Wave-number Independent Preconditioning for GMRES Time-spectral Solvers

Nathan L. Mundis * Dimitri J. Mavriplis †

Department of Mechanical Engineering, University of Wyoming, Laramie, Wyoming 82071-3295

The time-spectral method applied to the Euler equations theoretically offers significant computational savings for purely periodic problems when compared to standard time-implicit methods. However, attaining superior efficiency with time-spectral methods over traditional time-implicit methods hinges on the ability rapidly to solve the large non-linear system resulting from time-spectral discretizations which become larger and stiffer as more time instances are employed. In order to increase the efficiency of these solvers, and to improve robustness, particularly for large numbers of time instances, the Generalized Minimal Residual Method (GMRES) is used to solve the implicit linear system over all coupled time instances. The use of GMRES as the linear solver makes the time-spectral methods more robust, allows them to be applied to a far greater subset of time-accurate problems, including those with a broad range of harmonic content, and vastly improves the efficiency of time-spectral methods. However, it has been shown in previous work that when the number of time instances and/or the reduced frequency of motion increases (i.e. the maximum resolvable wave-number increases), the convergence degrades rapidly, requiring many more total preconditioning iterations to reach a converged solution. To alleviate this convergence degradation, this work formulates a wave-number independent preconditioner by inverting the spatial-temporal diagonal blocks in the preconditioner instead of the spatial diagonal blocks individually, as has been done previously. The solver utilizing this wave-number independent preconditioner is shown to be more efficient than past solvers under all conditions, but especially for high reduced frequencies and large number of time instances, i.e. conditions with high maximum wave numbers.

I. Introduction

For problems with strong periodic content, such as turbomachinery flows or rotorcraft aerodynamics, time-spectral methods can be used to substantially reduce the cost of computing the full, time-dependent solution for a given level of accuracy. In many cases, time-spectral methods using only a small number of time instances per period can provide equivalent or superior accuracy, at substantially reduced cost, compared to traditional time-implicit solutions using hundreds of time steps per period. In other cases, many time instances per period may be needed to resolve both high and low frequency periodic content simultaneously.

Both the time-spectral and harmonic-balance methods are derived using discrete Fourier analysis. These methods, developed by Hall,[1] McMullen,[2,3] and Gopinath,[4,5] transform the unsteady equations in the physical domain to a set of steady equations in the frequency domain and then use the time-discretization operator to transform the frequency content back into a discrete number of time instances that reside in the time domain. Each of these time instances is coupled to all other time instances through the time-discretization operator, and the entire system is solved as a single, large, steady-state problem. In the literature, the time-spectral method has been shown to be faster than the dual-time stepping implicit methods using backwards difference time formulae for time periodic computations, such as turbomachinery flows,[2,5] oscillatory pitching airfoil/wing cases,[4,6] flapping wing,[7] helicopter rotor[8,9] and vortex shedding problems,[3] which all use lower numbers of time instances. However, for problems with large numbers of time instances, it is not yet clear if the theoretical efficiency gains of TS methods over time-implicit methods can be realized.

In order to converge the time-spectral system with large numbers of time instances efficiently and robustly, the Krylov subspace, Generalized Minimal Residual method[10] is utilized. By using GMRES, the disparate time instances are much more strongly coupled than they would be if a stationary iterative method such as the Jacobi method were

*Post-doctoral Research Associate, AIAA Member; email: nmundis@uwyo.edu.

†Professor, AIAA Associate Fellow; email: mavripl@uwyo.edu.

used as the main, linear solver. However, time-spectral problems can still become difficult to converge when the maximum resolvable wave-number becomes large, which occurs as the reduced frequency of motion increases and/or the number of time instances used is increased. To remedy this convergence degradation, a preconditioning method that allows the pseudo-time step size to be independent of wave-number is developed. This method inverts the complete spatial-temporal diagonal block at each preconditioning iteration. This is in contrast to the previously developed solver, which inverts only the uncoupled spatial-diagonal blocks for each time instance individually.[11,12] The present work focuses on how this newly developed preconditioner improves the convergence of the time-spectral method for problems with high maximum resolvable wave-numbers.

In the following sections, the necessary components of the time-spectral time discretization and the application of the new preconditioner paradigm to this system are presented. First, the governing equations and the base solver for the Euler equations are outlined. Then, the additions to the flow solver required to implement the time-spectral method are discussed. Next, the linearization of the time-spectral Euler equations and the solution of the resultant linear system using GMRES are examined. Subsequently, the newly implemented wave-number independent preconditioner is explained. Following, results are presented: first, using all of the previous preconditioning methods and then for the new preconditioner. Additionally, the solver utilizing the spatial-temporal diagonal-block inversion preconditioner is compared, in depth, to the previously best solver.[11] Finally, conclusion about the performance of the new solver are drawn, its limitations are outlined, and improvements for the future are planned.

II. Governing Equations

A. Base Solver

The Euler equations in conservative form can be written as:

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

where $\mathbf{U}$ represents the vector of conserved quantities (mass, momentum, and energy) and $\mathbf{F}(\mathbf{U})$ represents the convective fluxes. Integrating over a (moving) control volume $\Omega(t)$, the following is obtained:

$$\int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) \cdot \tilde{\mathbf{n}}) dS = 0 \tag{2}$$

Using the differential identity

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV = \int_{\Omega(t)} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{\partial\Omega(t)} \mathbf{U}(\dot{\mathbf{x}} \cdot \tilde{\mathbf{n}}) dS \tag{3}$$

where $\dot{\mathbf{x}}$ and $\tilde{\mathbf{n}}$ are the velocity and normal of the interface $\partial\Omega(t)$, respectively, equation (2) becomes:

$$\frac{\partial}{\partial t} \int_{\Omega(t)} \mathbf{U} dV + \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \mathbf{U}\dot{\mathbf{x}}) \cdot \tilde{\mathbf{n}} dS = 0 \tag{4}$$

Considering $\mathbf{U}$ as cell averaged quantities, these equations are discretized in space as:

$$\frac{\partial}{\partial t}(V\mathbf{U}) + \mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)) = 0 \tag{5}$$

where $\mathbf{R}(\mathbf{U}, \dot{\mathbf{x}}, \tilde{\mathbf{n}}) = \int_{\partial\Omega(t)} (\mathbf{F}(\mathbf{U}) - \dot{\mathbf{x}}\mathbf{U}) \cdot \tilde{\mathbf{n}} dS$ represents the discrete convective fluxes in ALE form and V denotes the control volume. In the discrete form, $\dot{\mathbf{x}}(t)$ and $\tilde{\mathbf{n}}(t)$ now represent the time varying velocities and surface normals of the control-volume boundary faces.

The Euler equations are discretized by a central difference finite-volume scheme with additional matrix-based artificial dissipation on hybrid meshes which may include triangles and quadrilaterals in two dimensions. Second-order accuracy, on smoothly varying meshes, is achieved using a two-pass construction of the artificial dissipation operator, which corresponds to an undivided biharmonic operator. A single unifying face-based data-structure is used in the flow solver for all types of elements. For a given face, the residual contribution of that face can be written as:

$$\mathbf{R}_{1st O,ik}(\mathbf{U}, \dot{\mathbf{x}}(t), \tilde{\mathbf{n}}(t)) = (\mathbf{F}_i(\mathbf{U}_i) + \mathbf{F}_k(\mathbf{U}_k) - \mathbf{U}_{ik}\dot{\mathbf{x}}_{ik}) \cdot \tilde{\mathbf{n}}\Delta S + \kappa \underline{\mathbf{T}} |\underline{\Lambda}| \underline{\mathbf{T}}^{-1} (\mathbf{U}_i - \mathbf{U}_k) \tag{6}$$

for first-order matrix dissipation, where $\underline{\mathbf{T}}$ is the left-eigenvector matrix, $\underline{\Lambda}$ is the eigenvalue matrix, and $\underline{\mathbf{T}}^{-1}$ is the right-eigenvector matrix of the convective fluxes. For second-order matrix dissipation, the residual on a face can be written as follows:

$$\mathbf{R}_{2ndO,ik}(\mathbf{U},\dot{\mathbf{x}}(t),\tilde{\mathbf{n}}(t)) = (\mathbf{F}_i(\mathbf{U}_i)+\mathbf{F}_k(\mathbf{U}_k)-\mathbf{U}_{ik}\dot{\mathbf{x}}_{ik})\cdot\tilde{\mathbf{n}}\Delta S+\kappa\underline{\mathbf{T}}|\underline{\Lambda}|\underline{\mathbf{T}}^{-1}(\mathbf{L}_i(\mathbf{U})-\mathbf{L}_k(\mathbf{U})) \tag{7}$$

where $\mathbf{L_p}(\mathbf{U})$ is the undivided Laplacian operator, taken as:

$$\mathbf{L}_p(\mathbf{U}) = \sum_{q=1}^{neighbors}(\mathbf{U}_q-\mathbf{U}_p) \tag{8}$$

In both cases, κ is an empirical constant with a typical value of $1/2$ for first-order matrix dissipation and $1/8$ for second-order matrix dissipation.

B. Time-spectral Method

If the flow is periodic in time, the variables $\mathbf{U}$ can be represented by a discrete Fourier series. The discrete Fourier transform of $\mathbf{U}$ in a period of T is given by[4]

$$\widehat{\mathbf{U}}_k = \frac{1}{N}\sum_{n=0}^{N-1}\mathbf{U}^n e^{-ik\frac{2\pi}{T}n\Delta t} \tag{9}$$

where N is the number of time intervals and $\Delta t = T/N$. The Fourier inverse transform is then given as

$$\mathbf{U}^n = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\widehat{\mathbf{U}}_k e^{ik\frac{2\pi}{T}n\Delta t} \tag{10}$$

It should be noted that $\frac{N}{2}$ is an integer division operation. Also note that this corresponds to a collocation approximation, i.e. the function $\mathbf{U}(t)$ is projected into the space spanned by the truncated set of complex exponential (spectral) functions, and the expansion coefficients (in this case the $\widehat{\mathbf{U}}_k$) are determined by requiring $\mathbf{U}(t)$ to be equal to its projection at N discrete locations in time, as given by equations (9) and (10). Differentiating equation (10) in time, the following is obtained:

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \frac{2\pi}{T}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} ik\widehat{\mathbf{U}}_k e^{ik\frac{2\pi}{T}n\Delta t} \tag{11}$$

Substituting equation (9) into equation (11), the final form of the time-spectral time-derivative term is produced:[13,14]

$$\frac{\partial}{\partial t}(\mathbf{U}^n) = \sum_{j=0}^{N-1} d_n^j \mathbf{U}^j \tag{12}$$

where

$$d_n^j = \begin{cases} \frac{2\pi}{T}\frac{1}{2}(-1)^{n-j}\cot(\frac{\pi(n-j)}{N}) & n\neq j \\ 0 & n=j \end{cases} \tag{13}$$

for an even number of time instances and

$$d_n^j = \begin{cases} \frac{2\pi}{T}\frac{1}{2}(-1)^{n-j}\csc(\frac{\pi(n-j)}{N}) & n\neq j \\ 0 & n=j \end{cases} \tag{14}$$

for an odd number of time instances. Next, substitute equation (10) into equation (5), and require equation (5) to hold exactly at the same N discrete locations in time (i.e. multiply (5) by the dirac delta test function $\delta(t-t^n)$ and integrate over all time), which yields the following time-spectral governing equation:

$$\sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j + \mathbf{R}(\mathbf{U}^n,\dot{\mathbf{x}}^n,\tilde{\mathbf{n}}^n) = 0 \qquad n = 0,1,2,...,N-1 \tag{15}$$

This results in a system of N equations for the N time instances $\mathbf{U}^n$ which are all coupled through the summation over the time instances in the time derivative term. The spatial discretization operators remain unchanged in the time-spectral approach, with only the requirement that they be evaluated at the appropriate temporal location. Thus, the time-spectral method may be implemented without any modifications to an existing spatial discretization, requiring only the addition of the temporal discretization coupling term, although the multiple time instances must be solved simultaneously due to this coupling.

