s of the solution at time $t = t_n$.

$$\frac{q^{n,s+1} - q^{n,s}}{\Delta\tau} + \sum_{j=0}^{N-1} d_j^n q^{j,s+1} + \mathbf{R}\left(q^{n,s+1}\right) = 0, \quad \forall n \in \{0, \dots, N-1\} \tag{47}$$

We linearize the residual about $q^{n,s}$ and put Eq. 47 into *delta* form.

$$\frac{\Delta q^n}{\Delta\tau} + \sum_{j=0}^{N-1} d_j^n \Delta q^j + \left.\frac{\partial\mathbf{R}}{\partial q}\right|_{q=q^{n,s}} \Delta q^n = -\left[\mathbf{R}\left(q^{n,s}\right) + \sum_{j=0}^{N-1} d_j^n q^{j,s}\right], \quad \forall n \in \{0, \dots, N-1\} \tag{48}$$

We can express Eq. 48 in terms of the complete space-time domain where

$$\begin{aligned} \mathbf{\Delta q} &= \{\Delta q^0, \dots, \Delta q^{N-1}\} \\ \mathbf{R}\left(\mathbf{q}\right) &= \{\mathbf{R}\left(q^0\right), \dots, \mathbf{R}\left(q^{N-1}\right)\} \end{aligned}$$

are arrays of the update and spatial residual, respectively, over both space and time and D_N^{G} is the global temporal differentiation operator constructed via permutations of the local temporal differentiation operator, D_N.

$$\left[I + \Delta\tau D_N^{\mathrm{G}} + \Delta\tau\frac{\partial\mathbf{R}}{\partial q}\right]\mathbf{\Delta q} = -\Delta\tau\left[\mathbf{R}\left(\mathbf{q}^s\right) + D_N^{\mathrm{G}}\mathbf{q}^s\right] \tag{49}$$

Approximate Factorization between Time and Space

To avoid the need to solve the large global system spanning all of the degrees of freedom, we can apply an approximate factorization (AF) between time and space.

$$\left[I + \Delta\tau D_N^{\mathrm{G}}\right]\underbrace{\left[I + \Delta\tau\frac{\partial\mathbf{R}}{\partial q}\right]\mathbf{\Delta q}}_{\mathbf{\Delta\bar{q}}} = -\Delta\tau\left[\mathbf{R}\left(\mathbf{q}^s\right) + D_N^{\mathrm{G}}\mathbf{q}^s\right] \tag{50}$$

To begin each iteration, the flow solver evaluates the spatial residual, $\mathbf{R}\left(q^{n,s}\right)$, for each time sample concurrently and passes the residual and solution arrays to the PinT module. In the temporal coupling stage of the AF process, dense blocks of the local temporal operator, $I + \Delta\tau D_N$, are solved for an intermediate update, $\mathbf{\Delta\bar{q}}^n$, on a point-by-point basis in space.

$$\mathbf{\Delta\bar{q}} = -\Delta\tau\left[I + \Delta\tau D_N^{\mathrm{G}}\right]^{-1}\left[\mathbf{R}\left(\mathbf{q}^s\right) + D_N^{\mathrm{G}}\mathbf{q}^s\right] \tag{51}$$

Holding the pseudo time step constant for each of the time instances at a particular point in space ($\Delta\tau$ is still free to change in space to accommodate local time-stepping), permits the diagonalization of Eq. 51 via application of the discrete Fourier transform (DFT); rather than directly inverting the temporal operator in Eq. 51, a parallel DFT (matrix-vector product using N `MPI_reduce` calls) is applied to both components of the RHS of Eq. 50 (the solution and spatial residual[3]), and the first stage of the AF process is updated in the frequency domain using complex scalar division (the eigenvalues of D_N are $\lambda_k = i\omega k$).

$$\Delta\tilde{\bar{q}}_k = -\Delta\tau\frac{\tilde{R}_k + i\omega k\tilde{q}_k}{1 + \Delta\tau i\omega k}, \quad \forall k \in \mathcal{K} \tag{52}$$

After applying a parallel inverse DFT to the frequency-domain intermediate update, the second stage of the space-time approximate factorization process continues externally in the flow solver where the desired linear solver is applied to the remaining spatial problem at each time instance.

$$\left[I + \Delta\tau\frac{\partial\mathbf{R}}{\partial q}\right]\Delta q^n = \Delta\bar{q}^n, \quad \forall n \in \{0, \dots, N-1\} \tag{53}$$

This approximately-factored Time-Spectral scheme will be referred to herein as AF-TS.

---

[3] Alternatively, the TS source term can first be added to the spatial residual requiring only a single parallel DFT/IDFT.

Subiterative Space-Time Method

Employing a space-time AF scheme as a linear solver or preconditioner to a space-time GMRES formulation can greatly accelerate convergence by allowing for much larger pseudo time steps when a high frequency or a large number of time instances is desired. However, the reduced frequency ($\omega c/V_{\text{ref}}$) associated with several problems of interest are quite low; for example, the reduced frequency of a rotor blade with radius R, scales like c/R. For such problems, it may be desirable to employ a scheme that does not require the direct solution of an implicit temporal operator. Lakshminarayan et al. [62] developed a dual-pseudotime Time-Spectral scheme, using a subiteration procedure in a second level of pseudotime to avoid a costly inversion of the temporal operator while maintaining implicitness in the temporal dimension.

Starting with the expression for the Time-Spectral system of equations after applying backward Euler in pseudotime (Eq. 47), we can add and subtract an interim iterative state at $q^{n,k}$.

$$\frac{q^{n,s+1} - q^{n,k} + q^{n,k} - q^{n,s}}{\Delta\tau} + \sum_{j=0}^{N-1} d_j^n q^{j,s+1} + \mathbf{R}\left(q^{n,s+1}\right) = 0, \quad \forall n \in \{0,\ldots,N-1\} \tag{54}$$

Setting all of the $s+1$ terms to the subiteration index $k+1$ provides a dual-pseudotime subiteration procedure such that the subiterate $q^{n,k+1}$ converges sufficiently towards $q^{n,s+1}$ within each outer iteration to ensure stability.

$$\frac{q^{n,k+1} - q^{n,k}}{\Delta\tau} + \frac{q^{n,k} - q^{n,s}}{\Delta\tau} + \sum_{j=0}^{N-1} d_j^n q^{j,k} + \mathbf{R}\left(q^{n,k+1}\right) = 0, \quad \forall n \in \{0,\ldots,N-1\} \tag{55}$$

The time-derivative term is lagged at subiteration level k to avoid a costly inversion of the temporal operator while maintaining a degree of implicitness; often only a modest number of subiterations are required to maintain the stability of the iterative procedure. Linearizing the residual about $q^{n,k}$ and defining $\Delta q^n = q^{n,k+1} - q^{n,k}$ results in a dual-pseudotime space-time iterative scheme that can be used with either the TS or FDMT schemes.

$$\left[\frac{1}{\Delta\tau_{lhs}} I + \frac{\partial \mathbf{R}}{\partial q}\right]\Delta q^n = -\left[\mathbf{R}\left(q^{n,k}\right) + \sum_{j=0}^{N-1} d_j^n q^{j,k} + \frac{q^{n,k} - q^{n,s}}{\Delta\tau_{rhs}}\right], \quad \forall n \in \{0,\ldots,N-1\} \tag{56}$$

The pseudo time steps on the left and right-hand sides, $\Delta\tau_{lhs}$ and $\Delta\tau_{rhs}$, respectively, can be taken to be equivalent or the RHS can be accelerated by selecting $\Delta\tau_{rhs} > \Delta\tau_{lhs}$. This approach is currently the default method of solving the FDMT system of equations.

Because this is a nonlinear iteration, this scheme is not directly applicable to GMRES as a preconditioner. Research is ongoing on incorporating this process as a preconditioner for a space-time GMRES solver.

Space-Time GMRES Solver

Su and Yuan [63] and Mundis and Mavriplis [61] recognized that the decreased diagonal dominance of the implicit system resulting from the Time-Spectral discretization greatly degrades stability and convergence when using a large maximum wave-number, ωN; each elected to overcome this deficiency by incorporating a solution approach that does not rely upon such a property and implemented Time-Spectral GMRES solvers for the Euler equations on both structured and unstructured meshes, respectively, with various preconditioners. They successfully demonstrated wave-number independence of the resulting schemes [64] providing enhanced convergence for cases involving high frequencies, a large number of time instances or both. Following their success, we extended the GMRES module from the flow solver to enable Time-Spectral calculations and employed the space-time approximate factorization, which had been our core solver, as the preconditioner. Mundis and Mavriplis [65] have also implemented their Time-Spectral GMRES solver with a space-time approximate factorization preconditioner and have demonstrated its superior performance over an array of previously developed preconditioners [61, 64, 66]. Currently our GMRES implementation is limited to the original Time-Spectral method, but we are interested in extending it to the sub-spectral FDMT approach. Application of the direct space-time approximate factorization for this scheme would undo the parallel efficiency gains afforded by the reduced bandwidth of the temporal differentiation operator as the inverse of the circulant operator is dense. Su and Yuan demonstrated a multigrid preconditioner for their GMRES implementation and our goal is to include a temporal multilevel approach to precondition the GMRES-FDMT solver to provide a robust scheme capable of handling large wave-numbers while still providing an algorithm that scales weakly with respect to time-to-solution; the temporal multilevel approach will also provide a means to maintain the space-time approximate factorization, which greatly simplifies integration of flow solvers with a standalone PinT module.

V.D. Parallel Space and Time Domain Connectivity

The Helios off-body solver SAMcart employs the LLNL SAMARAI code [67] to facilitate refinement and the PUNDIT [53] code supplies efficient, implicit, domain connectivity between all grids. In the current work we are restricted to fixed grid refinement using the Topology Independent Overset Grid Assembler (TIOGA) [54] for domain connectivity between different grid levels. For a mesh system that uses a hierarchy of nested Cartesian grids at different levels, three different MPI communicators are used; the first manages the spatial data exchange between blocks of the same grid (e.g. halo data at the block boundaries), the second manages the spatial data exchanges between grids at different levels and the third manages the temporal data exchange between corresponding grid blocks at different time instances. This approach ensures that the overset grid connectivity happens only on a purely spatial problem and the entire execution is fully partitioned and distributed in both space and time.

V.E. Actuator Line Module

Actuator Line/Disk approaches are reduced order models where lifting surfaces such as wings and rotor blades are modeled using source terms in the governing equations. This approach has gained wide uses in both rotorcraft [68–70] and wind turbine [71–75] calculations. In general, actuator line methods rely on reduced order aerodynamics that use a combination of lifting line theory, airfoil tables and unsteady/steady stall models to compute the forces generated by lifting surfaces. The coupling between the lower order aerodynamics and higher fidelity CFD is as follows: Lower order aerodynamic models use the velocity field from the flow solution to compute metrics such as angle of attack, Mach number and Reynolds number and subsequently use them to determine sectional forces using a multi-dimensional airfoil table lookup. These sectional forces ($\vec{F}$) and rate of work performed by them ($\vec{F} \cdot \vec{v}$) are included as source terms to the momentum and energy equations. Data exchange between CFD and lower-order models happen every non-linear iteration such that the predicted loading, source terms and velocity field are consistent with one another. The following works [70, 71, 73, 75] are recommended to the reader for detailed descriptions of the formulations and nuances of the actuator line approach. The implementation followed by Gopalan et al. [74] is used in this work. In general, the actuator line method provides a simpler analysis framework to replicate the flow signature created by a spinning rotor blade, that involves high frequency perturbations that repeat every blade passage. These perturbations originate in the concentrated tip vortices trailed by the rotor blades and create Gaussian like perturbations in density and pressure and an anti-symmetric twin-Gaussian like perturbations in the velocity field. The ability of time-parallel methods to accurately resolve this type of time-varying flow field is key to extending this approach to more realistic rotary-wing calculations that use body-conforming grids for the rotor blades. The actuator line model also has the added benefit of avoiding the complexities associated with applying the Time-Spectral method to dynamic overset meshes where complete time-histories of the solution are not necessarily available throughout the domain [45, 47, 76].

VI. Numerical Results

The ultimate objective of this work is to develop a methodology to periodic flows using PinT time integration in a fraction of the execution time required for the corresponding SinT calculation. The Time-Spectral method has demonstrated success in reducing the execution time for periodic flows, but has typically, albeit not exclusively, used a limited number of harmonics as the algorithm scales as $N \log N$ at best using the Fast Fourier Transform to evaluate the TS source term and invert the implicit temporal operator [45]. The PinT version of the method, introduced by Mavriplis and Yang [25] reduced this scaling by exploiting an additional level of parallelism in time by distributing the time samples and performing the spatial processes concurrently. However, the density of temporal coupling operator limits the scalability as the bandwidth grows linearly with N.

In this section, we introduce the problem of the NREL Phase VI wind turbine and apply the periodic PinT Time-Spectral method. It becomes readily apparent that a large number of time samples are required to adequately resolve the flow. The section continues with a simplified problem of a periodic density pulse that is used to investigate the accuracy and scalability of the PinT TS scheme and the sub-spectral, algebraically accurate FDMT method.

