An Order NlogN Parallel Time Spectral Solver For Periodic and Quasi-Periodic Problems

Donya Ramezanian
PhD Advisor: ***Dr. Dimitri Mavriplis***
Department of Mechanical Engineering
University of Wyoming

12th July 2019

Outline

- Introduction
- Governing Equations
- Challenges
- Novelty
- Results
- Summary and Conclusions
- Future Work

- Introduction
- Governing Equations
- Challenges
- Novelty
- Results
- Summary and Conclusions
- Future Work

Introduction

- The simulation of unsteady phenomena typically demands large computational investments to achieve suitable accuracy

- Temporally periodic problems are one of the sub-categories of unsteady problems, that have a broad range of applications in the industry.

- These include wind-turbine flows, rotorcraft flows, turbomachinery flows, and vortex shedding problems

- Traditionally, time-marching methods were employed for unsteady flow problems including temporally periodic problems

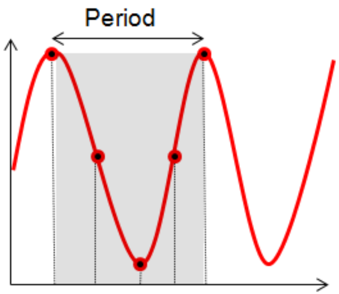
- Time-marching methods solve the problem for several periods until the initial transient part is resolved, and periodic steady state is obtained

Introduction

- In most realistic problems solving the transient part is very time consuming, making time-marching methods inevitably expensive
- Frequency-domain methods directly solve for the periodic solution and avoid the transient parts
- Time-spectral methods (TS) are among the frequency-domain methods that avoid resolving the transient parts and are more favorable in purely-periodic problems
- A hybrid backward difference time-spectral (BDFTS) discretization is an extension of the time-spectral approach for quasi-periodic problems

TS Introduction

- Span the characteristic time period with time instances
- Represent the time derivatives in governing equations as linear combinations of corresponding values in other time instances



- All the time instances are coupled. (Solve for all time instances simultaneously)
- Because of spectral convergence due to Fourier series, limited number of temporal DOF results in accurate solutions
- The time instances are computed in parallel. Exploit more parallelism by parallelizing temporal part, each time instance is assigned to an individual processor

BDFTS Introduction

- Time-spectral methods are only applicable in the presence of fully periodic flows, which represents a severe restriction for many aerospace engineering problems

- Quasi-periodic problems are problems that include a slow transient in addition to strong periodic behavior

- Applications in transient turbofan simulation, maneuvering rotorcraft calculations, …

- A hybrid backward difference time-spectral (BDFTS) discretization is an extension of the time-spectral approach for quasi-periodic problems

- Introduction
- **Governing Equations**
- Challenges
- Novelty
- Results
- Summary and Conclusions
- Future Work

Base Solver

- Inviscid compressible flow
- Arbitrary-Lagrangian-Eulerian (ALE) form of Euler equations

$$\frac{\partial U}{\partial t} + \nabla . F(U) = 0$$

$$\frac{\partial}{\partial t} \int_{\partial\Omega(t)} (F(U) - U\dot{x}).\vec{n} ds = 0$$

$$\frac{\partial (UV)}{\partial t} + R(U, \dot{x}, \vec{n}) = 0$$

- Central difference finite volume cell based in space
- Time discretization: BDF1, BDF2, TS, BDFTS

Temporal Derivative : BDF

- First-order backward difference scheme (BDF1)

$$\frac{\partial U}{\partial t} = \frac{U^{n+1} - U^{n}}{\Delta t} \qquad \mathrm{O}(\Delta t)$$

- Second-order backward difference scheme (BDF2)

$$\frac{\partial U}{\partial t} = \frac{3U^{n+1} - 4U^{n} + U^{n-2}}{2\Delta t} \quad O(\Delta t^{2})$$

$U^{n+1}$is the solution t current time-step

U^{n} is the solution at the previous time-step

U^{n-1} is the solution at the two time-steps ago

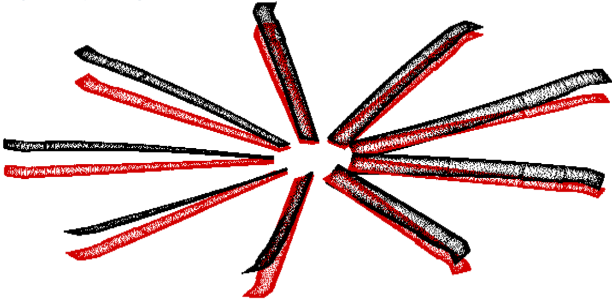
Temporal Derivative : TS

- Used in temporal purely periodic problems

Time Spectral temporal Discretization:
Collocation method using harmonic basis functions in time

First derivative in time: **CFD**

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



Discrete Euler Equation Becomes:

$$\sum_{j=0}^{N-1} d_n^j V^j U^j + \underbrace{R(U^n, \dot{x}^n, \vec{n})} = 0$$

No change in spatial discretization

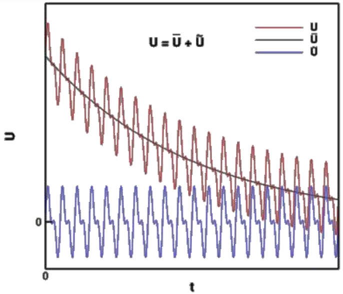
Coefficients (d_j^n) derived analytically using convolution of Fourier transform and synthesis.

Temporal Derivative : BDFTS

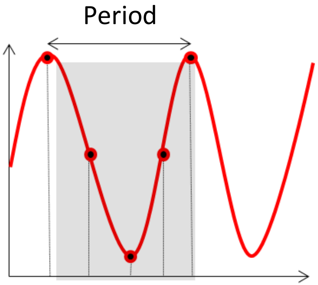
- Problems with a slow transient in addition to a strong periodic behavior in time (quasi-periodic problems)

$$U(t) = \tilde{U}(t) + \overline{U}(t)$$

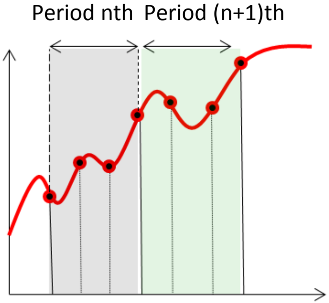
$$\underset{\text{periodic}}{\nearrow} \qquad \underset{\text{Mean}}{\uparrow}$$



Concept of polynomial subtraction for spectral methods(Gottlieb and Orzag (1977), Lanczos)



TS



BDFTS

Temporal Derivative : BDFTS

BDF1TS derivative:

$$\frac{\partial U^n}{\partial t} = \sum_{j=1}^{N-1} d_n^j U^j - \left(\sum_{j=1}^{N-1} d_n^j \varphi_{12}(t_j) - \varphi'_{12}(t_n)\right)U^{m+1} - \left(\sum_{j=1}^{N-1} d_n^j \varphi_{11}(t_j) - \varphi'_{11}(t_n)\right)U^m$$

n=1,2,…,N

φ_{11} and φ_{12} are the linear interpolation functions

$\sum_{j=1}^{N-1} d_n^j U^j$ Time spectral derivative

U^{m+1} Ending point of the period (Unknown)

U^m Beginning point of the period (known)

BDFTS derivation can be reformulated as:

$$[D_{qp}]\vec{U} = [D_{pp}]\vec{U} + [\mathrm{Mat}_{r1}]\vec{U} + \mathrm{co}\vec{\mathrm{n}}\mathrm{st.}$$

Spectral Matrix

Rank-1 Matrix

TS Solvers : Approximate Factorization

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n})$ = 0

TS Solvers : Approximate Factorization

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = 0$

- The residual is obtained from: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = \text{Res}$

Obtained from any formulation

TS Solvers : Approximate Factorization

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = 0$

- The residual is obtained from: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = \text{Res}$

Obtained from TS or BDFTS formulations

- The entire non-linear space-time system of equations is linearized by Newton-Raphson method

$$[A]\Delta U = -Res$$

[A] is the complete time-spectral Jacobian matrix

Res is the total residual of time-spectral system

TS Solvers : Approximate Factorization

- Approximates [A] as: $[A] \approx ([Temporal\ Part])([Spatial\ Part])$

- Separates spatial and temporal parts

- Not exact and include an error which is $\Delta\tau J[D]$

  J is the Jacobian of the spatial part of the system
  [D] is the TS or BDFTS derivative matrix

- Solves for ΔU in two steps:

  - ✓ solve the spatial part to find intermediate value, $\Delta\Delta U$ using direct or iterative methods e.g. block Gauss-Seidel
  - ✓ Using $\Delta\Delta U$, the temporal matrix is inverted to find ΔU

Improvement in TS solver

- Factorization error depends on the pseudo-time step
- Using AF as the solver suffers from requiring a small pseudo-time step or CFL number

Improvement in TS solver

- Factorization error depends on the pseudo-time step
- Using AF as the solver suffers from requiring a small pseudo-time step or CFL number

Using AF as a preconditioner in the context of the Newton-Krylov method

Newton-Raphson Method

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = 0$

- The residual is obtained from: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = \text{Res}$

  Obtained from TS or BDFTS formulations

- The entire non-linear space-time system of equations is linearized by Newton-Raphson method

$$[A]\Delta U = -Res$$

[A] is the complete time-spectral Jacobian matrix

Res is the total residual of time-spectral system

Newton-Raphson Method

- The non-linear space time system is: $\frac{\partial V U}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = 0$

- The residual is obtained from: $\frac{\partial V U}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = \text{Res}$

Obtained from TS or BDFTS formulations

- The entire non-linear space-time system of equations is linearized by Newton-Raphson method

$$[A]\Delta U = -Res$$

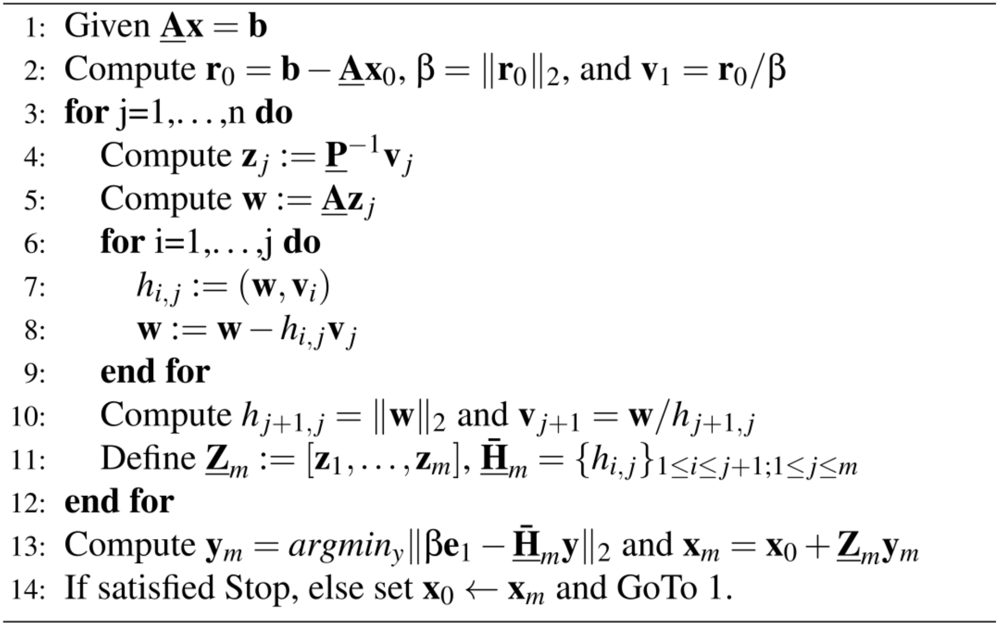
The linear system over all time and space at each step of Newton solution is solved to a specified linear tolerance using a Krylov method (GMRES)

