General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
DEVELOPMENT OF AN EFFICIENT COMPUTER CODE TO SOLVE
THE TIME-DEPENDENT NAVIER-STOKES EQUATIONS

FINAL REPORT

REPORT NO. SR-26

DECEMBER 1975

(Thermo Mechanical Systems Co., Canoga Park) 18 p HC $3.50

CSCL 20D 63/34 25203

THERMO MECHANICAL SYSTEMS COMPANY
7252 Remmet Ave
Canoga Park, California, 91303

APR 1976
RECEIVED
NASA STI FACILITY
INPUT BRANCH
DEVELOPMENT OF AN EFFICIENT COMPUTER CODE TO SOLVE
THE TIME-DEPENDENT NAVIER-STOKES EQUATIONS

FINAL REPORT

REPORT NO. SR-26
DECEMBER 1975

by
James L. Harp, Jr.
Thomas P. Oatway

Prepared For
National Aeronautics and Space Administration
Ames Research Center
Moffett Field, California 94035

Contract No. NAS2-8471

Thermo Mechanical Systems Company
7252 Remmet Avenue
Canoga Park, California 91303
ACKNOWLEDGEMENT

The authors wish to acknowledge and express appreciation for the major effort contributed by Mr. John J. Germroth on the program reported herein.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>2.0 INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>3.0 SYMBOLS</td>
<td>4</td>
</tr>
<tr>
<td>4.0 NON-UNIFORM TIMESTEP FOR IMPROVEMENT IN COMPUTATIONAL EFFICIENCY OF NAVIER STOKES SOLUTIONS</td>
<td>5</td>
</tr>
<tr>
<td>4.1 Treatment of Timestep Intermesh Boundaries</td>
<td>5</td>
</tr>
<tr>
<td>4.2 Development of Non-Uniform Timestep Logic</td>
<td>7</td>
</tr>
<tr>
<td>4.3 Development of Conservation Logic</td>
<td>7</td>
</tr>
<tr>
<td>4.4 Checkout and Debugging of Non-Uniform Timestep Coding</td>
<td>9</td>
</tr>
<tr>
<td>5.0 CONCLUDING REMARKS</td>
<td>10</td>
</tr>
<tr>
<td>6.0 REFERENCES</td>
<td>11</td>
</tr>
<tr>
<td>APPENDIX: Subroutine ZONEL</td>
<td>12</td>
</tr>
</tbody>
</table>
1. **SUMMARY**

A research effort was conducted with the goal of reducing computer time of a Navier Stokes Computer Code for prediction of viscous flow fields about lifting bodies. A two-dimensional, time-dependent, laminar, transonic computer code (STOKES) was modified to incorporate a non-uniform timestep procedure. The non-uniform timestep requires updating of a zone only as often as required by its own stability criteria or that of its immediate neighbors. In the uniform timestep scheme each zone is updated as often as required by the least stable zone of the finite difference mesh. Because of less frequent update of program variables it was expected that the non-uniform timestep would result in a reduction of execution time by a factor of five to ten. Available funding was exhausted prior to successful demonstration of the benefits to be derived from the non-uniform timestep method.
2.0 INTRODUCTION

Two elements lie at the core of any satisfactory numerical procedure for computing laminar and/or turbulent transonic airfoil flows, namely, (a) a general method for integrating the equations of continuum mechanics, to which equations governing the evolution of turbulence variables can be added, and (b) a mathematical model that describes the fluctuating component of an arbitrary turbulent field. To the extent possible, both the numerical method and the turbulence model should offer proven reliability; the choices proposed are believed to best meet such a criterion, namely:

1. A flow field computer code, which integrates the Navier-Stokes equations in two spatial dimensions, and
2. The Saffman Model of turbulent flow.