VI.A. NREL Phase VI Wind Turbine

The two-bladed NREL Phase VI wind turbine [77] has been studied extensively and used here as an a representative test case of the the types of periodic flows that will ultimately be computed using the periodic PinT approach. The rotor has a radial frequency of $\omega = 7.54$ rad/s and radius of 5.03 m with a root cut out of 2.0 m. The freestream Mach number corresponds to an inflow velocity of 10 m/s. A relatively coarse two-level grid system spans approximately 7.5 blade radii (R) in the y- and z-directions and approximately 37 R in the axial x-direction; each $375 \times 75 \times 75$ grid contains just over 2 million points for a total of 4,218,750 points. The fine-level grid spacing of $\Delta s = 0.25$ m is only slightly smaller than the tip chord of $c = 0.36$ m, but provides a reasonably resolved vortical wake structure while maintaining a manageable grid size for PinT calculations with our current computational allocation limits. The fine grid spans approximately 17 R behind the turbine before transitioning to the coarse grid.

Figure 3 depicts the flow field of several Time-Spectral and BDF2 turbine simulations; contours of axial velocity are plotted in addition to iso-contours of vorticity magnitude. Its is clear that a large number of modes are required to resolve this flow, as the $N = 33$ Time-Spectral case still exhibits aliased information in the plane of the blades. This aliasing problem is analyzed further by examining the transient flow solution at a sensor point. The sensor point is located at $(-0.1, 0.0, 4.5)$ m relative to the turbine origin and is traversed by the blades close to their tips. Figure 4 plots instantaneous streamwise momentum, ρu, computed with the BDF2 scheme in addition to the discrete solution values of Time-Spectral calculations for $N \in \{3, 5, 9, 17, 33\}$; reconstructions of the TS solutions are plotted as curves in the corresponding color. The high-frequency signal transferred from the blade passages poses resolution problems for the TS calculations, as discussed in [45, 46], including the $N = 33$ case that exhibits significant Gibbs' phenomenon. This is demonstrated further in Fig. 5 where the TS cases using $N \in \{5, 9, 17, 33\}$ are plotted independently with the transient solution; filtered BDF2 solutions to $K = (N-1)/2$ modes are also plotted and reveal that the TS solutions are plagued by aliasing errors from the unresolved frequencies. We continue in the following section with a simplified problem of a periodic density pulse that requires a smaller grid system enabling more extensive analysis.

VI.B. Periodic Density Pulse

A problem of an inviscid periodic density pulse was selected as a simplified problem, absent moving geometry, to analyze parallel scalability of the TS and FDMT methods without adding additional complexities that are independent of the core algorithm. The pulse is prescribed via a dynamic inflow boundary condition with a freestream Mach number of $M_\infty = 0.5$.

The dynamic inflow boundary is defined by as a two-dimensional Gaussian function scaled by a time-varying kernel, $\gamma(t)$, which is added to the density on the inflow plane, x_0.

$$\rho(x_0, y, z, t) = \rho_0 + \alpha e^{-\beta\left(y^2+z^2\right)}\gamma(t) \tag{57}$$

Pressure is extrapolated from the domain interior and all other flow quantities are prescribed. The density perturbation coefficient, α, is taken as $\rho_0/10$.

Two kernels are investigated; the first case includes just a single forcing frequency, $\gamma(t) = \sin(\omega t)$, and the second case is a more complex function, $\gamma(t) = \frac{2}{5+4\cos(2\omega t)} - 1$, that includes additional frequency content and a two-per-period forcing to mimic the two-per-period blade passages of the wind turbine described in the previous section. BDF2 calculations employ a physical time step of $\Delta t = T/4096$ and both test cases settle quickly into a periodic steady-state. Figure 6 plots density on a constant y-plane for both the single frequency (Figure 6a) and two-per-period (Figure 6b) cases on 65^3 meshes.

Figure 7 plots the convergence of the space-time approximate factorization Time-Spectral scheme for the single-frequency periodic density pulse for a high reduced frequency of 1. Each 50^3 grid was partitioned into fifteen blocks for a total of $15N$ processors for each calculation for $N = 4$ through 1024. While the iterative convergence of the AF-TS solver is independent of N for a fixed pseudo time step (Figure 7a), the wall time-to-solution scales linearly with N (Figure 7b); the scaling is more clearly exhibited in Fig. 7c which demonstrates how the residual versus wall time per time instance curves for each value of N nearly collapse onto each other.

In order to determine convergence dependence on the fundamental frequency for the available schemes, we selected three frequencies based on integer multiples of the turbine frequency on the 65^3 mesh: ω_0, $2\omega_0$ and $4\omega_0$. Figures 8 and 9 demonstrate the linear scaling of the periodic PinT Time-Spectral method using an approximate-factorization as the primary solver and GMRES preconditioner, respectively. Figure 8a plots space-time residual versus wall time and clearly demonstrates that higher frequencies reduce the convergence rate. Higher frequencies of more than 4 times the baseline value of ω_0 are not resolvable on the selected spatial grid but we expect the trend to continue for even higher frequency values. While iterative convergence is essentially independent of N (Figure 8c), the linear scaling of the method is demonstrated in Fig. 8b which plots space-time residual versus wall time per time instance. Applying the space-time approximate factorization scheme as a GMRES preconditioner significantly improves convergence by removing the frequency dependence exhibited previously. While the GMRES AF-TS scheme still exhibits linear scaling with respect to wall time (Figures 9a and 9b), Fig. 9c demonstrates essentially identical iterative convergence for each case, independent of N or ω.

Figure 10 provides BDF2 and periodic PinT solutions for the TS and FDMT-p methods for $p \in \{2, 4, 8\}$ for the single-frequency case using the space-time subiterative scheme; instantaneous values of density are sampled in the center of the domain. Because of the simple waveform, the TS method (Figure 10d) requires only a single harmonic to resolve the flow. The FDMT-2 case (Figure 10a) demonstrates a phase lag for $N=$ 4 and 8 but manages to captures the shape of the signal successfully. Higher resolution cases of $N = 32$ and above match the transient solution within the scale of the figure. The FDMT-4 case (Figure 10b) matches the transient signal for $N = 16$ and above and the FDMT-8 (Figure 10c) cases matches the transient signal for each value of N.[4] Figure 12a demonstrates the convergence rates of each scheme; the Time-Spectral method converges rapidly as only a single mode is required and the FDMT-p methods each exhibit the expected p-th order convergence rates (the black hashed lines correspond to the second-, fourth- and eighth-order convergence). Error is evaluated by taking the norm between each solution vector and the *exact* TS solution, q_{ex}, at $N = 512$ at time $t = 0$.

Figure 11 provides BDF2 and periodic PinT solutions for the TS and FDMT-p methods for $p \in \{2, 4, 8\}$ for the more complex forcing kernel that is designed to mimic the types of signals produced in turbine or rotor cases with multiple blade passages. The FDMT-2 method (Figure 11a) fails to match the BDF2 solution within the scale of the figure for even $N = 128$ case and the significantly under-resolved solutions demonstrate wild oscillations. The fourth- (Figure 11b) and eighth-order (Figure 11c) accurate discretizations naturally perform better but even the spectral scheme (Figure 11d) requires a large number of modes to adequately resolve the signal. The convergence rates of the schemes presented in Fig. 12b demonstrate the expected performance, but asymptotic convergence is delayed for the algebraic cases. The spectral case requires more than one hundred time samples to converge and the FDMT-8 scheme performs nearly as well.

Because the Time-Spectral method requires significant temporal communication, we investigated how the two-dimensional processor layout affects execution time. Processors can be distributed such that temporal or spatial degrees of freedom are clustered. Figure 13 plots the wall time per time step for a set of time instances for both topology options and as expected, the temporal clustering offers significant savings, but does not affect the asymptotic linear scaling. The calculations were performed on a machine with twenty-four cores per node; the temporal clustering cases corresponding to $N = 4, 8, 12$ and 24 each exhibit significantly faster execution times, even compared to other temporal clustering cases. Complete time histories of these cases can be packed onto a single node significantly increasing the communication speed for the temporal coupling stages. In general, temporal clustering offers roughly a 40% reduction in execution time as compared to the spatial clustering topology. In light of these results, we may explore the use of GPUs or Many-in-core (MIC) chips to enable more time samples to be packed onto a single processor or node to use shared memory for temporal coupling. Instantiating several time samples on a single core may also provide savings. Nevertheless, we do not expect a general relaxation of the linear scaling of the Time-Spectral method and will therefore focus our efforts on schemes with a fixed amount of sequential temporal coupling with the goal of achieving massive scaling for realistic rotorcraft and wind turbine calculations. The use of shared memory and temporal

---

[4] The FDMT-4 scheme is only applied to cases with $N > 4$ and the FDMT-8 scheme is only applied to case with $N > 8$ because there are not enough sample points to form the derivative with the desired accuracy below those values.

clustering should also benefit compactly-supported approaches where only fringe communication would be required, thereby reducing execution time even further.

Figure 14 provides the time per iteration of the subiterative space-time scheme for the Time-Spectral and FDMT methods. As anticipated, the wall clock time of the FDMT method remains nearly constant with increasing N, and a corresponding increase in the number of processors. This demonstrates near-optimal weak scaling of the wall clock time per iteration. In contrast, the TS method shows a linear increase in wall clock time per iteration once N grows large enough such that the temporal communication dominates the overall cost (approximately $N = 16$). Combining the data presented in Figs. 12 and 14 demonstrates the wall time per iteration required to achieve a given level of accuracy for each method. Figure 15 plots error versus wall time per iteration. Each successive data sample corresponds to a doubling of N. For the single frequency case (Figure 15a), there is not much difference in wall time per iteration between the FDMT and TS approaches, as only a few time samples are required to reduce the error of the Fourier representation to within machine precision. However, the more complex signal corresponding the to multiple frequency kernel requires a substantial number of harmonics to reduce the error to within machine precision. Figure 15b demonstrates that a given level of accuracy can be achieved by the FDMT method with a wall time per iteration that is many times lower than the wall time per iteration taken by the TS method to achieve the same level of accuracy. Since the wall time per iteration is essentially fixed for the FDMT methods, the speedup of the FDMT method over TS increases with the complexity of the solution.

The favorable scalability of the FDMT method suggests that it may successfully provide periodic PinT calculations with a similar level of accuracy in a fraction of the execution required for SinT time advancement. However, this does require a more robust solver that does not impose severe restrictions of the time step, which is currently the case. Even though the wall time per iteration is essentially constant for the FDMT methods, large values of N require a substantially smaller pseudo time step, preventing weak scaling with respect to time-to-solution. Development of a more robust solver is a primary focus of future work.

VII. Summary and Future Work

A solver framework has been introduced for the rapid development, testing and analysis of time-parallel calculations for time-periodic flows. The framework that currently includes a Cartesian flow solver and a standalone Parallel-in-Time library can be coupled to additional flow solvers in the future to compute more complex flows including curvilinear and unstructured body-fitted grids. The methodology developed can ultimately be incorporated into full-scale production codes such as Helios to facilitate realistic time-parallel calculations of realistic rotorcraft and wind turbine configurations.

A wind turbine case was introduced to motivate the need for many time instances in PinT calculations as the signals induced from rotating blades onto stationary Cartesian background grids include sharp disturbances that require high resolution, even for discrete Fourier methods such as the Time-Spectral scheme. An algebraically-accurate analog to the TS method was introduced and applied to a simplified periodic density pulse case. The new formulation demonstrates weak scalability as the temporal resolution is increased dramatically. This demonstration provides encouragement that this methodology will be able to scale to millions of processors on real problems.

Future work includes the development of a GMRES solver using the FDMT approach to provide scalable time-to-solution; i.e. the same time to solution is achieved for a problem size twice as large on twice as many cores by removing the time step restriction for increasing N. Other compactly-supported discretizations may also be explored. Because rotorcraft in hover and wind turbines involve a slow freestream, if any at all, multigrid and multilevel schemes will be explored as GMRES preconditioners to more rapidly propagate the solution throughout the domain. A temporal multigrid scheme may provide the robustness desired for the FDMT approach while retaining the ability to factor the implicit operator between time and space in order to maintain the portability of the PinT library. As demonstrated, temporal communication is greatly reduced when all of the temporal degrees of freedom are situated close to each other within the processor topology map; we will investigate additional ways to leverage processor proximity by instantiating multiple time samples on each process, which would allow the use of shared memory within a node and further reduce temporal communication. We would also like to add adaptive temporal mesh refinement with proper load balancing because the highest temporal resolution is primarily required in the vicinity of the moving geometry and we can more efficiently allocate the resources provided by coarsening the temporal resolution away from dynamic regions of the flow. Various sensors can initially drive coarsening and refinement of the temporal mesh, but ultimately the goal is to use a fully-automated, adjoint-based error estimation to adapt the complete space-time mesh with sophisticated dynamic load balancing to further reduce execution times.

Acknowledgments

Material presented in this paper is a product of the CREATE-AV$^{\text{TM}}$ Element of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office.

References

[1] Nievergelt, J., "Parallel methods for integrating ordinary differential equations," *Comm. ACM*, Vol. 7, 1964, pp. 731–733.

[2] Gear, C. W., "Parallel methods for ordinary differential equations," *Calcolo*, Vol. 25, No. 1–2, 1988, pp. 1–20.

[3] Evans, D. J. and Sanugi, B. B., "A Parallel Runge-Kutta Integration Method," *Parallel Computing*, Vol. 11, 1989, pp. 245–251.

[4] Womble, D. E., "A Time-Stepping Algorithm for Parallel Computers," *SIAM J. Stat. Comput.*, Vol. 11, No. 5, 1990, pp. 824–837.

