CONTROL SOFTWARE FOR TWO DIMENSIONAL AIRFOIL TESTS USING A SELF-STREAMLINING FLEXIBLE WALLED TRANSONIC TEST SECTION

S. W. D. Wolf

UNIVERSITY OF SOUTHAMPTON
Southampton, England

Grant NSG-7172
July 1982

LIBRARY COPY

12 JUL 1982

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA
1. Introduction

2. Control System Hardware Outline

3. Control Software
   3.1. Program TSWT
   3.2. Subroutine DATA
   3.3. Subroutine REDUCE
   3.4. Subroutine WAS
   3.5. Subroutine STAR
   3.6. Subroutine SUME
   3.7. Subroutine FORCE
   3.8. Subroutine SET
   3.9. Subroutine WALL
   3.10. Program REAN
   3.11. Program RUN

4. Symbols

5. References

Figures

Appendix A
The operating procedure of a self-streamlining wind tunnel has already been discussed and is summarised in the flow diagram shown on Figure 1. The iterative nature of the streamlining process, requiring numerous measurements and calculations coupled with the need for a rapid and continual exchange of data between wind tunnel and computer, makes mandatory the use of a computer.

The current operation of the Transonic Self-Streamlining Wind Tunnel (TSWT) involves on-line data acquisition with automatic wall adjustment. A tunnel run consists of streamlining the walls from known starting contours in iterative steps and acquiring model data. Each run performs what is described as a streamlining cycle. The associated control software is presented here.

Development of the on-line control system for TSWT continues. It is anticipated that modifications to the existing software package will consist of minor changes to some of the subroutines.

The introduction of further subroutines for

(a) Calculation of model forces from the 'wall data' (wall pressures and positions).

(b) Solution of mixed imaginary flowfields with a shock present.

(c) Prediction of wall shapes, to minimise boundary interference on a three dimensional model in a two dimensional test section.

(d) Assessment of residual boundary interferences on a three dimensional model in a two dimensional test section.

is planned as they become available. The modular architecture of the software will allow these additions to be made easily.

The software package has been developed for simple application to other self-streamlining wind tunnels. The modular architecture allows individual control system subroutines to be utilised with existing users software. Also the software package is written in a general manner to minimise new application modification.

During TSWT development numerous programmes have been written to check sections of the control software. A number of these remain in use to assist with TSWT operation as follows:
i) Set both walls to known contours together or individually.

ii) Allow operator modification of known wall contours.

iii) Display current position of both walls.

iv) Display and/or load contents of any specified data file record.

In addition programs have been written to command a Tektronix 4662 plotter to display model pressure distributions, flexible wall Mach number distributions and wall shapes.
2. CONTROL SYSTEM HARDWARE OUTLINE

An overview of the current control system is shown in Figure 2. Macroscopically, the system involves the interaction of the tunnel operator with the wind tunnel and computer to generate the required test data.

The on-line control system has two distinct functions:

1) to streamline the flexible walls which includes assessment of residual boundary interferences.
2) to acquire and reduce test data from a model.

These functions are achieved by two control loops between wind tunnel and computer linked to processing software for data manipulation. The control loops are for wall shape control and pressure data acquisition (Scanivalve control). Data acquisition is automatic since the Scanivalve is stepped by computer commands and analogue pressure data is fed direct to the computer.

The control system hardware has evolved about these two control loops using both analogue and digital data transfer. Microscopically, the control system becomes complex as shown in Figure 3.

The system hardware consists of the computer and its peripherals communicating with a series of control and signal conditioning sub-systems housed in a control cabinet by the wind tunnel. This cabinet is then connected to the test section wall position sensors (linear potentiometers) and wall jack stepper motors.

The hardware will perform four functions:

1) wall movement.
2) wall and model pressure measurements.
3) wall position sensing.
4) system monitoring (not yet existing).

The wall movement function involves the loading of forty motor latch boards with direction information (stop, forward-go or reverse-go). Then the sending of a 'go' pulse to the Pulse Sequence Generator (PSG), starts actual wall movement. The PSG generates control pulses at a fixed frequency to the forty Motor Drive Boards (MDBs). The MDB, using signals from the motor latch board and the power
supply, controls the sequence of power pulses transmitted to the 3-phase stepper motor powering each wall jack. After a pre-determined and variable time interval the PSG is switched off and wall movement ceases. The wall movement control sequence is then repeated until satisfactory wall contours are achieved.

The wall and model pressure measurements involve the driving of the scanivalve system by step pulses, to a required sampling port. Then the analogue signals from the pressure transducers are sampled by the computer after a suitable settling time (i.e. 50 milli-seconds).

The wall position sensing function involves the sampling of forty analogue signals from the linear potentiometers attached to each wall jack mechanism. These signals are continuously available at the computer peripherals, but they have been found to be susceptible to electronic interference when the jack motors are switched on. Hence the wall position is only sampled with the walls stationary.

The system monitor is a necessary part of a practical digital control system. When this hardware becomes available it will provide information on the status of hardware components, to allow quick error diagnostics during tunnel operations.

The conditioned analogue signals from the test section (within the range ± 5 volt) are fed via the control cabinet to a DEC AD11-K Module for 12-bit analogue to digital conversion. This module is combined with a DEC AM11-K Multiplexer to make 64 channels available for input signals.

The digital signals, at 0 and 5 volts, are transmitted to the wind tunnel in a code described in Appendix A. Device selection is by means of a 'telephone exchange' called the "address decoder". In fact, commands to the test section are sent to all devices but only one device is enabled to read the information, by address decoder selection. The versatility of a digital control system is well known and the reduction in interface wiring compared with an analogue system is significant. The address decoder has the capability of addressing 64 devices.

The control system hardware layout in Figure 3 has been simplified for clarity. In practice there are numerous synchronisation signal paths between devices, to ensure correct operation sequencing and to prevent 'race' problems.

The tunnel operator monitors the control system from a computer VDU consol and inputs test parameters. The consol allows the real time display of test section and model data which is stored on the computer mass storage device. This data can subsequently be drawn in graphical form on an XY Plotter (Tektronix 4662).
Hard copy of summarised run data is printed in real time on a line printer. The control of a model wake traverse mechanism has been incorporated in the system, utilising the wall movement technique already described.

The control system hardware has excess capability with 11 spare digital address slots and 15 spare analogue input channels. This spare capacity may be used for the control of tunnel Mach number and model attitude in the future.
Computer software has been developed for the on-line TSWT control system using a versatile modular architecture. Hence the program has been reduced to a collection of manageable subprograms which can be combined to control the wind tunnel and output real time results or provide more detailed off-line re-analysis of previously acquired data.

An overview of the control software package is shown below.

<table>
<thead>
<tr>
<th>File Type &amp; Name</th>
<th>File Storage Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Program</td>
<td>OFLEX</td>
<td>i) Control and sequence subroutine calls.</td>
</tr>
<tr>
<td>(TSWT)</td>
<td></td>
<td>ii) Read test parameters from the operator.</td>
</tr>
<tr>
<td>Subroutine 1</td>
<td>OAD</td>
<td>Acquire pressure data from the wind tunnel.</td>
</tr>
<tr>
<td>(DATA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine 2</td>
<td>ODR</td>
<td>Read tunnel data from disc storage and reduce raw pressure data from the wind tunnel.</td>
</tr>
<tr>
<td>(REDUCE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine 3</td>
<td>OJUDD</td>
<td>Perform wall setting calculations.</td>
</tr>
<tr>
<td>(WAS)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine 4</td>
<td>ODST</td>
<td>Calculate local boundary layer displacement thickness and Mach number along each wall.</td>
</tr>
<tr>
<td>(STAR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine 5</td>
<td>OERR</td>
<td>Assess wall induced interferences at the model.</td>
</tr>
<tr>
<td>(SUME)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine 6</td>
<td>{ OWING ONPL }</td>
<td>Calculation of model forces for NACA 0012-64 and NPL 951O sections respectively.</td>
</tr>
<tr>
<td>(FORCE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subroutine 7</td>
<td>OUT</td>
<td>i) Store run data on disc</td>
</tr>
<tr>
<td>(SET)</td>
<td></td>
<td>ii) Output data to the terminal and/or the plotter.</td>
</tr>
<tr>
<td>Subroutine 8</td>
<td>OADJ</td>
<td>Move the walls to new contours</td>
</tr>
<tr>
<td>(WALL)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This breakdown of the software into modules has been extremely useful, particularly for storage, editing and debugging purposes.

The software written in FORTRAN IV language is run on a DEC 11/34 with a DEC RT-11 V4 operating system. The software is linked to a system library and a FORTRAN library to access functions and system subroutines and a Real Time System Library (RTSL) to access peripheral control subroutines. The complete compiled and linked program requires over 100 blocks (25.6K words) of memory spare.

Current 16-bit computer processors are only capable of addressing 32K words (64K bytes) of real memory space. But of this, only 22K words is available for a user's program, depending on the size of the operating system. Therefore to run the TSWT control software on a 16-bit machine a technique of overlaying has to be used, so that only part of the software is stored in the real memory at any instant during execution.