[A] is the complete time-spectral Jacobian matrix

Res is the total residual of time-spectral system

TS Solvers : GMRES

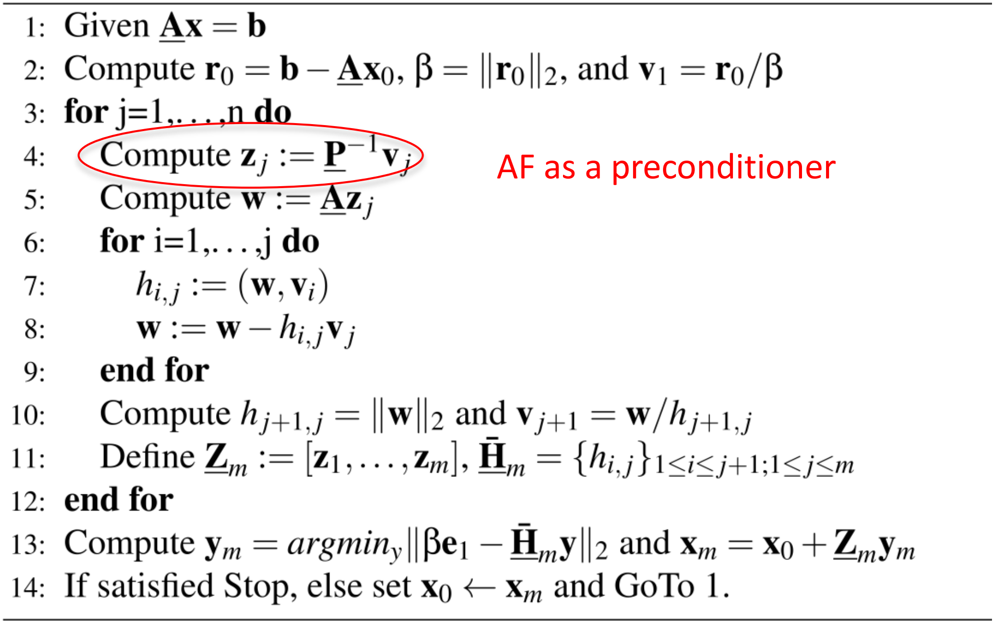
- Flexible GMRES algorithm that allows an iterative method as a preconditioner has been described by Saad:



```
 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 $\underline{\mathbf{Z}}_m := [\mathbf{z}_1, \ldots, \mathbf{z}_m]$, $\underline{\bar{\mathbf{H}}}_m = \{h_{i,j}\}_{1 \leq i \leq j+1; 1 \leq j \leq m}$
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.
```

TS Solvers : GMRES/AF

- Flexible GMRES algorithm that allows an iterative method as a preconditioner has been described by Saad:
- AF solver is used as a preconditioner in line 4 of the algorithm



```
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.
```

AF as a preconditioner

- Introduction
- Governing Equations
- Challenges
- Novelty
- Results
- Summary and Conclusions
- Future Work

Drawbacks of TS

- Traditionally, TS formulation was based on Discrete Fourier Transform

$$\frac{\partial U^n}{\partial t} = \frac{2\pi}{T}\sum_{k=\frac{-N}{2}}^{k=\frac{N}{2}-1} ik\hat{U}_k e^{ikn\Delta t\frac{2\pi}{T}} = \sum_{j=0}^{N-1} d_n^j U^j$$

$$U^n = \sum_{k=-\frac{N}{2}}^{k=\frac{N}{2}} \hat{U}_k e^{ikn\Delta t\frac{2\pi}{T}}$$

Fourier Inverse Transform

$$\hat{U}_k = \frac{1}{N}\sum_{n=0}^{N-1} U^n e^{-ikn\Delta t\frac{2\pi}{T}}$$

Fourier Transform

How many operations are involved?

Drawbacks of TS

Based on Discrete Fourier Transform

$$\frac{\partial U^n}{\partial t} = \frac{2\pi}{T} \sum_{k=\frac{-N}{2}}^{k=\frac{N}{2}-1} ik\widehat{U}_k e^{ikn\Delta t\frac{2\pi}{T}} = \sum_{j=0}^{N-1} d_n^j U^j$$

Rewriting the summation results in dense matrix $[D_{TS}]$

$$\begin{bmatrix} \dot{U}_1 \\ \dot{U}_2 \\ \dots \\ \dot{U}_{N-1} \end{bmatrix} = \begin{bmatrix} d_0^0 & d_0^1 & \dots & d_0^{N-1} \\ d_1^0 & d_1^1 & \dots & d_1^{N-1} \\ \dots & \dots & \dots & \dots \\ d_{N-1}^0 & d_{N-1}^1 & \dots & d_{N-1}^{N-1} \end{bmatrix}_{(N\times N)} \begin{bmatrix} U_0 \\ U_1 \\ \dots \\ U_{N-1} \end{bmatrix}$$

Total number of operations: $O(N^2)$

Wall clock time scales linearly with number of time instances (running in parallel) which is not desirable.

- Introduction
- Governing Equations
- Challenges
- **Novelty**
- Results
- Summary and Conclusions
- Future Work

Novelty

We tried to overcome this challenge:

- In this work, time-spectral method is implemented based on the parallel fast Fourier transform (FFT)
- The parallel FFT-based AF and GMRES/AF are implemented

Fast Fourier Transform (FFT)

Assuming the number of samples: $N = 2^L$

Considering Discrete Fourier Transform Formulation:

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

Splitting the summation in two parts:

$$\widehat{U}_k = \frac{1}{N}\left(\sum_{n=0}^{\frac{N}{2}-1} U^{2n} e^{-ik(2n)\Delta t\frac{2\pi}{T}} + \sum_{n=0}^{\frac{N}{2}-1} U^{2n+1} e^{-ik(2n+1)\Delta t\frac{2\pi}{T}}\right)$$

$$\widehat{U}_k = \frac{1}{N}\left(\sum_{n=0}^{\frac{N}{2}-1} U^{2n} e^{-ikn\frac{2\pi}{N/2}} + e^{-ik\frac{2\pi}{N}} \sum_{n=0}^{\frac{N}{2}-1} U^{2n+1} e^{-ikn\Delta t\frac{2\pi}{N/2}}\right) = \frac{1}{N}\left(e_k + w^k o_k\right)$$

DFT of even sequence $\{U^{2n}\}$

DFT of odd sequence $\{U^{2n+1}\}$

Fast Fourier Transform (FFT)

Recursively split each part to even /odd groups until each group has only one member.

$$\hat{U}_k = \frac{1}{N}\sum_{n=0}^{N-1} U^n e^{-ikn\Delta t\frac{2\pi}{T}}$$

$$\hat{U}_k = \frac{1}{N}\Big(\sum_{n=0}^{\frac{N}{2}-1} U^{2n} e^{-ikn\frac{2\pi}{N/2}} + e^{-ik\frac{2\pi}{N}}\sum_{n=0}^{\frac{N}{2}-1} U^{2n+1} e^{-ikn\Delta t\frac{2\pi}{N/2}}\Big)$$

$$\hat{U}_k = \frac{1}{N/2}\Big(\sum_{n=0}^{\frac{N}{4}-1} U^{2n} e^{-ikn\frac{2\pi}{N/4}} + e^{-ik\frac{2\pi}{N/2}}\sum_{n=0}^{\frac{N}{4}-1} U^{2n+1} e^{-ikn\Delta t\frac{2\pi}{N/4}}\Big)$$

$$\hat{U}_k = \frac{1}{N/2}\Big(\sum_{n=0}^{\frac{N}{4}-1} U^{2n} e^{-ikn\frac{2\pi}{N/4}} + e^{-ik\frac{2\pi}{N/2}}\sum_{n=0}^{\frac{N}{4}-1} U^{2n+1} e^{-ikn\Delta t\frac{2\pi}{N/4}}\Big)$$

Number of divisions : $\boldsymbol{L = log_2^N}$

Parallel FFT Communication Count

For each $\widehat{U}_k$ in each level e_k and o_k are needed;

each member in each level requires the data of another member to calculate its share

$$\widehat{U}_k = \frac{1}{N}(e_k + W^k o_k)$$

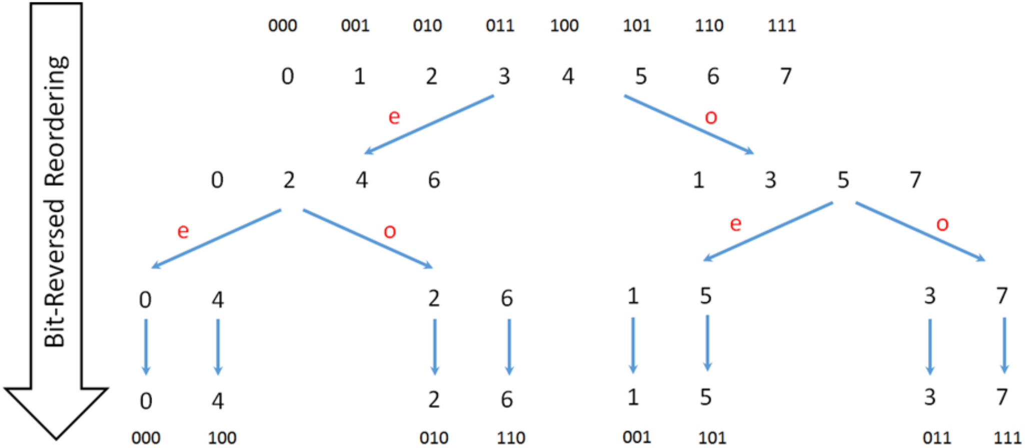
In each level *N* communication occurs.

Total number of communication $O(N\log_2 N)$

N	N^2	$\mathrm{N}\log_2 N$	$\frac{N^2}{\mathrm{N}\log_2 N}$
512	2^{18}	$2^9 * 9$	56.888
1024	2^{20}	$2^{10} * 10$	102.4
2048	2^{22}	$2^{11} * 11$	186.181

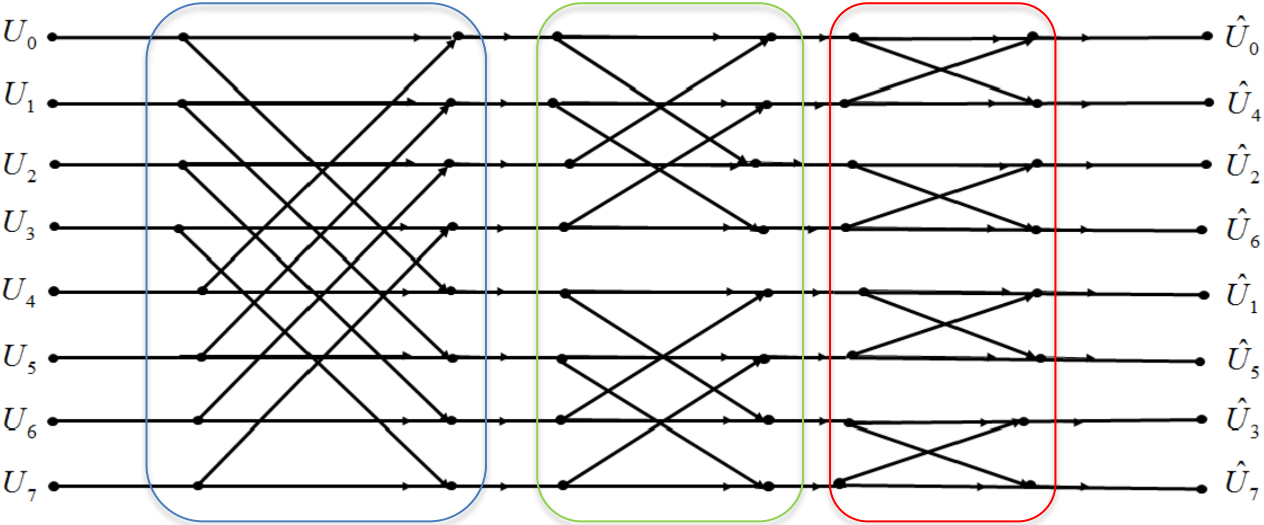
Reordering

- Splitting into odd/ even groups changes the order of samples
- Danielson-Lanczos lemma is used to find odd/even reordering pattern of samples
- The new ordering is obtained by bit-reversal of the original sample.



Recursive subdivision of N=8 sample set and corresponding bit-reversal ordering

Communication Pattern of FFT



- Communication pattern for all levels for 8 number of samples
- The levels in which further processors should communicate are more expensive

TS Derivative based on FFT

- Calculate FFT of samples ($O(logN)$ communication)
- Multiply $\widehat{U}_k$ into corresponding ik (No communication)
- Calculate the inverse of $ik\widehat{U}_k$($O(logN)$ communication)

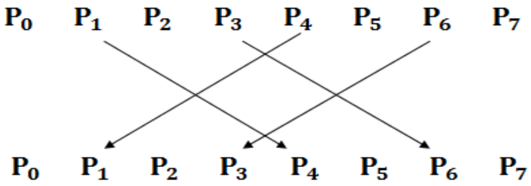
TS Derivative based on FFT

- Calculate FFT of samples ($O(logN)$ communication)
- Multiply $\widehat{U}_k$ into corresponding ik (No communication)
- Calculate the inverse of $ik\widehat{U}_k$($O(logN)$ communication)

The number of communication in FFT-based TS is $O(logN)$

No Reordering Required for FFT-TS

- Standard parallel FFT requires final reordering of data
  - Entire spatial grid from each core.

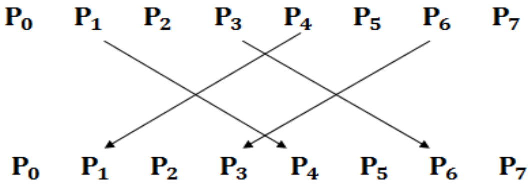


Pattern of communication due to reordering for 8 number of samples in parallel FFT routine.

No Reordering Required for FFT-TS

- Standard parallel FFT requires final reordering of data
  - Entire spatial grid from each core.

