N93-1...

1992

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA

ACCELERATION OF FDNS FLOW SIMULATIONS USING INITIAL FLOWFIELDS GENERATED WITH A PARABOLIZED NAVIER-STOKES METHOD

Prepared By:	Peter W. TenPas, Ph.D.
Academic Rank:	Assistant Professor
Institution and Department:	The University of Kansas Department of Mechanical Engineering
NASA/MSFC:	
Office:	Structures & Dynamics Laboratory
Division:	Aerophysics
Branch:	Computational Fluid Dynamics
MSFC Colleagues:	Ten-See Wang, Ph.D. Paul K. McConnaughey, Ph.D.

INTRODUCTION

The capability of computational fluid dynamics (CFD) analysis to predict complex flowfields has been greatly advanced by the widespread use of flow simulation programs based on the Navier-Stokes (NS) equations. The flow physics are theoretically well represented, and with proper care the numerical solution should not introduce appreciable uncertainties. However, the computational cost of a typical simulation in terms of both memory and execution time are large by current standards. Therefore, the efficiency of the numerical algorithm in solving the set of model equations is one factor determining the usefulness of CFD tools. The objective of the present study is to reduce the execution time of the FDNS flow simulation code by using the parabolized Navier-Stokes (PNS) equations to provide a "good" starting condition. The technique is not universal, however the PNS model can be applied to convection dominated flows with moderate deflection in geometries that are free of large obstructions.

The FDNS computer code developed by Chen *et al.* (1990) is a general-purpose CFD program for solving the Navier-Stokes equations in finite-difference form. The method is widely used at the Marshall Space Flight Center for modeling flows in propulsion systems. CFD analysis of a given problem involves a sequence of steps, including: generation of a mesh to represent the physical geometry, specification of the boundary conditions on the flowfield, estimation of an initial flowfield from which to start the simulation, calculation of a converged solution to the model equations, analysis of the solution by graphical as well as numerical techniques, revision of the numerical model if necessary, and finally presentation of the results. The calculation stage of the analysis requires the greatest computational resources. Additionally, considerable human effort is invested in all other phases of the analysis. Effective computer tools can relieve some of the burden in performing these pre- and post-processing tasks. A "front end" code to aid the analyst in estimating an initial flowfield may reduce the engineering time needed to set up an FDNS case.

NUMERICAL METHOD

The FDNS code uses an iterative method to solve a large set of non-linear equations. The total execution time is the product of the number of nodes, the calculation time per node, and the number of iterations. The number of nodes is primarily determined by the complexity of the given problem. The calculation time per node depends upon the computer hardware and the details of the FDNS coding. Finally, the number of iterations is a function of the nature of the errors present in the starting field and the error reduction properties of the FDNS solver. Since a large effort is needed to validate modifications to the FDNS code, it is not practical to make internal changes without good cause. But, the starting field is an arbitrary program input prepared by the user, and the converged solution is independent of the starting condition. Thus, by providing a starting condition with low initial error, the execution time may be reduced without altering the end results.

A procedure for generation of the starting flowfield is constrained by several factors, primarily quick execution time, generality and ease of use. Since parabolic equations may be solved at a much lower computational cost than elliptic equations, the PNS equations have been chosen to generate the initial flowfield. Furthermore, the PNS equations apply to a wide range of flow conditions and geometries without introducing many physical approximations. The decomposition of the pressure field is the most significant approximation for the cases studied. To facilitate use of the code, the standard FDNS input is utilized.

The PNS equations are a sub-set of the full Navier-Stokes (NS) equations derived by the following steps, which isolate the solution from the influence of downstream flow conditions.

1) Neglect streamwise diffusion terms.
2) Neglect streamwise convection in regions of reversed flow.
3) For subsonic flow, assume a mean streamwise pressure gradient applies at all points across any given flow cross-section.
4) For supersonic flow, solve for the pressure field in supersonic regions and impose this pressure on the subsonic boundary layer.
5) Apply boundary conditions for a well-posed parabolic problem.