Each subroutine is a self contained program communicating with the main program via common data blocks. So in theory only one subroutine is required in the real memory at any one time for execution. In practice, the subroutines have been grouped together to minimise the number of overlays, thereby reducing the time required for overlaying. The overlaying structure of the control software is shown below

<table>
<thead>
<tr>
<th>SEGMENT 1</th>
<th>SEGMENT 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAY Subroutine 1</td>
<td>Subroutine 5</td>
</tr>
<tr>
<td>REGION Subroutine 2</td>
<td>Subroutine 6</td>
</tr>
<tr>
<td></td>
<td>Subroutine 7</td>
</tr>
<tr>
<td></td>
<td>Subroutine 8</td>
</tr>
<tr>
<td></td>
<td>7762 words</td>
</tr>
<tr>
<td>ROOT Main Program</td>
<td>System Library</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>FORTRAN Library</td>
</tr>
<tr>
<td></td>
<td>RTSL Library</td>
</tr>
<tr>
<td>9470 words</td>
<td>7762 words</td>
</tr>
</tbody>
</table>

This program structure is implemented at 'link' time during the program generation cycle as below
These commands generated a runnable program called OFLEX. The control software memory requirement was reduced from 27k words to 17.3k words.

At 'run' time the program OFLEX requires four data files to exist on the computer mass storage device. Data file ADC.DAT receives the raw analogue-to-digital counts of the 'wall and model data' for each streamlining iteration; PAD.DAT provides and receives sets of wall contours and the associated external imaginary wall velocity distributions; NPL.DAT or WING.DAT receives Cps from the NPL 9510 and NACA 0012-64 models respectively, for each streamlining iteration; TWST.DAT holds all fixed tunnel data, i.e. jack positions, potentiometer calibrations, scaling and coupling factors, matrix coefficients for camber interference assessment and boundary layer information. RUN.DAT holds run data, i.e. ambient temperature and pressure, run number, iteration record number, and number of model tappings. The data files ADC.DAT, PAD.DAT and NPL.DAT/WING.DAT each hold 50 records: Records 1 to 3 in PAD.DAT hold data on the three aerodynamically straight contours (i.e. position and boundary layer thickness). Records 1 and 2 of NPL.DAT and WING.DAT files are used to store model tapping X and Y co-ordinates relative to the model leading edge. Records 4 to 50 are available to store data from each streamlining iteration. Hence iteration record numbers range from 4 to 50. When the iteration record number equals 50, ADC.DAT, PAD.DAT and NPL.DAT/WING.DAT must be copied to a data bank since the original data is then overwritten by subsequent iterations, starting with record 4. The upper limit on the iteration record number has been chosen to keep the data files in manageable proportions (i.e. 25.6k words maximum size). The total storage requirement of data files to run OFLEX in 61.7k words.

The data file RUN.DAT must be loaded with current run data before the control software is activated. This operation is performed by running a program OSTART, the software of which, called RUN, is described in section 3.11.

A complete listing of the control software is described in the following sections. Where possible standard FORTRAN has been used but peripheral control commands are peculiar to the DEC system used. These subroutine calls can be grouped into Analogue to Digital sampling commands (ADC and RTS), and programmable clock commands (SETR and LWAIT). In addition there are calls to the system library routines (IPEEK and IPOKE) for digital input and output operations.
An example of the minimal printout from OFLEX associated with a typical streamlining cycle, involving two wall adjustments, is shown on Figure 5. The walls were initially set to contours stored in record 30 of file PAD.DAT and the run finished with wall contours as stored in record 32. The model had no pressure tappings and therefore no wing performance data was presented in the print-out. Note that the print-out from OFLEX is sent to a line printer (logical Unit 7) while operator information is sent to a VDU Consol (Logical Unit 5). The operator information consists of prompts to indicate the stages reached in the streamlining cycle and error warnings.

The versatility of the software has allowed simple generation of programs for particular tasks such as tunnel data re-analysis. Using the existing subroutines as building blocks, a new program has been made up of a series of these subroutines linked to a new main program. For example, data re-analysis is achieved by running the program ORLEX. The main program REAN is a modification of TSWT with different subroutine calls and an extended print-out demanded as described in section 3.10. The program structure is very similar to that of the control software and is generated using the following link command with a memory requirement of 17.1k words.

```
R LINK
*ORLEX = OREF,FORLIB,RTSL/C
*OAD,ODR,OJUDD/O:1/C
*OERR,ODST,OUT/O:1/C
```

where OREF is the file storage name of program REAN

An example of the extended printout is shown on Figure 6 for a typical re-analysis of raw TSWT data for one iteration of run 389.

Should any new analysis technique become available, then a new subroutine could replace or supplement the existing subroutines, and be incorporated in OFLEX and/or ORLEX by minor adjustments to the main program and the LINK commands.
3.1. Program TSWT

The main control program listed on Figure 4.1 reads tests parameters from the tunnel operator and sequences subroutine calls:-

Lines 0002 - 0009

Define all common data blocks used thus:-

Block 1

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NJ</td>
<td>Effective number of jacks per wall (real plus dummy).</td>
</tr>
<tr>
<td>MT</td>
<td>Number of model pressure tappings.</td>
</tr>
<tr>
<td>NR</td>
<td>Number of tunnel reference samples during a scanivalve scan.</td>
</tr>
<tr>
<td>CL1</td>
<td>Printout control value.</td>
</tr>
<tr>
<td>Bl</td>
<td>Prandtl-Glauert scaling factor $= \sqrt{1-M^2}$</td>
</tr>
<tr>
<td>PR1</td>
<td>For development only.</td>
</tr>
<tr>
<td>AK1</td>
<td>Aerodynamic coupling factor.</td>
</tr>
<tr>
<td>AK3</td>
<td>Wall movement scaling factor.</td>
</tr>
<tr>
<td>AN</td>
<td>Model angle of attack (degrees).</td>
</tr>
<tr>
<td>R3</td>
<td>Chord Reynolds number.</td>
</tr>
<tr>
<td>PP2</td>
<td>Dynamic pressure allowing for compressibility.</td>
</tr>
<tr>
<td>ITRN</td>
<td>Run number.</td>
</tr>
</tbody>
</table>

Block 2

<table>
<thead>
<tr>
<th>Array</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN</td>
<td>Bottom wall potentiometer outputs, volts.</td>
</tr>
<tr>
<td>RS</td>
<td>Top wall potentiometer outputs, volts.</td>
</tr>
<tr>
<td>W</td>
<td>Bottom wall imaginary wall velocities.</td>
</tr>
<tr>
<td>X</td>
<td>Top wall imaginary wall velocities.</td>
</tr>
<tr>
<td>P</td>
<td>Top wall real wall static pressures.</td>
</tr>
<tr>
<td>Cl</td>
<td>Bottom wall real wall static pressures.</td>
</tr>
<tr>
<td>RD</td>
<td>Development data.</td>
</tr>
</tbody>
</table>
Block 3  Array   D  =  Measuring point co-ordinates.
          "  WTY  =  Top wall movement from the straight
          "  WBK  =  Bottom wall movement from the
          "  WL  =  Effective panel length per wall
                  tapping.
          "  B  =  Model Cps.
          "  PD  =  Transducer Cps.

Block 4  Array   E  =  Calculated imaginary external top
          "  H  =  Calculated imaginary external bottom
          "  Y  =  Predicted top wall movements
                  required (Inches).
          "  G  =  Predicted bottom wall movements
                  required (Inches).
          "  WI  =  Real top wall velocities squared
                  ($V/U_\infty$)$^2$
          "  XI  =  Real bottom wall velocities squared
                  ($V/U_\infty$)$^2$

Block 5  Array   U  =  Difference between real and imaginary
          "  V  =  Difference between real and imaginary
                  top wall velocities ($u/U_\infty$)
          "  DS  =  'Straight' wall local $\delta^*$ values.
          "  DET  =  Local $\Delta\delta^*$ values.
          "  CS  =  Matrix coefficients for camber interference assessment.
Initialise Arrays X & DET.

Set streamlining trigger (PR1) to zero (i.e. walls not streamlined).

Set automatic in-file selection trigger (IAM) to zero (i.e. Non automatic selection).

Define File 5 as data file RUN.DAT.

Read run data from RUN.DAT and load ITRN with the run number, IFN with iteration record number, TAB with ambient temperature, AMP with ambient pressure and MT with number of mode 1 taps.

Set extended printout trigger (CL1) to zero (i.e. minimal print-out required).

NR = Number of tunnel reference samples in a pressure scan.

NJ = Number of measuring points per wall (See Figure 7).

Input test parameter:

Angle of attack (AN)

Subroutine call sequence.

If no model tappings (i.e. MT=0) do not call Subroutine FORCE.

If walls streamlined (i.e. PR1=1) do not call Sub-routine WALL
When the Iteration Record number is 50 and jump out of streamlining loop.

Increment the Iteration Record number

If walls are un-steamlined (i.e. PR1=0) continue the streamlining cycle.

Load run data array TUN and load file RUN.DAT with new values of TUN.