- Time spectral implementation always requires the application of a forward FFT followed by an Inverse FFT



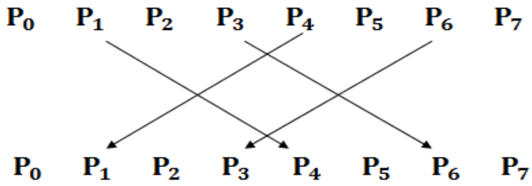
Pattern of communication due to reordering for 8 number of samples in parallel FFT routine.

No Reordering Required for FFT-TS

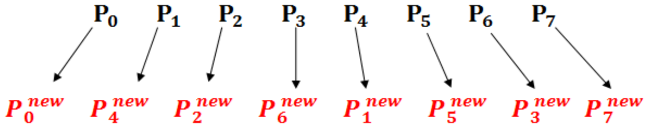
- Standard parallel FFT requires final reordering of data
  - Entire spatial grid from each core.

- Time spectral implementation always requires the application of a forward FFT followed by an Inverse FFT

- There is no need to reorder data
  - All that is required is k in the IFFT and the address of each core for the communication pattern at each level


P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7
P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7

Pattern of communication due to reordering for 8 number of samples in parallel FFT routine.


P_0 P_1 P_2 P_3 P_4 P_5 P_6 P_7
P_0^{new} P_4^{new} P_2^{new} P_6^{new} P_1^{new} P_5^{new} P_3^{new} P_7^{new}

New addressing of cores for 8 number of samples to avoid extra communication in parallel TS routine

$$\frac{\partial U^n}{\partial t} = \frac{2\pi}{T} \sum_{k=\frac{-N}{2}}^{k=\frac{N}{2}-1} \boxed{ik}\hat{U}_k e^{ikn\Delta t \frac{2\pi}{T}}$$

Extension of FFT Application

- Time-spectral method is implemented based on base-3 FFT

- The number of operations reduces from $O(N^2)$ to $O(2Nlog_3^N)$

Extension of FFT Application

- Implementation of FFT-based second-order time-spectral derivative

- The number of operations reduces from $O(N^2)$ to $O(NlogN)$

- In aero-structural problems such as flutter problems, …

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

FFT-AF in Purely Periodic Problems

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n})$ = 0
- The residual is obtained from: $[D_{PP}]VU + R(U^n, \dot{x}^n, \vec{n})$ = Res
- The entire non-linear space-time system of equations is linearized by Newton-Raphson method

$$[A]\Delta U = -Res$$

$$[A] = [\frac{V}{\Delta\tau} + J + V[D_{PP}]]$$

[A] is the complete time-spectral Jacobian matrix

Res is the total residual of time-spectral system

V is the cell volume

$\Delta\tau$ is the AF pseudo time-step

J is the Jacobian of spatial part

$[D_{PP}]$ is the spectral matrix

FFT-AF in Purely Periodic Problems

- Approximates [A] as: $[A] \approx \underbrace{([I] + \Delta\tau[D_{PP}])}_{\text{Temporal Part}}\underbrace{(\frac{V}{\Delta\tau}[I] + [J])}_{\text{Spatial Part}}$

- Find intermediate value $\Delta\Delta U$ by solving the spatial part, using any direct or iterative solver

- Solve temporal part
  - Take FFT of $\Delta\Delta U$ to find $\Delta\Delta\widehat{U}_k$
  - Multiply $\Delta\Delta\widehat{U}_k$ by $\frac{1}{1+iwk\Delta\tau}$ to find $\Delta\widehat{U}_k$
  - Take IFFT of $\Delta\widehat{U}_k$ to find ΔU

FFT-AF in Quasi-Periodic Problems

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n})$ = 0
- The residual is obtained from: $[D_{qp}]VU + R(U^n, \dot{x}^n, \vec{n})$ = Res

$$[D_{qp}]VU = [D_{PP}]VU + [Mat_{r1}]VU + const.$$

- The entire non-linear space-time system of equations is linearized by Newton-Raphson method

$$[A]\Delta U = -Res = -[D_{qp}]VU - R(U^n, \dot{x}^n, \vec{n})$$

[A] is the complete time-spectral Jacobian matrix

Res is the total residual of time-spectral system

V is the cell volume

$\Delta\tau$ is the AF pseudo time-step

J is the Jacobian of spatial part

$[D_{qp}]$ is the quasi-periodic matrix

$$[A] = [\frac{V}{\Delta\tau} + J + V[D^*_{qp}]]$$

$$[D^*_{qp}] = [D_{PP}] + [Mat_{r1}]$$

FFT-AF in Quasi-Periodic Problems

- Approximates [A] as: $$[A] \approx \underbrace{([I] + \Delta\tau[D_{qp}^{*}])}_{\text{Temporal Part}}\underbrace{(\frac{V}{\Delta\tau}[I] + [J])}_{\text{Spatial Part}}$$

$$[D_{qp}^{*}] = [D_{pp}] + [Mat_{r1}]$$

- Find intermediate value $\Delta\Delta U$ by solving the spatial part, using any direct or iterative solver

- Using the intermediate value, $\Delta\Delta U$ the temporal matrix is inverted to find ΔU

$$\Delta U = [[I] + \Delta\tau\left[D_{qp}^{*}\right]]^{-1}\Delta\Delta U$$

FFT-AF in Quasi-Periodic Problems

- Approximates [A] as: $$[A] \approx \underbrace{([I] + \Delta\tau[D_{qp}^{*}])}_{\text{Temporal Part}}\underbrace{(\frac{V}{\Delta\tau}[I] + [J])}_{\text{Spatial Part}}$$

$$[D_{qp}^{*}] = [D_{pp}] + [Mat_{r1}]$$

- Find intermediate value $\Delta\Delta U$ by solving the spatial part, using any direct or iterative solver

- Using the intermediate value, $\Delta\Delta U$ the temporal matrix is inverted to find ΔU

$$\Delta U\text{=}[[I] + \Delta\tau\left[D_{qp}^{*}\right]]^{-1}\Delta\Delta U$$

Using FFT?

FFT-AF in Quasi-Periodic Problems

- Calculation of the temporal part of AF can be done much easier in frequency domain
- The temporal equation:

$$(I+\Delta\tau[D^{*}_{qp}])\Delta U = (I+\Delta\tau[D_{pp}]+\Delta\tau[Mat_{r1}])\Delta U = \Delta\Delta U$$

FFT in Approximate Factorization Scheme

- Calculation of the temporal part of AF can be done much easier in frequency domain
- The temporal equation:

$$(I+\Delta\tau[D^*_{qp}])\Delta U = (\underbrace{I+\Delta\tau[D_{pp}]}_{[D^*_{PP}]}+\Delta\tau[\underbrace{Mat_{r1}}_{[Mat_{r1}]=[\vec{u}\vec{v}^T]}])\Delta U = \Delta\Delta U$$

- Easy to find the inverse of $[D^*_{PP}]$ in the FD
  - Spectral matrix is diagonal in the FD
- $[D^*_{PP}]$ is modified by a rank-1 matrix
- The inverse of the temporal matrix is calculated using the **Sherman Morrison** formulation
- Two times inversion of the $[D^*_{PP}]$ is required in this process.

$$([D^*_{PP}]+\vec{u}\vec{v}^T)^{-1} = [D^*_{PP}]^{-1} - \frac{[D^*_{PP}]^{-1}\vec{u}\vec{v}^T[D^*_{PP}]^{-1}}{\underbrace{1+\vec{v}^T[D^*_{PP}]^{-1}\vec{u}}_{\text{Scalar}}}$$

FFT in Approximate Factorization Scheme

- Find the intermediate value, $\Delta\Delta U$ by solving the spatial part
- Solve the temporal part:

I. Find FFT of $\Delta\Delta U$ to find $\Delta\Delta\hat{U}_k$

II. Find $\Delta\hat{U}_k$ by taking the inverse of the temporal matrix using Sherman-Morrison formulation

III. Transfer back the result to time domain using IFFT to obtain ΔU

Newton-Raphson Method

- The non-linear space time system is: $\frac{\partial VU}{\partial t} + R(U^n, \dot{x}^n, \vec{n}) = 0$

- The residual is obtained from: $\boxed{\frac{\partial VU}{\partial t}} + R(U^n, \dot{x}^n, \vec{n}) = \text{Res}$

Obtained from TS or BDFTS formulations

- The entire non-linear space-time system of equations is linearized by Newton-Raphson method

$$[A]\Delta U = -Res$$

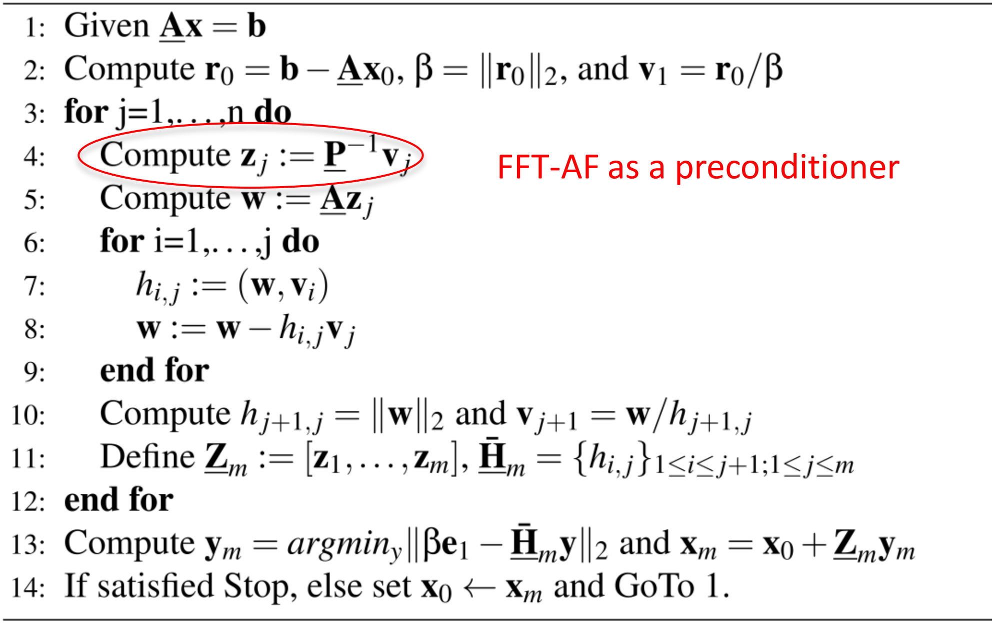
The linear system over all time and space at each step of Newton solution is solved to a specified linear tolerance using a Krylov method (GMRES)

[A] is the complete time-spectral Jacobian matrix

Res is the total residual of time-spectral system

FFT-based GMRES/AF

- Flexible GMRES algorithm that allows an iterative method as a preconditioner has been described by Saad:
- AF solver is used as a preconditioner in line 4 of the algorithm



```
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.
```

FFT-AF as a preconditioner

More on GMRES/AF

- Two pseudo-time terms are used in GMRES:
  - The constant pseudo-time term in the preconditioner:

$$[A] \approx [[I] + \boxed{\Delta\tau_{AF}}[D_{TS}]][\frac{V}{\boxed{\Delta\tau_{AF}}}[I] + [J]]$$

  - The growing pseudo-time term in the space-time Jacobian of the GMRES:

$$[A] = [\frac{V}{\boxed{\Delta\tau_{Newton}}} + J + V[D_{TS}]]$$

- The pseudo-time term in the FGMRES grows rapidly so that an exact Newton method can be recovered.