[5] Jackson, K. R., "A Survey of Parallel Numerical Methods for Initial Value Problems for Ordinary Differential Equations," *IEEE Trans. Magnetics*, Vol. 27, No. 5, 1991, pp. 3792–3797.

[6] Horton, G. and Knirsch, R., "A Space-Time Multigrid Method for Parabolic Partial Differential Equations," *Parallel Computing*, Vol. 18, 1992, pp. 21–29.

[7] Horton, G. and Vandewalle, S., "A Space-Time Multigrid Method for Parabolic Partial Differential Equations," *SIAM J. Sci. Comput.*, Vol. 16, No. 4, 1995, pp. 848–864.

[8] Lions, J. L., Maday, Y., and Turinici, G., "Résolution d'EDP par un schéma en temps pararéel," *C.R.Acad Sci. Paris Sér. I Math*, Vol. 332, 2001, pp. 661–668.

[9] Sterck, H. D., Manteuffel, T. A., McCormick, S. F., and Olson, L., "Least-Squares Finite Element Methods and Algebraic Multigrid Solvers for Linear Hyperbolic PDEs," *SIAM J. Sci. Comput.*, Vol. 26, No. 1, 2004, pp. 31–54.

[10] Sterck, H. D., Manteuffel, T. A., McCormick, S. F., and Olson, L., "Numerical Conservation Properties of H(div)-Conforming Least-Squares Finite Element Methods for the Burgers Equation," *SIAM J. Sci. Comput.*, Vol. 26, No. 5, 2005, pp. 1573–1597.

[11] Garrido, I., Lee, B., Fladmark, G. E., and Espedal, M. S., "Convergent Iterative Schemes for Time Parallelization," *Math. Comp.*, Vol. 75, No. 255, 2006, pp. 1403–1428.

[12] Gander, M. J. and Vandewalle, S., "Analysis of the Parareal Time-Parallel Time-Integration Method," *SIAM J. Sci. Comput.*, Vol. 29, No. 2, 2007, pp. 556–578.

[13] Gander, M. J. and Hairer, E., "Nonlinear Convergence Analysis for the Parareal Algorithm," *Domain Decomposition Methods in Science and Engineering XVII*, edited by U. Langer, M. Discacciati, D. E. Keyes, O. B. Widlund, and W. Zulehner, Vol. 60 of *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, 2008, pp. 193–200.

[14] Maday, Y., "The parareal in time algorithm," June 2008.

[15] Minion, M. L. and Williams, S. A., "Parareal and Spectral Deferred Corrections," *Numerical Analysis and Applied Mathematics*, edited by T. E. Simos, No. 1048 in AIP Conference Proceedings, AIP, 2008, pp. 388—391.

[16] Amodio, P. and Brugnano, L., "Parallel solution in time of ODEs: some achievements and perspectives," *Appl. Numer. Math.*, Vol. 59, No. 3–4, 2009, pp. 424—435.

[17] Christlieb, A. J., Macdonald, C. B., and Ong, B. W., "Parallel High-Order Integrators," *SIAM J. Sci. Comput.*, Vol. 32, No. 2, 2010, pp. 818–835.

[18] Minion, M. L., "A Hybrid Parareal Spectral Deferred Corrections Method," *Comm. App. Math. and Comp. Sci.*, Vol. 5, No. 2, 2010, pp. 265–301.

[19] Speck, R., Ruprecht, D., Krause, R., Emmett, M., Minion, M., Winkel, M., and Gibbon, P., "A massively space-time parallel N-body solver," 2012, Proc. of the SC'12 International Conference for High Performance Computing, Networking, Storage and Analysis.

[20] Emmett, M. and Minion, M., "Toward an Efficient Parallel in Time Method for Partial Differential Equations," *Communications on Pure and Applied Mathematics and Computational Science*, Vol. 7, No. 1, 2012, pp. 105–132.

[21] Dai, X. and Maday, Y., "Stable Parareal in Time Method for First- and Second-Order Hyperbolic Systems," *SIAM J. Sci. Comput.*, Vol. 35, No. 1, 2013, pp. A52–A78.

[22] Gander, M. J. and Güttel, S., "PARAEXP: A Parallel Integrator for Linear Initial-Value Problems," *SIAM J. Sci. Comput.*, to appear.

[23] Gander, M. J., *50 Years of Parallel Time Integration*, Springer Verlag, to appear 2015.

[24] Ruprecht, D., Speck, R., Emmett, M., Bolten, M., and Krause, R., "Poster: Extreme-scale space-time parallelism," *Proceedings of the 2013 Conference on High Performance Computing Networking, Storage and Analysis Companion*, SC '13 Companion, Denver, Colorado, USA, 2013.

[25] Mavriplis, D. J. and Yang, Z., "Time Spectral Method for Periodic and Quasi-Periodic Unsteady Computations on Unstructured Meshes," AIAA Paper 5034, Chicago, Illinois, June 2010.

[26] McMullen, M., Jameson, A., and Alonso, J., "Demonstration of Nonlinear Frequency Domain Methods," *AIAA Journal*, Vol. 44, No. 7, July 2006, pp. 1428–1435.

[27] Canuto, C., Hussaini, M., Quarteroni, A., and Zang, T., *Spectral Methods: Fundamentals in Single Domains*, Scientific Computation, Springer, Berlin, 2006.

[28] Kennedy, C. A., Carpenter, M. H., and Lewis, R. M., "Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations," *Applied numerical mathematics*, Vol. 35, No. 3, 2000, pp. 177–219.

[29] Jameson, A., "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 1596, Honolulu, Hawaii, June 1991.

[30] Hall, K., Thomas, J., and Clark, W., "Computation of Unsteady Nonlinear Flows in Cascades using a Harmonic Balance Technique," *9th International Symposium on Unsteady Aerodynamics, Aeroacoustics and Aeroelasticity of Turbomachines*, Lyon, France, September 2000.

[31] Hall, K. C., Thomas, J. P., and Clark, W., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, Vol. 40, May 2002, pp. 879–886.

[32] McMullen, M. and Jameson, A., "The computational efficiency of non-linear frequency domain methods," *Journal of Computational Physics*, Vol. 212, 2006, pp. 637–661.

[33] Nadarajah, S. K., McMullen, M. S., and Jameson, A., "Optimum Shape Design for Unsteady Flows Using Time Accurate and Non-Linear Frequency Domain Methods," AIAA Paper 3875, Orlando, Florida, June 2003.

[34] Murman, S. M., "Reduced-Frequency Approach for Calculating Dynamic Derivatives," *AIAA Journal*, Vol. 45, No. 6, June 2007, pp. 1161–1168.

[35] Gopinath, A. K. and Jameson, A., "Time Spectral Method for Periodic Unsteady Computations over Two- and Three-Dimensional Bodies," AIAA Paper 1220, Reno, Nevada, January 2005.

[36] Gopinath, A. K. and Jameson, A., "Application of the Time Spectral Method to Periodic Unsteady Vortex Shedding," AIAA Paper 0449, January 2006.

[37] Blanc, F., Roux, F.-X., Jouhaud, J.-C., and Boussuge, J.-F., "Numerical Methods for Control Surfaces Aerodynamics with Flexibility Effects," IFASD 2009, Seattle, WA, June 2009.

[38] McMullen, M. S., *The Application of Non-Linear Frequency Domain Methods to the Euler and Navier-Stokes Equations*, Ph.D. thesis, Stanford University, Stanford, CA, March 2003.

[39] Yang, Z., Mavriplis, D. J., and Sitaraman, J., "Prediction of Helicopter Maneuver Loads Using BDF/Time Spectral Method on Unstructured Meshes," AIAA Paper 1122, Orlando, Florida, January 2011.

[40] Mundis, N. L. and Mavriplis, D. J., "Finite-element Time Discretizations for the Unsteady Euler Equations," AIAA Paper 0569, University of Wyoming, Kissimmee, Florida, January 2015.

[41] Choi, S., Potsdam, M., Lee, K., Iaccarino, G., and Alonso, J. J., "Helicopter Rotor Design Using a Time-Spectral and Adjoint-Based Method," AIAA Paper 5810, Victoria, British Columbia, September 2008.

[42] Peyret, R., *Spectral Methods for Incompressible Viscous Flow*, Applied Mathematical Sciences, Springer, New York, 2002.

[43] Hesthaven, J. S., Gottlieb, S., and Gottlieb, D., *Spectral Methods for Time-Dependent Problems*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2007.

[44] Huang, H. and Ekici, K., "Stabilization of High-Dimensional Harmonic Balance Solvers Using a Temporal Spectral Viscosity Operator," AIAA Paper 0354, Grapevine, Texas, January 2013.

[45] Leffell, J., *An Overset Time-Spectral Method for Relative Motion*, Ph.D. thesis, Stanford University, June 2014.

[46] Leffell, J. I., Murman, S. M., and Pulliam, T. H., "Time-Spectral Rotorcraft Simulations on Overset Grids," AIAA Paper 3258, Atlanta, GA, June 2014.

[47] Leffell, J. I., Murman, S. M., and Pulliam, T. H., "An Extension of the Time-Spectral Method to Overset Solvers," AIAA Paper 0637, Grapevine, Texas, January 2013.

[48] Buelow, P., Venkateswaran, S., and Merkle, C., "Stability and convergence of implicit upwind schemes," *Computers & Fluids*, Vol. 30, 2001, pp. 961–988.

[49] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. ScI. STAT. COMPT.*, Vol. 7, No. 3, July 1986, pp. 856–869.

[50] Lakshminarayan, V. K., Roget, B., and Sitaraman, J., "A Multi-Strand Mesh Generation and Solver Framework for Complex Aerodynamics Calculations," Abstract Submitted for AIAA Aerospace Science Meeting, San Diego, California, January 2016.

[51] Sitaraman, J., Katz, A., Jayaraman, B., Wissink, A. M., and Sankaran, V., "Evaluation of a Multi-Solver Paradigm for CFD using Overset Unstructured and Structured Adaptive Cartesian Grids," AIAA Paper 0660, Reno, Nevada, January 2008.

[52] Wissink, A., Sitaraman, J., Roget, B., Potsdam, M., Lakshminarayan, V., Jain, R., Forsythe, J., and Bauer, A., "Recent Advancements in the Helios Rotorcraft Simulation Code," AIAA Paper TBD, San Diego, CA, January 2016.

[53] Roget, B. and Sitaraman, J., "Robust and efficient overset grid assembly for partitioned unstructured meshes," *Journal of Computational Physics*, Vol. 260, 2014, pp. 1 – 24.

[54] Brazell, M. J., Sitaraman, J., and Mavriplis, D. J., "An Overset Mesh Approach for 3D Mixed Element High Order Discretizations," AIAA 2015-1739, Kissimmee, Florida, January 2015.

[55] Thomas, J. P., Custer, C. H., Dowell, E. H., and Hall, K. C., "Unsteady Flow Computation Using a Harmonic Balance Approach Implemented about the OVERFLOW 2 Flow Solver," AIAA Paper 4270, San Antonio, Texas, June 2009.

[56] Custer, C. H., *A Nonlinear Harmonic Balance Solver for an Implicit CFD Code: OVERFLOW 2*, Ph.D. thesis, Duke University, 2009.

[57] Thomas, J. P., Custer, C. H., Dowell, E. H., Hall, K. C., and Corre, C., "Compact Implementation Strategy for a Harmonic Balance Method Within Implicit Flow Solvers," *AIAA Journal*, Vol. 51, No. 6, June 2013, pp. 1374–1381.

[58] "libPFASST," `https://bitbucket.org/memmett/libpfasst`.

[59] Minion, M., Speck, R., Bolten, M., Emmett, M., and Ruprecht, D., "Interweaving PFASST and Parallel Multigrid," *SIAM Journal on Scientific Computing*, Vol. 37, No. 5, 2015, pp. 244–263.

[60] Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B., "Parallel Time Integration with Multigrid," *SIAM J. Sci. Comp.*, Vol. 36, No. 6, 2014, pp. C635–C661.

[61] Mundis, N. L. and Mavriplis, D. J., "GMRES applied to the Time-spectral and Quasi-periodic Time-spectral Methods," AIAA Paper 3084, San Diego, California, June 2013.

[62] Lakshminarayan, V. K., Duraisamy, K., and Alonso, J. J., "Adjoint-Based Estimation and Control of Spatial, Temporal and Stochastic Approximation Errors for Aerodynamic Applications," AIAA Paper 0518, Grapevine, Texas, January 2013.

[63] Su, X. and Yuan, X., "Implicit solution of time spectral method for periodic unsteady flows," *International Journal for Numerical Methods in Fluids*, Vol. 63, No. 7, 2010, pp. 860–876.

[64] Mundis, N. L. and Mavriplis, D. J., "Wave-number Independent Preconditioning for GMRES Time-spectral Solvers," AIAA Paper 0568, University of Wyoming, Kissimmee, Florida, January 2015.

[65] Mundis, N. L. and Mavriplis, D. J., "Toward an Optimal Solver for Time-spectral Solutions on Unstructred Meshes," AIAA Paper TBD, San Diego, CA, January 2016.

[66] Mundis, N. L. and Mavriplis, D. J., "An Efficient Flexible GMRES Solver for the Fully-coupled Time-Spectral Aeroelastic System," AIAA Paper 1427, National Harbor, Maryland, January 2014.