3.2. Subroutine DATA (Incorporating Subroutines GAD and STEP)

This subroutine primarily acquires all the tunnel pressures either from the tunnel via an automatically stepped scanivalve or from disc storage. Some data is then reduced.

The option to acquire old data stored on disc allows re-analysis of previous runs. All analogue to digital conversions associated with the pressure data are handled by software in Subroutine GAD. Automatic control of the Scanivalve is achieved by software in Subroutine STEP. The listing on Figure 4.2 can be broken down thus

Dimension of data blocks DA, DB, DC and IDATA.
Define only common data blocks required in Subroutine DATA.
Initialise the first row of 2D-array IDATA.
CHD = Model chord = 4 inches (NACA 0012-64)
= 6 inches (NPL 9510)
NJ1 = number of wall jacks.
NR = number of tunnel reference samples in a pressure scan.
IPL = Approximate low value of scanivalve position encoder output at pressure scan start condition.
Transducer calibration valves (cm Hg per A-D count).
If a re-analysis (i.e. CL1=500), open data file 4, select an existing record (IR) and fill IDATA with raw transducer A-D counts from a previous TSWT run.

Load pressure channel zeroes into IDATA

Identify all passes of the data acquisition cycle other than the first and print a prompt.

Initiate DO loop to acquire tunnel pressures using the scanivalve system.

If the first data acquisition cycle (i.e. X(30)=0) read the four pressure transducer zero signals and halt the software until a 'CR' is received from the terminal - transmitted when the tunnel flow is stabilised by the operator.

Wait for a change in the encoder output generated by a scanivalve step using Subroutine STEP. If there is a step error halt program.

Sample four transducer outputs using Subroutine GAD and load array IDATA.

Load pressure channel zeroes into array IX

Open a data file 4 and fill a selected record (IOT) with 192 raw A-D count values from the iteration, stored in IDATA.

Convert raw data into pressures relative to tunnel reference pressure (cm Hg), and load into array PD.

Load arrays P and Q with wall jack centreline pressures with dummy measuring points included (See Figure 7).

Load arrays P and Q with mid-jack centreline pressures.

RD (NR + 1) = test section total pressure reading.

RD (NR + 2) = Ambient pressure (cm Hg).

RD (NR + 3) = Ambient temperature (deg C).

RL = Stagnation pressure (cm Hg)
Calculate tunnel reference Mach numbers at regular intervals through the pressure scan, and look for excessive fluctuations (i.e. $\Delta M > .01$).

Calculate selected tunnel pressures for possible calibration checks.

Store average tunnel reference static pressure in Bl.

Subroutine GAD

Subroutine label with data transfer of value N and array IDATA.

Dimension of IDATA (Average A-D counts) and IDT (Raw A-D counts) arrays.

NS = Number of samples per channel

Time delay of 50 milli-seconds using the programmable clock (Subroutine SETR).

Take NS samples of four analogue input channels at 1kHz and store digital results in IDT.

Convert all raw A-D counts from packed integer format to real number format (Subroutine CVSWG) then take the average of the NS samples per channel and store in IDATA.

Stop the programmable clock.

Subroutine STEP

POKE digital output register with the command number 16384 (See Appendix A). Note no scanivalve address is required with the present hardware configuration.

Time delay of 20 milli-seconds using the programmable clock.

POKE digital output register with a zero value to clear.
3.3. Subroutine REDUCE

This subroutine performs data reduction operations on tunnel pressure information and acquires further tunnel data from data files held on the mass storage device. The software listed on Figure 4.3. is described thus:

Lines 0004 - 0010 Define only common data blocks required in subroutine REDUCE.

Lines 0011 - 0015 Set variables, as previously described for subroutine DATA.

Lines 0016 - 0025 Calculate the average freestream Mach number of the run.

Line 0026 PP2 = qI/qc compressibility correction to tunnel q.

Lines 0027 - 0041

Lines 0042 - 0054 Convert tunnel pressure data into Cp using the tunnel reference pressure associated with each group of six readings.

Lines 0055 - 0080 Load model pressure coefficients into array B from transducer channels 1 and 3 data. Note this software is model dependent and configured for use of the NPL 9510 section.

Lines 0083 - 0092 Specify IWF corresponding to 'Straight Wall' base data for Mach number panels M < .725, .725 < M > .825 or M > .825 for re-analysis case only. Otherwise a dummy data base is used (i.e. IWF=1).

Lines 0094 - 0096 Load arrays DA, DB and PC with data from File 2 (TUN.DAT).

Lines 0097 - 0098 Define File 3 as data file PAD.DAT.

Lines 0099 - 0109 Define wall contour record number automatically (i.e. IAM=1) or by operator input.

Line 0110 Load array DC with 'wall data' from File 3.
3.4. Subroutine WAS

The wall adjustment strategy\(^1,2,3\) (WAS) manipulates the calculated imbalance between real and imaginary wall velocities to generate a wall movement which will give zero wall loading or vorticity.

The strategy requires interpolation of real wall velocities at regular intervals along each wall. To ensure accurate interpolation at the wall ends using curve fitting to the wall velocity distribution, two straight dummy wall extensions are added to each flexible wall, 15.25 cm (6 inches) and 22.86 cm (9 inches) in length respectively. The dummy wall measuring points introduced are points 1 and 2 upstream where the wall velocity is assumed free-stream and points 22, 23 and 24 downstream, which have a wall velocity equal to that of measuring point 21 or jack 19. Measuring point 1 is also assumed to be the origin of the wall boundary layer. Measuring points 3 to 21 are real and correspond to the position of wall jacks. The software representation of each flexible wall, in terms of measuring point is shown in Figure 7. The software listed on Figure 4.4. is described below.

Define only common data required in subroutine.
Dimension working arrays.

Message to tunnel operator.

Equalising top and bottom wall coupling and scaling factors.

Compute external velocities for next wall shape where

\[ Q_1 = \text{compressible dynamic pressure} \]
\[ \text{TEMP}1 \, \text{&} \, \text{TEPL} = \text{incompressible top and bottom wall Cps.} \]
\[ U \, \& \, V = \text{imbalance between real and imaginary wall velocities (Top and bottom walls).} \]
\[ E \, \& \, H = \text{external velocity perturbations computed for the next top and bottom wall shapes} \left( \frac{v}{u_\infty} \right) \]
\[ XI \, \& \, WI = \text{real wall velocities squared} \left( \frac{v}{u_\infty} \right)^2 \]

Load Z with interjack streamwise spacing co-ordinates.

Set up DO loop for both walls.

Load A & XB with D \& U or V data in sets of four values for top or bottom wall.

Cubic spline fit to each set of data to obtain wall velocity imbalance between jacks.

Summation of velocity induced by the vorticity distributed along a wall at each jacking point. The result is stored in arrays S for the top wall and T for the bottom wall.

End of wall velocity analysis repeated for each wall.

Numerical integration of top and bottom wall normal velocity components to generate jack movement demands.

Raw movements are stored in TT and R. Arrays Y \& G store the effective jack movements after scaling.

Likewise E and H (the external velocity perturbations) are modified by scaling in lines 0133 \& 0134. In lines 0135 \& 0136 values in arrays Y and G are converted to compressible values using \( B_1 = \sqrt{1-M^2} \).
3.5. Subroutine STAR

The function of STAR is to calculate Mach number and boundary layer displacement thickness values for each wall jacking point necessary only for re-analysis information. The boundary layer calculations use a numerical solution of the Von Karman Momentum Integral equation for a turbulent boundary layer. The software listing is shown on Figure 4.5.

Line 0002  Dimension of array DELTA for $\delta^*$ storage.
Lines 0003 - 0007  Define only required common data blocks.
Lines 0008 - 0011  Calculation of isentropic flow relationships.
Lines 0012 - 0013  Calculation of air density (kg/ft$^3$)
Lines 0014 - 0015  Calculation of air temperature (K).
Lines 0016 - 0017  Calculation of air viscosity (lb/ft$^2$sec$^1$).
Line 0018  CLI value selects extended printout when equal to 500
Lines 0027 - 0145  Overall Do loop performs calculations for each jacking on both walls using sets of three jacks, labelled 0, 1 and 2.
Lines 0044 - 0045  Load X1 and X2 with jacking 1 and 2 co-ordinates relative to jacking 0.
Lines 0046 - 0051  Ensure that the correct tunnel reference pressure is used with each wall pressure.
Lines 0053 - 0062  Load SP1, SP2 and SP3 with top or bottom wall velocity perturbations.
Lines 0063 - 0068  Calculate wall velocities U0, U1 and U2 (ft/sec)
Lines 0069 - 0077  Calculate local wall mach number and store in P (top wall) or Q (bottom wall).
Lines 0081 - 0087  For the second measuring point assume a turbulent boundary layer growth according to $\delta^* = \frac{.0213x}{0.1428Rx}$.
Lines 0088 - 0092  Calculate the velocity gradient at jacking 1 and store value in D2.
3.6. Subroutine SUME

SUME assesses the residual interferences, due to any residual wall loading, along the tunnel centre-line and the model chord line. The three components of the interferences are related to estimated changes in $C_L$. The quality of streamlining is then assessed from these interferences and a measure of the average pressure imbalance across the walls. A listing is shown on Figure 4.6.