- Here we employed an inexact Newton approach for efficiency reasons.
  - Linear tolerance of 0.1

- Introduction
- Governing Equations
- Challenges
- Novelty
- **Results**
- Summary and Conclusions
- Future Work

Test Cases

- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**

Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion

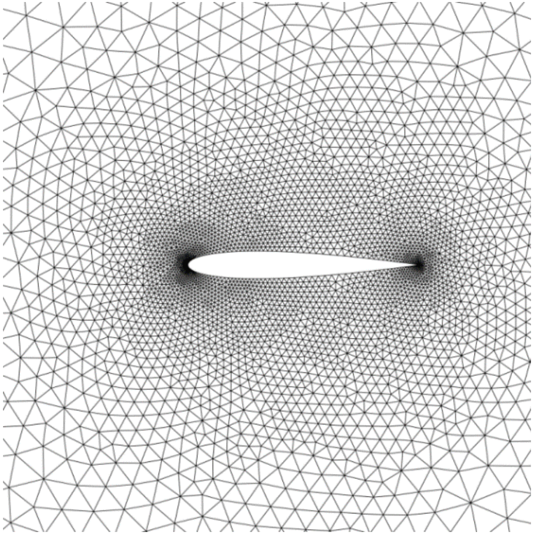
- Naca-0012 Airfoil
- 15573 triangular elements
- Free stream Mach = 0.755
- Prescribed pithing motion:

$$\alpha_t = \alpha_0 + \alpha_A Sin(\omega t)$$

$$\alpha_0 = 0.016^{\circ} \quad \alpha_A = 2.51^{\circ}$$

- ω is specified via reduced frequency

$$k_c = 0.0814 - 0.1628$$

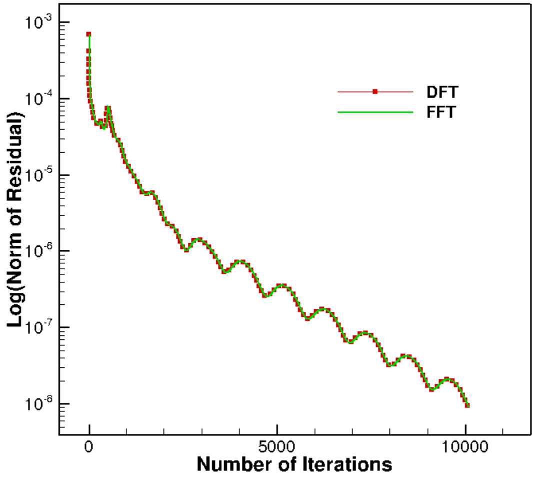


Test Cases

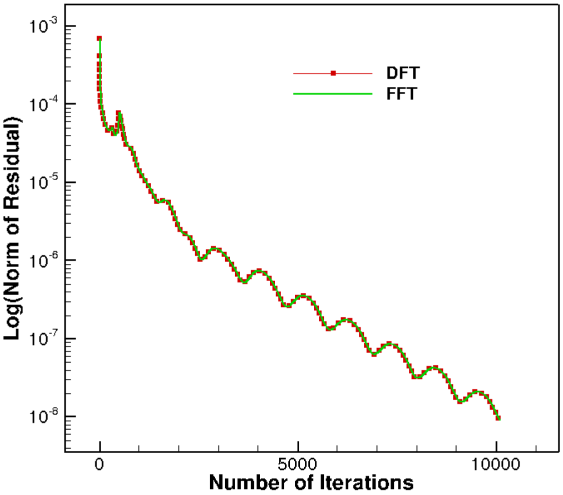
- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - **Case 1-1 : Testing the performance of FFT based AF for Case 1**

Case1-1 : AF Residual Validation

- DFT and FFT based AF solver

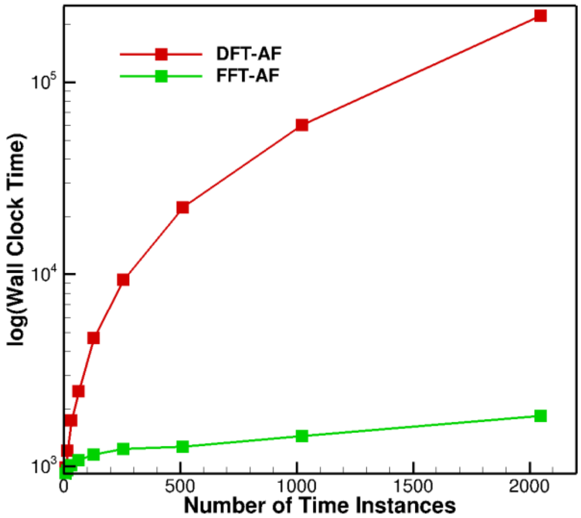


$N = 256$

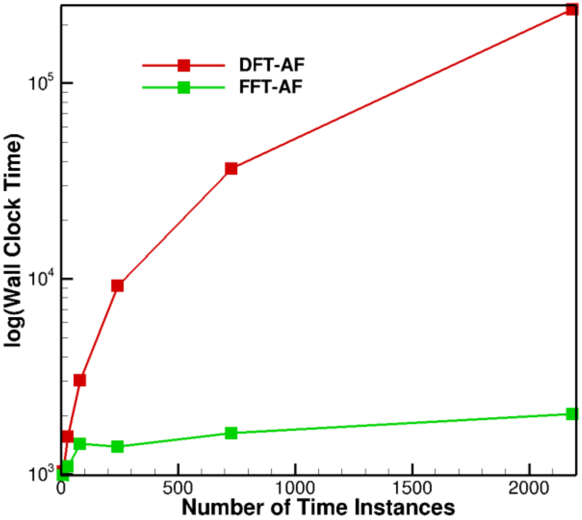


$N = 243$

Case 1-1 : AF Performance comparison



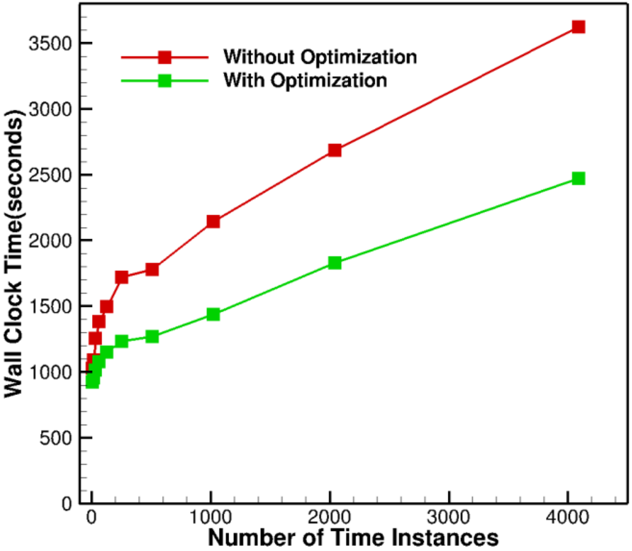
Even



Odd

- DFT- and FFT- based AF solvers
- Even number of samples up to 2048, odd number of samples up to 2187

Case 1-1: Optimization for Real Valued Samples

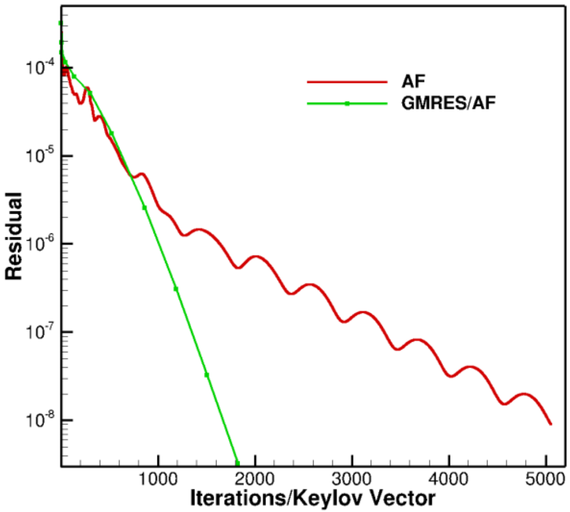


➢ Wall clock time versus number of time instances for original complex FFT and real-data split FFT implementation

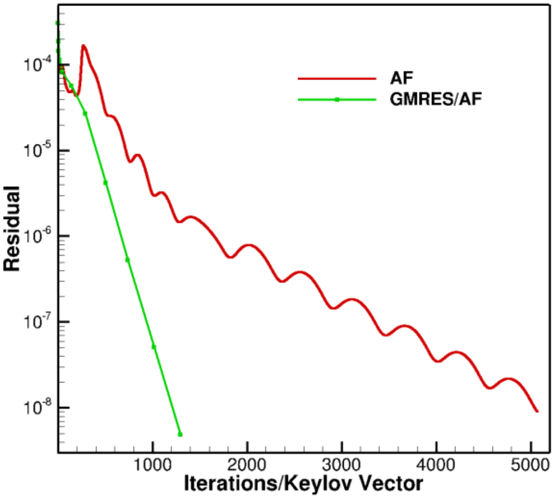
Test Cases

- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - **Case 1-1 : Testing the performance of FFT based AF for Case 1**
  - **Case 1-2 : Testing the performance of FFT based GMRES/AF for Case 1**

Case 1-2: FFT-GMRES/AF solver performance



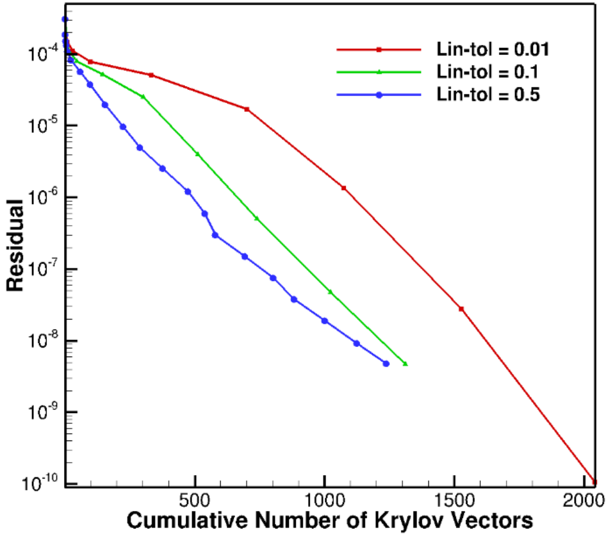
N = 8



N = 1024

- Comparison of the non-linear residual versus iterations for the AF solver and versus Krylov vectors for the GMRES/AF solver with 8 and 1024 number of time instances.

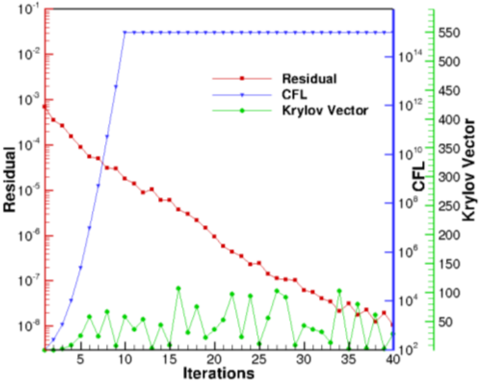
Case 1-2: Study of Linear Tolerance



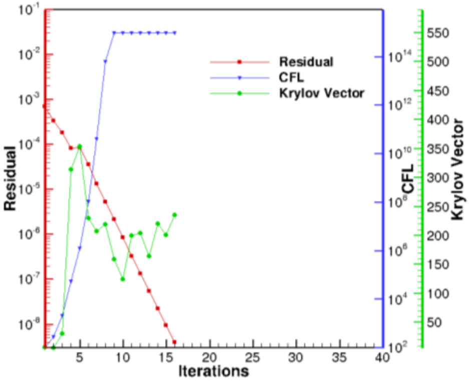
N = 256

- ➢ Residual versus Krylov vectors for different linear tolerances for 256 number of time instances.

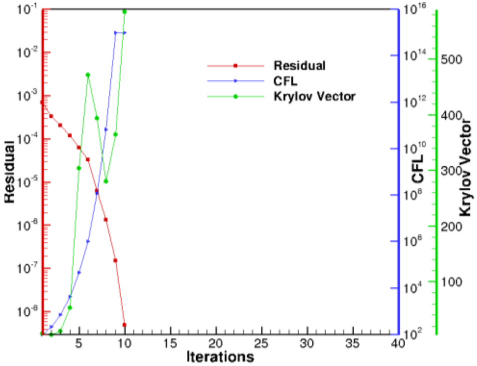
Case 1-2: Study of Linear Tolerance



Linear-tol = 0.5



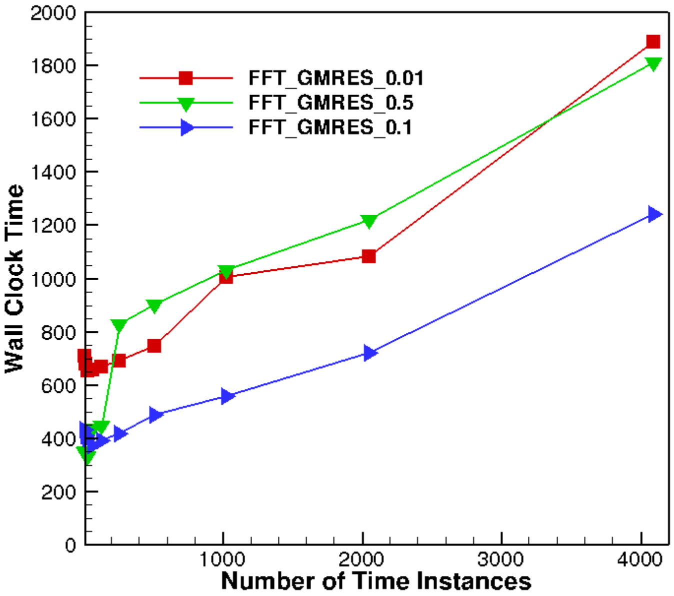
Linear-tol = 0.1



Linear-tol = 0.01

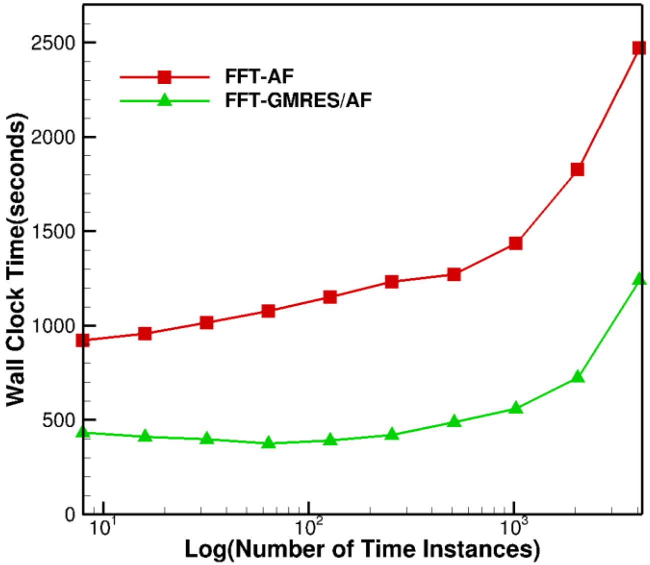
- ➢ Non-linear convergence, CFL history, and number of Krylov vectors in each iteration for linear tolerance of 0.5 (left plot), 0.1 (middle plot), 0.01 (right plot).