C. Fully Implicit Method

A common approach for solving the system of equations resulting from the time-spectral method (c.f. equation (15)) consists of adding a pseudo-time term as:

$$\frac{\partial}{\partial \tau}(V^n \mathbf{U}^n) + \sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j + \mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n) = 0 \tag{16}$$

and time-stepping these equations until a pseudo-time steady state is achieved. However, for explicit pseudo-time stepping approaches, it has been shown that the pseudo-time step is limited by stability considerations as:[5]

$$\Delta\tau_n = CFL \frac{V^n}{\| \lambda \| + V^n k'} \tag{17}$$

where λ is the spectral radius of the spatial discretization operator $\mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n)$ and k' represents the largest wave-number that can be resolved by the specified N time instances at the given period T:

$$k' = \begin{cases} \frac{\pi N}{T} & \text{if } N \text{ is even} \\ \frac{\pi (N-1)}{T} & \text{if } N \text{ is odd.} \end{cases} \tag{18}$$

This restriction on $\Delta\tau_n$ results in convergence degradation as the number of time instances is increased or the period decreased. The impact of this restriction can be greatly lessened by resorting to an implicit approach in pseudo-time. Such an approach has been derived in reference[15] using a first-order backwards difference scheme in pseudo-time.

A more general strategy consists of devising a Newton approach for solving the fully coupled non-linear equations at all time instances given by equation (15) or (16). The Newton scheme takes the form:

$$[\mathbf{A}]\,\Delta\mathbf{U} = -\sum_{j=0}^{N-1} d_n^j V^j \mathbf{U}^j - \mathbf{R}(\mathbf{U}^n, \dot{\mathbf{x}}^n, \tilde{\mathbf{n}}^n) \tag{19}$$

with the resulting Jacobian matrix given by:[15]

$$[\mathbf{A}] = \begin{bmatrix} \frac{V^0}{\Delta\tau_0}\underline{\mathbf{I}} + \underline{\mathbf{J}}_0 & V^1 d_0^1 \underline{\mathbf{I}} & \dots & V^{N-1} d_0^{N-1} \underline{\mathbf{I}} \\ V^0 d_1^0 \underline{\mathbf{I}} & \frac{V^1}{\Delta\tau_1}\underline{\mathbf{I}} + \underline{\mathbf{J}}_1 & \dots & V^{N-1} d_1^{N-1} \underline{\mathbf{I}} \\ \vdots & \vdots & \dots & \vdots \\ V^0 d_{N-1}^0 \underline{\mathbf{I}} & V^1 d_{N-1}^1 \underline{\mathbf{I}} & \dots & \frac{V^{N-1}}{\Delta\tau_{N-1}}\underline{\mathbf{I}} + \underline{\mathbf{J}}_{N-1} \end{bmatrix} \tag{20}$$

where a diagonal pseudo-time term can be included as shown for enhanced diagonal dominance of the Jacobian matrix. In the above matrix, $\underline{\mathbf{J}}_j$ corresponds to the Jacobian of the spatial-discretization operator evaluated at time instance j. For a first-order spatial discretization, $\underline{\mathbf{J}}_{j,1stO}$ is as follows:

$$\underline{\mathbf{J}}_{j,1stO} = \frac{\partial \mathbf{R}_{1stO}}{\partial \mathbf{U}} \tag{21}$$

For a second-order spatial discretization, $\underline{\mathbf{J}}_{j,2ndO}$ contains many more entries, as each element of the mesh is not only influenced by its nearest neighbors, but also by the neighbors of its neighbors. The second-order Jacobian is derived as follows:

$$\underline{\mathbf{J}}_{j,2ndO} = \left.\frac{\partial \mathbf{R}_{2ndO}}{\partial \mathbf{U}}\right|_{L=constant} + \left.\frac{\partial \mathbf{R}_{2ndO}}{\partial \mathbf{L}}\right|_{U=constant} \cdot \frac{\partial \mathbf{L}}{\partial \mathbf{U}} \tag{22}$$

Returning to equation (19), each non-linear Newton iteration requires solving the linear system

$$[\mathbf{A}]\,\Delta\mathbf{U} = -\mathbf{R}_{\mathbf{TS}}(\mathbf{U}) \tag{23}$$

where $\mathbf{R}_{\mathbf{TS}}(\mathbf{U})$ represents the residual of the complete time-spectral system (i.e. right hand side of equation (19)), and $[\mathbf{A}]$ is a large matrix spanning all spatial and temporal degrees of freedom. Since direct inversion of $[\mathbf{A}]$ is generally intractable, an inexact Newton scheme can be formulated using an approximate representation of $[\mathbf{A}]$ which

is simpler to invert. One possible simplification is to replace the exact spatial Jacobian in each diagonal block $\underline{\mathbf{J}}_i$ by the corresponding first-order Jacobian $\underline{\mathbf{J}}_{i,1stO}$ as is typically done for steady-state solvers. Another simplification consists of dropping all the off-diagonal terms representing the coupling between different time instances. When this is done, the linear system becomes decoupled among time instances and can be written as:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,1stO}\right]\Delta\mathbf{U}_i=-\mathbf{R}_{\mathbf{TS}}(\mathbf{U}) \tag{24}$$

for each time instance $i=0,1,2,...,N-1$. Alternatively, the diagonal blocks in the $[\mathbf{A}]$ matrix may be retained and the system solved in a block Jacobi fashion following:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,1stO}\right]\Delta\mathbf{U}_i^{l+1}=-\mathbf{R}_{\mathbf{TS}}(\mathbf{U})-\sum_{j\neq i}\left[V^j d_i^j\underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l \tag{25}$$

where the block size corresponds to the entire spatial domain or each time instance and where l denotes the block Jacobi iteration index.

D. Block-colored Gauss-Seidel Linear Solver

As can be seen, both approaches require the inversion the first-order spatial Jacobian (augmented with a pseudo-time term) at each iteration. This may be accomplished using a suitable iterative solver such as block colored Gauss-Seidel (BCGS). In this case, the block now corresponds to the $[4\times4]$ block diagonal matrix at each cell of the mesh, and the iterative scheme can be written as follows, with the spatial off-diagonal terms retained:

$$\left[\frac{V^i}{\Delta\tau_i}\underline{\mathbf{I}}+\left[\underline{\mathbf{D}}_{i,1stO}\right]\right]\Delta\mathbf{U}_i^{l+1}=-\mathbf{R}_{\mathbf{TS}}(\mathbf{U})-\sum_{j\neq i}\left[V^j d_i^j\underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l-\left[\underline{\mathbf{O}}_{i,1stO}\right]\Delta\mathbf{U}_i^l \tag{26}$$

where $\underline{\mathbf{D}}_{i,1stO}$ denotes the $[4\times4]$ block matrix for the current cell, and $\underline{\mathbf{O}}_{i,1stO}$ refers to the off-diagonal blocks for neighboring mesh cells. Although this equation describes a block Jacobi iteration (at the mesh cell level), a block-colored Gauss-Seidel scheme can be recovered with a few simple modifications. First, the computational elements are divided into computational "colors" such that no two adjacent elements are the same color. This coloring allows the Gauss-Seidel method to be run in parallel. The BCGS method then updates (in parallel) all elements of each individual color (with the different colors updated in sequence), such that each update uses the newest information available for all other colors. The BCGS algorithm, written in residual form for a generic linear system is given in Algorithm (1).

Algorithm 1 : Block-colored Gauss-Seidel

1: Given $\underline{\mathbf{A}}\mathbf{x}=\mathbf{b}$
2: **for** i=1,...,ζ **do**
3: **for** j=1,...,n_{colors} **do**
4: **for** k=1,...,n_{e_j} **do**
5: Compute $\mathbf{r}_{i,k}=\mathbf{b}-\underline{\mathbf{A}}\mathbf{x}_{current}$
6: Compute $\Delta\mathbf{x}_{i,k}=\underline{\mathbf{D}}_k^{-1}\mathbf{r}_{i,k}$
7: Update $\mathbf{x}_{i+1,k}=\mathbf{x}_{i,k}+\Delta\mathbf{x}_{i,k}$
8: **end for**
9: **end for**
10: Compute $R_{L,i}=\|\mathbf{r}_i\|_2$
11: If satisfied Stop.
12: **end for**

In Algorithm (1), ζ is the maximum number of BCGS iterations allowed, n_{colors} is the number of colors into which the elements have been divided, and n_{e_j} is the number of elements having the jth color. Additionally, $\underline{\mathbf{D}}_k$ is the diagonal block of the Jacobian for element k, and is inverted directly using LU-decomposition. The specification $\mathbf{x}_{current}$ is used to indicate that, some colors will have information from the previous iteration $\mathbf{x}_i$ while other colors might have already been updated during the current iteration $\mathbf{x}_{i+1}$; in other words, whether from the previous or current iteration, the most up-to-date information at the time of the evaluation of line 5 is used.

As noted earlier, the $\underline{\mathbf{A}}$ matrix given in lines 1 and 5 of the algorithm can either include or exclude the off-diagonal terms given in equation (20). When these time-coupling terms are excluded, this corresponds to a BCGS solver with explicit time treatment, which is abbreviated "BCGS-EX." When the off-diagonal terms are included, implicit time treatment is used in the BCGS solver, and this method is abbreviated "BCGS-IM."