Lines 0002 - 0007 Define only required Common Data blocks.

Lines 0008 Define model in use
IW $T = 0$ for NACA 0012-64
IW $T = 1$ for NPL 9510

Line 0010 CHD = Model chord (inches)

Line 0013 OR = X co-ordinate of model pivot point which is at $\frac{1}{4}$ chord and nominally mid way between the straight walls (Inches).
Calculate model chord line X and Y increments and store in AX and AH respectively.

IPP = Position of model pivot point as a number of chords.

OR = X co-ordinate of the model ¼ chord point - the origin.

Subroutine print trigger (OT) set to zero for print-out.

Eliminate dummy jacks by reducing number of jacks per wall by four.

Y3 = nominal tunnel semi-height (Inches).

Calculate imaginary wall velocity squared and sum the absolute value of the load difference between real and imaginary velocities squared for each jack. Store the results in EE (top wall) and F (bottom wall).

Calculate the average $C_p$ difference between real and imaginary flows for each wall (called E) and store as ET and CEB, and print-out.

If minimal print-out is specified (i.e. CLI = 0) jump to line 0083

For 37 points along the tunnel centreline summate the effect of wall voricity, at the model and along the tunnel centreline.

X1 = X co-ordinate of measuring point.

Sum the velocity components of the wall voricity at co-ordinate X1 on the tunnel centreline and store results in arrays UT and VT.

X2 - Horizontal displacement between jack and analysis point (Inches).

Y2 = Displacement of bottom wall downwards from the centreline (Inches).
Calculate the horizontal induced velocity perturbation component due to vorticity at measuring points n. (See Figure 7)

\[ U_l = \frac{Z_n}{2\pi} \times \sum_{\text{Top}} \frac{Y_n}{X_n^2 + Y_n^2} \times \Gamma_n \]

where \( \Gamma_n \) = Vortex strength
- Local imbalance of wall real and imaginary velocities.
- \( X_n \) = horizontal separation of analysis point and a vortex assumed at a measuring point n.
- \( Y_n \) = vertical separation of same.
- \( Z_n \) = panel length = distance between mid-jack points spanning measuring point n.

In software notation

\[ U_l = \frac{W_l}{T_o p_i} \times ((U \times R2) + (V \times R3)) \]

where \( W_l \), \( U \), and \( V \) are arrays.

Lines 0067

Calculate the vertical component of induced velocity perturbation \( V_l \) due to vorticity at measuring points n

\[ V_l = \frac{Z_n}{2\pi} \times \sum_{\text{Top}} \frac{X_n}{X_n^2 + Y_n^2} \times \Gamma_n \]

Lines 0076 - 0079

If \( O_T = O \) print-out results.

Lines 0083 - 0119

Calculate velocity perturbations for nine equidistant points along the model chord line.

Line 0091

Calculate induced flow angles along the model chord line (AA2).
Line 0092  Calculate induced $C_p$ (CP1).
Line 0093  Store $\frac{1}{4}$ chord point $C_p$ in address CP.
Line 0103 - 0105  Load $A1$ and $A2$ with leading edge and trailing edge
induced flow angles.
Line 0107  $SP = \text{Contribution to induced camber.}$
Lines 0110 - 0112  Differentiate between odd and even numbered measuring
points.
Lines 0114 - 0118  Load accumulators for Simpsons rule numerical integra-
tion technique.
Line 0120  $CPE = \text{Average } C_p \text{ errors induced along model chord.}$
Line 0123  $P3 = \text{Induced camber in terms of } \Delta C_L.$
Line 0124  $A3 = \text{Induced camber (degrees).}$
Line 0128  $CL = \text{Induced Angle of Attack in terms of } \Delta C_L.$
Line 0129  $A1 = \text{Induced Angle of attack (degrees).}$
Lines 0130 - 0139  If a re-analysis print-out all residual interference
information (i.e. $CL1 = 500$).
Lines 0140 - 0142  Print abbreviated version of residual interference
information if $CL1 = 0$.
Lines 0145 - 0154  Increment ISC for each quality of streamlining satis-
fied.
Line 0156  If all the qualities of streamlining are satisfied
(i.e.$ISC = 5$) $PR1 = 1$, the walls are streamlined.
Line 0160 - 0163  If the walls are streamlined print messages to the
operator and the line printer.

3.7. Subroutine FORCE (Based on a NASA numerical technique)

FORCE integrates the model pressure distribution to give coefficients $C_L$,
$C_D$ and $C_M$. 

- 23 -
3.7.1. NACA 0012-64 version - listed on Figure 4.7a.

Lines 0002 - 0006 Define required Common Data blocks.

Lines 0007 Dimension working arrays.

Line 0010 \( MT = \) Number of model tappings +2 for dummy leading edge tappings.

Lines 0011 - 0012 Open data file 2 (WING.DAT)

Line 0013 Load array AG with X co-ordinates of model tappings (upper surface followed by lower surface).

Line 0014 Load AJ with Y co-ordinates as for the X co-ordinates.

Line 0015 \( IHT = \) Number of model tappings per surface.

Lines 0016 - 0023 Titling and test parameter input from the terminal (AN = angle of attack - degrees).

Lines 0024 - 0029 Calculate an effective vertical component for each model tapping, and store in array AH.

Lines 0031 - 0040 Initiate a DO loop with special processing for first and last tappings on each surface and different analysis for each surface.

Line 0042 \( WF = \) Weighting Factor, in this case for \( C_N \).

Line 0043 \( L6 = \) Local components of \( C_N \) for upper surface.

Line 0044 \( L5 = \) Sum of \( C_N \) components.

Line 0045 Store negative L6 in array A.

Lines 0047 - 0050 Calculate the upper surface \( C_N \) component for the dummy leading edge tap.

Lines 0051 - 0056 Calculate the \( C_N \) component for the last downstream tap on the upper surface.

Line 0055 \( C_N = \) Upper surface \( C_N \).

Lines 0057 - 0061 Calculate the lower surface \( C_N \) component for the dummy leading edge tap.

Lines 0062 - 0067 Calculate \( C_N \) component for the last downstream tap on the lower surface.
CNL = Lower surface $C_N$.  
$L5$ = Total model $C_N$.  

Calculate $C_N$ components for lower surface tappings.  

Do loop to calculate $C_L$ components for each tapping.  

$C5$ = Upper surface $C_c$ component.  

Store $C5$ in array $AH$.  

$C3$ = Sum of $C_c$ components.  

$CCP$ = Super surface $C_c$.  

$C5$ = Lower surface $C_c$ component.  

$St$ = Lowers surface model $C_c$.  

$C3$ = Total $C_c$.  

Do loop to calculate $C_m$ components about the model leading edge.  

$BB$ = Upper surface $C_m$ components.  

$C9$ = Sum of upper surface $C_m$ components.  

TIT = Tapping position (% chord).  

$BB$ = Lower surface $C_m$ components.  

$C8$ = Sum of lower surface $C_m$ components.  

$ST1$ = Total $C_m$.  

Output forces $C_N$, $C_c$, $C_m$.  

Calculate coefficients $C_L$ and $C_D$ from $C_N$, $C_c$ and angle of attack.  

Output wing performance coefficients.  

Close data file.  

3.7.2. NPL 9510 version - listed on Figure 4.7b

This version of FORCE is the same as the NACA 0012-64 version in all respects, except for the number of model tappings (50 in total, with a split 32:18 on upper and lower model surfaces respectively) and the labelling associated with the print-
out. Also notice that print-out control commands have been inserted at lines 0015, 0092, 0108, 0112, 0120 and 0126. Setting CLI = 0 suppresses all print-out except data associated with the label 'WING PERFORMANCE'. NPL 9510 model Cps are stored in NPL.DAT for subsequent plotting (See next section).

3.8. Subroutine SET

SET controls the output of tunnel data to the VDU consol and the line printer and the output of results to disc storage. This software is configured for testing the NPL 9510 section. A listing is shown on Figure 4.8.

Line 0002 Dimension data array DA.

Line 0003 - 0009 Define required Common Data blocks.

Line 0010 Initialise some array elements for print-out purposes.

Lines 0011 - 0012 Open wall output file 3 with 50 records each 512 words in length (PAD.DAT).

Lines 0013 - 0014 Open wing output file 2 with 50 records each 128 words in length (NPL.DAT).

Line 0015 \( \text{NJL} \) = Number of wall jacks.

Line 0016 Define model used: IWT = 0 for NACA 0012-64;
IWT = 1 for NPL 9510.

Line 0017 CLI = 0 for minimal print-out, so jump to line 0055

Lines 0019 - 0035 Output of pressure transducer Cps, external wall velocity for each jack and the predicted jack movement for zero wall interference.

Lines 0036 - 0039 Print-out top wall labels.

Lines 0042 - 0043 Define File 6 as WALM.DAT

Lines 0044 - 0053 Output wall Mach number distributions stored in arrays P and \( \theta \) into an operator set (IMR) record of File 6 for subsequent plotting.