[67] Wissink, A. M., Hornung, R. D., Kohn, S. R., Smith, S. S., and Elliott, N., "Large scale parallel structured AMR calculations using the SAMRAI framework," *Supercomputing, ACM/IEEE 2001 Conference*, IEEE, 2001, pp. 22–22.

[68] Rajagopalan, R. G. and Fanucci, J. B., "Finite difference model for vertical axis wind turbines," *Journal of Propulsion and Power*, Vol. 1, No. 6, 1985, pp. 432–436.

[69] Gupta, V., *Quad tilt rotor simulations in helicopter mode using computational fluid dynamics*, Ph.D. thesis, 2005.

[70] O'Brien Jr, D. M., *Analysis of computational modeling techniques for complete rotorcraft configurations*, Ph.D. thesis, 2006.

[71] Troldborg, N., *Actuator Line Modeling of Wind Turbine Wakes*, Ph.D. thesis, Technical University of Denmark, 2008.

[72] Churchfield, M. J., Lee, S., Moriarty, P. J., Martinez, L. A., Leonardi, S., Vijayakumar, G., and Brasseur, J. G., "A large-eddy simulation of wind-plant aerodynamics," *AIAA paper*, Vol. 537, 2012.

[73] Jha, P. K., Churchfield, M. J., Moriarty, P. J., and Schmitz, S., "Guidelines for volume force distributions within actuator line modeling of wind turbines on large-eddy simulation-type grids," *Journal of Solar Energy Engineering*, Vol. 136, No. 3, 2014, pp. 031003.

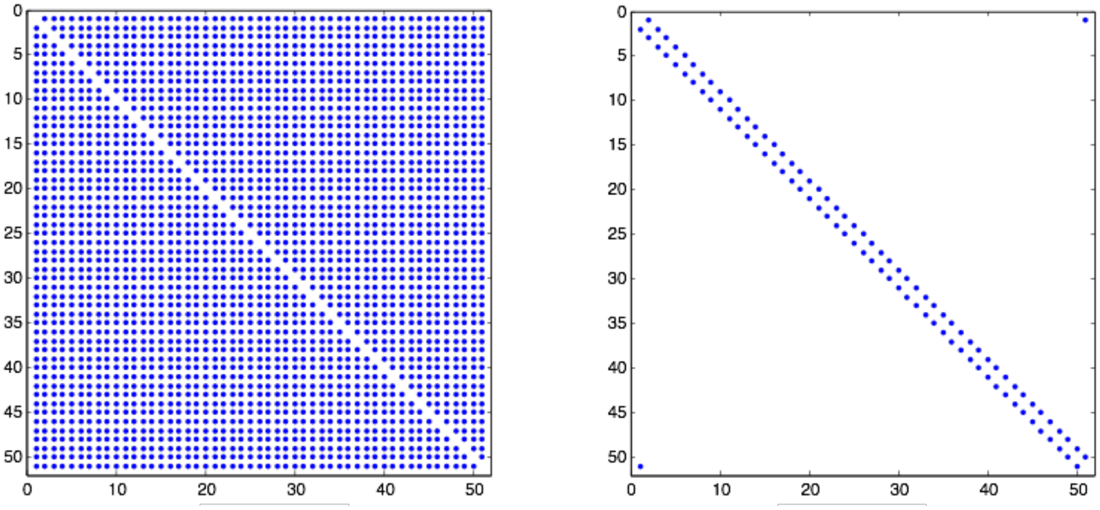
[74] Gopalan, H., Gundling, C., Brown, K., Roget, B., Sitaraman, J., Mirocha, J. D., and Miller, W. O., "A coupled mesoscale–microscale framework for wind resource estimation and farm aerodynamics," *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 132, 2014, pp. 13–26.

[75] Gundling, C., Sitaraman, J., Roget, B., and Masarati, P., "Application and validation of incrementally complex models for wind turbine aerodynamics, isolated wind turbine in uniform inflow conditions," *Wind Energy*, 2014.

[76] Mavriplis, D. J., Yang, Z., and Mundis, N., "Extensions of Time Spectral Methods for Practical Rotorcraft Problems," AIAA Paper 0423, Nashville, Tennessee, January 2012.

[77] Hand, M., Simms, D., Fingersh, L., Jager, D., Cotrell, J., Schreck, S., and Larwood, S., "Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns," NREL/TP-500-29955, National Renewable Energy Laboratory, December 2001.

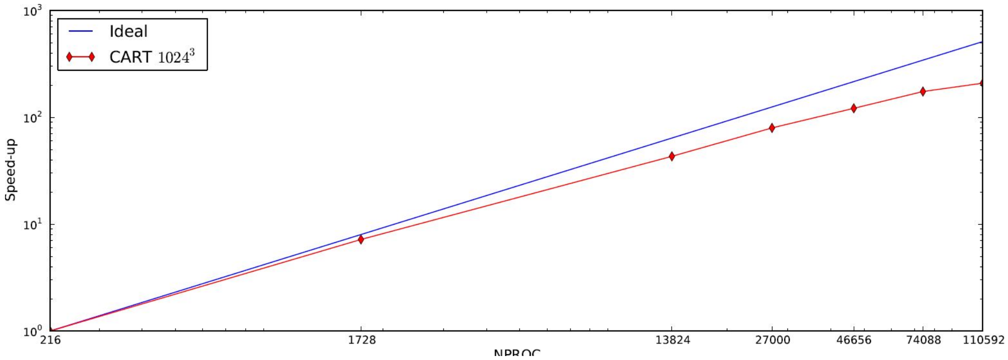
[78] Debonis, J. R., "Solutions of the Taylor-Green Vortex Problem Using High Resolution Explicit Finitie Difference Methods," AIAA Paper 0382, NASA Glenn Research Center, Grapevine, Texas, January 2013.



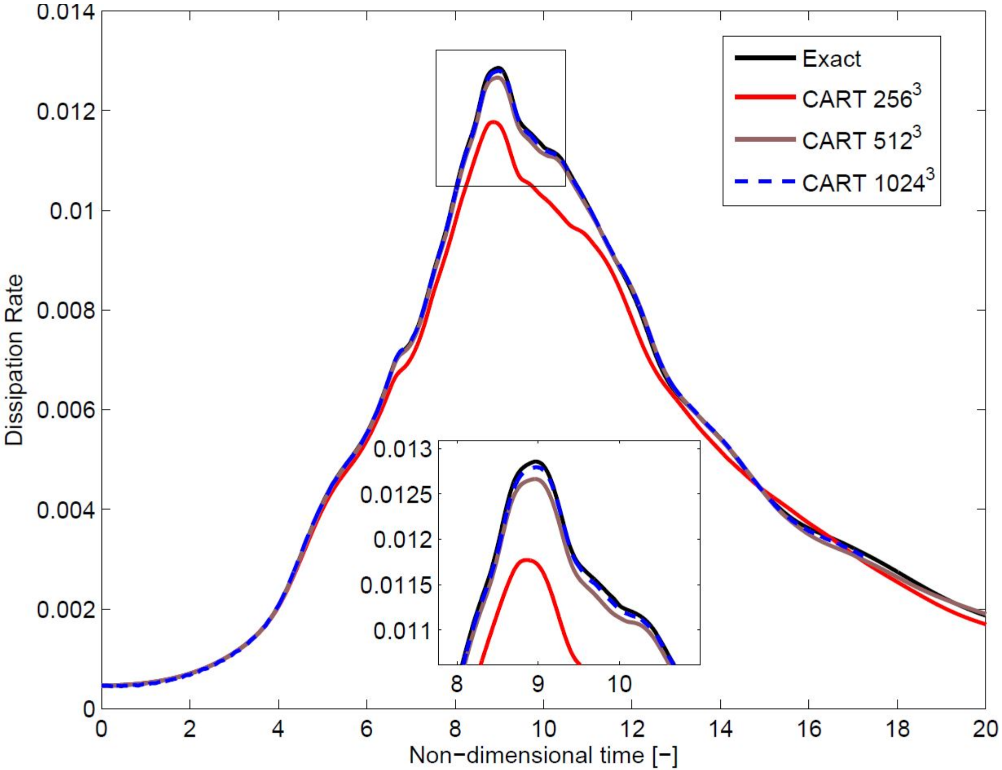
(a) Time-Spectral

(b) FDMT-2

Figure 1: Nonzero entries for temporal differentiation operators. In the current PinT framework, each nonzero entry requires inter-processor communication. The bandwidth of the TS operator grows with N but the bandwidth is fixed to $p/2$ for a p-th order central difference operator enabling the algorithm to scale weakly.

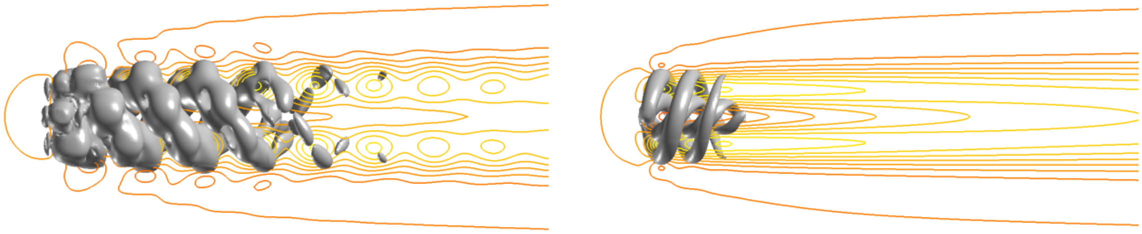


(a) Speedup



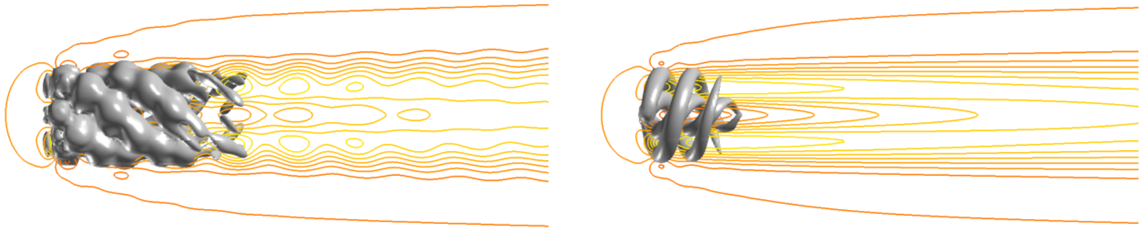
(b) High-order Resolution

Figure 2: Cartesian flow solver (CART) applied to canonical Taylor-Green vortex problem using a high-order discretization (sixth-order accurate inviscid flux and fourth-order viscous flux discretizations) with RK3 time integration. (a) Parallel speedup for a grid consisting of more than one billion points $\left(1024^3\right)$ up to 110,592 cores. (b) High-order resolution approaching *exact* spectral solution [78].



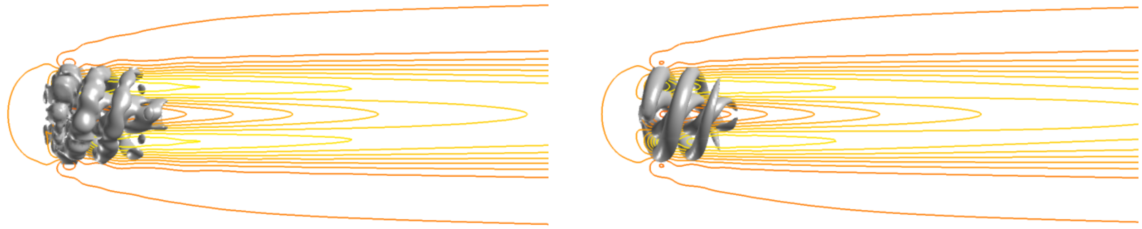
(a) TS, $N = 3$

(b) BDF2, $\Delta t = T/180$



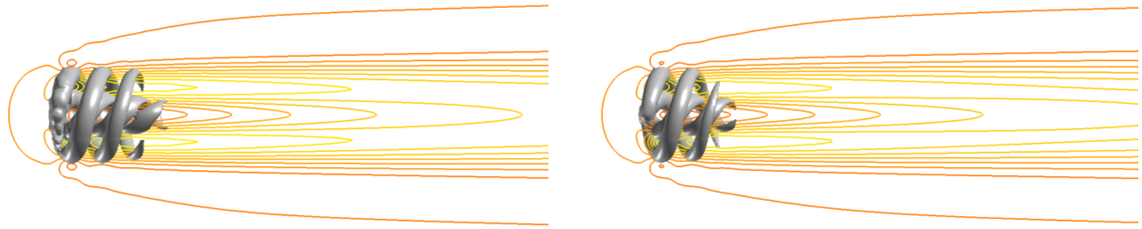
(c) TS, $N = 5$

(d) BDF2, $\Delta t = T/360$



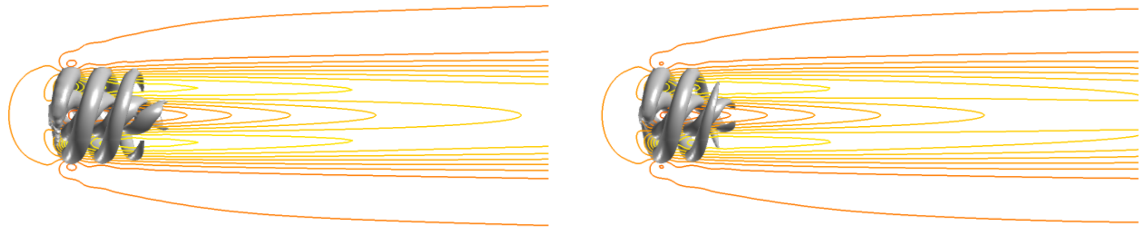
(e) TS, $N = 9$

(f) BDF2, $\Delta t = T/720$



(g) TS, $N = 17$

(h) BDF2, $\Delta t = T/1440$



(i) TS, $N = 33$

(j) BDF2, $\Delta t = T/2880$

Figure 3: NREL Phase VI Wind Turbine. Time-Spectral and BDF2 Solutions. Axial velocity contours and iso-contours of vorticity magnitude for Time-Spectral calculations using $N \in \{3, 5, 9, 17, 33\}$ and BDF2 calculations using a physical time step, $\Delta t = T/N$, for $N \in \{180, 360, 720, 1440, 2880\}$.