For each of the many laminar flows (References 1-7) to which it has been applied, the presently operational flow field computer code has been shown to yield solutions accurate enough for any engineering purpose. Furthermore, in the two-dimensional test calculations completed to date, the less proven Saffman turbulence (Ref. 8-9) model, in combination with the flow field computer code, has provided an accurate solution to the turbulent flat plate problem, (Ref.10) a turbulent boundary layer-shock wave interaction (Ref.10) and the separation phenomena that occur on a compression corner in supersonic flow. (Ref.10)

A version of the Afton plane symmetric code (A2P) was developed for the NASA Ames Research Center to solve a laminar transonic airflow problem. The flow field was computed about the lifting NACA 64A410 airfoil at Mach .72, Reynolds number (based on chord length) of $1.75 \times 10^6$ and at an angle of attack of $4^\circ$ (ref 11). Starting from the impulsive conditions the calculation was carried out to the time it takes a freestream particle to travel one chord length. The calculation required slightly less than five hours of CDC 7600 computer time. In the course of the calculation the flow separated on the leeward trailing edge, a leeward compression wave formed, and approximately 50% of the measured lift was computed.
However, it was estimated that an additional ten hours on the CDC 7600 computer would be required to complete the problem.

Computing times of the order of fifteen hours on the CDC 7600 to complete one transonic airfoil calculation make transonic flow computations about lifting bodies impractical for design purposes. A speed-up factor of the order of 10 would be necessary for numerical computation of transonic flow to be in the practical range. Coupled with a better estimate of the initial conditions than impulsive initial conditions, and with a speed-up factor of ten, transonic lifting airfoil problems can be solved in less than an hour on the CDC 7600 computer.

The main objective of the work conducted in this program was to develop a computer code which would be approximately an order of magnitude faster than the present computer program. This was to be accomplished by use of a "Non-Uniform Timestep" in the computation procedure to improve computation efficiency.
3.0 **SYMBOLS**

E Specific internal energy  
j Index specifying streamline-like direction  
k Index specifying potential-like direction  
M Mass  
n Time index for finite difference equation  
p Pressure  
$\Delta t$ Timestep increment
4.0 NON-UNIFORM TIMESTEP FOR IMPROVEMENT IN COMPUTATIONAL EFFICIENCY OF NAVIER STOKES SOLUTIONS

The STOKES computer code was developed for the NASA Ames Research Center from an AFTON plane symmetric code (A2P). It is a two-dimensional time-dependent code used to predict laminar transonic flows about lifting bodies. In common with other AFTON codes, a uniform timestep was incorporated in the original version of the program. The uniform timestep procedure requires that all of the variables of motion in every zone of the mesh be updated or calculated as often as required by stability considerations for the least stable zone of the mesh. The timestep is dependent upon the time required for the speed of sound to cross a zone. It was anticipated that significant computer time savings would be achieved if the non-uniform timestep procedure were incorporated in the code. This allowed the variables of motion to be calculated for a given zone of the finite difference mesh only as often as required by considerations of stability of that zone and its immediate neighbors. It was estimated that a speed-up factor between five and ten would result from the incorporation of non-uniform timestep procedure.

4.1 Treatment of Timestep Intermesh Boundaries

One of the most crucial aspects of the non-uniform timestep procedure relates to the way boundaries between adjacent zones having different timesteps are treated. In order to maintain AFTON conservation properties across the interface it had been previously determined by Trulio (Ref. 12) that correct time-centering of the pressures used to update the momenta was essential. As initially conceived, the interface treatment would assume that the adjoining large zone
would be treated in nearly all respects as the small timestep level zone (a micro zone) and updated with a small timestep. In this macro boundary zone the explicitly calculated pressure available from the first microstep update (there would be two such calculations in order to bring the zone correctly forward in time equivalent to a single macrostep calculation) is the correct pressure to use in updating the adjacent momentum zones. This pressure would be used for both microstep momentum updates and the single macrostep momentum update. This procedure would have correctly treated the momentum conservation constraints in the transition from a micro zone to a macro zone.