Lines 0055 - 0080 Store potentiometer values for new wall contours in array DA. If CLI = 500 print-out information on jack X & Y co-ordinates, local Mach number, current potentiometer volt values, predicted potentiometer volt values for the next iteration.
WM = Effective position of top wall contour allowing for boundary layer growth along the wall.

If movement demands for the downstream jacks (No. 16-20) exceed mechanical limits in terms of pot volts, set to safe minimum values.

If CLI = 500 print out the local wall Mach Number between jacks.

Repeat analysis and print-out for the bottom wall.

Calculation of wall Mach number standard deviation for top and bottom walls for use in special re-analysis of tests with an empty test section.

Set KT to equal the total number of jacks.

Store top wall external velocity perturbations in array DA.

Print prompts on wall and model output records.

Write array DA into record IFN of File 3 on disc.

Define the number of model tappings and store model Cps in record IFN of File 2 on disc.

Close data files 2 and 3.

3.9. Subroutine WALL (incorporating Subroutines INIT, START, MOVE and DIO)

This subroutine controls the test section wall adjustments. Both walls are moved simultaneously in variable increments of movement. Each jack is commanded to move away from the model in its last adjustment to ensure the walls are rigid to air pressure loads. Numerous safety checks are included to guard against jacks jamming or jacks out of control.

Subroutine INIT initialises the control system. Subroutine START switches the jacks on and off for a time interval proportional to the average position error of the wall adjacent to the model pressure surface. Subroutine MOVE loads each jack with direction information and checks the data is loaded. Subroutine DIO handles all the Digital input and output operations between wind tunnel and computer.

The software is dependant on the tunnel hardware and is listed on Figure 4.9
and described thus

Line 0002 Define Common data required.
Line 0003 NWJ = total number of wall jacks.
Lines 0004 - 0005 Define arrays thus
   PC = potentiometer calibration (volts per inch)
   IM = stores jack movement demands for checks.
   ADC = stores wall pot. volts for movement checks.
   PV = wall contour pot. volts.
   PL = minimum safe pot. volt readings for downstream
   jacks.
Line 0006 Load array PL with values
Lines 0007 - 0010 Define File 3 as PAD.DAT and load array PV with a set
   of wall contour pot. volts from record IFN of File 3.
Line 0011 IF = 1 for the first pass through the wall adjust-
   ment cycle
Line 0012 IP = anti-backlash overshoot increment.
Line 0013 Initialise the control system.
Lines 0014 - 0016 Load variables
   IW = Wall number (0:top/l:bottom)
   ICNT = No. of attempts to move one jack.
   IS = No. of jacks correctly positioned.
Line 0017 ITOL = Wall setting tolerance band in pot. volts.
Line 0018 - 70 Calculate movement required for each jack from its
   current location and load each jack with direction
   information.
Lines 0019 - 0021 Read the potentiometer output for jack (L), convert
   the result to pot. volts, and store in IPV.
Line 0022 IMOVE = Actual jack movement from previous position.
   Ignored on first pass.
Line 0023 Store current jack pot. volts (IPV) in array ADC.
Lines 0024 - 0027 If IPV > 980 (i.e. jack approaching mechanical limit)
   then stop.
Single out downstream jacks (i.e. No. 16-20 and 36-40) and ensure that the demanded pot. volts are greater or equal to the minimum safety values.

IPD = Demanded jack pot. volts adjusted to overshoot the wall position, towards the model, by an increment IP.

IMV = Required jack movement in pot. volts. (+ve towards the model and -ve away from the model).

If the wall is being adjusted from its overshoot position (i.e. IP = 0) only allow the wall to move away from the model (i.e. IMV < 0). Set IMV = 0 if the jack tries to move towards the model. This technique ensures the wall is rigid to air pressure loads by eliminating jack mechanism backlash in the correct sense.

Change the sign of all bottom wall movement demands for hardware compatibility.

Change the sign of top wall movement demands for all odd numbered jacks to accommodate the hardware configuration for the top wall control.

Jump to line 0063 if this is the first pass (i.e. IF = 1).

For jacks 1 and 21 do not check for jack jamming due to the small movement demands on these jacks positioned close to the wall anchor point (See figure 7).

If the demanded jack movement (IML(L)) is more than 50 pot. volts and the jack potentiometer reading shows only a change of 10 or less pot. volts then warn the operator and abort the wall adjustment.

If only one jack has yet to be correctly adjusted (IS = 39) increment ICNT for each pass through the adjustment cycle.
If IS = 39 and ICNT = 6 then warn the operator and abort the wall adjustment. One jack may have moved out of its potentiometer measuring range.

If the change in jack pot. volts between position samples is greater than 50 warn the operator and abort the wall adjustment. A jack may be moved the wrong way.

Load array IM with values of required jack movement.

If the required movement is less than or equal to ITOL go to statement 40 (line 0067).

Increment IS by one and set IMV = 0.

Load jack (L) with direction information using Subroutine MOVE.

Set IF = 0 to indicate the first pass is complete.

If IS = 40 the wall is correctly adjusted.

Move the walls using Subroutine START.

If IS less than 40 repeat the wall adjustment cycle

If the wall has been correctly adjusted to the demanded contour (IP = 0) finish. If the wall has been correctly adjusted to the overshoot contour (IP = 5) repeat the wall adjustment cycle with IP = 0.

Set IW = 0 for top wall condition.

For all downstream jacks (Nos. 16-20 and 36-40) warn the operator when their pot. limit has been reached.

Subroutine INIT

NWJ = 40 = Number of wall jacks.

Set data operation as write before read IDI = 128.
Line 0004 - 0010
Generate command numbers ICOM for a motor stop directive to each jack and send them individually to the wind tunnel using Subroutine D10.

Line 0007
Check the data has been correctly loaded.

Subroutine START

Line 0001
Subroutine statement with data transfer of the demanded jack pot. volts (PV) and the required movements or position errors (IM).

Line 0004 - 0006
Determine the average position error of the bottom wall in pot. volts and store the result in IAM.

Line 0007
RCT = number of clock counts between motor power on and off.

Line 0008
The upper limit of RCT is 600. Set by wall damage considerations.

Line 0010
The lower limit on RCT is 25. Set by resolution of the wall position measurements and speed considerations.

Line 0012 - 0014
If RCT > 50 load jacks 1 and 21 with a motor stop because they will overshoot by too large an amount.

Line 0016 - 0019
Generate a command number ICOM to start the motors (See Appendix A) and send it to the wind tunnel using Subroutine D10.

Line 0020 - 0022
Variable time delay set by number of clock counts (RCT) at 100 Hz.

Line 0023
Stop all motors

Subroutine MOVE

Line 0001
Subroutine statement with data transfer of IJ (Jack number) and IMV (required jack movement).

Line 0002
Determine whether IMV is negative, positive or zero.
Subroutine DIO

This subroutine performs the handshaking between computer and wind tunnel to transfer data on the digital I/O lines.

Line 0001 Subroutine statement with data transfer ICOM(Command number), IDI (Data operation) and INPUT (Data check)

Line 0002 Load Digital output buffer (address 167774) with command number 45 to ensure all motors are switched off.

Line 0003 Send command number ICOM to the wind tunnel.

Line 0004 Halt program until an external data accept signal is received on the digital I/O status register (handshaking between wind tunnel & computer).

Line 0006 Reset the Digital I/O status register to zero.

Line 0007 If IDI = 0 (Write only operation) finish.

Line 0009 Halt the program until an external data ready pulse is received on the digital I/O status register.

Line 0011 Store the contents of the Digital input register in INPUT.

Line 0012 Initialise the status register.

3.10 Program REAN

The main re-analysis program listed on Figure 4.10 reads test parameters and sequences subroutine calls. The program is very similar to TSWT, the main differences are:
1. The printout trigger is set to give an extended print-out (i.e. 
   CL1 = 500).

2. All test parameters and run details are provided by the operator.

3. The program contains no loops and will run through only once per 
   activation.

4. Subroutine WALL is not included.

5. Subroutine STAR is included to provide additional information during 
   the re-analysis.

3.11 Program RUN

This program has to be activated before each session of TSWT runs, to ensure 
the data file RUN.DAT contains the current ambient conditions, run and file 
numbers.

The software is shown on Figure 4.11 and discussed below.