E. Generalized Minimal Residual Method

Despite the additional stability the above, implicit method affords, the time-spectral Euler equations still become difficult to solve as the number of time instances increases. With an increasing number of time instances, the Jacobian given in equation (20) proceeds farther and farther from diagonal dominance. To restore diagonal dominance, a decreasingly small pseudo-time-step size must be used. Thus, the potential efficiency gains of time-spectral methods over time-implicit methods begin to evaporate.

To regain the efficiency improvements afforded by time-spectral methods, a linear solver that does not require diagonal dominance of the Jacobian should be used. The Generalized Minimum Residual method is such a solver. A welcome byproduct of this solver choice is that GMRES also more fully couples the various time instances. This additional coupling arises because the Hessenberg matrix that is directly inverted as part of GMRES is constructed using the full Jacobian of the linear system coupled over all time instances; whereas, the matrix that is directly inverted during each iteration of a stationary iterative method is some easily invertible part of the full Jacobian (usually the spatial-diagonal blocks). In other words, stationary iterative methods ignore some information (and thereby some coupling), but GMRES uses this information and preserves a greater degree of coupling.

A flexible variant of the GMRES algorithm as described by Saad[10] is used. This flexible variant allows the use of an iterative method as preconditioner. The flexible GMRES (FGMRES) algorithm proceeds as given in Algorithm (2).

Algorithm 2 : Flexible GMRES

1: Given $\underline{\mathbf{A}}\mathbf{x} = \mathbf{b}$
2: Compute $\mathbf{r}_0 = \mathbf{b} - \underline{\mathbf{A}}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$
3: **for** j=1,...,n **do**
4: Compute $\mathbf{z}_j := \underline{\mathbf{P}}^{-1}\mathbf{v}_j$
5: Compute $\mathbf{w} := \underline{\mathbf{A}}\mathbf{z}_j$
6: **for** i=1,...,j **do**
7: $h_{i,j} := (\mathbf{w}, \mathbf{v}_i)$
8: $\mathbf{w} := \mathbf{w} - h_{i,j}\mathbf{v}_j$
9: **end for**
10: Compute $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$
11: Define $\underline{\mathbf{Z}}_m := [\mathbf{z}_1, \ldots, \mathbf{z}_m]$, $\underline{\bar{\mathbf{H}}}_m = \{h_{i,j}\}_{1 \le i \le j+1; 1 \le j \le m}$
12: **end for**
13: Compute $\mathbf{y}_m = argmin_y \|\beta\mathbf{e}_1 - \underline{\bar{\mathbf{H}}}_m\mathbf{y}\|_2$ and $\mathbf{x}_m = \mathbf{x}_0 + \underline{\mathbf{Z}}_m\mathbf{y}_m$
14: If satisfied Stop, else set $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ and GoTo 1.

In this description, $\underline{\mathbf{A}}$ corresponds to the full and exact time-spectral Jacobian matrix $[\mathbf{A}]$ defined in equation (20) with the second-order accurate spatial Jacobian, which may or may not be augmented with a pseudo-time-step term, $\mathbf{b}$ corresponds to the negative of the non-linear time-spectral residual $-\mathbf{R}_{\mathbf{TS}}(\mathbf{U})$, and x is the non-linear update $\Delta\mathbf{U}$ to be computed.

Preconditioning is applied in line 4 of the algorithm. The BCGS linear solver is used for preconditioning in several different configurations as covered in the next subsection.

A pseudo-time step must be applied to the BCGS system (i.e. added to $\underline{\mathbf{P}}$) to ensure diagonal dominance and convergence. To solve the minimization problem in line 13 of the algorithm, QR-factorization by means of Givens rotations is utilized.[10] Finally, line 14 of the algorithm indicates that this algorithm is, in fact, truncated, restarting GMRES using n Krylov vectors per restart.

Although Algorithm (2) shows the minimization problem outside the loop over Krylov vectors, in fact, this minimization is updated as each additional Krylov vector is added. This is done so that the current value of the linear residual is known for each iteration j. The FGMRES algorithm exits whenever this residual has either converged a specified amount or has converged to machine zero.

A pseudo-time step is used in the matrix $\underline{\mathbf{A}}$ in the FGMRES algorithm as well. By using this pseudo-time step within FGMRES itself the routine is modified positively as follows: an update that will avoid over-correction issues is used; the pseudo-time step is allowed to grow as the residual decreases so quadratic convergence can be retained; and the convergence rate of the FGMRES linear solver itself is increased (i.e. fewer Krylov vectors and/or restarts are required to converge the linear system) because the pseudo-time step term makes the Jacobian better conditioned for Krylov subspace methods. Thus, two different pseudo-time steps are used: one in the second-order Jacobian $\underline{\mathbf{A}}$ used by the FGMRES algorithm itself and another in the first-order Jacobian $\underline{\mathbf{P}}$ used in the BCGS portion of the preconditioner. Because the Jacobian used in the BCGS preconditioner must be diagonally dominant, the preconditioner pseudo-time

step is almost always smaller and grows more slowly than the FGMRES pseudo-time step. Conversely, the pseudo-time step used in the FGMRES algorithm grows rapidly such that the diagonal term becomes vanishingly small and an exact Newton method is recovered after several orders of magnitude decrease in the non-linear residual.

1. *Preconditioning Methods for FGMRES*

As mentioned above, all preconditioning methods make use of the block-colored Gauss-Siedel solver in one form or another. The first preconditioning method uses BCGS with explicit treatment of time. In other words, Algorithm (1) is used, with a specified number of iterations ζ on each Krylov vector. Here the preconditioner matrix $\underline{\mathbf{P}}$ corresponds to the $[\mathbf{A}]$ matrix, given by equation (20), using the first-order spatial Jacobian with a pseudo-time step that is different, and smaller, than the pseudo-time step used in FGMRES. Additionally, this preconditioner matrix does not include the off-diagonal, time-coupling terms. GMRES using this preconditioner is abbreviated as "GMRES-EX" in the remainder of this work.

The next preconditioner used is BCGS with implicit treatment of time. This preconditioner uses Algorithm (1) with a specified number of iterations ζ on each Krylov vector. The full $[\mathbf{A}]$ matrix, including the temporal off-diagonal time-spectral coupling terms, is used, but again, the first-order spatial Jacobian is used with a smaller pseudo-time step. This solver/preconditioner combination is abbreviated "GMRES-IM".

Both of the preconditioners presented thus far have a serious limitation: the pseudo-time step size needed to preserve diagonal-dominance of the Jacobian and an appropriately scaled update remains relatively small. In order to bypass this limitation, a preconditioner for FGMRES is formed as a defect correction method applied to the residual of equation (25) as:

$$\left[\frac{V^i}{\Delta\tau_{BCGS}}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,1stO}\right](\Delta\mathbf{U}_i^{l+1}-\Delta\mathbf{U}_i^l)=-\mathbf{R_{TS}}(\mathbf{U})-\sum_{j\neq i}\left[V^j d_i^j\underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l-\left[\frac{V^i}{\Delta\tau_{FGMRES}}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,1stO}\right]\Delta\mathbf{U}_i^l \tag{27}$$

where the right-hand-side corresponds to the residual of equation (25). A dual iteration strategy is required for this preconditioner (thus the term defect-correction). Inner BCGS iterations are used to invert the left-hand side matrix providing an updated value for $\Delta\mathbf{U}_{i+1}^l$, which is then substituted into the right-hand side (RHS) terms, and the process is repeated, effectively driving the right-hand-side residual to zero after a number of outer iterations. The advantage of this approach comes from the fact that the pseudo-time step on the right-hand side residual is generally much larger than that required for stability of the BCGS iterative scheme. This solver/preconditioner combination, which uses the first-order Jacobian on the RHS, is abbreviated "GMRES-DC1". The same defect-correction method can also use the second-order Jacobian on the RHS as follows:

$$\left[\frac{V^i}{\Delta\tau_{BCGS}}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,1stO}\right](\Delta\mathbf{U}_i^{l+1}-\Delta\mathbf{U}_i^l)=-\mathbf{R_{TS}}(\mathbf{U})-\sum_{j\neq i}\left[V^j d_i^j\underline{\mathbf{I}}\right]\Delta\mathbf{U}_j^l-\left[\frac{V^i}{\Delta\tau_{FGMRES}}\underline{\mathbf{I}}+\underline{\mathbf{J}}_{i,2ndO}\right]\Delta\mathbf{U}_i^l, \tag{28}$$

which will be abbreviated as "GMRES-DC2".

All of the preconditioners outlined thus far require a pseudo-time step as defined by equation (17), which becomes smaller as the maximum wave-number resolvable by the time-spectral system increases. This wave-number increases, as can be seen from equation (18), as the length of the period of interest decreases or the number of time instances used increases. It was originally believed that the spatial spectral radius component in equation (17) would be dominant in almost all cases. Recent work,[11,12] however, has indicated that the wave-number component dominates much more frequently than had been anticipated, especially for problems using large numbers of time instances such as those investigated in this and previous works.[16,17,11,12] As such, a wave-number independent preconditioner is derived by taking the GMRES-DC2 preconditioner, discussed above, and making one change: the $\underline{\mathbf{D}}_k$ given in line 6 of the BCGS algorithm, Algorithm (1) are changed to be the spatial-temporal diagonal blocks of the full, first-order accurate time-spectral Jacobian. These spatial-temporal diagonal blocks look similar to the time-spectral Jacobian given by equation (20), as follows:

$$\underline{\mathbf{D}}_{ST,k}=\begin{bmatrix}\frac{V_k^0}{\Delta\tau_0}\underline{\mathbf{I}}+\underline{\mathbf{D}}_{S,k,0} & V_k^1 d_0^1\underline{\mathbf{I}} & \cdots & V_k^{N-1}d_0^{N-1}\underline{\mathbf{I}}\\ V_k^0 d_1^0\underline{\mathbf{I}} & \frac{V_k^1}{\Delta\tau_1}\underline{\mathbf{I}}+\underline{\mathbf{D}}_{S,k,1} & \cdots & V_k^{N-1}d_1^{N-1}\underline{\mathbf{I}}\\ \vdots & \vdots & \cdots & \vdots\\ V_k^0 d_{N-1}^0\underline{\mathbf{I}} & V_k^1 d_{N-1}^1\underline{\mathbf{I}} & \cdots & \frac{V_k^{N-1}}{\Delta\tau_{N-1}}\underline{\mathbf{I}}+\underline{\mathbf{D}}_{S,k,N-1}\end{bmatrix} \tag{29}$$

where $\underline{\mathbf{D}}_{ST,k}$ is the full spatial-temporal diagonal block of the time-spectral Jacobian for spatial element k and $\underline{\mathbf{D}}_{S,k,j}$ and V_k^j are the spatial diagonal block and element volume, respectively, for spatial element k and time instance j.