Case 1-2: Study of Linear Tolerance



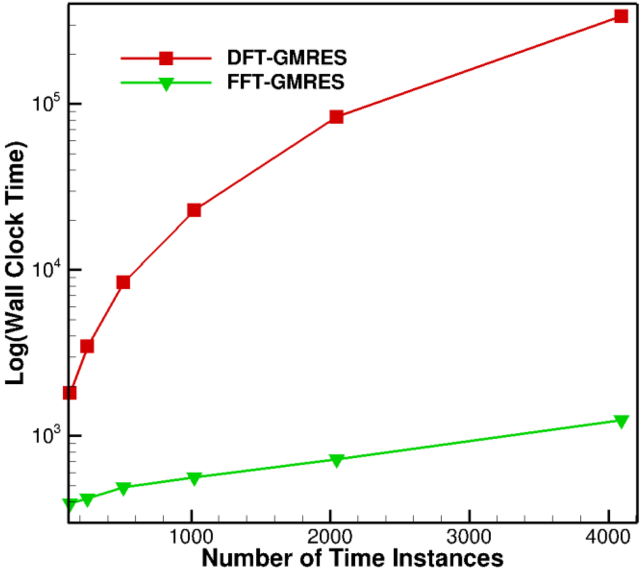
- ➢ Wall-clock time versus number of time instances for different linear tolerances .

Case 1-2: Performance of FFT-based AF and GMRES/AF



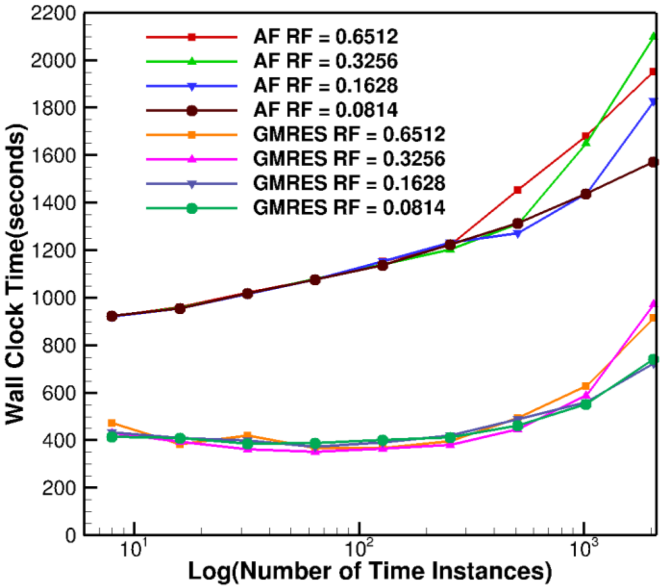
- ➢ Wall-clock time versus number of time instances for FFT based GMRES/AF and FFT based AF solver for up to 2048 number of time instances

Case 1-2: Performance of DFT and FFT based GMRES/AF



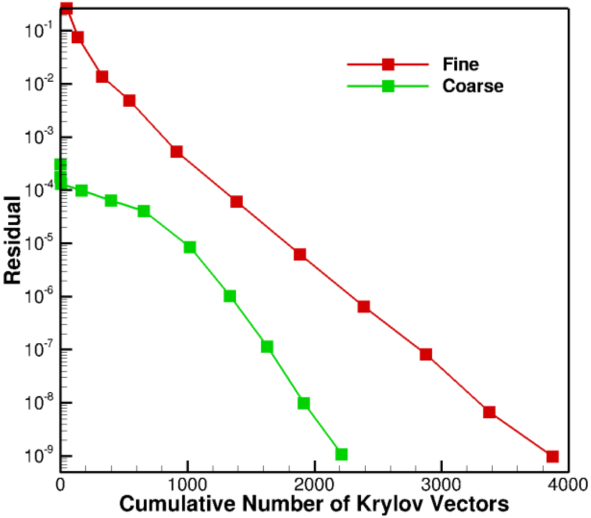
- ➢ Wall-clock time for DFT and FFT based GMRES/AF solvers for up to 2048 number of time instances.

Case1-2: Solver Characteristic

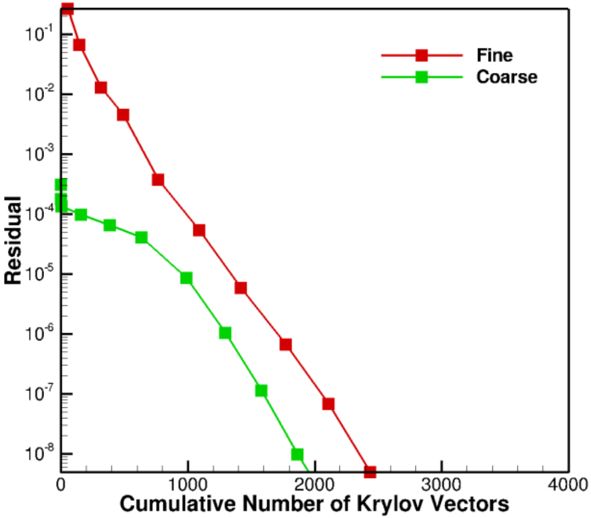


➢ Wall-clock time versus number of time instances for FFT based GMRES/AF and FFT based AF solver using different reduced frequencies

Case1-2: Mesh Resolution Study



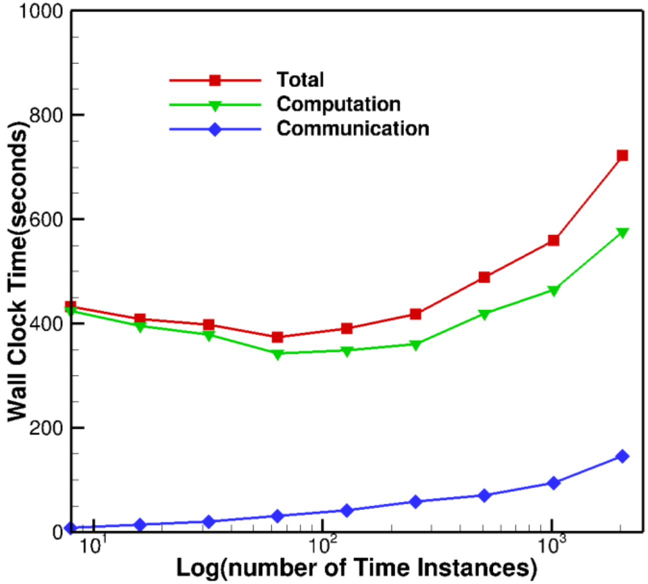
20 block-Jacobi sweeps



Solving Jacobi to machine zero

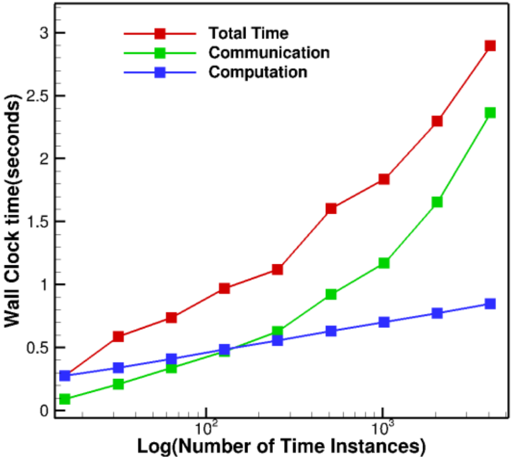
➢ Convergence study of GMRES/AF solver using 64 number of time instances and linear tolerance of 0.1, with: 20 block-Jacobi sweeps in the preconditioner(Left) solving Jacobi to machine zero in the preconditioner(Right)

Wall Clock Time due to Communication and Computation



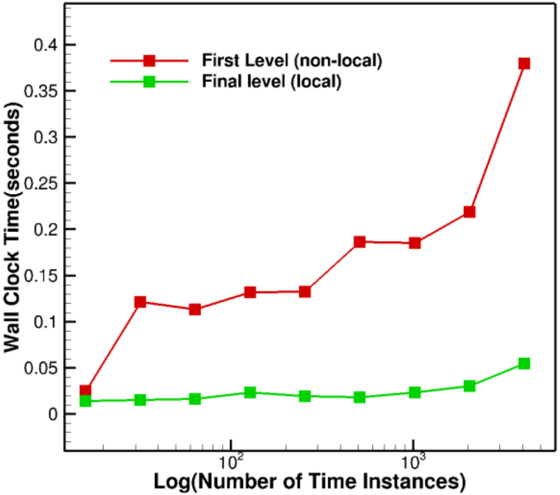
➢ Breakdown of wall-clock time for computation and communication of the solver running on NCAR Wyoming Yellowstone supercomputer using up to 2048 processors

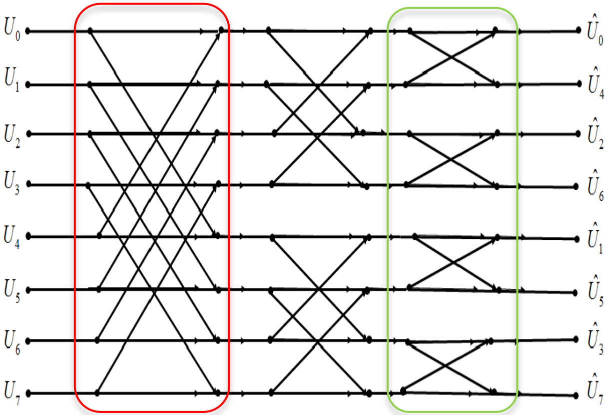
Wall Clock Time due to Communication and Computation



- Breakdown of wall-clock time for computation and communication within parallel FFT routine running on NCAR- Wyoming Yellowstone supercomputer using up to 4096 processors
- Computation displays expected O(logN) weak scaling
- The wall clock grows faster than expected due to pattern of communication each level

Wall Clock Time for First and Last Level



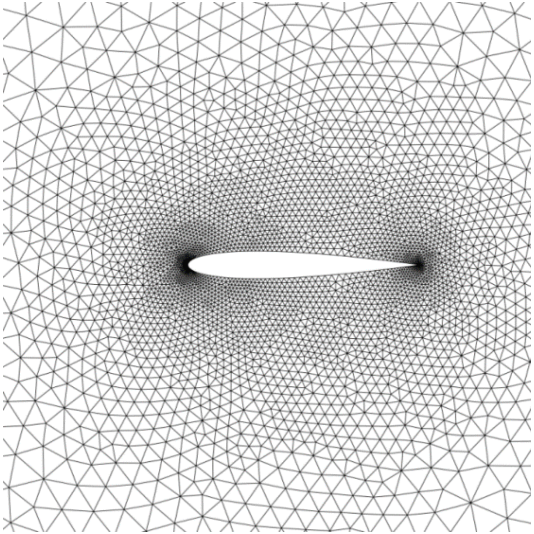


- Comparison of communication time for first and last level of parallel FFT routine using up to 4096 processors
- Difference in wall clock time due to non local communication. (Verified by NWSC- Yellowstone system staff)

Test Cases

- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - **Case 1-1 : Testing the performance of FFT based AF for Case 1**
  - **Case 1-2 : Testing the performance of FFT based GMRES/AF for Case 1**
- **Case 2: Purely Periodic Pitching Airfoil with Gaussian Bump Motion**