Line 0002  Define array TUN to hold run data.
Lines 0003 - 0004  Define File 2 as data file RUN.DAT.
Line 0005  Load array TUN with run data stored in RUN.DAT, record 
  1.
Lines 0007 - 0009  Indicate if only an increment of the record number is 
   required (Yes: INC = 1).
Lines 0010 - 0011  Read thermocouple A-P channel and load TUN (3) with 
   the current ambient temperature.
Line 0012  If INC = 1 jump to line 0025.
Line 0014 - 0021  Input ambient pressure (cm Hg), the run number and the 
   file number as requested.
Lines 0022 - 0023  Load TUN (1) & TUN (2) with new values.
Line 0025  Increment the record number by one.
Line 0026  Write the modified contents of array TUN to RUN.DAT 
   on disc.
Lines 0027 - 0028  Print the contents of array TUN as a check.
\textbf{SYMBOLS}

\begin{itemize}
  \item $V$ = wall velocity perturbation ($V = U - U_\infty$).
  \item $U$ = local wall velocity.
  \item $U_\infty$ = freestream velocity.
  \item $\delta^*$ = boundary layer displacement thickness.
  \item $x$ = horizontal distance from the boundary layer origin (Inches).
  \item $R_X$ = Reynolds number based on boundary layer length ($x$).
  \item $Y$ = vertical distance from the tunnel centreline (positive upwards).
  \item $C_L$ = lift coefficient.
  \item $C_D$ = pressure drag coefficient.
  \item $C_m$ = pitching moment coefficient about the airfoil leading edge.
  \item $C_N$ = normal force coefficient.
  \item $C_C$ = chordwise force coefficient.
  \item $M$ = Freestream Mach number.
  \item $X$ = horizontal distance from leading edge (Inches).
  \item $Y$ = vertical distance from the chord line (Inches).
\end{itemize}
5. REFERENCES

1. M.J. Goodyer and S.W.D. Wolf
   'The Development of a Self-Streamlining Flexible Walled Transonic Test Section',

2. S.W.D. Wolf and M.J. Goodyer
   'Self-Streamlining Wind Tunnel-Low Speed Testing and Transonic Test Section Design',

3. M. Judd, S.W.D. Wolf and M.J. Goodyer
   'Analytical Work in Support of the Design and Operation of Two-Dimensional Self-
FIG. 1 SELF-STREAMLINING OPERATING PROCEDURE
FIG. 2  TSWT CONTROL SYSTEM OUTLINE
FIG. 3 T.S.W.T. CONTROL SYSTEM HARDWARE
Figure 4 Listings of TSWT Control Software
SUBROUTINE WAS

COMMON/ONE/NJ, MT, NR, CL1, B1, PR1, AK1, AK3, AN, R3, PP2, ITRN
COMMON/TWO/RN(30), RS(30), W(30), X(30), P(30), D(30), RD(20)
COMMON/THREE/D(60), WTY(30), WBY(30), WL(30), B(50), PD(40,4)
COMMON/FOUR/E(30), H(30), Y(30), G(30), WI(30), XI(30)
COMMON/FIVE/U(30), V(30), DS(60), DET(60), CS(20)
DIMENSION A(4), XB(4), C(30,4), Z(30), S(30), T(30)
REAL N

WRITE(7,5)
FORMAT('WAS COMPUTING NOW !!', 1115X)
AK2 = AK1
AK4 = AK3
NJ1 = NJ-4
MM = 1
M = NJ
DO 100 I = 1, NJ
IF(MM.EQ.NR) GO TO 10
NT = (I-5)/6
NM = NT * 6
IF((I-5).EQ.NT) MM = MM + 1
Q1 = (RD(MM) - RD(NR + 1))/PP2
TEMP = -P(I)/Q1
TEMP1 = B1*TEMP
TEMP = SORT(1-TEMP1)
U(I) = TEMP-X(I)
E(I) = (AK3*U(I)/2)+X(I)
TEP = -0(I)/Q1
TEP1 = B1*TEP
TEMP = (SORT(1-TEMP1)
V(I) = W(I)-TEP
H(I) = W(I)-(AK4*V(I)/2)
WI(I) = (TEP+l)*(TEP+l)
XI(I) = (TEMP+l)*(TEMP+l)
CONTINUE
L = M-2
DO 110 I = 1, L
Z(I) = (D(I)+D(I+1))/2
DO 175 NN = 1, 2
NC = NN-l
DO 120 J = 1, 4
KI = I+J
XB(J) = V(I+J)
IF(NC.EQ.0) GO TO 25
XB(J) = U(I+J)
CONTINUE
VO = (XB(3)-XB(2))/(A(3)-A(2))
V1 = XB(2)-VO*A(2)
V2 = 1/(A(4)-A(1))
V3 = (XB(4)-VO*A(4)-V1)/((A(4)-A(2))*(A(3)-A(4)))
CALL REDUCE
CALL WAS
CALL SUME
IF(MT.EQ.0) GO TO 100
CALL FORCE
100 CALL SET
IF(PR1.EQ.1.) GO TO 200
CALL WALL
200 IAM=1
IF(IFN.EQ.50) WRITE(7,400)
400 FORMAT(/60('#')/' REMEMBER TO COPY DATA FILES BEFORE NEXT RUN'
C /60('#'))
IF(IFN.EQ.50) GO TO 250
IFN=IFN+1
IF(PR1.EQ.0.) GO TO 500
250 TUN(1)=ITRN+1
TUN(2)=IFN
IFN=IFN+1
TUN(1)=ITRN
TUN(2)=4.
510 CALL ASSIGN(4,'RUN.DAT',0,'OLD',)
DEFINE FILE 4 (6r256,u,ijr)
WRITE(41)(TUN(N),N=1,6)
CALL CLOSE(4)
END

Figure 4.1.2
SUBROUTINE DATA

AUTO SCAN

DIMENSION DA(100),DB(100),DC(100),SD(4)
DIMENSION IDATA(49,4)
COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RSC30),WC30),XC30),PSC30),O(40),RDC20)
COMMON/THREE/D(60),WTY(30),WBY(30),WL(30),B(50),PB(48,4)
COMMON/FOUR/E(30),H(30),G(30),W(30),XI(30)
COMMON/FIVE/U(30),V(30),DSC60),DETC60),CSC20)
COMMON/SEVEN/PCC60)
COMMON/EIGHT/TAB,AMP,IFN,IAM
COMMON/NINE/I(1,4)
DATA IDATA(1,1),IDATA(1,2),IDATA(1,3),IDATA(1,4)/0,0,0,0/
DATA ACQUISITION

CHD = 4.
NJ1 = NJ-4
M = NJ
NR = 8
IP1 = 2050
SD(1) = .0331668
SD(2) = .0142486
SD(3) = .0134168
SD(4) = .0141158
IP0=0
IF(C11, EQ, 0) GO TO 36
WRITE(5,31)
FORMAT('DATA INPUT FILE = ')
CALL ASSIGN (4,-1,'OLD',)
DEFINE FILE 4 (50,512,U,IJR)
WRITE(7,32) IFN
FORMAT(40('*')/' ITERATION RECORD NO. = ','I4/40('*'))
READ(4,'IFN')((IDATA(N,J),N=1,49),J=1,4)
CALL CLOSE(4)
GO TO 44
IDATA(1,1)=IX(1)
IDATA(1,2)=IX(2)
IDATA(1,3)=IX(3)
IDATA(1,4)=IX(4)
IF(C11, EQ, 0) AND, X(30).EQ.1.)WRITE(5,800)
FORMAT(' SCANIVALVE GO'/13('='
DO 25 N=1,19
IF(N.GT.1) GO TO 40
IF(X(30).EQ.1.)GO TO 25
CALL GAP(N,IDATA)
WRITE(5,15)
FORMAT('TURN ON WIND')
READ(5,35) ISTOP
GO TO 25
FORMAT(13=''))
DO 25 N=1,19
IF(N.GT.1) GO TO 40
IF(X(30).EQ.1.)GO TO 25
CALL GAP(N,IDATA)
WRITE(5,15)
FORMAT('TURN ON WIND')
READ(5,35) ISTOP
GO TO 25
FORMAT(13=''))

Figure 4.2.1.
CALL STEP
IENC=CVSWG(IADC(20,1),1)
IF2=IABS(IENC-IP1)
IF1 = IENC
IF(IP2.GT.200) GO TO 30
WRITE(7,43)
FORMAT('STEP ERROR')
GO TO 180
CALL GADeN,IDATA)
CONTINUE
IX(1)=IDATA(1,1)
IX(2)=IDATA(1,2)
IX(3)=IDATA(1,3)
IX(4)=IDATA(1,4)
CALL ASSIGN(4,'ADC.DAT',0,'OLD',)
DEFINE FILE 4(50,512,U,1JR)
WRITE(7,42) IFN
IFN
.ITERATION RECORD NO. = ',I4)
WRITE(4'IFN)(IDATA(N,J),N=1,49),J=1,4)
CALL CLOSE(4)
C
DATA REDUCTION
C
DO 45 J = 1,4
DO 5 N = 2,49
IDATA(N,J)= IDATA(1,J)-IDATA(N,J)
NN = N-1
PD(NN,J)= SO(J)*IDATA(N,J)
CONTINUE
DO 110 N = 1,2
P(N) = 0.0
Q(N) = 0.0
CONTINUE
DO 115 N = 22,24
P(N) = PD(19,2)
Q(N) = PD(19,4)
CONTINUE
DO 120 N = 3,21
IA = N-2
P(N) = PD(IA,2)
Q(N) = PD(IA,4)
CONTINUE
IT=21
DO 210 N=25,33
IT=IT+1
IF(N.EQ.28) IT=IT+1
P(N)=PD(IT,2)
Q(N)=PD(IT,4)
CONTINUE
RD(NR+1) = PD(48,1)
RD(NR+2) = AMP
RD(NR+3) = TAB
AB1 = 0.