The method described above does not correctly conserve the total energy in the same transition region. For a macro level zone, $j$, in the Lagrangian case,

$$M_j (E_j^{n+1} - E_j^{n-1}) = -p_j^n \Delta V_j^n$$  \hspace{1cm} (1)

must be true to conserve total energy; where $M_j$ is the zone mass, $E_j$ the specific internal energy, $P_j$ the zone pressure and $\Delta V_j$ the change in volume. Superscripts refer to centering in time; $n$ is a microstep increment. Macrozone energy is updated in time from $n - 1$ to $n + 1$, an interval of $2n$ or two microsteps. Had we treated this boundary macro zone as a micro zone, as described in Paragraph 2, two such updates would be required.

$$M_j (E_j^n - E_j^{n-1}) = p_j^n \Delta V_j^n$$  \hspace{1cm} (2)

$$M_j (E_j^{n+1} - E_j^n) = -p_j^n \Delta V_j^n$$  \hspace{1cm} (3)

The pressure would no longer be correctly centered in relation to the internal energy. The resulting change in internal energy from $E_j^{n-1}$ to $E_j^{n+1}$ would not be the same as that calculated by Equation (1).

The interface was revised to incorporate a different method which preserves the general conservation properties. Macro zones in the interfaces were to be treated as macro zones in general respects; an intermediate pressure centered at $n$ is calculated in the macro zone and used to update the adjoining momentum zones. This pressure at $n$ is used with Equation (1) to explicitly calculate the internal energy at $n + 1$ and so preserve total energy conservation. Logic and coding were developed and inserted in the updated program.
4.2 Development of Non-Uniform Timestep Logic

The logic for controlling the zone properties updating sequence and the assignment of timestep levels for thermodynamic and momentum zones was included in two new subroutines inserted in the code. These subroutines, ZONEL and TMESTP are shown in the program flow diagram presented in Figure 1. ZONEL is the routine which assigns timestep levels to the thermodynamic and momentum zones. It is called from FLOW at the beginning of a new macrocycle. At that point all zones have been updated to the same time and new levels must be assigned for the new updating sequence.

The TMESTEP subroutine calculates and stores the actual timestep, based upon stability criteria, for each zone. It replaces coding formerly in MVS. This routine is also called from FLOW, but after MVS and after the final macrolevel update. The timesteps are then available for ZONEL to perform its required testing and assignment at the beginning of the next macrocycle. A listing of subroutine ZONEL is included as an Appendix to this report.

The program, as written, is currently restricted to only two timestep levels, although additional time reduction should result from multiple time levels. Thus, zones which are at least twice as large as the smallest zone require updating only every other cycle. Zones which are four times larger than the smallest zone would normally require updating only every fourth cycle, etc. A temporary restriction in timestep levels was inserted to simplify checkout and debugging. Coding is in place which will eventually allow up to five timestep levels.

4.3 Development of Conservative Logic

In order to maintain the conservative aspects of the AFTON numerical method an approach was taken which utilized momentum and mass accumulators to determine mesh boundary fluxes. This approach, because of its inherent simplicity, offered substantial savings in execution time in addition to the savings which were expected to result from the non-uniform timestep.
Figure 1: General Flow Diagram for Computer Program With Non-Uniform Timestep Scheme.
Complete conservation also required calculation of the work done on the surfaces of any body placed in the flow field. The AFTON differencing scheme results in a half momentum zone around each of the mesh boundaries and around any body in the mesh. Logic had to be implemented which accounted for the fact that the work calculation for the body surfaces included the momentum transported across these half-zone boundaries.

The various difficulties which were encountered while incorporating the conservation logic eventually led to exhaustion of the available funding for this program. It is estimated that about one-half of the conservation logic had been inserted in the code. Substantial additional programming and debugging effort is required to complete this task and insure that all of the conservation checks of the AFTON method are preserved.

4.4 Checkout and Debugging of Non-Uniform Timestep Coding

A test problem was run to check out the code for laminar flow over a right circular cylinder. Three cycles were run with timestep restricted to two timestep levels. The program appeared to execute normally. However, validation of the changes requires completion of the conservation logic and coding.
5.0 CONCLUDING REMARKS

An effort was undertaken to incorporate a non-uniform timestep in a two-dimensional, time-dependent, Navier-Stokes solution of the laminar, transonic flow over a lifting body. Available funding was exhausted during the implementation of the conservation logic. Without the conservation properties the accuracy of the numerical calculation is uncertain. Successful demonstration of the non-uniform timestep as a method for improving the efficiency of Navier-Stokes solutions requires additional programming and debugging effort.