Case 2: Purely Periodic Pitching Airfoil with Gaussian Bump Motion



- Naca-0012 Airfoil
- 15573 triangular elements
- Free stream Mach = 0.755
- Prescribed pithing motion:

$$\alpha_t = \frac{1}{\sqrt{20\pi}} e^{-\frac{(t-10)^2}{2}}$$

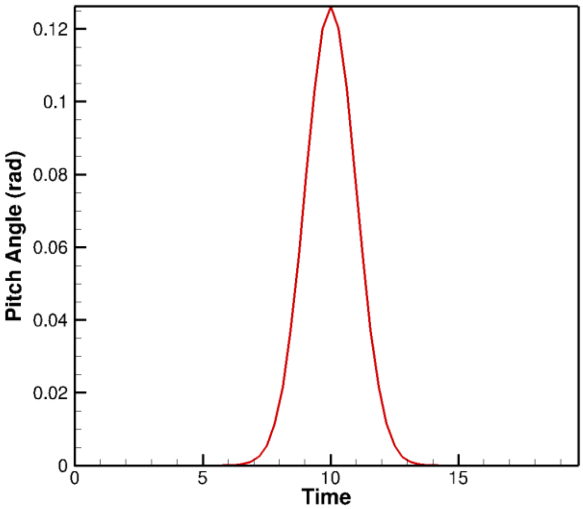
- ω is specified via reduced frequency

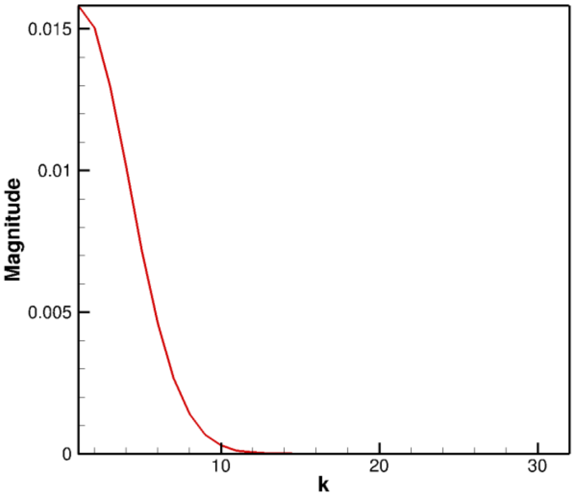
$$k_c = 0.208$$

Test Cases

- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - **Case 1-1 : Testing the performance of FFT based AF for Case 1**
  - **Case 1-2 : Testing the performance of FFT based GMRES/AF for Case 1**
- **Case 2: Purely Periodic Pitching Airfoil with Gaussian Bump Motion**
  - **Case 2-1 : Comparison of the performance of FFT based GMRES/AF and BDF2 for Case 2**

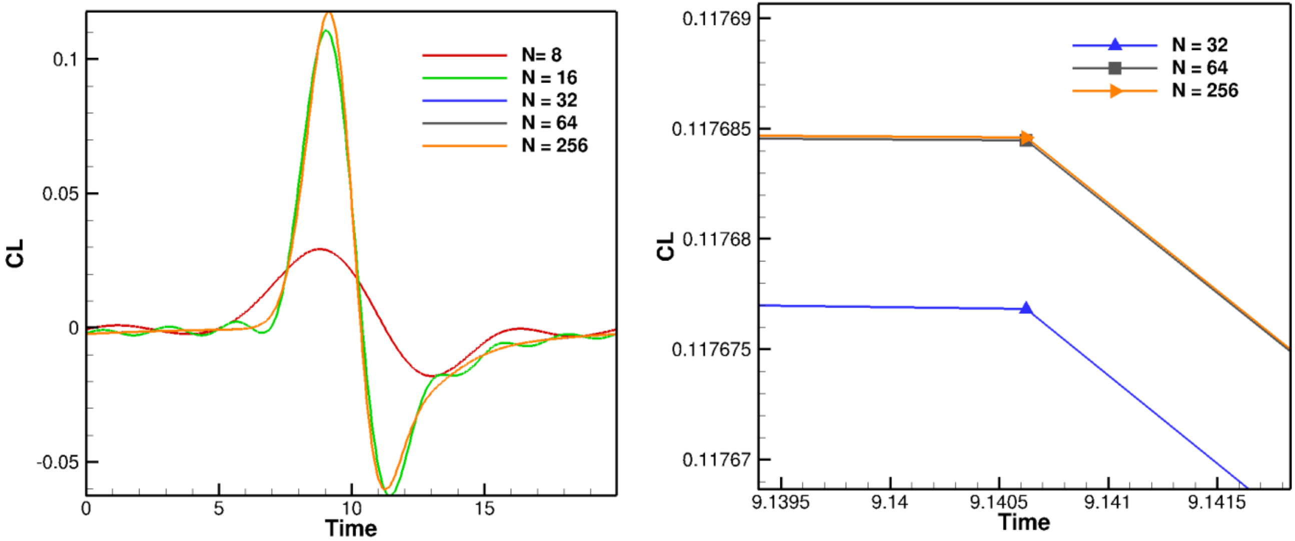
Case2-1: Gaussian Bump Pitching Motion





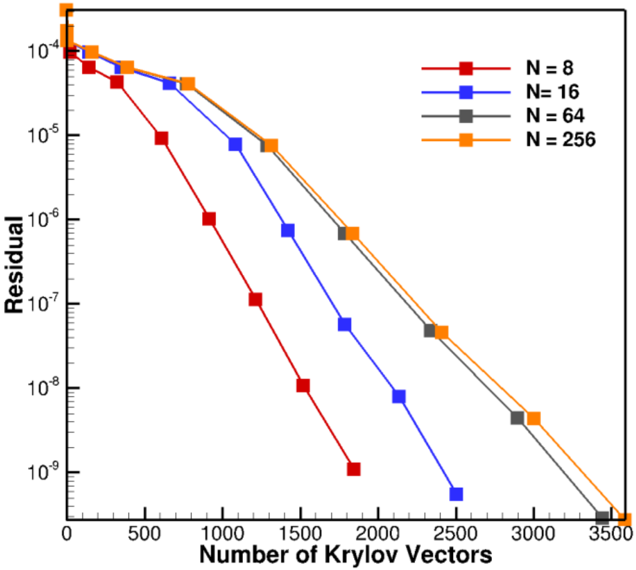
- Time history of Gaussian bump prescribed pitching motion and (Left) and frequency content of prescribed motion signal (Right)

Case2-1: TS Solution



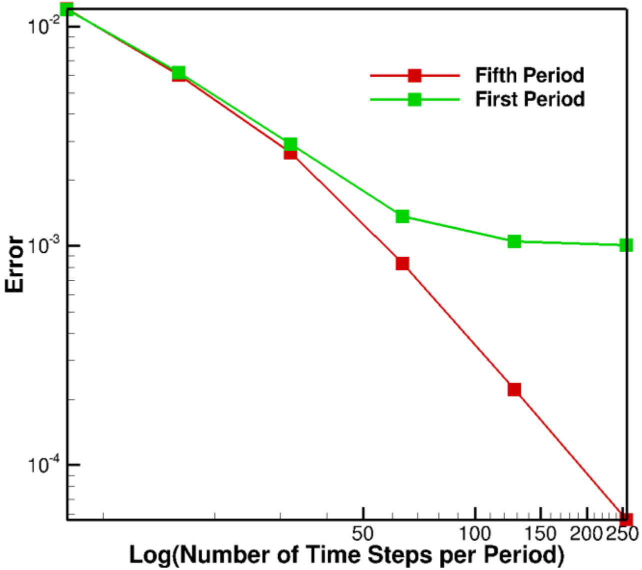
- Computed lift coefficient history using TS solver with different number of time instances (Left) and details of differences between TS solutions for N = 32, 64 and 256 (Right)

Case2-1: FFT- GMRES/AF Convergence



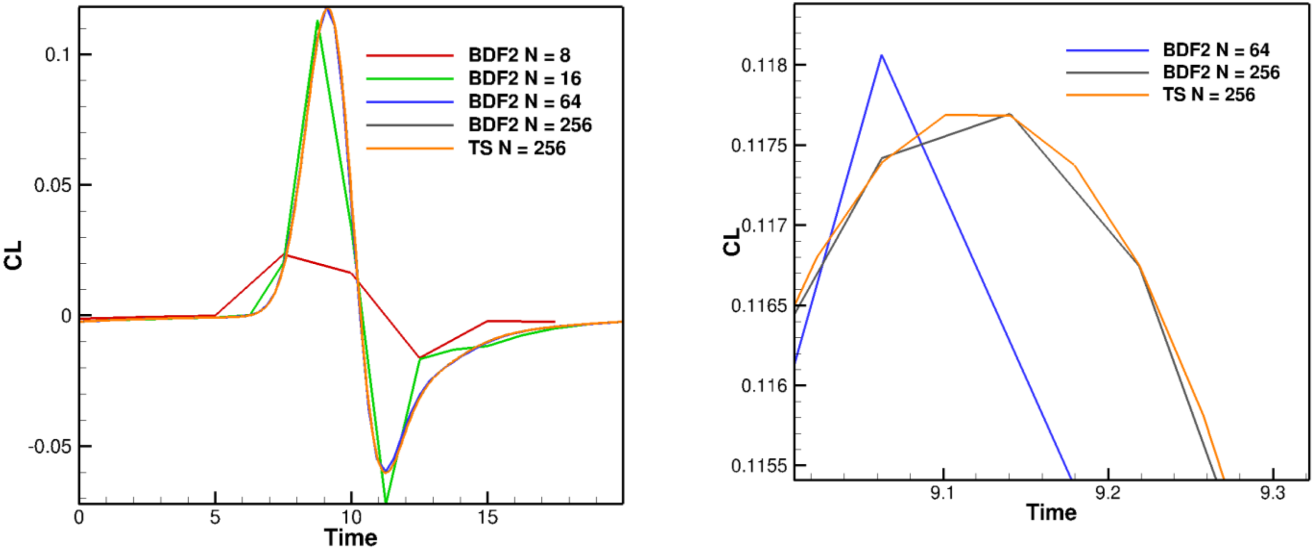
➢ Convergence histories for TS solver as measured by residual versus cumulative number of Krylov vectors, using different number of time-instances

Case2-1: BDF2 Error Study



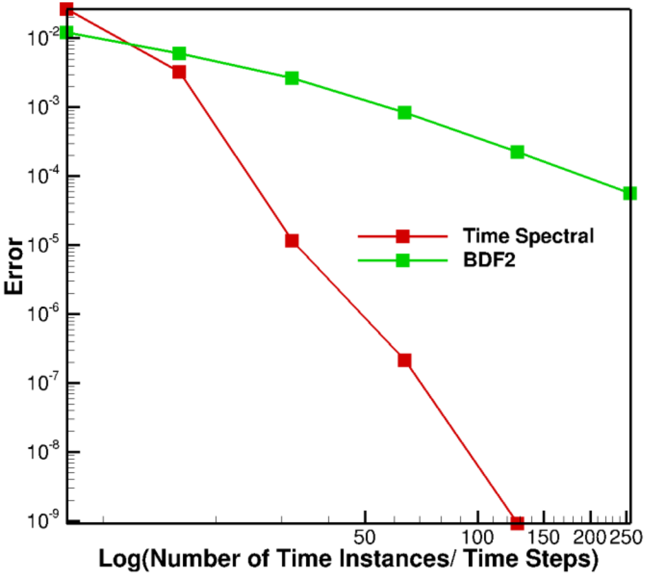
- Temporal error of BDF2 solution for the first and fifth periods using different number of time-steps

Case2-1: BDF2 Solution



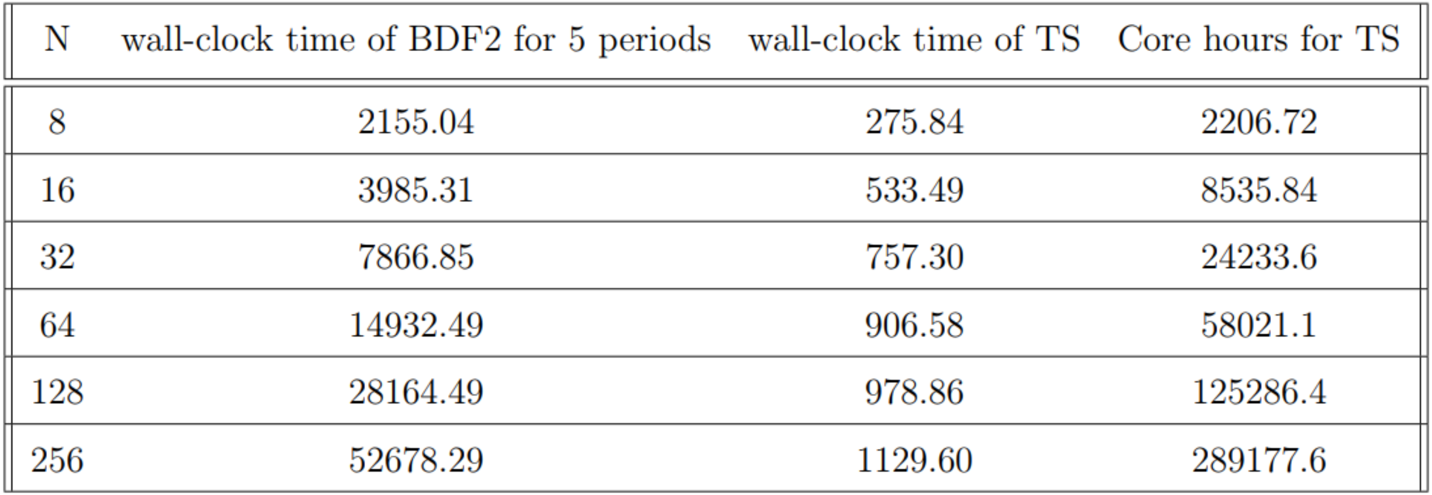
➢ Computed lift coefficient time histories using the BDF2 scheme over last of 5 periods for different numbers of time steps (Left) and detail of time histories near peak CL value (Right)