Figure 4.2.2.
0106  R1 = RD(NR+2) - RD(NR+1)
0107  DO 55 J = 1,NR
0108  ITR = 1 + ((J-1)*6)
0109  RD(J) = PD(ITR,1)
0110  AD1 = AD1 + RD(J)
0111  R2 = RD(NR+2) - RD(J)
0112  R3 = 0.28571*ALOG(R1/R2)
0113  PP1 = 5.0 * (EXP(R3)-1)
0114  TFM = SORT(PP1)
0115  IF(J.EQ.1) TFM1=TFM
0117  TR = RD(J)/2.54
0118  IF(J.EQ.1) GO TO 55
0120  IF(ABS(TFM-TFM1).GT.0.01) WRITE(7,56)
0122  56  FORMAT(/' MACH NO. ERROR'/20('*'))
0123  55  CONTINUE
0124  H1 = PD(1,2) + RD(1)
0125  H2 = PD(43,2) + RD(6)
0126  H3 = PD(20,2) + RD(4)
0127  H4 = PD(1,4) + RD(1)
0128  H5 = PD(43,4) + RD(6)
0129  H6 = PD(20,4) + RD(4)
0130  H7 = PD(2,1)
0131  H1 = H1/2.54
0132  H2 = H2/2.54
0133  H3 = H3/2.54
0134  H4 = H4/2.54
0135  H5 = H5/2.54
0136  H6 = H6/2.54
0137  H7 = H7/2.54
0138  AS = AD1/NR
0139  TR = AS/2.54
0140  SP = RD(NR+1)/2.54
0141  BI=AS
0142  180  RETURN
0143  END

DATA

Figure 4.2.3.
SUBROUTINE GAD (N, IDATA)
DIMENSION IDATA(49,4), IDT(500)
NS=15
IEND = 0
CALL SETR(4,0,50,,IEND)
CALL LWAIT(0,IEND)
IEND = 0
IRTS=0
CALL RTS(IDT,60,,NS,16,4,,2,IRTS,IBUM)
CALL SETR(4,9,1,.,IEND)
CALL LWAIT(0,IRTS)
DO 5 J = 1,4
DO 70 NN = 1,NS
IS = ((NN-1)*4)+J
IDT(IS) = CVSG(IDT(IS),1)
SUM = SUM + IDT(IS)
CONTINUE
70 IDATA(N,J) = SUM/NS
CONTINUE
5 CALL SETR(-1,,)
CALL RTS(IDT,60,,NS,16,4,,2,IRTS,IBUM)
CALL SETR(4,9,1,.,IEND)
CALL LWAIT(0,IRTS)
DO 5 J = 1,4
DO 70 NN = 1,NS
IS = ((NN-1)*4)+J
IDT(IS) = CVSG(IDT(IS),1)
SUM = SUM + IDT(IS)
CONTINUE
70 IDATA(N,J) = SUM/NS
CONTINUE
5 CALL SETR(-1,,)
RETURN
END
GAD

SUBROUTINE STEP
CALL IPOKE("167774,16384)
IEND=0
CALL SETR(4,0,20,,IEND)
CALL LWAIT(0,IEND)
CALL IPOKE("167774,0)
RETURN
END
STEP
SUBROUTINE REDUCE

DIMENSION DA(100), DB(100), DC(100), SD(4)

DIMENSION IDATA(49, 4)

COMMON/ONE/NJ, NT, NR, CL1, B1, PR1, AK1, AK3, AN, R3, PP2, ITRN

COMMON/TWO/RD(30), W(30), X(30), P(40), R(40), RD(20)

COMMON/THREE/O(60), WTY(30), WBY(30), UM(30), B(50), PD(48, 4)

COMMON/FOUR/E(30), H(30), Y(30), G(30), WI(30), XI(30)

COMMON/FIVE/U(30), V(30), D(60), DET(60), CS(20)

COMMON/SEVEN/PC(50)

COMMON/EIGHT/TAB, AMP, IFN, IAM

DATA ACQUISITION

AS = B1
CHD = 4.
NJ1 = NJ - 4
M = NJ
NR = 8
MM = 1
R1 = RD(NR+2) - RD(NR+1)
R2 = RD(NR+2) - AS
R3 = 0.28571 * ALOG(R1/R2)
PP1 = 5.0 * (EXP(R3) - 1)
AM1 = SQRT(PP1)
B1 = SQRT(1 - PP1)
WRITE(5, 90) AM1
WRITE(7, 90) B1
FORMAT(7, 90) AM1

FORHAT(// ', MACH NO. = ', F8.4)
PP2 = 1.0 + (0.25 * PP1) + (0.025 * PP1 * PP1)
CC1 = 1 + (0.2 * PP1)
CC2 = 2.5 * ALOG(CC1)
CC3 = EXP(CC2)
D1 = RD(NR+2) * 8.998E-3 / (273.15 + RD(NR+3))
D1 = D1 / CC3
AT = RD(NR+3) + 273.15
AT = AT / CC1
CC4 = 0.92 * ALOG(AT / 273.15)
V1 = 11.52 * EXP(CC4)
PP3 = (AS - RD(NR+1)) / PP2
U0 = SQRT(56.369 * PP3 / D1)
R2 = U0 * D1 * 3218E4 / V1
R3 = R2 * CHD / 12.
WRITE(7, 95) R3

FORMAT(7, 95) R3

DO 60 J = 2, 4

DO 65 N = 1, 48
IF(NM.EQ.NR) GO TO 170
NT = (N - 5) / 6
R2 = (RD(NM) - RD(NR+1)) / PP2
PD(N, J) = -PD(N, J) / Q1
CONTINUE

Figure 4.3.1.
CONTINUE
DO 500 J=16,24
IJ=J+8
JB=J+29
IJB=J+22
B(IJ)=PD(J,3)
IF(J.GT.21) GO TO 500
B(JB)=PD(IJB,3)
CONTINUE
MM=1
NP=1
DO 600 K=2,42
NT=(K-5)/6
NT=NT*6
NJ=(K-7)/6
NJ=NJ*6
IF((K-5).EQ.MM) MM=MM+1
IF((K-7).EQ.NJ) GO TO 600
Q1=(RD(MH)-RD(NR))/PP2
B(NP)=(RD(MH)-PD(K,1))/Q1
NP=NP+1
IF(K.EQ.25) NP=33
CONTINUE
CALL ASSIGN(2,'TUN.DAT',0,'OLD',)
DEFINE FILE 2 C12,256,U,IJR)
IF(CL1.EQ.0.) IWF=1
IF(CL1.EQ.0.) GO TO 77
WRITE(7,75)
FORMAT('FILE NO. ?/ 1 FOR 725/ 2 FOR 725<625')
WRITE(7,76)
FORMAT('3 FOR M=>.25 ANS = ')}
READ(7,35) IWF
READ(2,12) (DA(J),J=1,80)
READ(2,11) (DB(J),J=1,33)
NJ2 = NJ1 + NJ1
READ(2,12) (PC(J),J=1,NJ2)
CALL ASSIGN(3,'PAD.DAT',0,'OLD',)
DEFINE FILE 3 C50,512,U,INR)
IF(IAM.EQ.1) GO TO 20
WRITE(5,140)
FORMAT('/S WALL RECORD? = ')}
READ(5,35) IR
IF(CL1.EQ.0.) GO TO 41
GO TO 40
IR=IFN-1
WRITE(7,30)
FORMAT('WALL CONTOURS RECORD = ',I4)
READ(3,14) (DC(J),J=1,80)
DO 150 K = 1,80
111 IF(K.LT.20) UTY(K) = DA(K)
112 IF(K.LE.40.AND.K.GT.20) WBY(K-20) = DA(K)
113 IF(K.GT.40) D5CK-40) = DACK)
CONTINUE
Figure 4.3.2.
DO 155 J = 1, 33
IF(J.LE.9) CS(J) = DB(J)
IF(J.GT.9) D(J-9) = DB(J)
CONTINUE
READ(2, 10) AK1, AK3
DO 156 J = 1, NJ1
NJJ = J + 2
WL(J) = (D(NJJ+1)-D(NJJ-1))/2
CONTINUE
DO 160 K = 1, 88
IF(K.LE.20) RS(K) = DC(K)
IF(K.GT.20 . AND. K.LE.40) RN(K-20) = DC(K)
IF(K.GT.40 . AND. K.LE.64) X(K-40) = DC(K)
IF(K.GT.64) W(K-64) = DC(K)
CONTINUE
DO 175 J = 1, NJ1
WTY(J) = WTY(J) - RS(J)
WTY(J) = WTY(J) / PC(J)
WBY(J) = RN(J) - WBY(J)
WBY(J) = WBY(J) / PC(J+NJ1)
CONTINUE
CALL CLOSE(3)
CALL CLOSE(2)
X(30) = 1.
RETURN
END
SUBROUTINE WAS

COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(30),Q(30),RD(20)
COMMON/THREE/D(60),WTY(30),WBY(30),WL(30),B(50),PD(48,4)
COMMON/FOUR/E(30),V(30),DS(60),DET(60),CS(20)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
DIMENSION A(4),XB(4),C(30,4),ZC(30),S(30),T(30)
REAL N
WRITE(7,5)
FORMAT(1X,'WAS COMPUTING NOW!!')