Thus, for the two-dimensional Euler equations, each $\underline{\mathbf{D}}_{ST,k}$ is of the size $[(4\times N)\times(4\times N)]$. By inverting the spatial-temporal diagonal blocks in the BCGS algorithm, the pseudo-time-step size needed to restore diagonal dominance in the preconditioner becomes as follows:

$$\Delta\tau_n = CFL\frac{V^n}{\|\lambda\|} \tag{30}$$

which, as can be seen, has achieved the goal of wave-number independence. The solver/preconditioner combination which utilizes spatial-temporal diagonal-block inversion is abbreviated "GMRES-STI". Table 1 summarizes the various solvers and solver/preconditioner combinations presented in the current work and the abbreviations used to describe them.

Table 1. Summary of Solvers and Preconditioners

Linear Solver	Preconditioner	Abbreviation
Time-explicit BCGS	N/A	BCGS-EX
Time-implicit BCGS	N/A	BCGS-IM
FGMRES	Time-explicit BCGS	GMRES-EX
FGMRES	Time-implicit BCGS	GMRES-IM
FGMRES	Def. Corr., BCGS-IM, 1stO Jac.	GMRES-DC1
FGMRES	Def. Corr., BCGS-IM, 2ndO Jac.	GMRES-DC2
FGMRES	Def. Corr., Spatial-Temporal Inv.	GMRES-STI

F. Implementation

The various time instances in the time-spectral approach are coupled and must be solved simultaneously. This coupling can be implemented serially, whereby a single time-instance is solved at any given moment, and then transmits its update to the next time instance, which is then solved and the process repeated sequentially until all time instances have been updated. However, since the coupling occurs as a source term in the residual, each individual time instance may be solved in parallel with the other time instances. This introduces an additional dimension for achieving parallelism compared to time-implicit computations, where progress in the time dimension is necessarily sequential. In this implementation, two levels of parallelism are introduced, the first in the spatial dimension, and the second in the time dimension where the various time instances are solved by spawning multiple instances of the spatial solver on a parallel computing cluster. The implementation uses MPI for parallelism in the time dimension and OpenMP for additional parallelism in the space dimension. This additional parallelism in time may prove to be particularly enabling with the advent of rapidly expanding massively parallel computing clusters, particularly for cases where parallelism in the spatial dimension has been exhausted (perhaps due to the adequacy of moderate grid sizes).

One of the drawbacks of time-spectral methods is that each time instance must broadcast its entire solution field to all other time instances, which can result in a significant amount of communication. Various strategies for communicating the different time instances to all processors have been investigated. Currently, a Round-Robin approach is implemented, where each processor sends its time instance to a single neighboring processor. The received time instance is added to the time derivative source term on the local processor, and then passed on to the next processor. By repeating this procedure $N-1$ times, where N is the number or time instances, the complete time derivative involving summations from all time instances is accumulated without the requirement of creating a local temporary copy of all the additional time-instance solution vectors or performing any communication intensive broadcast operations.

The aforementioned communication strategy works in all cases except the newly implemented spatial-temporal diagonal-block inversion solver, GMRES-STI, where it is necessary not just to include the contributions of other time instances in the residual, but in the spatial-temporal diagonal blocks of the Jacobian as well. As such, after the Jacobian has been calculated during each non-linear iteration and before the GMRES linear solve begins, the $[4\times 4]$ diagonal blocks of the spatial Jacobian for a single grid cell are all transmitted to a selected MPI process on which the spatial-temporal diagonal block, of size $[(4\times N)\times(4\times N)]$ is assembled and inverted. The spatial-cells are distributed as evenly as possible among the N computational cores available, one core each for the N time instances. How evenly the cells can be distributed among the N cores, however, is complicated by the use of the BCGS preconditioner, which necessitates that each cell color, in fact, be split as evenly as possible. Evenly splitting each color is more difficult than evenly splitting the entire grid because, since a typical grid composed of triangles and quadrilaterals is split into four

colors, the actual groups being split are only one-fourth the grid size, if not smaller. It turns out that there has been no reason to color grids into nearly equally sized color groups up to this point, so the current cell coloring algorithm usually produces one color with only a few cells. Specifically, the 8,747 cells of the grid used in the case studies presented later in this work is split into four colors, the smallest of which has only 167 cells. A more even splitting of the cells among the colors could slightly increase efficiency when small grids and large number of time instances are used, but remains to be implemented in future work

In addition to the transmission of the spatial-temporal diagonal blocks among the cores during each non-linear iteration, the BCGS residual vector must also be transmitted to its corresponding MPI process for each BCGS preconditioning iteration. Once all the residual vectors are assembled on the correct processes, the BCGS solution vector is found from this residual vector and the spatial-temporal diagonal block, which was LU-decomposed earlier, at the start of the current non-linear iteration. Then, the BCGS solution vector is divided and each piece returned to the MPI process on which its time instance usually resides. As can be seen, the GMRES-STI solver has more than twice as much communication per preconditioner iteration as the GMRES-DC1 and GMRES-DC2 solvers. Additionally, it requires the inversion of a $[(4\times N)\times(4\times N)]$ matrix for every spatial grid cell during each non-linear iteration. It turns out this matrix inversion step becomes the most time consuming step in the GMRES-STI solver for large numbers of time instances. Methods to improve the efficiency of the spatial-temporal diagonal-block inversion step will be investigated in the future.

III. Results

A two-dimensional inviscid flow test case is constructed with the forced pitching oscillation of a NACA-0012 airfoil at a Mach number of 0.755 and a mean incidence α_0 of 0.016 degrees. The pitching motion is prescribed about the quarter chord of the airfoil as follows:

$$\alpha(t) = \alpha_0 + \alpha_A \sin(\omega t). \tag{31}$$

The reduced frequency k_c is equal to 0.0814 and the pitching amplitude α_A is equal to 2.51 degrees. The unstructured, computational mesh utilized consists of 4471 nodes and 8747 triangles. Figure (1) shows the near field mesh. This test case corresponds to the AGARD test case No. 5.[18] Figure (2a) shows the comparison of the lift coefficient versus non-dimensional time between a reference solution obtained using a second-order accurate time-implicit solver with $\Delta t = T/4096$ (where T denotes the period of airfoil motion) and the time-spectral method with $N = 3, 7, 15$, and 47 time instances. For this case, the lift computed by the time-spectral method with even 1 harmonic or 3 time instances shows reasonable agreement with the reference solution. Figure (2b) shows the comparison of the moment coefficient versus non-dimensional time for the same reference and time-spectral solutions. The moment history contains multiple harmonics and thus is not captured accurately with $N = 3$ in the TS method; in fact, a rather large number, $N = 47$, of time instances is needed to produce near-exact agreement. Figure (2c) shows the comparison of the drag coefficient versus non-dimensional time for those same solutions. It should be noted that, since the results use the Euler equations, the drag shown is pressure drag and does not contain viscous effects. The drag coefficient shows good agreement for the $N = 47$ solution. The most difficult areas of the pressure drag curve to resolve occur around $t = 7$s and $t = 28$s, where the $N = 47$ and reference solutions show a slight reversal. It can be seen that the results of the TS method converge to the reference solution as the number of time instances increases.

A. Convergence of the AGARD5 Test Case

While the accuracy of the time-spectral method is tantamount, equally important is the efficiency of the solver. The present work is primarily intended to discuss advancements made in solver efficiency and capability as work with time-spectral methods has proceeded. Figure (3) plots the convergence of the time-spectral method for the AGARD 5 test case using $N = 3$ time instances for the first five of the different solver configurations summarized in Table (1). In order to resolve any details for the convergence of the GMRES based solvers, it was necessary to scale the x-axis such that the full convergence history of both BCGS-EX and BCGS-IM are not visible. Table (2) summarizes the convergence of the various solvers that were shown in Figure (3) in terms of the non-linear iteration number, cumulative preconditioning iterations, cumulative Krylov vectors, and wall-clock time needed to produce a converged solution.

It can be seen that the GMRES-DC1 solver is the most computationally efficient overall of those presented herein. When comparing the BCGS-EX solver to the BCGS-IM solver, the latter reduces the number of non-linear iterations, the number of BCGS iterations, and the wall-clock time all by more than a factor of 10. These gains are accomplished

Table 2. Convergence of the AGARD 5 Test Case with $N = 3$ Time Instances

Solver	Non-linear Iter.	BCGS Iter.	Krylov Vectors	Wall-clock Time (s)
BCGS-EX	106,844	213,687	N/A	2,317
BCGS-IM	8,744	17,487	N/A	214
GMRES-EX	151	9,205	4,564	72
GMRES-IM	25	6,101	2,061	37
GMRES-DC1	21	6,800	540	30

almost exclusively by the increased CFL number allowed by the BCGS-IM. A CFL number of only 0.25 can be used for BCGS-EX, while a CFL number more than $10\times$ higher, i.e. 2.6, can be used for the BCGS-IM solver.

The GMRES-EX method gains over the BCGS-IM by halving the number of BCGS iterations used, while decreasing the number of non-linear iterations by a factor of 58 and the wall-clock time by a factor of almost 3. The gains of GMRES-EX are summarized as follows: each BCGS iteration is more efficient when used as a preconditioner rather than as the main linear solver and GMRES allows the use of more BCGS iterations for every non-linear iteration, reducing overhead costs. The further efficiency increases provided by the GMRES-IM solver over the GMRES-EX solver occur because of the implicit coupling among time instances, in the same way that BCGS-IM gains over BCGS-EX. Emphasis should be placed on just how much the use of GMRES improves the efficiency of the solution: the GMRES-IM solver cuts the wall-clock time by over 5.7 times and the BCGS iterations by over 2.8 times when compared to the BCGS-IM solver; the GMRES-DC1 solver saves a further 7 seconds or 19% of the wall-clock time over the GMRES-IM solver.

The gains of adding a defect-correction step to the GMRES preconditioner accrue mainly because of the decreased numbers of non-linear iterations and Krylov vectors, which decreases the total overhead cost of convergence to the final solution. This is evidenced by the number of BCGS iterations increasing slightly as compared to the GMRES-IM method while the number of non-linear iterations and the number of Krylov vectors used decreases.