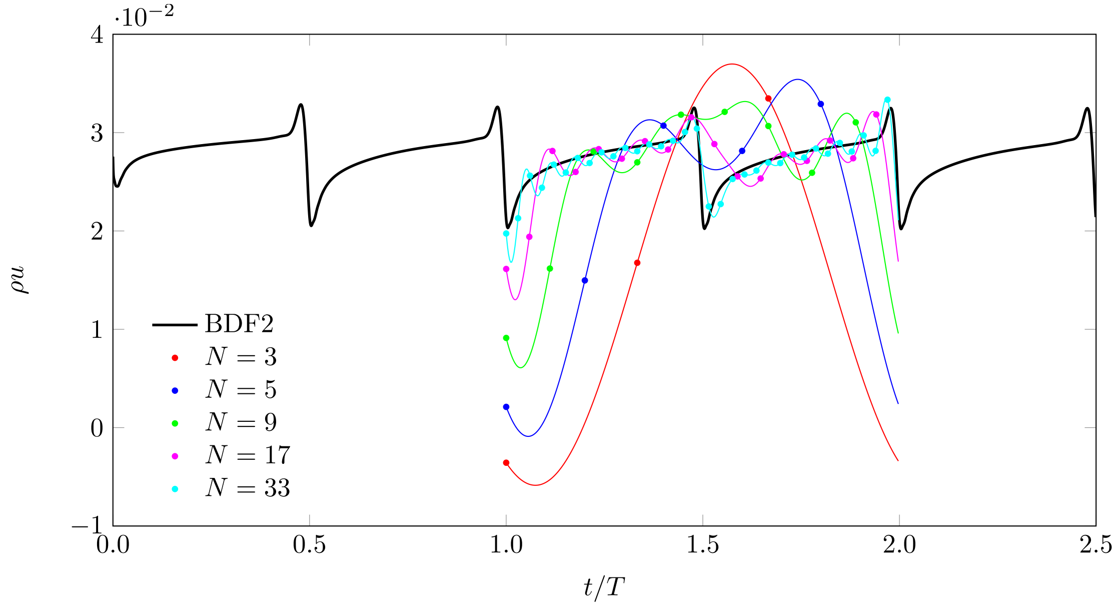
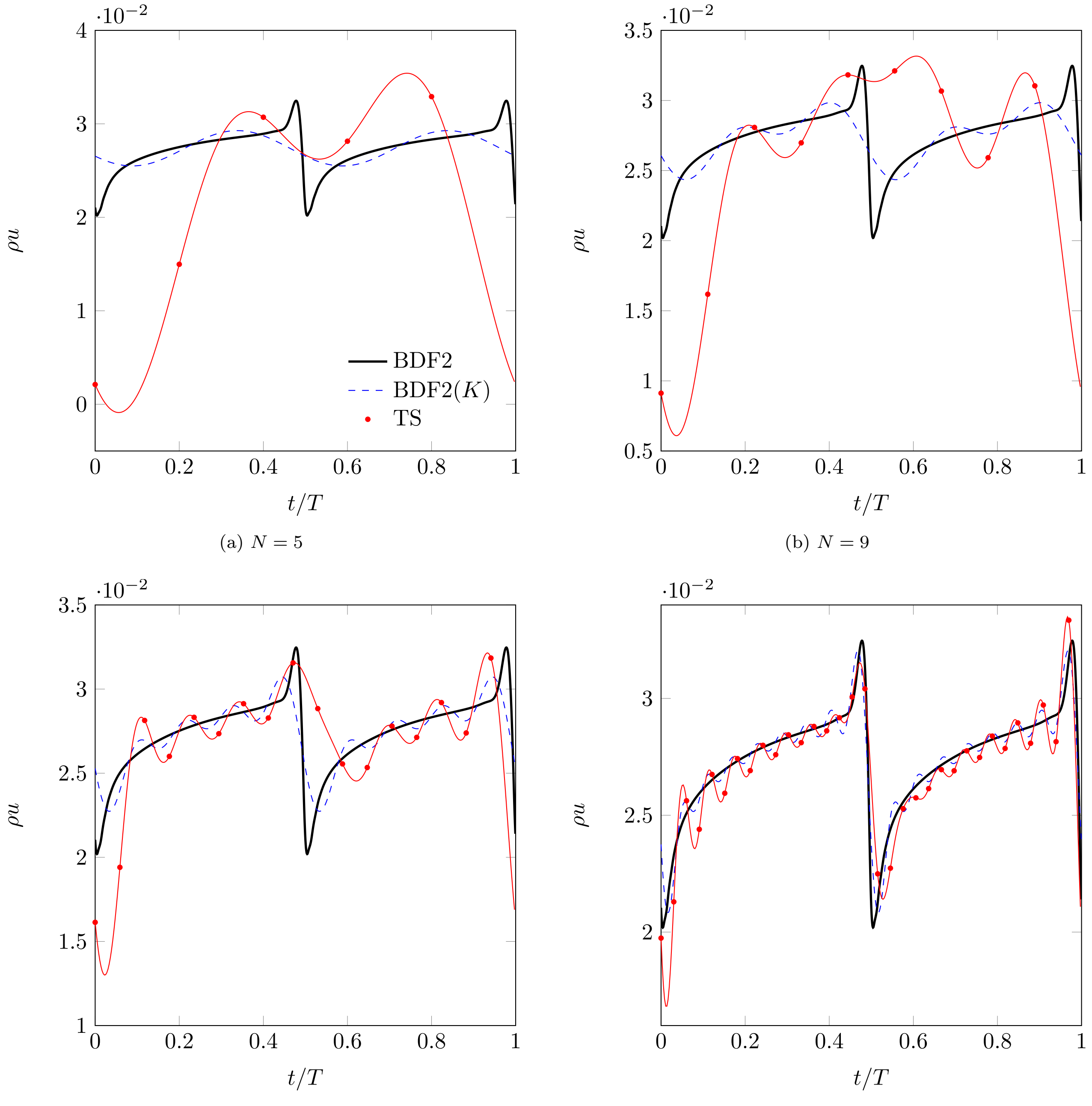


Figure 4: NREL Phase VI Wind Turbine. Space-Time Approximate Factorization Time-Spectral Method. Values of streamwise momentum, ρu, sampled at a sensor point close to the rotor plane are plotted for $N \in \{3, 5, 9, 17, 33\}$ at their respective collocation points in addition to their Fourier reconstructions with solid lines of the corresponding color. The BDF2 solution is plotted for the first 2.5 periods of turbine rotation.



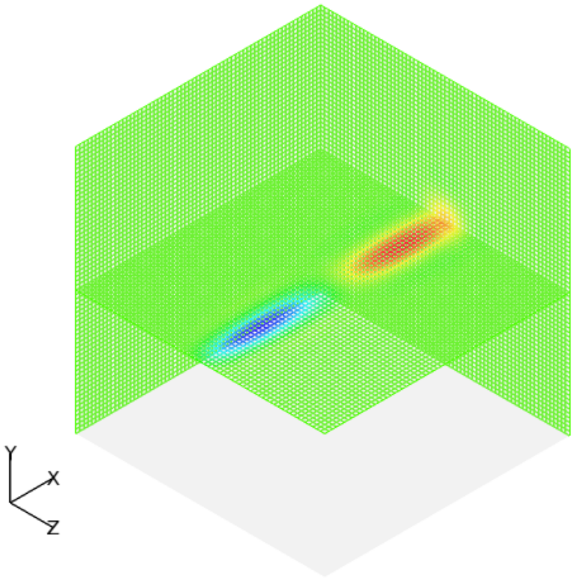
(a) $N = 5$

(b) $N = 9$

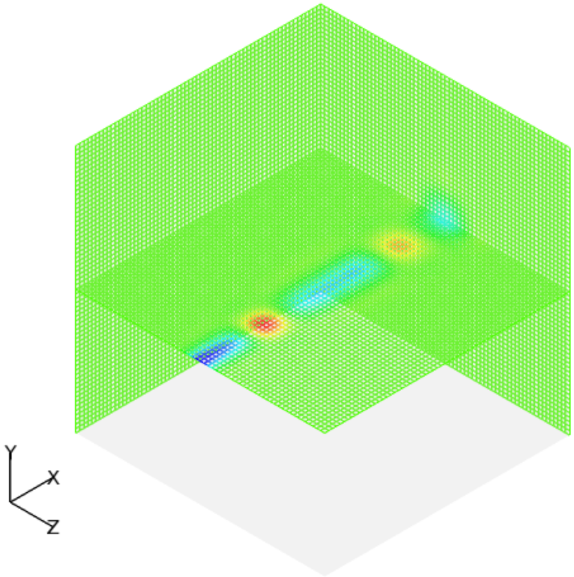
(c) $N = 17$

(d) $N = 33$

Figure 5: NREL Phase VI Wind Turbine. Space-Time Approximate Factorization Time-Spectral Method. Values of streamwise momentum, ρu, sampled at a sensor point close to the rotor plane are plotted for $N \in \{5, 9, 17, 33\}$ at their respective collocation points in addition to their Fourier reconstructions with solid red lines. The black curve represents the BDF2 solution and the hashed blue represents the filtered BDF solution to $K = (N-1)/2$ modes.