The numerical solution is obtained by approximating the PNS equations by finite-differences in transformed coordinates in a manner similar to the FDNS program. First-order approximations for streamwise derivatives are used in the present code since high accuracy is not needed. The set is solved by a single space-marching sweep through the grid. The kernel of the solver is the coupled space-marching method of TenPas and Pletcher (1991), which advances the solution one grid step in the primary flow direction based on the known values upstream. A distinguishing feature of the algorithm is the coupled solution of the momentum and continuity equations, with Newton linearization of the non-linear convective terms. This formulation permits the velocity components and pressure to be solved for directly. With the velocity field known at a given marching step, parabolized transport equations are independently solved for the enthalpy and turbulence properties as necessary. Local iteration is used to converge the coefficients of the linearization and to update large changes in properties. The space-marching calculation proceeds step-by-step until the exit boundary is reached. In the course of this study it was found useful to perform a final smoothing procedure on the pressure field. A pressure Poisson equation formulated as a linear combination of the momentum equations is solved by line iterative method to distribute sharp pressure changes among surrounding nodes.

To evaluate the potential benefit of this procedure a flow initialization computer code has been developed for incompressible two-dimensional flow. The FDNS source code is used as the basis for the PNS code. The FDNS input and initialization modules are used to input data defining the geometry and flow conditions. One additional file is read to define the PNS solution parameters. The solver portion of FDNS is replaced by the space-marching solver, and the modules that evaluate the finite-difference equations have been modified to permit access by the new solver. The non-linear terms are reworked to implement the Newton linearization. The output routines from FDNS are kept intact. The PNS code is executed starting from FDNS input files, and produces a flowfield output file containing the PNS solution. Depending upon the convergence limits for the Newton and local property iteration loops, the entire PNS flow initialization solution is executed in the time required for between one and ten FDNS iterations. The FDNS program is then executed using the PNS output file as the starting condition.

RESULTS

The PNS flow initialization method was tested on two flow geometries for incompressible flow. Both laminar and turbulent flows were simulated. Tables 1 and 2 present comparisons of the iterations required to converge the FDNS solution based on a simple starting condition versus the PNS estimate. The convergence rate of the FDNS program is dependent upon several factors. The input values used for various constants are: REC=0.5, BETAP=1.0, PSMO=0.005, and ICOUP=2. The convergence rate is also sensitive to the time step. The time step shown for the baseline FDNS calculations was determined by trial and error to minimize the number of iterations. The same time step was then used for the PNS initialized cases. The study time was not sufficient to evaluate the time step sensitivity using the PNS initialization. The convergence tolerance was selected to represent a compromise between accuracy and total execution time. The fine grid cases in particular are difficult to converge to a high level of precision, and the laminar flows may also be unsteady to a small degree. Therefore, the iteration count comparison is rather arbitrary as the value depends upon the convergence tolerance. However, since the iteration counts are roughly proportional for various levels of convergence, the percentage of the baseline execution time is not strongly related to the magnitude of the convergence tolerance.

The first geometry is a 180 degree turn-around-duct (TAD) similar to that studied by Monson *et al.* (1990). A fully developed laminar or turbulent profile is specified upstream. Uniform grids are used for the laminar flow runs. For the turbulent cases the mesh is compressed near the channel walls to grid sizes of 0.01 and 0.002, to yield y^+ values of 50 and 75 for Reynolds numbers of 10^5 and 10^6, respectively. In each case the 180 degree turn is subdivided into 40 marching steps with equally spaced grids in the straight sections upstream and downstream. To test the range of application of the PNS approximation the tightness of the turn was varied from a gentle bend with outer to inner radius ratio of 5/4 to a very tight ratio of 3/1. The PNS approximation for the gentle turn is relatively good and significant reductions in the FDNS execution time are obtained for both the laminar and turbulent case. For a tighter turn the PNS approximation does not model the actual flow well, and the change in FDNS execution time diminishes.