Results for the various solvers when $N = 15$ are presented in Table (3) except BCGS-EX is omitted as it is so clearly non-competitive. As can be seen, the same trends hold for the $N = 15$ solutions as were discussed for the

Table 3. Convergence of the AGARD 5 Test Case with $N = 15$ Time Instances

Solver	Non-linear Iter.	BCGS Iter.	Krylov Vectors	Wall-clock Time (s)
BCGS-IM	12,992	27,653	N/A	2,430
GMRES-EX	125	22,197	11,098	873
GMRES-IM	124	11,551	3,922	440
GMRES-DC1	22	14,525	383	423

$N = 3$ solutions. Although not presented herein, the same trends continue to hold for increasing numbers of time instances. Specifically, the best GMRES solver, GMRES-DC1 takes one-fifth or less time than BCGS-IM for greater numbers of time instances. In fact, it has been found that the BCGS-IM solver cannot efficiently converge solutions with more than fifty time instances, as the CFL number needed to maintain diagonal dominance in the BCGS solver becomes prohibitively small. It should be noted that all of the solutions presented thus far have been converged to a convergence tolerance of 1×10^{-11} and utilize an aggressive GMRES CFL number growth strategy.

B. Spatial-temporal Diagonal-block Inversion

As mentioned above, by inverting the temporally-coupled diagonal blocks of the Jacobian, instead of just the uncoupled, spatial diagonal blocks, as has been done up to this point, in the BCGS preconditioner, the temporal restrictions on the CFL number can be eliminated. Theoretically, this should result in solution convergence that, in terms of non-linear iterations, Krylov vectors, and preconditioner iterations, is independent of the number of time instances used as well as the reduced frequency of the motion. All of the wall-clock times presented in this section are for solution

utilizing one computational core per time instance.

First, the same test case as was used above, i.e. the AGARD test case No. 5, is examined using the same computational mesh as previously. Figure (4) shows the convergence of this test case for the GMRES-DC1 solver by plotting the non-linear residual versus the number of non-linear iterations, cumulative preconditioning iterations, cumulative Krylov vectors, and wall-clock time, in that order for the four subfigures. Table (4) summarizes this same data for the converged solutions. It should be noted that the solutions in this section have all been converged to a convergence tolerance of 3×10^{-12} and all use the same GMRES CFL number growth strategy, which has not been optimized for efficiency. Additionally, they are run on better computational hardware than the previous results and thus they should not be directly compared to the results presented in the previous section. It should be further noted that, although not justified herein, the GMRES-DC1 solver is used because it was found to be generally more efficient for flow alone problems than the GMRES-DC2 solver, which has been shown to be better for aeroelastic problems.[12] As can be

Table 4. Convergence of the AGARD 5 Test Case using the GMRES-DC1 Solver

Num. Time Inst.	Non-linear Iter.	BCGS Iter.	Krylov Vectors	Wall-clock Time (s)	BCGS CFL
15	26	16,950	782	195	1.4
31	26	21,723	1,057	356	1.2
63	28	30,838	1,408	807	1.2
127	30	59,917	891	2,355	1.2
255	31	97,211	1,424	8,378	1.2
511	32	193,970	2,787	28,783	1.2

seen in this figure and table, all six solutions use about the same number of non-linear iterations, while the number of preconditioner iterations and the wall-clock time used increase with increasing number of time instances. These trends result from the smaller pseudo-time-step size that is allowed as the number of time instances increases, as discussed previously and given in equation (17). The fact that the maximum stable BCGS CFL number also decreases slightly with increasing number of time instances exacerbates the situation by limiting the pseudo-time-step size to an even smaller value. The Krylov vector trend appears a little odd as it seems to reset between 63 and 127 time instances. In order to ensure convergence, it was necessary to allow more preconditioning iterations per Krylov vector to be used for 127 and more time instances, as has been demonstrated in previous work.[11] When 15 through 63 time instances are used, only 10 defect correction steps are allowed per Krylov vector, while 30 defect corrections per Krylov vector are allowed between 127 and 511 time instances.

Table 5. Convergence of the AGARD 5 Test Case using the GMRES-STI Solver

Num. Time Inst.	Non-linear Iter.	BCGS Iter.	Krylov Vectors	Wall-clock Time (s)	BCGS CFL
15	24	8,905	437	137	2.5
31	24	8,850	435	245	2.5
63	24	8,951	440	458	2.5
127	24	8,970	441	1,085	2.5
255	24	9,029	444	3,514	2.5
511	24	9,109	448	9,881	2.5

On the other hand, when the GMRES-STI solver is applied to this same problem with the same numbers of time instances, as shown in Figure (5) and Table (5), all numbers of time instances use about the same number of preconditioner iterations and Krylov vectors, in addition to non-linear iterations. This result is because the pseudo time-step size used by the GMRES-STI solver is independent of the number of time instances and is fully determined by the spectral radius of the spatial discretization. It should be noted that a maximum of 10 defect-correction steps are allowed for each Krylov vector for all numbers of time instances. Wall-clock time, on the other hand, still increases for increasing numbers of time instances. However, the wall-clock time used is not only lower than that used by the GMRES-DC1 solver for all number of time instances considered, it also increases less rapidly when time instances

are added when using the GMRES-STI solver as compared to the GMRES-DC1 solver

To verify the wave-number independence of the GMRES-STI solver, the same test case is run except at twice the reduced frequency of the AGARD 5 test case, i.e. $k_c = 0.1628$ for the second test case, with all other test-case parameters remaining unchanged. Tables (6) and (7) convey the convergence data for this test case using the GMRES-DC1 and GMRES-STI solvers, respectively. As can be seen, increasing the number of time instances for the GMRES-

Table 6. Convergence of the $2k_c$ Test Case using the GMRES-DC1 Solver

Num. Time Inst.	Non-linear Iter.	BCGS Iter.	Krylov Vectors	Wall-clock Time (s)	BCGS CFL
15	23	19,595	752	208	1.4
31	25	26,152	1,206	412	1.2
63	26	46,502	2,090	1,138	1.2
127	26	116,520	1,691	4,592	1.2
255	29	142,955	2,054	10,979	1.2
511	29	284,551	3,561	43,684	1.0

Table 7. Convergence of the $2k_c$ Test Case using the GMRES-STI Solver

Num. Time Inst.	Non-linear Iter.	BCGS Iter.	Krylov Vectors	Wall-clock Time (s)	BCGS CFL
15	21	7,528	369	121	2.5
31	21	7,784	382	218	2.5
63	21	8,064	396	419	2.5
127	21	7,945	390	964	2.5
255	21	7,864	386	3,118	2.5
511	21	7,864	386	8,618	2.5

DC1 solver requires the use of more Krylov vectors and BCGS iterations, as expected, but to an even greater degree than was apparent for the first test case. On the other hand, if the GMRES-STI solver is used, the number of Krylov vectors and BCGS iterations remains about the same and in fact decreases slightly at the higher reduced frequency. Indeed, while the wall-clock time for convergence increases for all numbers of time instances when the reduced frequency is doubled using the GMRES-DC1 solver, it actually decreased slightly when the GMRES-STI solver is used. This unexpected result is likely produced by the stronger temporal coupling at higher reduced frequencies which, although it degrades performance of the GMRES-DC1 solver, it is fully taken into account in the GMRES-STI solver. Since the CFL number remains the same between the first and second test cases when using the GMRES-STI solver, the pseudo-time step size does not change as a result of the increased reduced frequency, while the period of airfoil motion is halved when the reduced frequency is doubled, it logically follows that fewer pseudo-time steps would be required, i.e. fewer non-linear iterations. Indeed, the number of non-linear iterations decreases from 24 for the first test case to 21 for the second, while the wall-clock time per non-linear iteration remains about the same between the two test cases (i.e. 412 seconds for the first test case and 410 seconds for the second test case, when using 511 time instances).

To further examine how the cumulative preconditioner iterations and wall-clock time needed to converge a solution change in response to reduced frequency, Figure (6) plots these two quantities against the reduced frequency for $N = 63$ and $N = 127$ time instances using the two solvers that are being compared. As can be seen in even more stark contrast, the GMRES-STI solver uses basically the same number of preconditioning iterations at all reduced frequencies and both numbers of time instances, while the number of preconditioning iterations increases rapidly at higher reduced frequencies when the GMRES-STI solver is used. These preconditioning trends affect wall-clock time required in the expected way for the GMRES-DC1 solver, whose solution time is roughly proportional to the number of preconditioning iterations used for a given number of time instances.[11] When using the GMRES-STI solver, while wall-clock time generally decreases as reduced frequency increases, it does increase when more time instances are used in large part because of the increased size of the spatial-temporal diagonal blocks that must be inverted.

Figure (7) shows the cumulative preconditioner iterations and wall-clock time needed to converge a solution versus the number of time instances for three different reduced frequencies using both GMRES-DC1 and GMRES-STI solvers. This figure clearly demonstrates the superior efficiency of the GMRES-STI solver at all reduced frequencies and for any number of time instances considered. It should be noted that GMRES-STI is also a more consistent solver, regardless of the number of time instances used, as can be seen by its steady trends with number of time instances, whereas the results for the GMRES-DC1 solver have three clearly anomalous points, one for each frequency. It should be noted that in the (b) subfigure, wall-clock time is on a logarithmic scale.

Finally, Figures (8) and (9) plot the number of preconditioning iterations and wall-clock time (on a log scale) versus the number of time instances for six reduced frequencies, from one-half the AGARD5 test case frequency to eight times that frequency. Figure (8) supports the conclusion that the GMRES-STI solver is wave-number independent at all frequencies, except the lowest. The results for the lowest frequency are particularly perplexing as, for a given number of time instances, the maximum resolvable wave number is the smallest for this frequency. At the present time, the reason that the number of preconditioning iterations needed increases exponentially for the lowest frequency while being nearly constant for the other four frequencies is not known. Despite the rapid increase in the number of preconditioning iterations for the lowest frequency, its wall-clock time to convergence follows the same trends as the results for the other frequencies, which helps to demonstrate that the limiting step in the GMRES-STI solver is the inversion of the spatial-temporal diagonal-blocks. While these blocks are inverted only once per non-linear iteration, they are inverted directly, a process that uses $O\left(4N^3\right)$ operations where N is the number of time instances. Given that the computational hardware used is scaling as $O(N)$, i.e. one node per time instance, an increase in the number of time instances results in $O\left(4N^2\right)$ work per computational core during the diagonal-block inversion step, assuming perfect scaling. This scaling is the primarily reason why the wall-clock time to convergence exhibits an increasing slope with increasing numbers of time instances, as shown in Figure (9).