Case2-1: Comparison of BDF2 and TS Error



- ➢ Temporal error of TS and BDF2 solutions as a function of the number of time-instances or time steps

Case2-1: Comparison of Run Time of BDF2 and TS



N	wall-clock time of BDF2 for 5 periods	wall-clock time of TS	Core hours for TS
8	2155.04	275.84	2206.72
16	3985.31	533.49	8535.84
32	7866.85	757.30	24233.6
64	14932.49	906.58	58021.1
128	28164.49	978.86	125286.4
256	52678.29	1129.60	289177.6

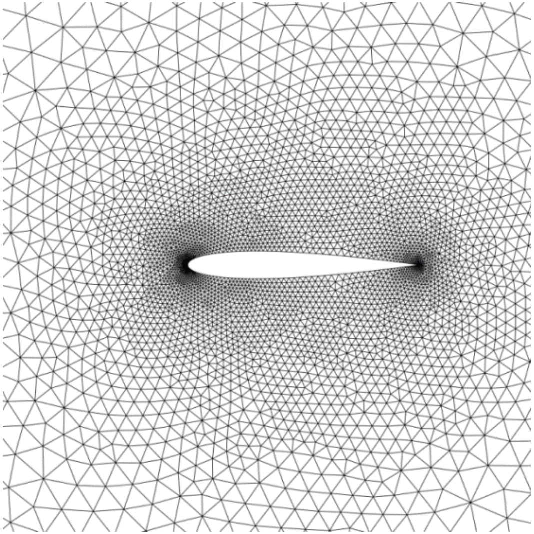
- ➢ Run time for solving the Gaussian bump problem using BDF2 solver for 5 periods, and TS solver for 8 to 256 time-steps per period or time-instances

Test Cases

- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - **Case 1-1 : Testing the performance of FFT based AF for Case 1**
  - **Case 1-2 : Testing the performance of FFT based GMRES/AF for Case 1**
- **Case 2: Purely Periodic Pitching Airfoil with Gaussian Bump Motion**
  - **Case 2-1 : Comparison of the performance of FFT based GMRES/AF and BDF2 for Case 2**
- **Case 3: Quasi-Periodic Pitching Airfoil with Single Frequency Prescribed Motion**

Case 3: Quasi-Periodic Pitching Airfoil with Single Frequency Prescribed Motion

- Naca-0012 Airfoil
- 15573 triangular elements
- Free stream Mach = 0.755
- Prescribed pithing motion:



$$\alpha_t = \alpha_0 + \bar{\alpha}(t) + \alpha_A \, Sin(\omega t)$$

$$\bar{\alpha}(t) = \begin{cases} 0 & t < t_1 \\ \alpha_m \frac{1}{2}(1 - \cos(\omega_m(t - t_1))) & t \geq t_1 \end{cases}$$

$$\alpha_A = 2.51^\circ$$
$$\alpha_0 = 0.016^\circ$$

$\omega_1 = 0.1628$ Periodic Content

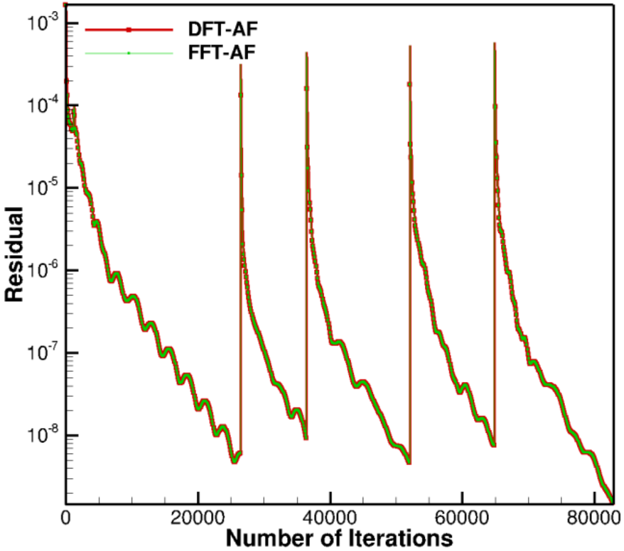
$\omega_m = 0.1\omega_1$ Slow Transient

→ Quasi- Periodic Problem

Test Cases

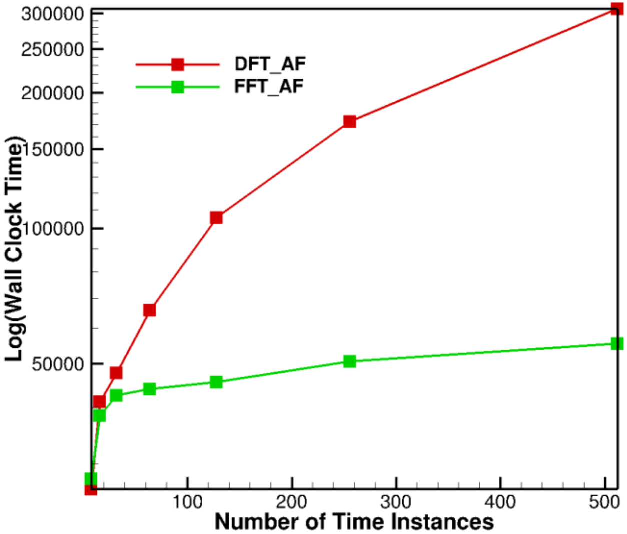
- Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion
  - Case 1-1 : Testing the performance of FFT based AF
  - Case 1-2 : Testing the performance of FFT based GMRES/AF
- Case 2: Purely Periodic Pitching Airfoil with Gaussian Bump Motion
  - Case 2-1 : Comparison of the performance of FFT based GMRES/AF and BDF2
- Case 3: Quasi-Periodic Pitching Airfoil with Single Frequency Prescribed Motion
  - Case 3-1 : Testing the performance of FFT based AF

Case 3-1: FFT-AF Residual Validation



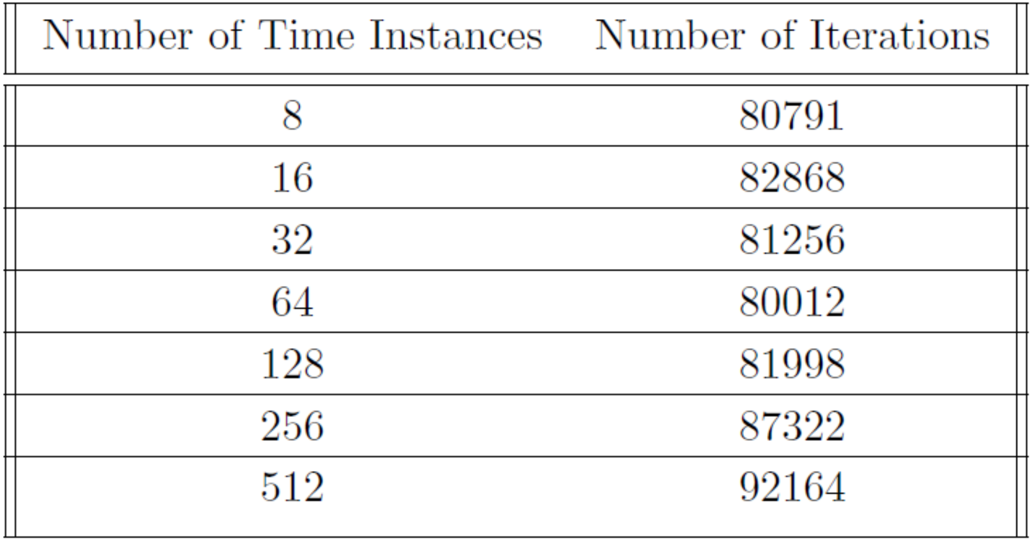
- ➢ Residual versus iterations for DFT and FFT based AF solver, using 16 time-instances per period for 5 periods

Case 3-1: Performance of DFT- and FFT- based AF



- Comparison of wall clock time versus number of time instances for DFT and FFT based AF solution of the problem.
- Even Number of Samples up to 512 time-instances/processors

Case 3-1: Comparison of Convergence Rate of AF



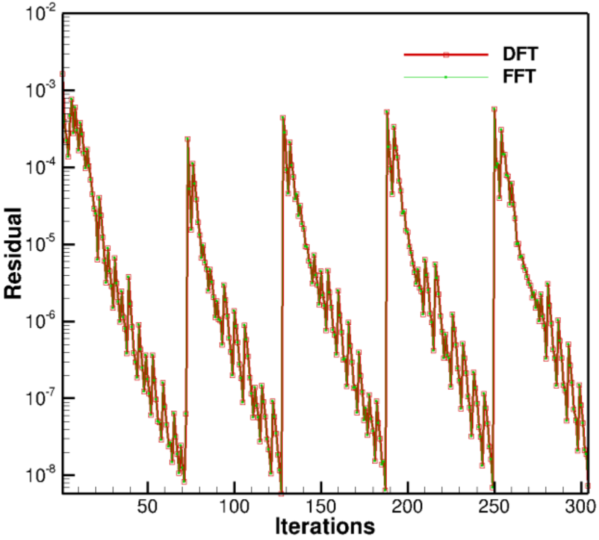
Number of Time Instances	Number of Iterations
8	80791
16	82868
32	81256
64	80012
128	81998
256	87322
512	92164

- Comparison of convergence rate of the quasi-periodic AF scheme over five periods for different number of time instances per period

Test Cases

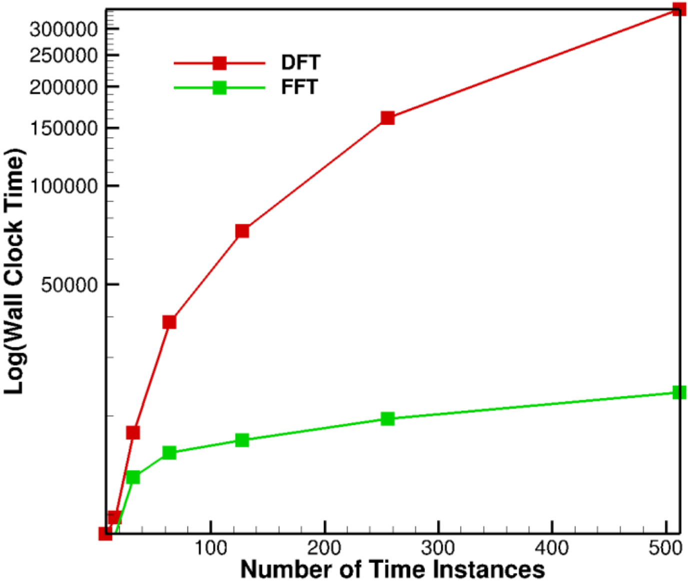
- **Case 1: Purely Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - Case 1-1 : Testing the performance of FFT based AF
  - Case 1-2 : Testing the performance of FFT based GMRES/AF
- **Case 2: Purely Periodic Pitching Airfoil with Gaussian Bump Motion**
  - Case 2-1 : Comparison of the performance of FFT based GMRES/AF and BDF2
- **Case 3: Quasi-Periodic Pitching Airfoil with Single Frequency Prescribed Motion**
  - Case 3-1 : Testing the performance of FFT based AF
  - Case 3-2 : Testing the performance of FFT based GMRES/AF

Case 3-2: FFT-GMRES/AF Residual Validation



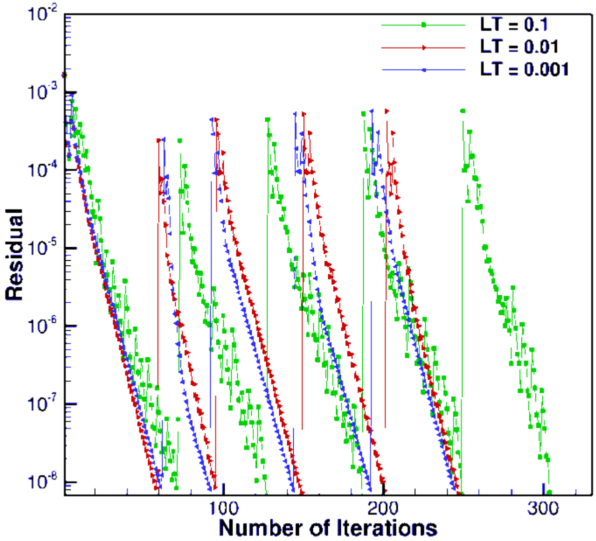
➢ Residual versus iterations for DFT and FFT based GMRES/AF solvers using 16 time-instances per period for 5 periods.