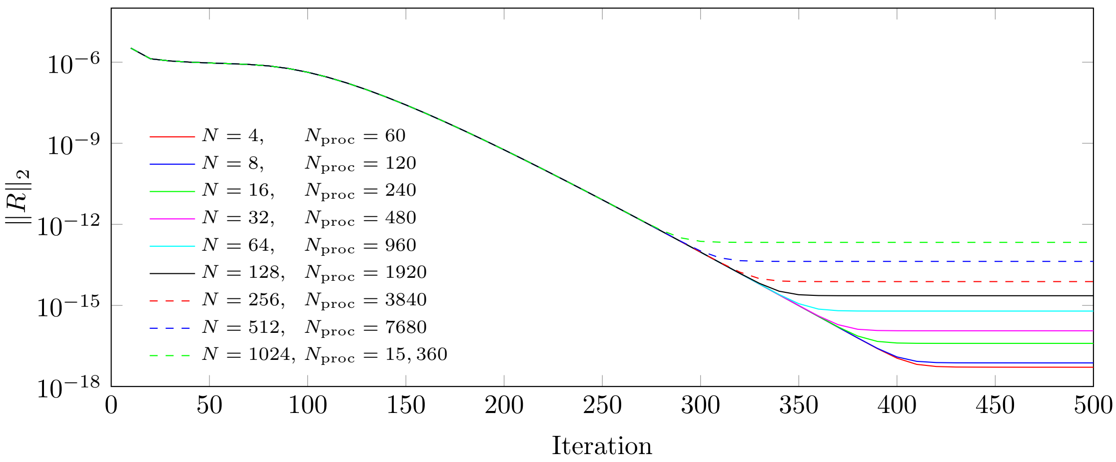


(a) Single-frequency kernel, $\gamma(t) = \sin(\omega t)$

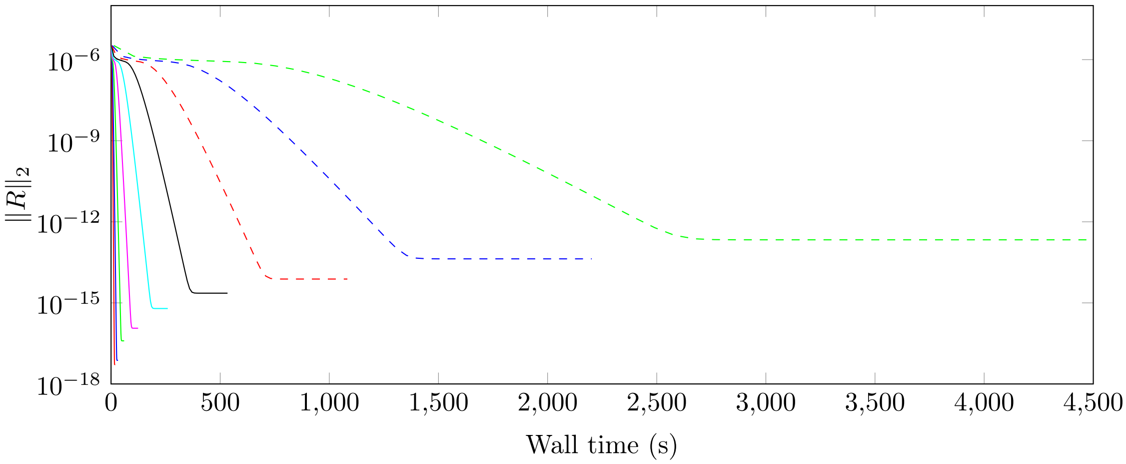


(b) Multiple-frequency kernel, $\gamma(t) = \frac{2}{5+4\cos(2\omega t)} - 1$

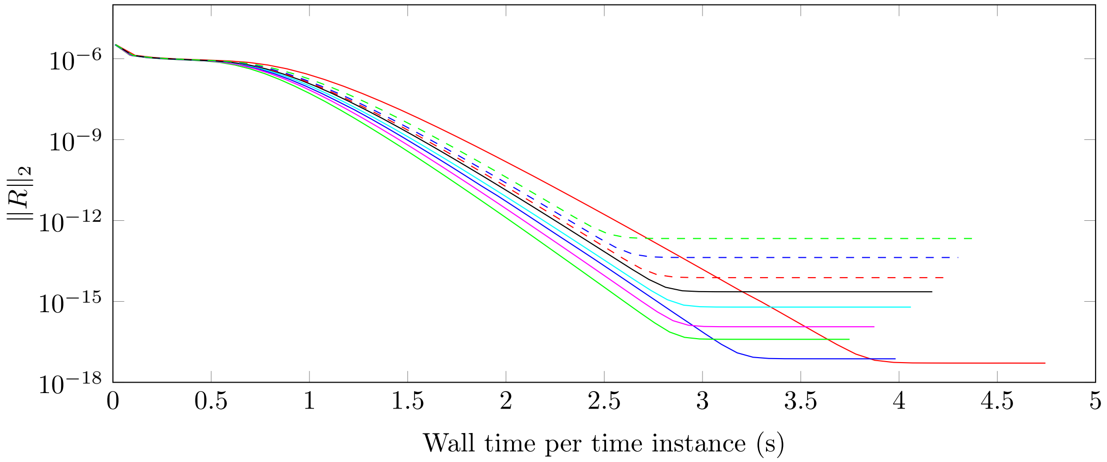
Figure 6: Periodic Density Pulse. Grid and contours of density on constant y-plane for both the (a) single frequency and (b) multiple frequency kernels at time $t = 0$.



(a) Space-time residual versus iteration

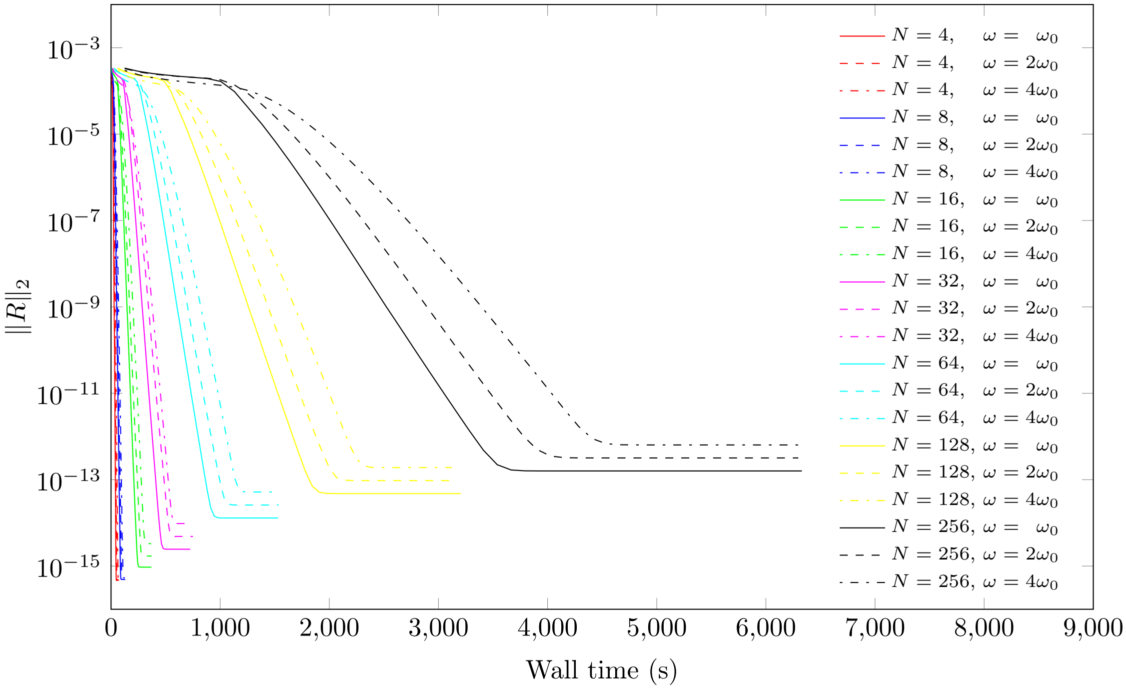


(b) Space-time residual versus wall time

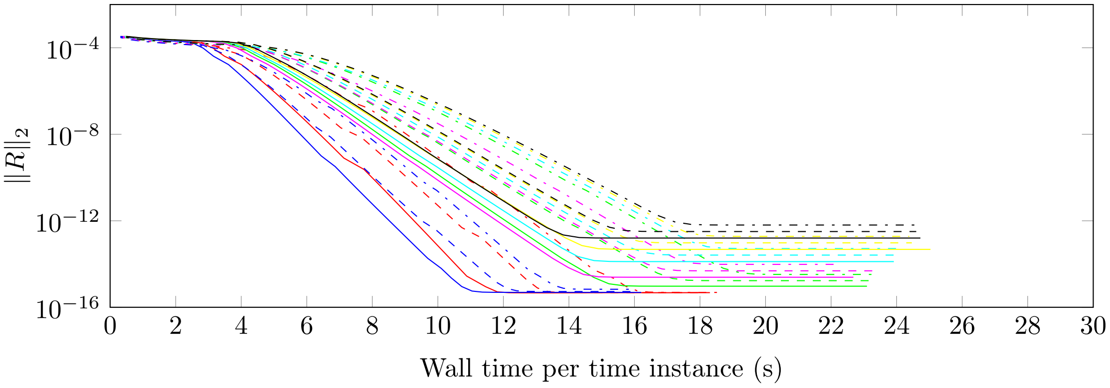


(c) Space-time residual versus wall time per time instance

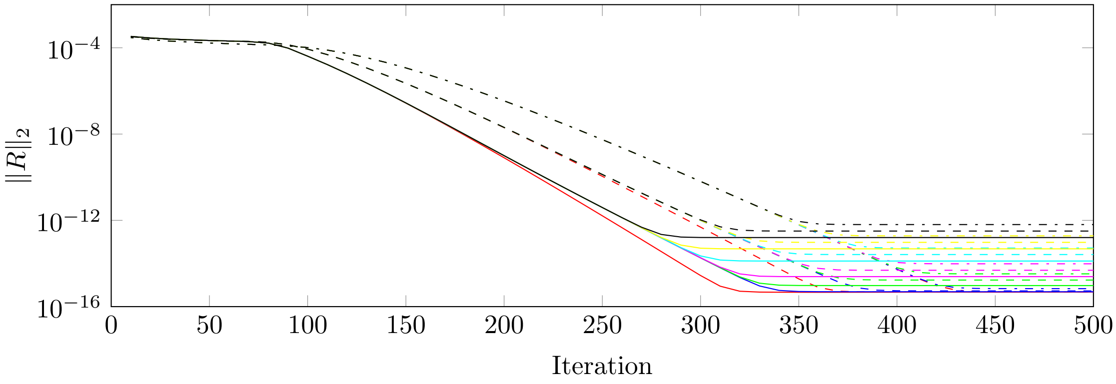
Figure 7: Periodic Density Pulse. Scaling of the Space-Time Approximate Factorization Time-Spectral Method applied to the single-frequency kernel case. Convergence of residual for different number of time instances, N. Residual versus (a) iteration (b) wall time and (c) wall time per temporal degree of freedom. This case used a reduced frequency of $\omega = 1$ on a 50^3 mesh partitioned into fifteen blocks. Thus the overall number of processors $N_{\text{proc}} = 15N$ for each calculation.



(a) Space-time residual versus wall time

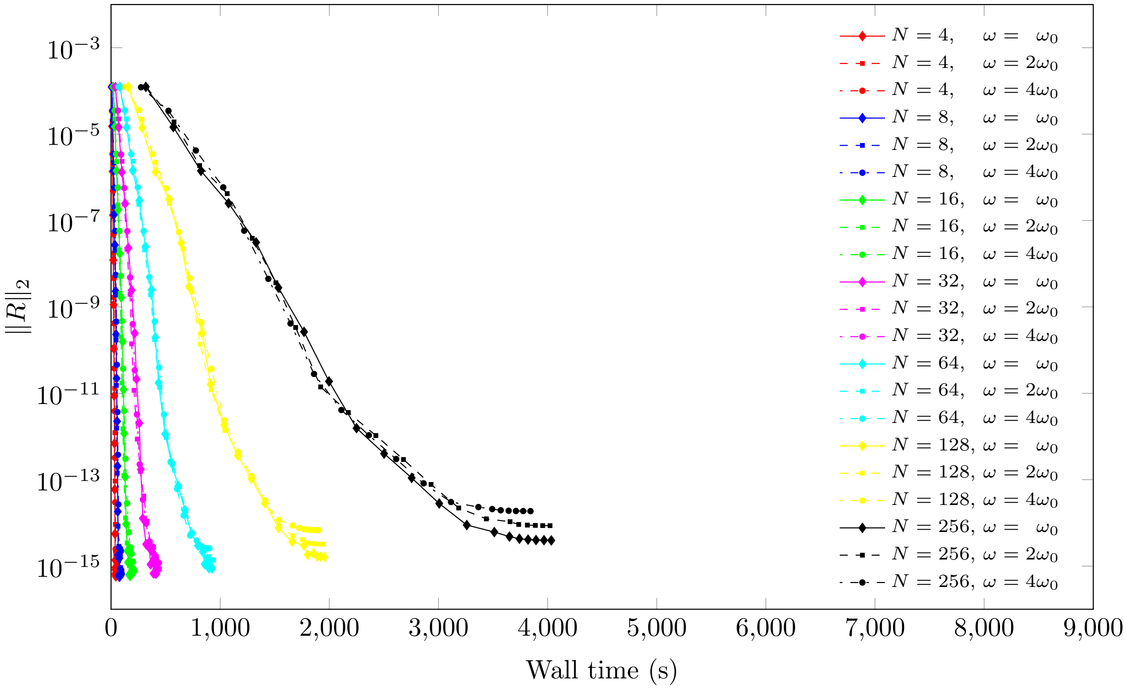


(b) Space-time residual versus wall time per time instance

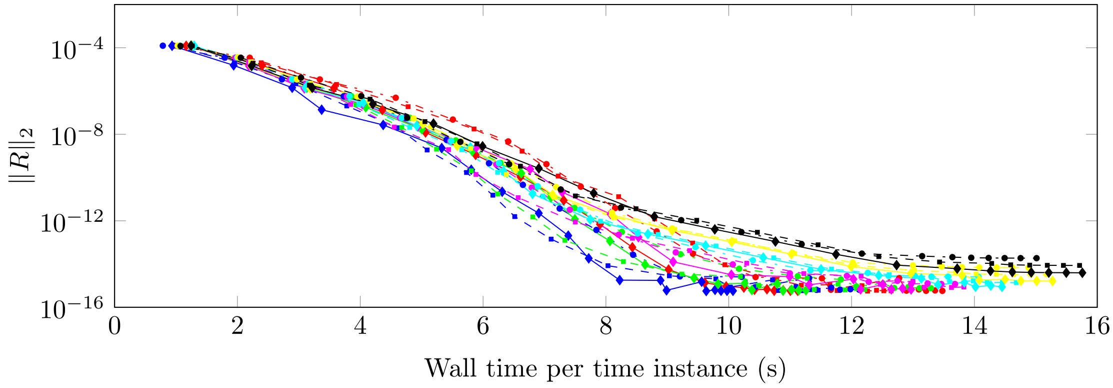


(c) Space-time residual versus iteration

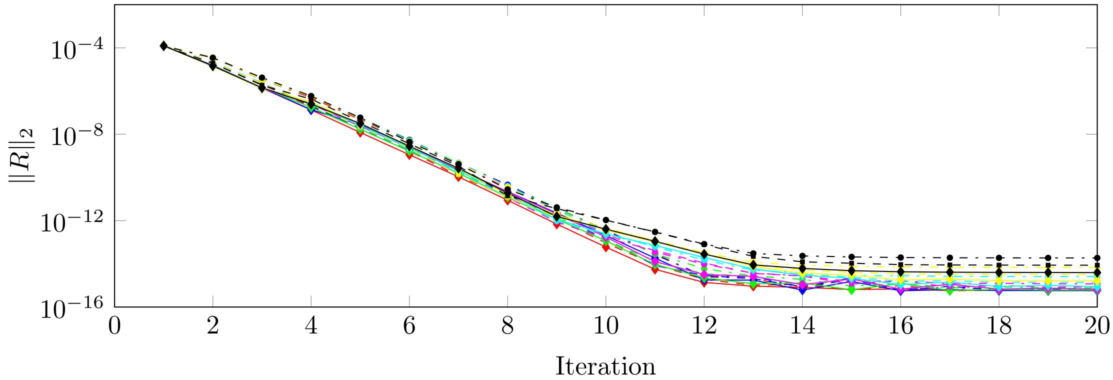
Figure 8: Periodic Density Pulse. Scaling of the Time-Spectral method using the space-time approximate factorization solver applied to the single-frequency kernel case. Convergence of residual for different number of time instances, N, and frequencies, ω. Space-time residual versus (a) wall time (b) wall time per time instance and (c) iteration. The base frequency is defined as: $\omega_0 = \omega/a$, where a is the speed of sound and $\omega = 7.54$ rad/s is the radial frequency of the NREL Phase VI wind turbine in §VI.A. The 65^3 mesh is partitioned into twelve blocks for a total of $12N$ processors for each calculation.