IV. Conclusions and Future Work

In the present work, a robust, wave number independent time-spectral solver has been developed. The flexible variant of the Krylov subspace, generalized minimal residual method is used to solve the fully-coupled linear system of time-spectral Euler equations for each non-linear iteration. By inverting the full spatial-temporal diagonal blocks of the Jacobian in the BCGS preconditioner, instead of merely the spatial-diagonal blocks for each time instance independently, as has been done previously,[11,12] wave-number independence of time-spectral solution convergence has been achieved. As compared to previous iterations of the time-spectral solver,[11,12] the GMRES-STI solver reduces the solution time for all reduced frequencies and numbers of time instances considered. This time reduction is as low as a 25% of the GMRES-DC1 time for $N = 15$ time instances and one-half the AGARD5 reduced frequency and as high as 80% of the GMRES-DC1 time for $N = 511$ time instances and two times the AGARD5 reduced frequency, i.e. the old GMRES-DC1 solver takes five times longer than the GMRES-STI solver for the highest wave-number test case where the two solvers are compared. It should be noted, however, that cases with as many as $N = 767$ time instances and at eight times the AGARD5 reduced frequency are also presented in this work. No comparison to the GMRES-DC1 solver is made for these higher wave-number cases, but if the reduced frequency and time instance trends exhibited for the GMRES-DC1 were to continue to hold, an even greater time reduction would be seen when comparing the GMRES-DC1 solver to the GMRES-STI solver. It should also be noted that the present work uses more time instances at higher reduced frequencies than has been found in the literature up to this point.

Although the GMRES-STI solver is much more efficient than the GMRES-DC1 solver it is still far from optimal. For an optimal solver, the wall-clock time to convergence should remain constant for time-spectral solutions with any number of time instances using one CPU core per time instance. As noted previously, the single most computationally intensive step in the GMRES-STI solver is the inversion of the spatial-temporal diagonal blocks at each pseudo-time step. As also noted, the GMRES-STI solver utilizes the full LU-decompositions of these blocks, which are calculated directly at the beginning of each non-linear iteration. Making this process more efficient is the obvious place to further improve the efficiency of the solver. As such, future work should be focused on exactly these upgrades. One possible improvement might be made by using an incomplete LU-factorization (ILU) in this step instead of the full LU-factorization. Another possibility would be to use an approximate factorization (AF) of the spatial-temporal diagonal blocks such that the spatial and temporal parts can be inverted independently of each other. Both ILU and AF applied to the spatial-temporal diagonal blocks will be implemented in future work. Overall, the wave number independence of the preconditioner used in the GMRES linear solver has proven to be a critical advance in the efficient solution of the time-spectral method.

References

[1]Hall, K. C., Thomas, J. P., and Clark, W. S., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, Vol. 40, No. 5, 2002, pp. 879–886.

[2]McMullen, M., Jameson, A., and Alonso, J. J., "Acceleration of Convergence to a Periodic Steady State in Turbomachineary Flows," AIAA Paper 2001-0152, Jan. 2001.

[3]McMullen, M., Jameson, A., and Alonso, J. J., "Application of a Non-Linear Frequency Domain Solver to the Euler and Navier-Stokes Equations," AIAA Paper 2002-0120, Jan. 2002.

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

[5]van der Weide, E., Gopinath, A. K., and Jameson, A., "Turbomachineary Applications with the Time Spectral Method," AIAA Paper 2005-4905, June 2005.

[6]Lee, K.-H., Alonso, J. J., and van der Weide, E., "Mesh Adaptation Criteria for Unsteady Periodic Flows Using a Discrete Adjoint Time-Spectral Formulation," AIAA Paper 2006-0692, Jan. 2006.

[7]Sankaran, S., Gopinath, A., Weide, E. V. D., Tomlin, C., and Jameson, A., "Aerodynamics and Flight Control of Flapping Wing Flight Vehicles: A Preliminary Computational Study," AIAA 2005-0841, Jan. 2005.

[8]Choi, S., Potsdam, M., Lee, K., Iaccarino, G., and Alonso, J. J., "Helicopter Rotor Design Using a Time-Spectral and Adjoint-Based Method," AIAA 2008-5810, Sep. 2008.

[9]Choi, S. and Datta, A., "CFD Prediction of Rotor Loads using Time-Spectral Method and Exact Fluid-Structure Interface," AIAA 2008-7325, Aug. 2008.

[10]Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.

[11]Mundis, N. L. and Mavriplis, D. J., "GMRES applied to the Time Spectral and Quasi-periodic Time Spectral Methods," AIAA Paper 2013-3084, June 2013.

[12]Mundis, N. L. and Mavriplis, D. J., "An Efficient Flexible GMRES Solver for the Fully-coupled Time-spectral Aeroelastic System," AIAA Paper 2014-1427, Jan. 2014.

[13]Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A., *Spectral Methods in Fluid Dynamics*, Springer, 1987.

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

[15]Sicot, F., Puigt, G., and Montagnac, M., "Block-Jacobi Implicit Algorithm for the Time Spectral Method," *AIAA Journal*, Vol. 46, No. 12, 2008, pp. 3080–3089.

[16]Mundis, N. L. and Mavriplis, D. J., "Quasi-Periodic Time Spectral Method for Aeroelastic Flutter Analysis," AIAA Paper 2013-0638, Jan. 2013.

[17]Mundis, N. L., Mavriplis, D. J., and Sitaraman, J., "Quasi-Periodic Time-spectral Methods for Flutter and Gust Response," 69th Forum of the American Helicopter Society, AHS International, Alexandria, VA, May 2013.

[18]AGARD, "Compendium of Unsteady Aerodynamic Measurements," Tech. Rep. No. 702, 1982.

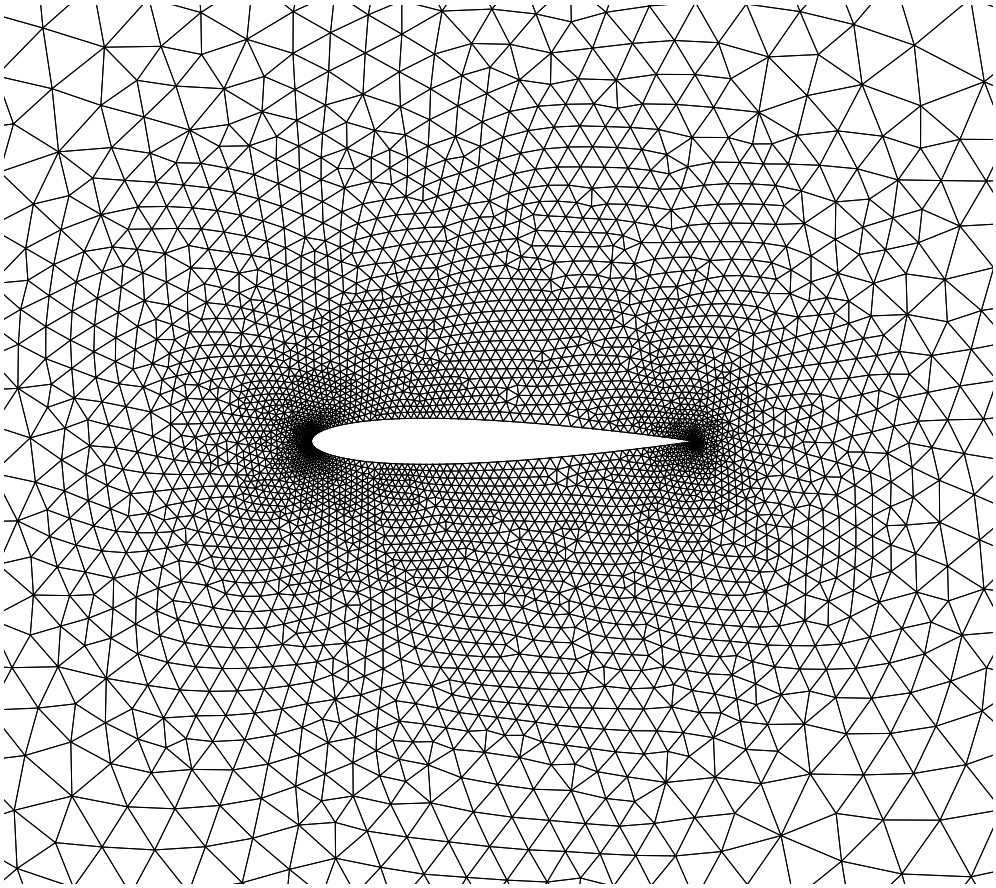
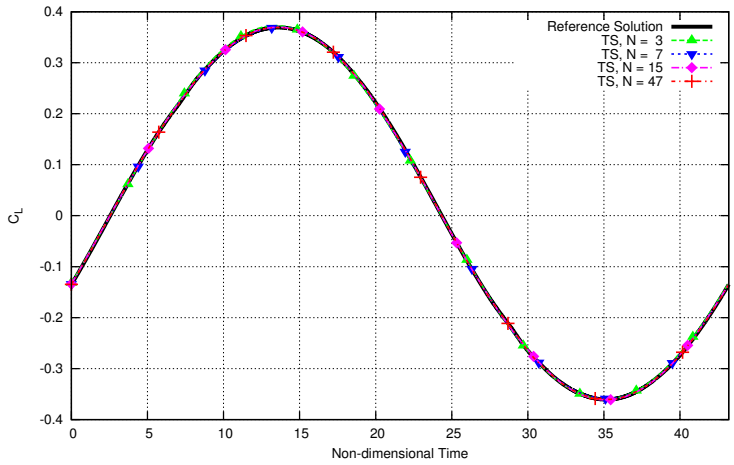
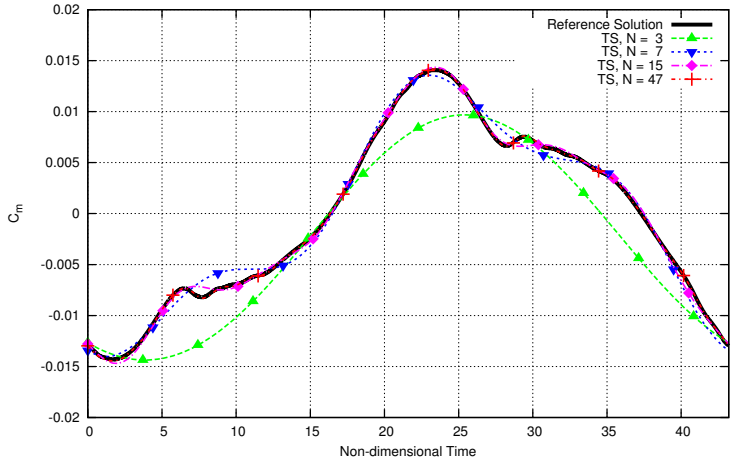