Case 3-2: Performance of DFT- and FFT-based GMRES/AF



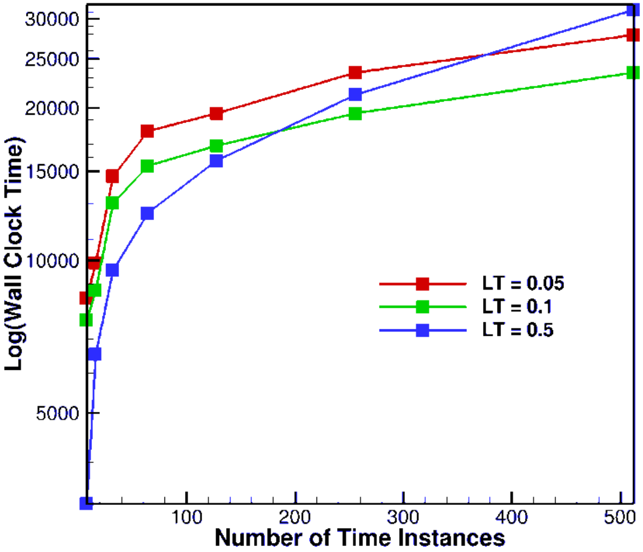
➢ Wall-clock time for DFT- and FFT- based GMRES/AF solvers for up to 512 number of time instances.

Case 3-2: Study of Linear Tolerance



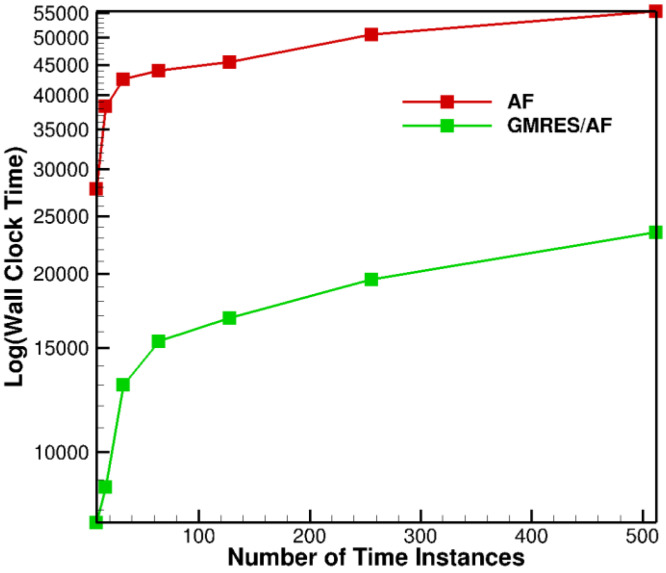
- ➢ Non-linear residual versus number of iterations for linear tolerance of 0.1, 0.01, 0.001
- ➢ Tighter linear tolerance results in greater wall lock time

Case 3-2: Study of Linear Tolerance



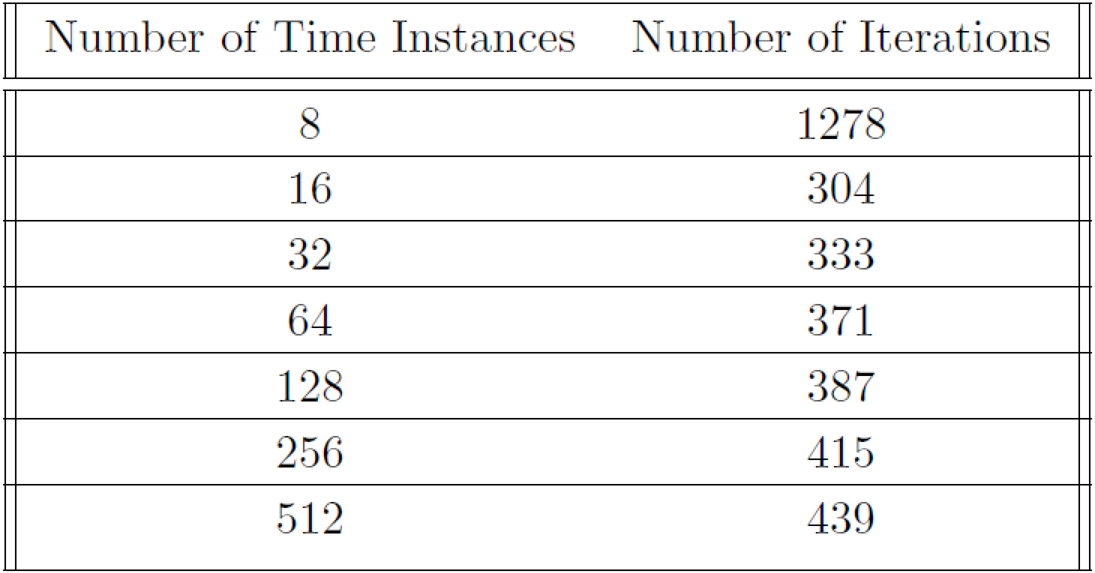
➢ Wall-clock time versus number of time instances for different linear tolerances

Case 3-2: Performance of FFT- based AF and GMRES/AF



- Wall-clock time versus number of time instances for FFT- based GMRES/AF and FFT- based AF solver for up to 512 number of time instances

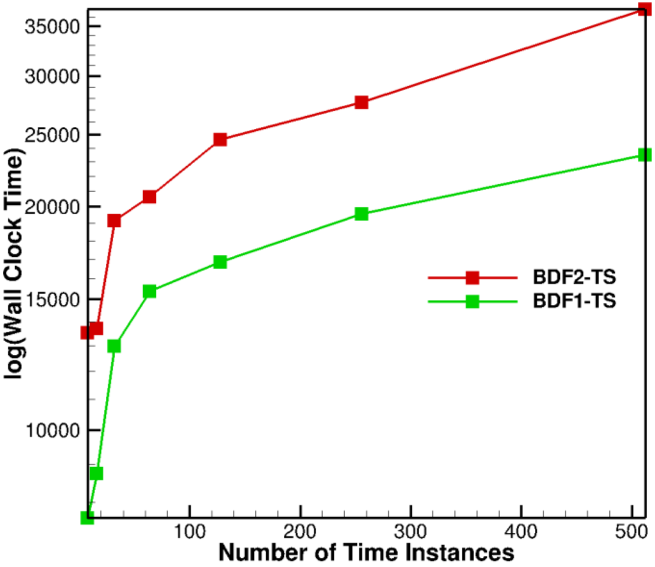
Case 3-2: Comparison of Convergence Rate of GMRES/AF



Number of Time Instances	Number of Iterations
8	1278
16	304
32	333
64	371
128	387
256	415
512	439

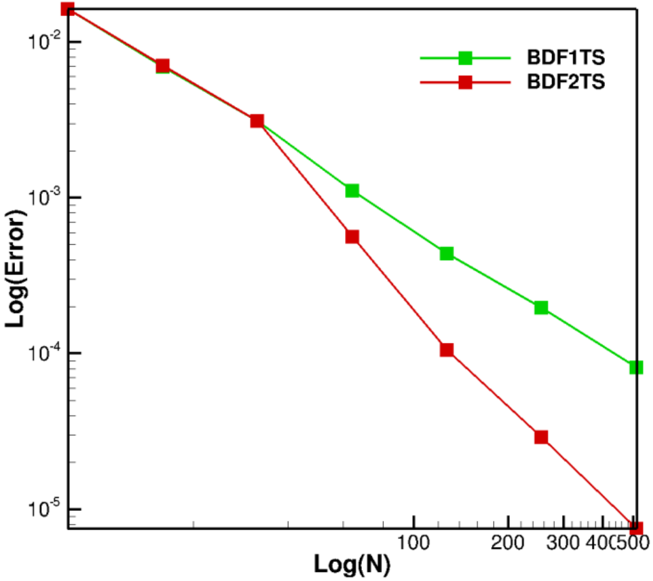
- Comparison of convergence rate of the quasi-periodic GMRES/AF scheme over five periods for different number of time instances per period

Case 3-2: Performance of BDF1TS and BDF2TS



- Wall-clock time for FFT- based BDF1TS and BDF2TS solvers for up to 512 number of time instances.

Case 3-2: Accuracy of BDF1TS and BDF2TS



- ➢ Lift coefficient error versus log of time instances using BDF1TS and BDF2TS solvers

- Introduction
- Governing Equations
- Challenges
- Novelty
- Results
- **Summary and Conclusions**
- Future Work

Summary and Conclusions (1/4)

- A new parallel time-spectral algorithm is developed for periodic and quasi-periodic problems
- The new implementation is based on the FFT and scales as $NlogN$ and results in significant savings compared to previous implementations in terms of wall-clock time which was based on the DFT and scales as $O(N^2)$
- An FFT-based AF algorithm is developed and used as the direct solver to solve purely periodic problems
- FFT-based AF is significantly more efficient than the DFT-based AF solver in terms of wall-clock time
  - $NlogN$ computation and communication versus $O(N^2)$

Summary and Conclusions (2/4)

- The FFT-based AF scheme reformulated as a preconditioner for GMRES
  - The GMRES/AF scheme is shown to be consistently and significantly more efficient than the AF scheme alone
  - 2 to 3 times speed up in GMRS/AF compared to AF
- The overall FFT-based GMRES/AF solver performance can be more than an order of magnitude more efficient than the previous DFT-based implementations
  - $NlogN$ computation and communication versus $O(N^2)$
- Both the AF scheme used directly as a solver and the GMRES/AF linear solver are relatively insensitive to the number of time-instances and to the reduced frequency of the problem

Summary and Conclusions (3/4)

- The performance of the FFT-based TS solvers is studied in problems with prescribed motion including a wide range of frequency spectrum

- The performance of the FFT-based time-spectral solvers is compared to the BDF2

- By improvements made in time-spectral solvers done in this work, these solvers can outperform the time-accurate solvers in problems with high frequency content as well as problems with few harmonic contents

Summary and Conclusions (4/4)

- The application of FFT-based time spectral method is extended to quasi-periodic problems, using BDFTS formulations
- FFT-based BDFTS formulations are dramatically more efficient than the DFT-based BDFTS approach
  - $NlogN$ computation and communication versus $O(N^2)$ in periodic component of the solver
- The BDFTS equations correspond to rank-1 update of the fully-periodic time-spectral equations and can be solved effectively by leveraging the FFT-based periodic AF solver using the Sherman-Morrison formulation
- Using parallel FFT- based AF as a preconditioner for GMRES results in 2 to 3 times more efficiency compared to AF alone as the solver.
- Although BDF2TS requires longer wall-clock time for convergence, it provides better accuracy for cases with larger number of time instance, compared to BDF1TS scheme

- Introduction
- Governing Equations
- Challenges
- Novelty
- Results
- Summary and Conclusions
- Future Work

Future Work

- **Three dimensional parallel in space and time problems**
  - The performance of the new approach was tested for 2D problems
  - The goal was to study the temporal efficiency of the solvers in all the test cases the spatial component was solved in serial
  - The 2D test cases with solution of the spatial part on one core are; representative of the size of a spatial portion in a parallel 3D run
  - By combining the temporal parallelism afforded by this approach with spatial parallelism, the solution of periodic and quasi-periodic problems of moderate spatial size can be effectively scaled to hundreds of thousands of cores
- **Extension to other flow regimes**
  - The solution of the Euler equations are presented in all the test cases
  - For turbulent flow problems, the spatial part becomes harder to solve, and requires more sophisticated spatial solvers
  - Other elaborate spatial solvers such as multigrid, … can make AF a stronger preconditioner for GMRES

Future Work

- **Studying the viability of BDFTS**
  - Unlike the TS method, BDFTS methods need to resolve the transient part. Majority of the CPU resources could be spent resolving the transient part of the solution
  - In most cases the problem needs the same number of periods as required in time-accurate methods to resolve the slow transient content
  - In the BDFTS method each period must be solved faster than time-accurate methods, in order to outperform them
  - Comparison of the performance of BDF1TS and BDF2TS in 3D problems

Acknowledgements

- **Committee:**

Dr. Dimitri Mavriplis (advisor)

Dr. Jayanarayanan Sitaraman

Dr. Joshua Leffell

Dr. Jonathan W. Naughton

Dr. Michael Stoellinger

Dr. Fred Furtado

- **Family and Friends**

- **Lab mates**

Acknowledgements

This research was sponsored by

- NASA's Transformational Tools and Technologies (TTT) Project of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate
- the Alfred Gessow Rotorcraft Center of Excellence through a subcontract with the University of Maryland

Computer resources were

- NCAR-Wyoming Supercomputer Center(NWSC) and by the University of Wyoming Advanced Research Computing Center(ARCC).

Thank You