The second case is flow over a backward-facing step as simulated by Chen (1988). A parabolic inlet profile is specified for laminar flow, and a plug flow profile with turbulent kinetic energy of 0.005 and dissipation rate of 0.0002 is specified for the turbulent case. In both cases the flow inlet plane is at the sudden expansion and the grid spacing is uniform across the channel The laminar flow case is a 2/1 expansion at a Reynolds number of 400, which is the upper limit of steady laminar flow for this geometry. The numerical grid extends 20 channel heights downstream to allow for the near approach to separation on the straight wall and subsequent flow redevelopment. Comparisons are made for a uniformly spaced coarse grid and a refined grid with grid compression near the expansion. Large reductions are achieved for the coarse grid case, with the FDNS convergence deteriorating for the refined mesh. A 3/2 expansion is used for the turbulent case, with a variable grid extending 10 channel heights downstream. As with the TAD case, the PNS initialization is less effective for the turbulent conditions. Even though the PNS flowfield looks reasonable for this case, the fine grids and the coupling between the velocity field and turbulence properties slows convergence. Under these conditions the FDNS code is slow to eliminate even relatively small errors in the solution.

Table 1. Comparison of FDNS iteration counts for TAD cases.

Re	R_2/R_1	Grid	Δt	Poiseuille V, Uniform P $U\Delta t/\Delta x$	ε	N	PNS V & P fields $U\Delta t/\Delta x$	ε	N	CPU %
500	5/4	101x21	0.72	2	10^{-4}	943	2	10^{-4}	250	27
500	2/1	101x21	0.08	3/4	"	513	3/4	"	390	76
500	3/1	201x21	0.08	1	"	767	1	"	715	93
10^5	5/4	101x21	0.18	1/2	10^{-3}	1000	1/2	10^{-3}	487	49
10^5	3/1	201x21	0.02	1/4	"	1019	1/4	"	988	97
10^6	3/1	201x41	0.01	1/8	$8x10^{-3}$	3000	1/8	$8x10^{-3}$	3000	100

Table 2. Comparison of FDNS iteration counts for backward-facing step cases.

Re	H_2/H_1	Grid	Δt	Static V, Uniform P $U\Delta t/\Delta x$	ε	N	PNS V & P fields $U\Delta t/\Delta x$	ε	N	CPU %
400	2/1	81x41	0.25	1	10^{-4}	1028	1	10^{-4}	197	19
400	2/1	121x81	0.25	3/2	10^{-3}	1000	3/2	10^{-3}	500	50
10^5	3/2	61x49	0.32	2	10^{-4}	2370	2	10^{-4}	1861	79

CONCLUSIONS

While the results obtained for laminar flow are encouraging, much smaller reductions were achieved for the turbulent flow cases. For fine grids or complicated flows the FDNS convergence history is most strongly determined by the slow decay of low frequency errors. In contrast, the flow initialization procedure is most likely to reduce high frequency errors. And, the influence of the starting condition diminishes as the iteration count becomes large. To achieve significant reduction in the iteration count for complex problems the specific properties of the finite-difference equations (e.g. truncation error properties, amount of damping) or the algebraic solver (e.g. matrix structure, limits on inner iterations) must be examined. Finally, the PNS solver may have some practical use, by saving engineering time in setting up an FDNS case.

REFERENCES

1. Chen, Y.S., "Viscous Flow Computations Using a Second-Order Upwind Differencing Scheme," AIAA Paper 88-0417, Jan. 1988.

2. Chen, Y.-S., Farmer, R.C. and Freeman, J.A., FDNS User's Manual (Version 1.0), SECA-TR-90-10, SECA, Inc., Huntsville, Alabama, Contract NAS8-37408, June 1990.

3. Monson, D.J., Seegmiller, H.L., McConnaughey, P.K. and Chen, Y.-S., "Comparison of Experiment with Calculations Using Curvature-Corrected Zero and Two-Equation Turbulence Models for a Two-Dimensional U-Duct," AIAA Paper 90-1484, June 1990.

4. TenPas, P.W. and Pletcher, R.H., "Solution of the Navier-Stokes Equations for Subsonic Flows using a Coupled Space-Marching Method," *AIAA Journal*, Vol. 29, no. 2, Feb. 1991, pp. 219-226.