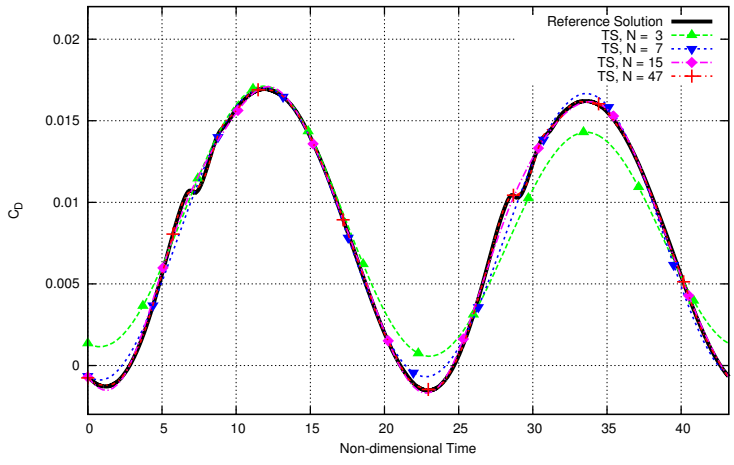
Figure 1. Near field mesh for the NACA-0012 airfoil



(a)



(b)



(c)

Figure 2. Comparison of computed lift coefficient (a), moment coefficient (b), and pressure drag coefficient (c) using the time-spectral method to a reference, time-implicit solution for the AGARD5 test case

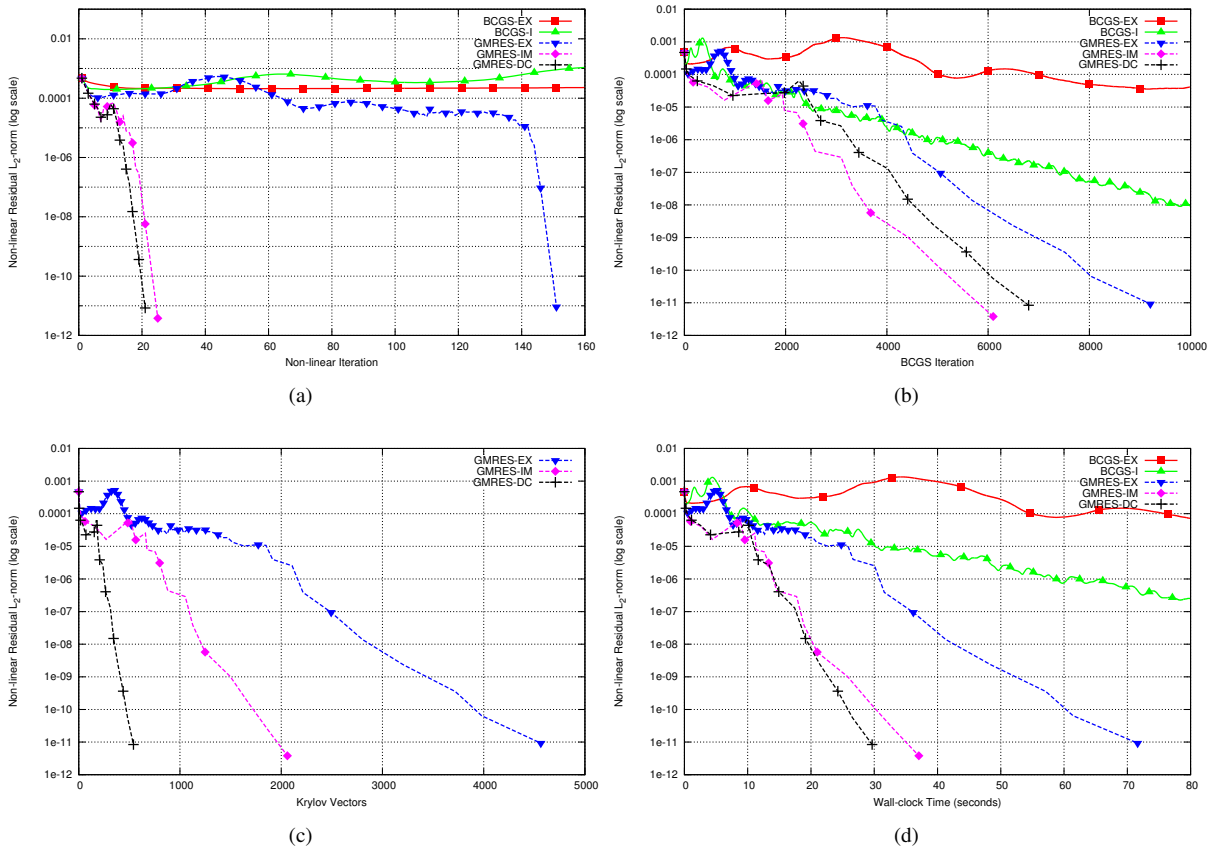


Figure 3. Comparison of convergence for the AGARD test case No. 5 with 3 time-spectral time instances using the first 5 of the different solvers discussed with Non-linear Iterations (a), BCGS Iterations (b), Krylov vectors (c), and wall-clock time (d) on the x-axis

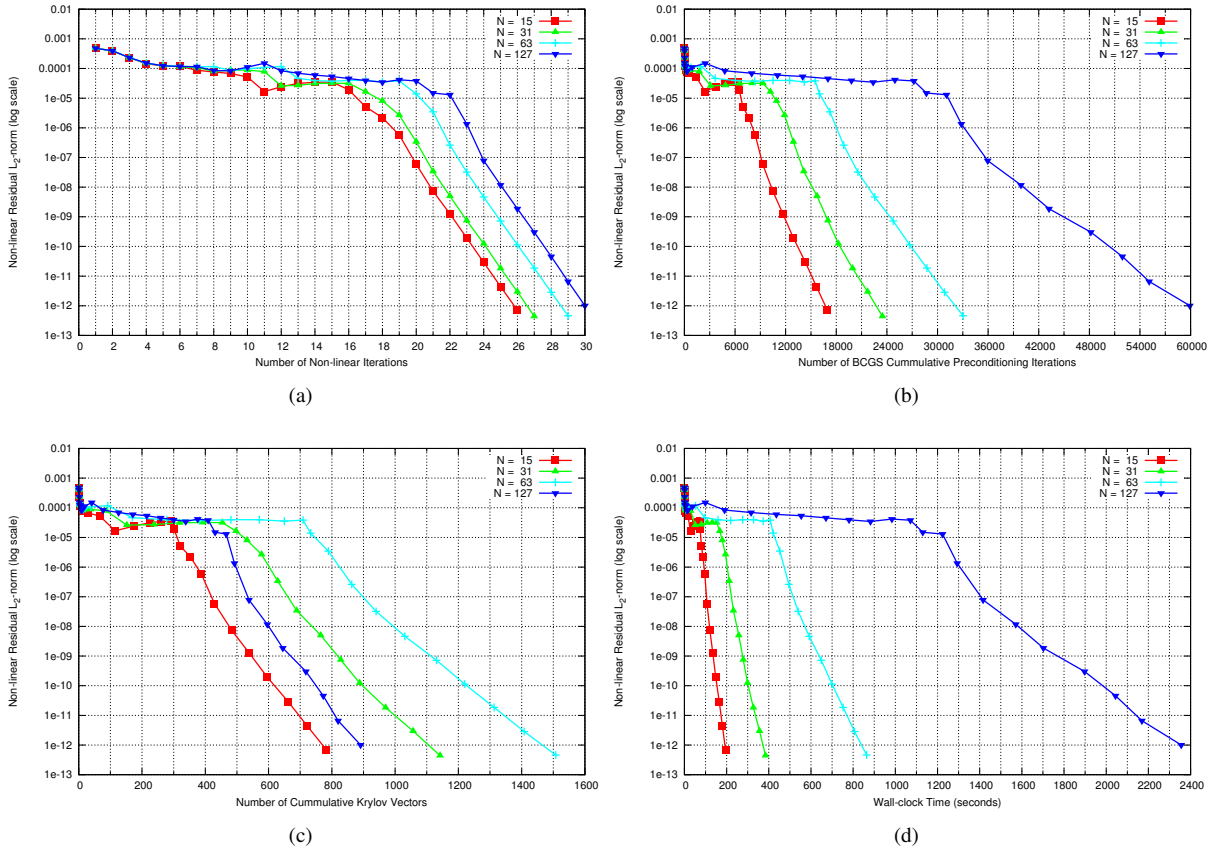


Figure 4. Comparison of convergence for the AGARD test case No. 5 with the indicated numbers of time instances using the GMRES-DC1 solver with Non-linear Iterations (a), BCGS Iterations (b), and Krylov vectors (c), and wall-clock time (d) on the x-axis

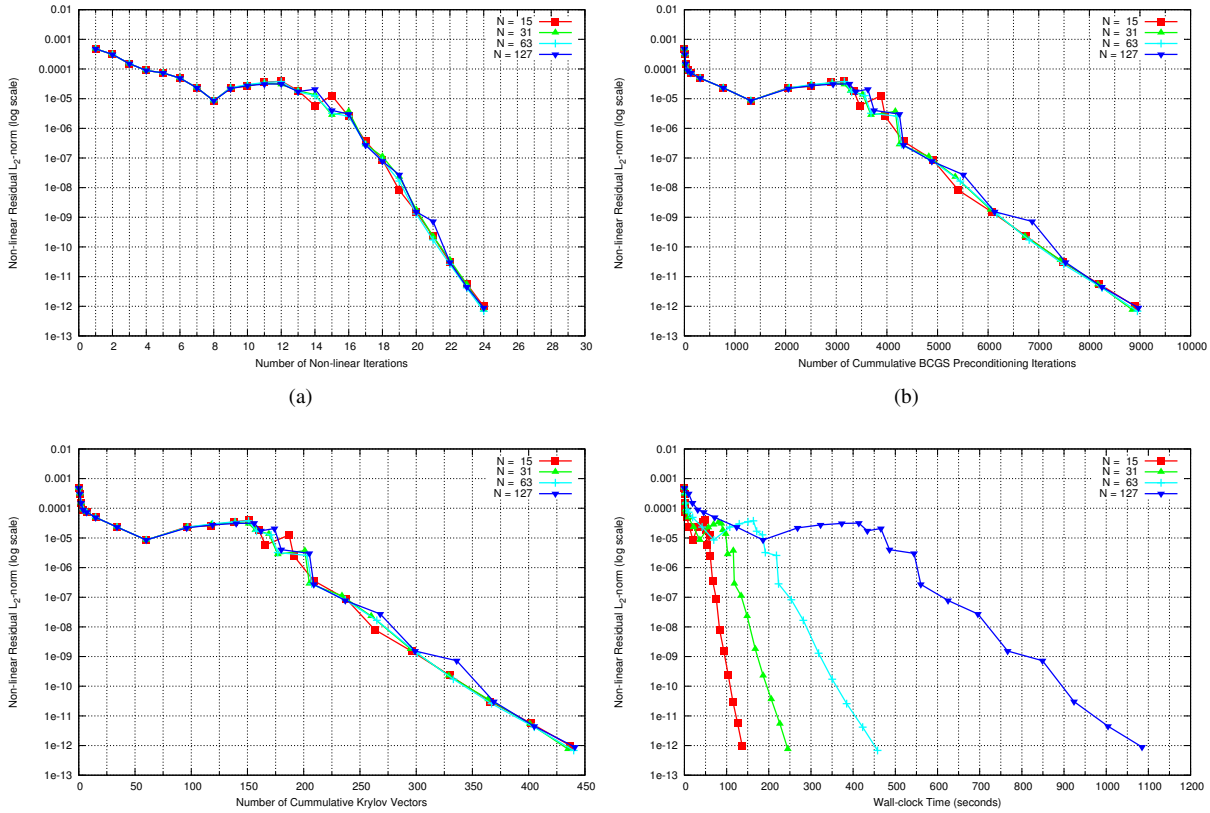


Figure 5. Comparison of convergence for the AGARD test case No. 5 with the indicated numbers of time instances using the GMRES-STI solver with Non-linear Iterations (a), BCGS Iterations (b), and Krylov vectors (c), and wall-clock time (d) on the x-axis