APPENDIX

Subroutine ZONEL

![Image of the subroutine ZONEL]
ZUNEL 37
41 15 CONTINUE
42 L = L + 1
43 L = L + 1
45 20 CONTINUE
46 LCT = LCT + 1
47 DO 50 J = JMIN, JH
51 IF(JLE,JHE) GO TO 50
52 IF(LOC(JL),GT.DLEV(2)) GO TO 25
53 JZL(JL) = 1
54 GO TO 50
C TEMPARY COMMENT OUT IF TRANSFER TO LIMIT TO TAU LEVELS
C25 IF(ZOT(JL),GT.DLEV(3)) GO TO 50
62 50 CONTINUE
63 JZL(JL) = 2
66 GO TO 50
67 30 IF(ZOT(JL),GT.DLEV(4)) GO TO 35
75 JZL(JL) = 3
76 GO TO 50
77 35 IF(ZOT(JL),GT.DLEV(5)) GO TO 40
105 JZL(JL) = 4
106 GO TO 50
107 40 CONTINUE
113 50 CONTINUE
120 JML(JM,JL) = MIN(JZL(JM,JL),JZL(JM,JL))
132 DO 90 J = JMP, JH
133 IF(JLE,JHE) GO TO 90
135 JML(JM,JL) = MIN(JZL(JM,JL),JZL(JM,JL))
140 90 CONTINUE
147 JML(JM,JL) = MIN(JZL(JM,JL),JZL(JM,JL))
150 IF(JLE,JHE) GO TO 90
151 JML(JM,JL) = MIN(JZL(JM,JL),JZL(JM,JL))
C SPECIAL TESTING FOR KNOT
C 210 62 CONTINUE
214 J2 = J2 + 2
215 J2 = J2 + 2
217 J3 = J3
220 621 CONTINUE
223 JML(JL) = MIN(JZL(JL),JZL(JL),JZL(JL),JZL(JL))
230 IF(JLNE,JH) GO TO 90
232 J2 = J2 + 1
234 J3 = J3 + 1
235 GO TO 621
C SPECIAL TESTING FOR KLINES AROUND THE BODY
C 240 63 CONTINUE
244 J1 = J1
245 J1 = J1

ORIGINAL PAGE IS
OF POOR QUALITY
ZONEL

240 041 CONTINUE
250 JML(J1,L) = \frac{\sin(J2(L),J1L(J2,L))}{JML(J1,L)}
260 IF(J1.NE.J) GO TO 66
270 J1 = J1 + 1
280 J2 = J2 + 1
290 CONTINUE ZJNEL.75

C SPECIAL TESTING FOR KT3P

C
270 04 CONTINUE
270 J1 = J1 - 2
280 J3 = J3 - 1
290 J1 = J1 + 1
300 CONTINUE ZJNEL.76
310 JML(J1,L) = MINO(J1L(J2,L),J1L(J3,L),J1L(J4,L))
320 IF(J1.NE.J) GO TO 66
330 J2 = J2 + 1
340 J3 = J3 + 1
350 J1 = J1 + 1
360 CONTINUE ZJNEL.77
370 IF(LCT,E7) GO TO 68
380 L = L - 1
390 J2 = L2
400 J1 = J1 - 1
410 CONTINUE ZJNEL.78
420 JML(J1,L) = \frac{\sin(J1L(J2,L),J1L(J3,L),J1L(J4,L))}{JML(J1,L)}
430 IF(J1.NE.J) GO TO 66
440 J2 = J2 + 1
450 J3 = J3 + 1
460 J1 = J1 + 1
470 CONTINUE ZJNEL.79
480 IF(LCT,E7) GO TO 15
490 RETURN
500 END