(a) Space-time residual versus wall time

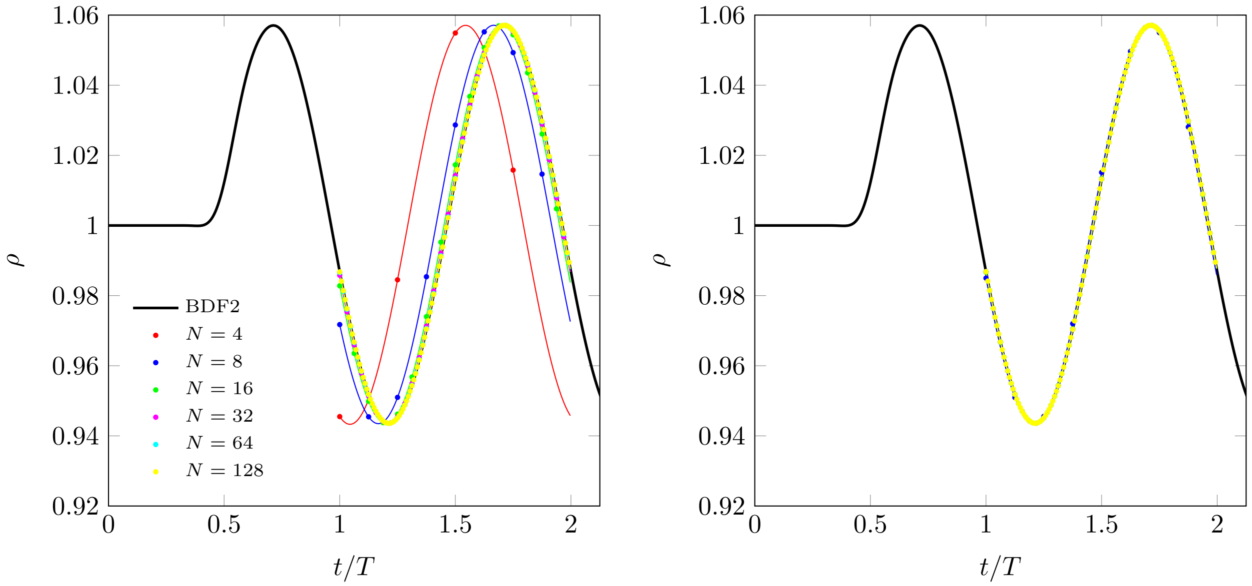


(b) Space-time residual versus wall time per time instance



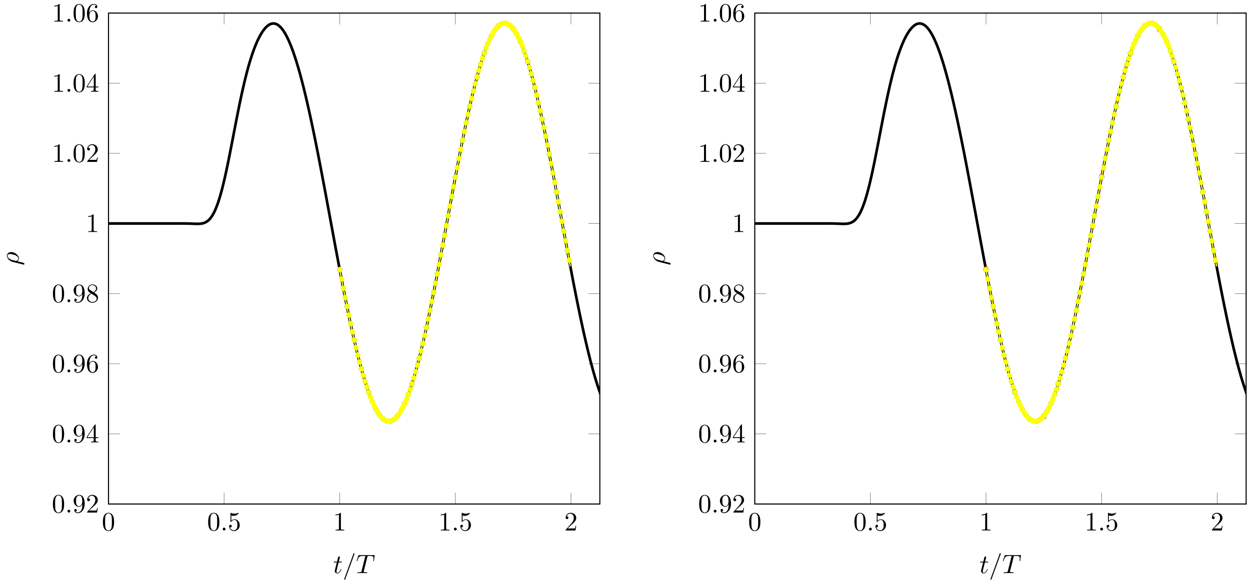
(c) Space-time residual versus non-linear iteration

Figure 9: Periodic Density Pulse. Scaling of the GMRES Time-Spectral method using a space-time approximate factorization preconditioner applied to the single-frequency kernel case. Convergence of residual for different number of time instances, N, and frequencies, ω. Space-time residual versus (a) wall time (b) wall time per time instance and (c) non-linear iteration. The base frequency is defined as: $\omega_0 = \omega/a$, where a is the speed of sound and $\omega = 7.54$ rad/s is the radial frequency of the NREL Phase VI wind turbine in §VI.A. The 65^3 mesh is partitioned into twelve blocks for a total of $12N$ processors for each calculation.



(a) FDMT-2

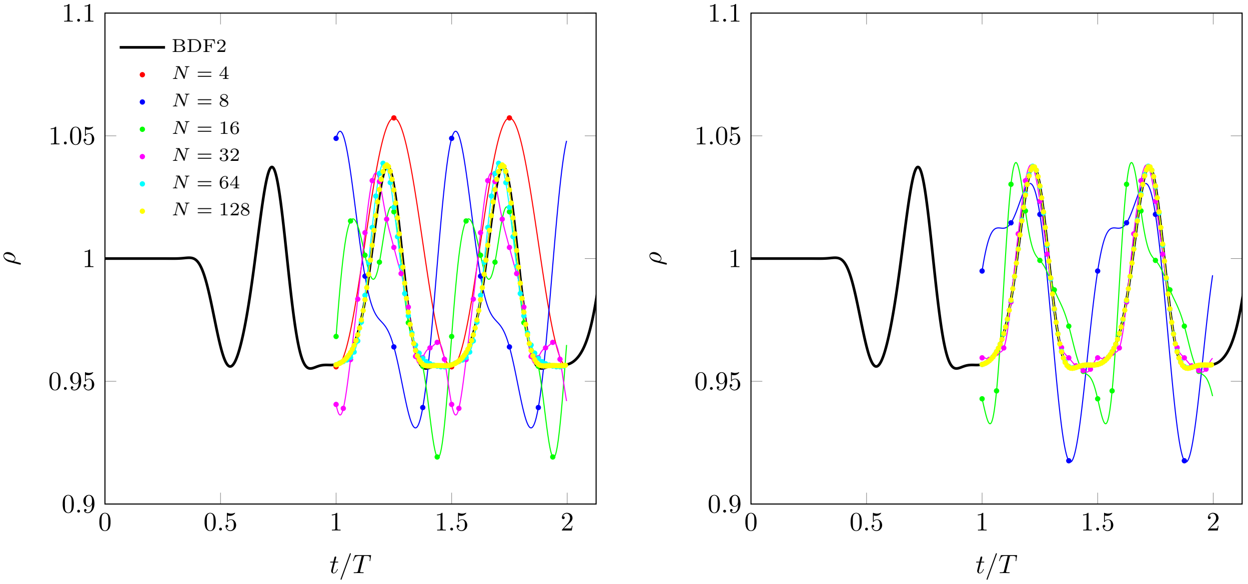
(b) FDMT-4



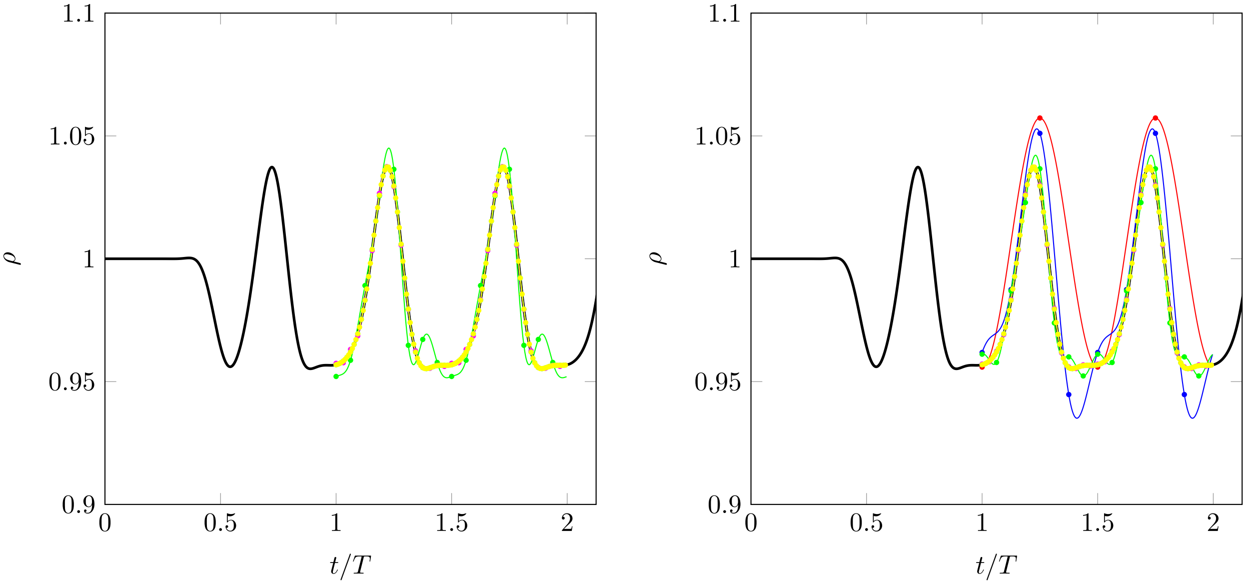
(c) FDMT-8

(d) TS

Figure 10: Periodic Density Pulse. Subiterative Space-Time Scheme applied to the single-frequency kernel, $\gamma(t) = \sin(\omega t)$. BDF2 (black curve), Time-Spectral and FDMT solutions for $N \in \{4, 8, 16, 32, 64, 128\}$ are plotted. Density is sampled at a sensor located in the center of the domain. The dots correspond to the solution at the N time instances and the solid lines represent the reconstruction using the discrete Fourier series for all cases. The $N = 4$ case and $N = 4, 8$ cases are not plotted in the respective FDMT-4 and FDMT-8 figures, respectively, as the bandwidth is not large enough for the specified number of time samples.