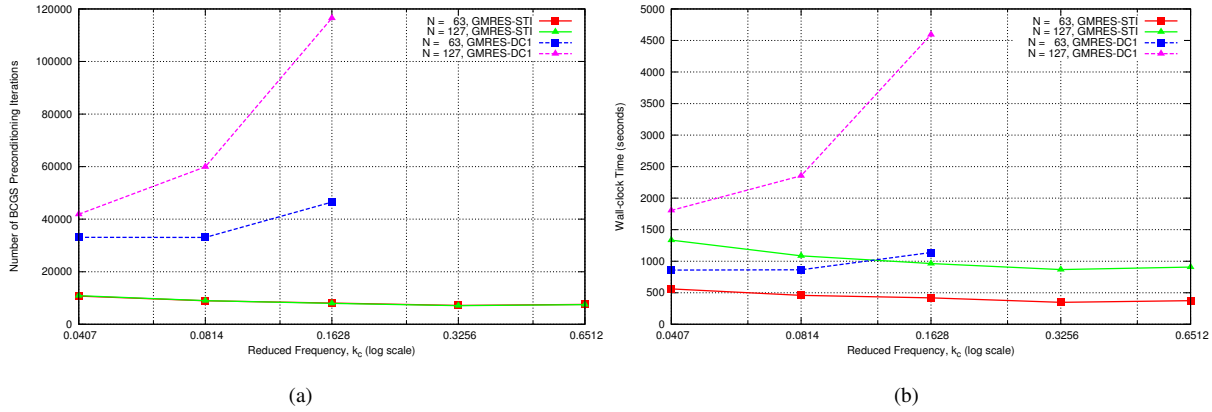
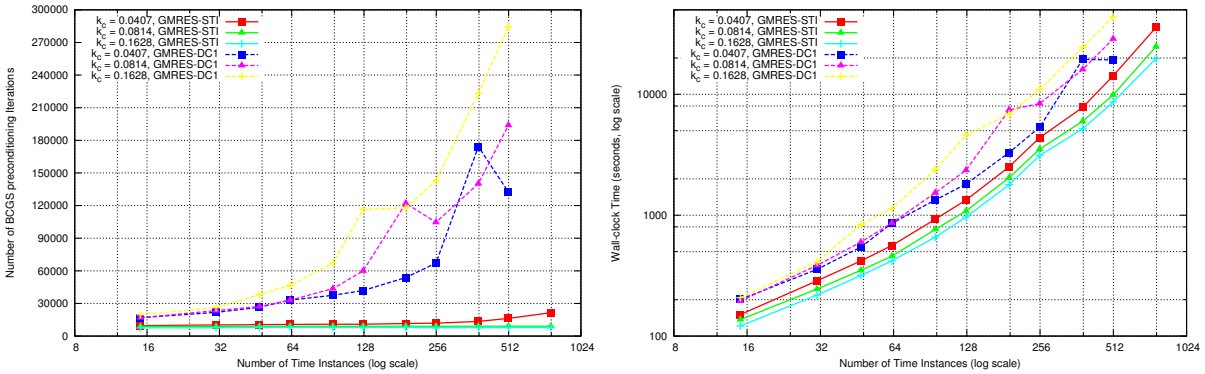


Figure 6. (a) BCGS preconditioner iterations and (b) wall-clock time needed for converged solutions versus reduced frequency for $N = 63$ and $N = 127$ time instances using the GMRES-DC1 and GMRES-STI solvers



(a) (b)

Figure 7. (a) BCGS preconditioner iterations and (b) wall-clock time needed for converged solutions versus number of time instances for three reduced frequencies – one-half the AGARD5 frequency, the AGARD5 frequency, and twice the AGARD5 frequency – using the GMRES-DC1 and GMRES-STI solvers

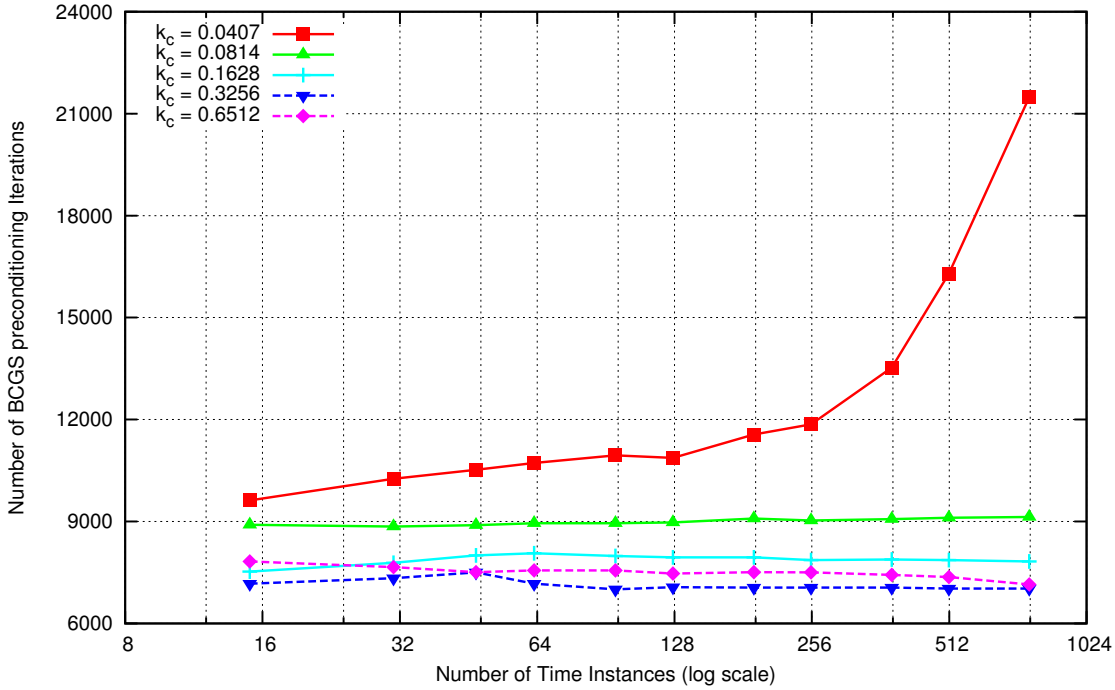


Figure 8. BCGS preconditioner iterations needed for converged solutions versus number of time instances for five reduced frequencies – from one-half the AGARD5 frequency to eight times the AGARD5 frequency, doubling the frequency each time– using GMRES-STI solver

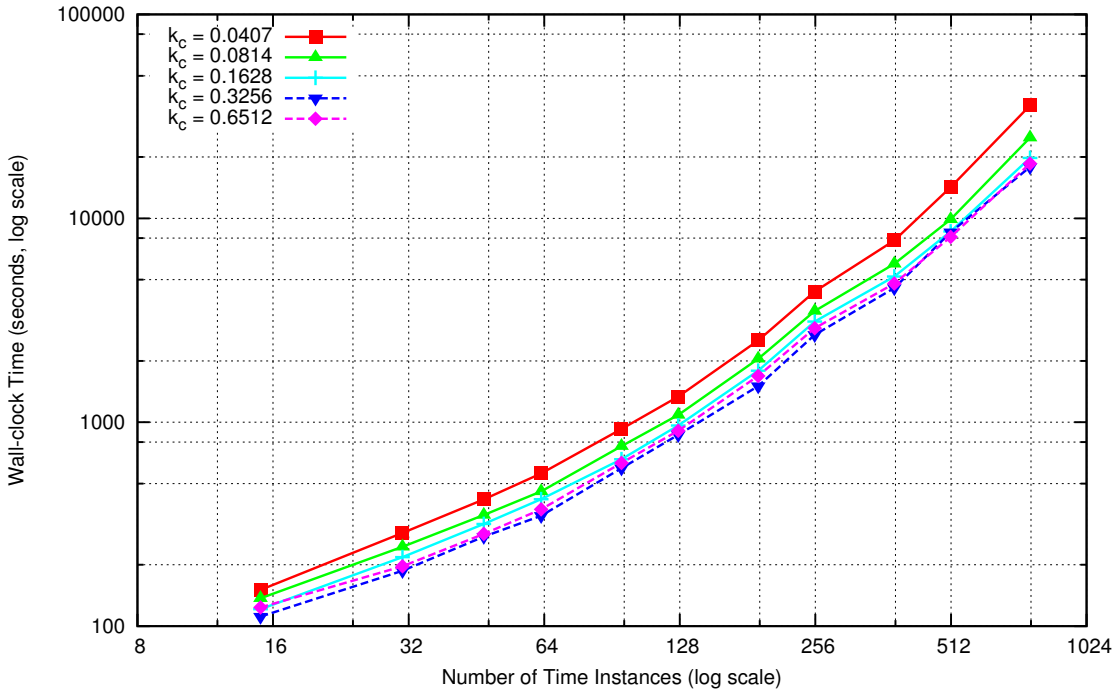


Figure 9. Wall-clock time needed for converged solutions versus number of time instances for five reduced frequencies – from one-half the AGARD5 frequency to eight times the AGARD5 frequency, doubling the frequency each time– using GMRES-STI solver