TFLO ALGORITHM DEVELOPMENT

Motivation

- TFLO has access to a variety of AEC computer facilities. Different processors can be used for different applications within the overall execution platform, but enough to test at the most important steps of the design.

- For TFLO to be applied as an effective tool for the design engineers, the execution time should be quantified in hours.

- TFLO currently has a fully implicit multigrid Kalman filter utilizing a second system of loops. The order of the algorithm is defined on the basis of the overall execution time in the initial design and the overall execution time of the overall algorithm.

- One of the advantages of TFLO is the ability to efficiently solve the time accuracy problem. This generally provides efficiency improvements in the time accuracy resolution of the solution (the real-time accuracy).

- Implementing the convergence of the multigrid solver, providing a potential reduction in the computational effort after step time (the multigrid solver).

- To take advantage of the broad period in the solution, and to use a linear frequency-domain method to provide an efficient initial solution for a fully time-accurate calculation.

Non-Linear Frequency Domain Methods

- Time accuracy solutions capture all arbitrary time history in the solution field. In time-accurate solutions, the only concern is to capture the time history of the solution.

- Convergent solution strategies for computational efficiency are described in the following section. In conservative calculations, the models are strongly affected by the inaccuracy of the solution. The time evolution of these waves back and forth through the machine effectively decreases the decay rate of the expansion terms.

- In engineering applications it appears to be possible to achieve sufficient accuracy in a single-time step with a small number of time steps, some-times only a single transient mode.

- The linear frequency-domain (LFD) method can be used to establish a converged solution at order time. The cost of this solution is in order magnitude less than the current state of the art in accuracy calculations. Traditional accuracy requirements are decoupled from the physics of the problem.

- Advantage: The cost of the LFD solver is a function of the number of temporal nodes multiplied by the cost of a single steady state computation. The computational cost is decoupled from the physics of the problem.

- Non-Linear Frequency Domain Methods: In the case of reduced-order modeling, because we are only interested in high temporal frequencies, the optimization efficiency is not provided by the time accuracy system.

Lower Upper Symmetric Gauss-Seidel

- The lower-upper symmetric Gauss-Seidel (LU-SGS) method can be used in the following equation: where

  \[
  \frac{1}{2} \left[ \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right] \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- This expression can be expressed as the minmod of a lower, diagonal, and upper limit. This solution is used to establish the first temporal node in the time evolution of the LFD method. The expression for the SGS Newton iteration is

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- The symmetric Gauss-Seidel scheme can be applied to the above equation. After multiple linear convergence, the above equation can be expressed as the following equation:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- This expression can be expressed as the minmod of a lower, diagonal, and upper limit. This solution is used to establish the first temporal node in the time evolution of the LFD method. The expression for the SGS Newton iteration is

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

LU-SGS Convergence

- The convergence rate of the global characteristics in the solution field are provided below. The data in the solution vary from the 2nd and 3rd zones, and the linear solver converges very fast after 3-5 cycles under the logarithmic scale.

- The CPU time per multiple cycle is about 50% less than that of the above mentioned method. The overall convergence solutions can be obtained with almost an order of magnitude less computation time. The data in the solution vary from the 2nd and 3rd zones, and the linear solver converges very fast after 3-5 cycles under the logarithmic scale.

- The non-linear LU-SGS scheme can also be applied to the steady flow calculations. The factorization and linearization errors can be eliminated from this algorithm. Furthermore, the computational cost of this scheme is equal to that of a single Runge-Kutta implicit time-marching scheme.

- Applying LU decomposition to the fully implicit backward difference formulation results in the following expression:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- To calculate the linear and nonlinear errors, the non-linear errors are eliminated at the introduction of additional terms. After some mathematical manipulation, the expression is simplified to:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- The LU decomposition value is about 50% less than that of the above mentioned method. The overall convergence solutions can be obtained with almost an order of magnitude less computation time.

- The LU-SGS scheme can also be applied to the steady flow calculations. The factorization and linearization errors can be eliminated from this algorithm. Furthermore, the computational cost of this scheme is equal to that of a single Runge-Kutta implicit time-marching scheme.

- Applying LU decomposition to the fully implicit backward difference formulation results in the following expression:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- To calculate the linear and nonlinear errors, the non-linear errors are eliminated at the introduction of additional terms. After some mathematical manipulation, the expression is simplified to:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

Non-Linear Equations Recast in the Frequency Domain

- The Navier-Stokes equations can be expressed in a simplified form:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- Using finite volume techniques the above equation can be rewritten as the sum of linear and non-linear terms. Applicating the volume integration to the product of the left and right side of the equation, we can definite the left side of the equation.

- Assuming that the solution is a harmonic component of the same period. We can rewrite the equation above into the frequency domain. Adding a phase to the time derivative term, we can express the following equation in the frequency domain:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- The non-linear term is expressed in an approximate form:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- Solution Contours: Contours of pressure are provided below at different instances in time corresponding to different angles of attack.

- Solution Contours: Contours of pressure are provided below at different instances in time corresponding to different angles of attack.

LU-SGS Dual Stepping Scheme

- The non-linear LU-SGS scheme can also be applied to the steady flow calculations. The factorization and linearization errors can be eliminated from this algorithm. Furthermore, the computational cost of this scheme is equal to that of a single Runge-Kutta implicit time-marching scheme.

- Applying LU decomposition to the fully implicit backward difference formulation results in the following expression:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- To calculate the linear and nonlinear errors, the non-linear errors are eliminated at the introduction of additional terms. After some mathematical manipulation, the expression is simplified to:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- The LU decomposition value is about 50% less than that of the above mentioned method. The overall convergence solutions can be obtained with almost an order of magnitude less computation time.

- The data in the solution vary from the 2nd and 3rd zones, and the linear solver converges very fast after 3-5 cycles under the logarithmic scale.

- Tests have been carried out on one-dimensional channel flow with area change. With the increase number of cycles, the non-linear convolution in the linear matrix, although the overall convergence rate remains very similar.

Non-Linear-Frequency Domain (NLFD) - Pitching Airfoil

- The Navier-Stokes equations are solved for cylinder flow and a Rayleigh number of 186. Numerical results for global accuracy are compared below to experimental data as a function of temporal resolution to compute the solution of the Navier-Stokes equations in the frequency domain.

- The results of this work indicate that the following expression can be used to solve the steady-state cylinder:

  \[
  \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \left( \frac{1}{2} \left( -A^{k+1} - A^{k+1} \right) \right) \]

- The evaluation of the modulated residual requires no more effort than regular time-domain stepping. The following solutions are used to establish the factorization error. Since each cycle only provides a single extra order accurate, only a small number of iterations should be required.

- The figures above illustrate specific engine contours computed using the lowest (1 mode) and highest resolution (7 modes) respectively.

Conclusions

- The Stanford AAG team has successfully developed three new numerical schemes, in a methodological approach to improve the efficiency of steady-state simulations. All NLFD schemes were developed to improve the convergence rate of linear iterations within the TFLO solver. A hybrid ADI solver has been developed to improve the efficiency of steady-state simulations. The NLFD schemes have been developed to improve the efficiency of steady-state simulations. The hybrid NLFD solver has been developed to improve the efficiency of steady-state simulations. The hybrid NLFD solver has been developed to improve the efficiency of steady-state simulations. The hybrid NLFD solver has been developed to improve the efficiency of steady-state simulations.