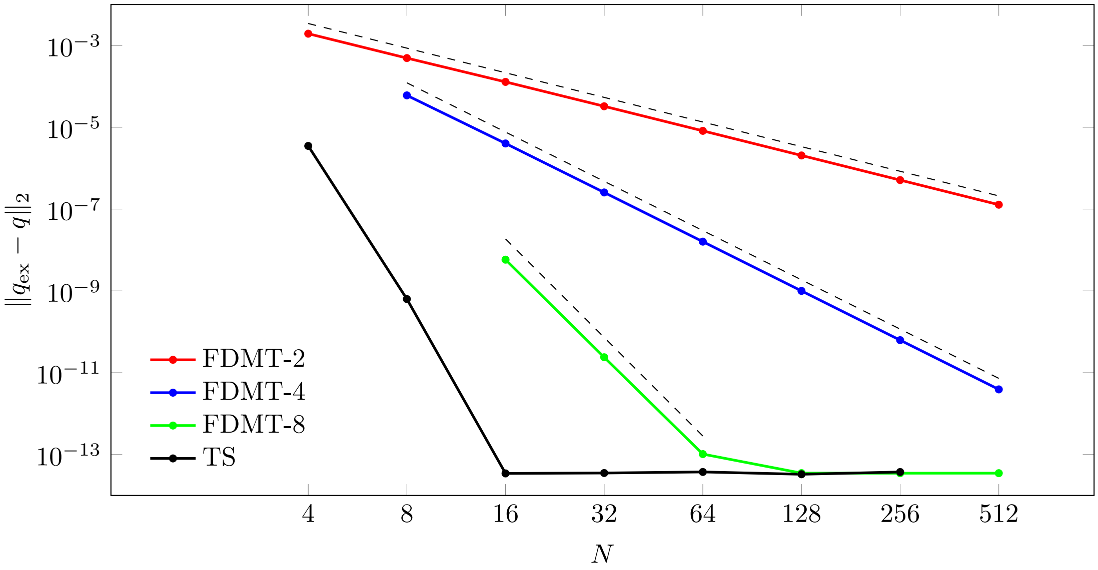


(a) FDMT-2 (b) FDMT-4

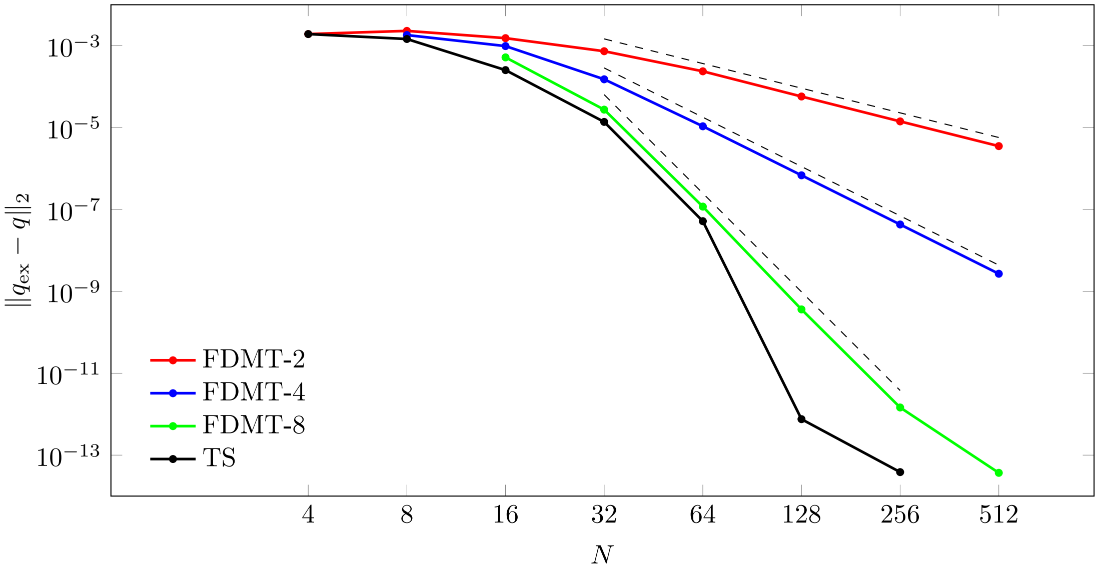


(c) FDMT-8 (d) TS

Figure 11: Periodic Density Pulse. Subiterative Space-Time Scheme applied to the multiple-frequency kernel, $\gamma(t) = \frac{2}{5+4\cos(2\omega t)} - 1$. BDF2 (black curve), Time-Spectral and FDMT solutions for $N \in \{4, 8, 16, 32, 64, 128\}$ are plotted. Density is sampled at a sensor located in the center of the domain. The dots correspond to the solution at the N time instances and the solid lines represent the reconstruction using the discrete Fourier series for all cases. The $N = 4$ case and $N = 4, 8$ cases are not plotted in the respective FDMT-4 and FDMT-8 figures, respectively, as the bandwidth is not large enough for the desired number of samples.



(a) Single frequency, $\gamma(t) = \sin(\omega t)$



(b) Multiple frequencies, $\gamma(t) = \frac{2}{5+4\cos(2\omega t)} - 1$

Figure 12: Periodic Density Pulse. Order-of-accuracy verification for the (a) single-frequency and (b) multiple frequency cases using the subiterative space-time scheme. The *exact* solution, q_{ex}, is taken as the TS solution for $N = 512$; error is evaluated at time $t = 0$. Hashed lines correspond to the expected order of accuracy. Each grid of 65^3 mesh points is partitioned into six blocks for a total of $6N$ processors for each calculation.

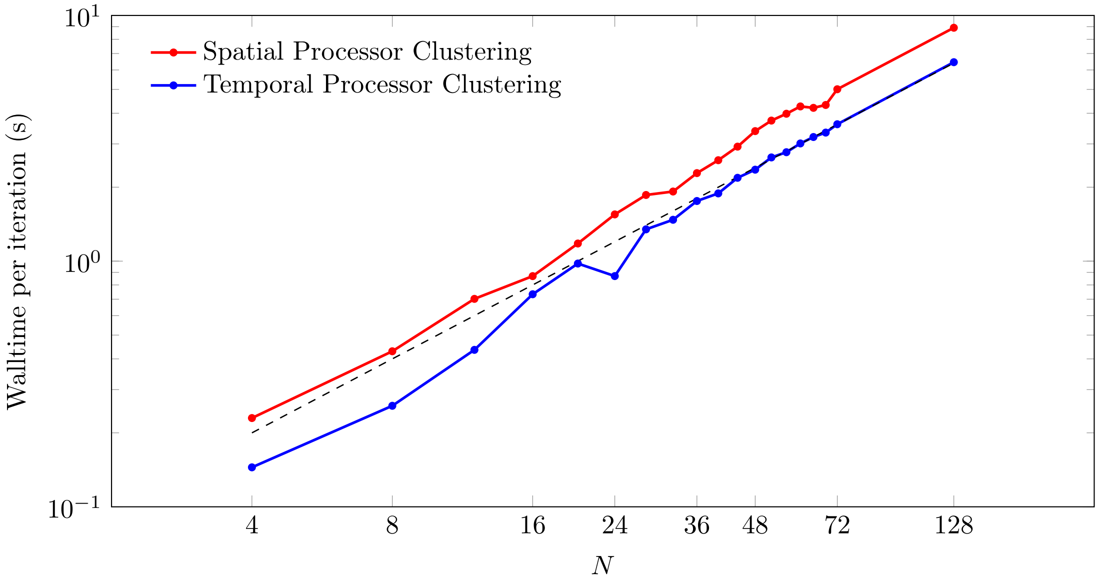


Figure 13: Periodic Density Pulse. Space-Time Approximate Factorization Time-Spectral Method. Wall time per iteration is plotted for both spatial and temporal processor clustering. All computation was executed on machine equipped with twenty-four cores per node. Each grid of 65^3 mesh points is partitioned into twelve blocks for a total of $12N$ processors for each calculation. While the temporal clustering exhibits better performance than the spatial clustering topology, the particular cases of $N = 4, 8, 12$ and 24 enable the entire time history of blocks to be packed into a single node and exhibit further reduction in wall time per iteration.

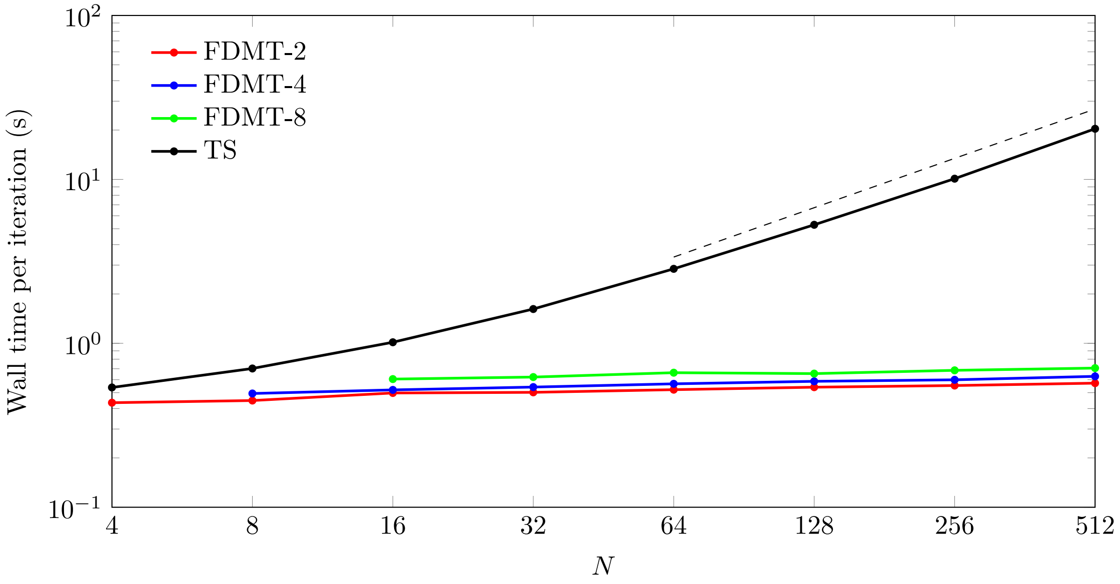
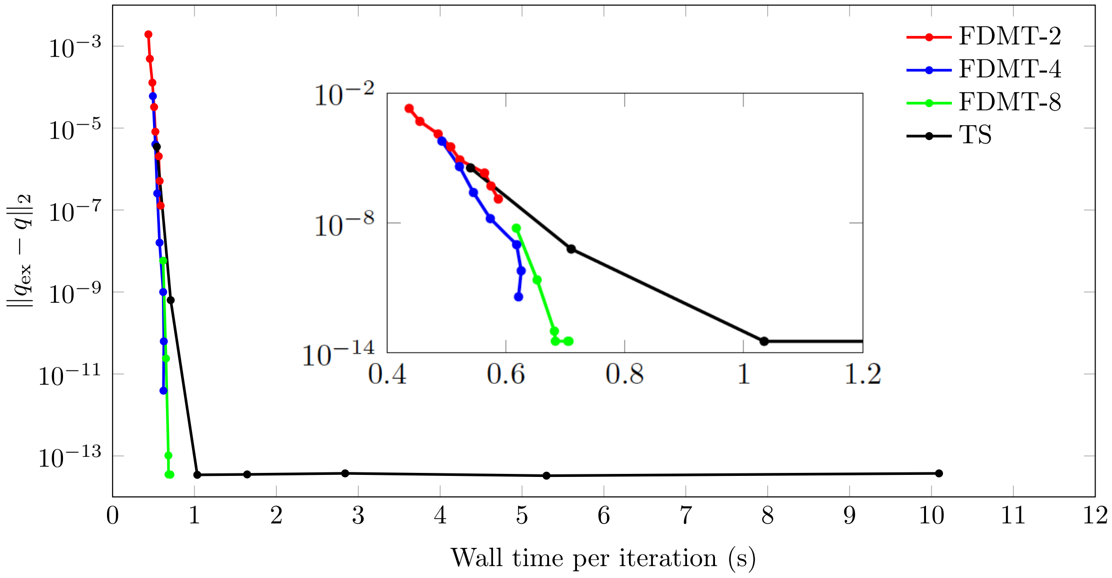
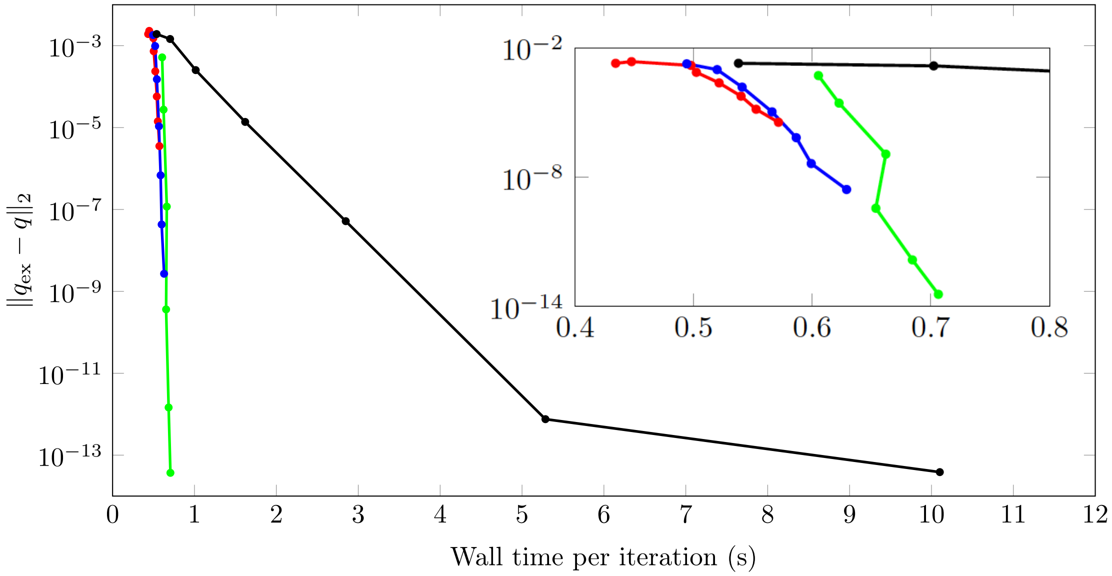


Figure 14: Periodic Density Pulse. FDMT versus Time-Spectral using the subiterative space-time solver. Wall time per iteration for $N \in \{4, 8, 16, 32, 64, 128, 256, 512\}$. The second-, fourth- and eighth-order accurate FDMT cases exhibit near-optimal weak scaling while the Time-Spectral case exhibits linear scaling (hashed black line). Each grid of 65^3 mesh points is partitioned into six blocks for a total of $6N$ processors for each calculation.



(a) Single frequency, $\gamma(t) = \sin(\omega t)$



(b) Multiple frequencies, $\gamma(t) = \frac{2}{5+4\cos(2\omega t)} - 1$

Figure 15: Periodic Density Pulse. Error versus wall time per iteration for the (a) single-frequency and (b) multiple frequency cases using the subiterative space-time scheme. The *exact* solution, q_{ex}, is taken as the TS solution for $N = 512$; error is evaluated at time $t = 0$. Each grid of 65^3 mesh points is partitioned into six blocks for a total of $6N$ processors for each calculation.