AK2 = AK1
AK4 = AK3
NJ1 = NJ-4
MM = 1
M = NJ
DO 100 I = 1,NJ
IF(MM.EQ.NR) GO TO 10
NT = (I-5)/6
IF(I-NT) MM = MM +1
01

TEMP = -P(I)/Q1
TEMP1 = B1*TEMP
TEMP = SQRT(1-TEMP1)
TEMP = TEMP-1
U(I) = TEMP-X(I)
E(I) = (AK3*U(I)/2)+X(I)
TEP = -Q(I)/Q1
TEP1 = B1*TEP
TEP = SQRT(1-TEP1)
TEP = TEP-1
V(I) = W(I)-TEP1
H(I) = W(I)-(AK4*X(I)/2)
W(I) = W(I)*X(I)
XI(I) = (TEMP+1)*(TEMP+1)
CONTINUE
L = M-2
DO 110 I = 1,L
Z(I) = (D(I)+D(I+1))/2
DO 175 NN = 1,2
NC = NN-1
DO 120 J = 1,4
KI = I+J
A(J) = D(I+J)
IF(NC.EQ.0) GO TO 25
XB(J) = V(I+J)
GO TO 120
XB(J) = U(I+J)
CONTINUE
V0 = (XB(3)-XB(2))/(A(3)-A(2))
V1 = XB(2)-V0*A(2)
V2 = 1/(A(4)-A(1))
V3 = (XB(4)-V0*A(4)-V1)/(A(4)-A(2))*A(3)-A(4))

Figure 4.4.1.
\[ V_4 = \frac{X_B(1) - V_0 A(1) - V_1}{(A(1) - A(2)) A(3) - A(1))} \]
\[ V_6 = V_2 (V_3 - V_4) \]
\[ V_5 = V_4 - V_6 A(1) \]
\[ I = I + 1 \]
\[ P_1 = A(2) + A(3) \]
\[ C(I, 1) = V_1 - A(2) A(3) V_5 \]
\[ C(I, 2) = V_0 + V_5 P_1 - V_6 A(2) A(3) \]
\[ C(I, 3) = V_6 P_1 - V_5 \]
\[ C(I, 4) = -V_6 \]
\[ \text{IF } (I, I.T, (M-3)) \text{ GO TO 35} \]
\[ L_1 = N - 2 \]
\[ D_0 = 130 \]
\[ J = 2, L_1 \]
\[ Z_0 = Z(J) \]
\[ Z_02 = Z_0 A_0 \]
\[ Z_03 = Z_02 A_0 \]
\[ S_M = 0 \]
\[ K = N - 3 \]
\[ D_0 = 140 \]
\[ I = 1, K \]
\[ Y_1 = D(I + 1) \]
\[ C_0 = C(I, 1) \]
\[ C_1 = C(I, 2) \]
\[ C_2 = C(I, 3) \]
\[ C_3 = C(I, 4) \]
\[ Y_2 = D(I + 2) \]
\[ Y_2 S_0 = Y_2 * Y_2 \]
\[ Y_1 S_0 = Y_1 * Y_1 \]
\[ S_0 = C_0 C_1 X 0 + C_2 (Z_02) + C_3 (Z_03) \]
\[ \text{TEMP} = \frac{A B S(Y_2 - Z_0)}{A B S(Y_1 - Z_0)} \]
\[ S_1 = A \log (\text{TEMP}) \]
\[ S_2 = (C_1 + C_2 Z_0 + C_3 Z_02) (Y_2 - Y_1) \]
\[ S_3 = (C_2 + C_3 Z_0) ((Y_2 S_0) - (Y_1 S_0)) / 2 \]
\[ S_4 = C_3 ((Y_2 S_0 - Y_2) - (Y_1 S_0 - Y_1)) / 3 \]
\[ S_M = S_M + S_0 S_1 + S_2 S_3 + S_4 \]
\[ \text{CONTINUE} \]
\[ \text{IF } (N.C. \text{EQ.} 1) \text{ GO TO 45} \]
\[ S(J) = S_M / 6.28319 \]
\[ \text{GO TO 130} \]
\[ T(J) = S_M / 6.28319 \]
\[ \text{CONTINUE} \]
\[ \text{CONTINUE} \]
\[ R = 0 \]
\[ T_1 = 0 \]
\[ S(I) = 0.0 \]
\[ T(I) = 0.0 \]
\[ D_0 = 150 \]
\[ I = 1 + N J_1 \]
\[ T_0 = S(I) \]
\[ R_0 = T(I) \]
\[ T_1 = Z(I) \]
\[ I_1 = I + 1 \]
\[ T_2 = D(I) \]
\[ T_3 = S(I) \]
\[ R_3 = T(I) \]
\[ T_4 = Z(I) \]
\[ I_2 = I + 2 \]
0114 T5 = D(I2)
0115 T6 = S(I2)
0116 R6 = T(I2)
0117 T7 = Z(I2)
0118 FS1 = (T6-T3)/(T7-T4)
0119 FS2 = (R6-R3)/(T7-T4)
0120 T2SQ = T2*T2
0121 T5SQ = T5*T5
0122 T8 = (FS1-(T0-T3)/(T1-T4))/(T7-T1)
0123 T9 = FS1 - T8 * T7
0124 P2 = (T3-T9*T4)*(T5-T2)+T8*((T5SQ*T5)-(T2SQ*T2))/3
0125 TT = TT+P2+(T9-T8*T4)*((T5SQ)-(T2SQ))/2
0126 R8 = (FS2-(R0-R3)/(T1-T4))/(T7-T1)
0127 R9 = FS2 - R8 * T7
0128 P3 = (R3-R9*T4)*(T5-T2)+R8*((T5SQ*T5)-(T2SQ*T2))/3
0129 F = E(I2)
0130 R = R+P3+(R9-R8*T4)*((T5SQ)-(T2SQ))/2
0131 Y(I2) = (AK3*TT)+(AK2*AK4*R)
0132 G(I2) = (AK4*R)+(AK1*AK3*TT)
0133 E(I2) = E(I2) + ((H(I2)-W(I2)+V(I2))*AK2)
0134 H(I2) = H(I2) +((F-U(I2)-X(I2))*AK1)
0135 Y(I2) = B1 * Y(I2)
0136 G(I2) = B1 * G(I2)
0137 150 CONTINUE
0138 RETURN
0139 END

Figure 4.4.3.
SUBROUTINE STAR

DIMENSION DELTA(60)

COMMON/ONE/NJ,MT, NR, CL1, B1, PR1, AK1, AK3, AN, R3, PP2, ITRN

COMMON/TWO/RN(30), RS(30), W(30), X(30), P(30), Q(30), RD(30)

COMMON/THREE/D(60), WTY(30), WBY(30), WL(30), B(50)

COMMON/FOUR/E(30), H(30), Y(30), G(30), WI(30), XI(30)

COMMON/FIVE/U(30), V(30), DS(60), DET(60), CS(20)

PP1 = 1 - (B1*B1)

CC1 = 1 + (0.2*PP1)

CC2 = 2.5 * ALOG(CC1)

CC3 = EXP(CC2)

CC4 = 2.5 * ALOG(CC3)

D1 = RD(NR+2)*8.998E-3/(273.15+RD(NR+3))

D1 = D1/CC3

AT = RD(NR+3)+273.15

AT = AT/CC1

CC4 = 0.92*ALOG(AT/273.15)

V1 = 11.52*EXP(CC4)

IF(CL1.EQ.0.) GO TO 56

WRITE(7,S)

FORMAT('///SX, 'DELTA STAR CALCS.'//')

WRITE(7,10)

FORMAT(2X,'TAP NO.',2X,'DU/DX',5X,'MACH NO.',6X,'D*',9X' DD*')

NJ2 = NJ * 2

RR1 = RD(NR+2) - RD(NR+1)

MM = 1

DO 2 N = 2,NJ2

IF(N.GT.NJ) NN = N-NJ

IF(N.GT.(NJ-2)) GO TO 2

IF(CL1.EQ.500 •• AND.N.EQ.(NJ+1)) WRITE(7,11)

IF(N.EQ.(NJ+1)) MM = MM+1

IF(N.EQ.NJ.OR.N.EQ.NJ2) GO TO 12

X1 = D(NN) - D(NN-1)

X2 = D(NN+1) - D(NN-1)

IF(MM.EQ.NT) GO TO 120

NTS = N-5

NT = NTS/6

NT = NT/6

IF(N.EQ.NT) MM = MM+1

IF(N.GT.NJ) GO TO 110

SP1 = P(N)

SP2 = P(N+1)

SP3 = P(N-1)

GO TO 100

IF(N.GT.NJ) GO TO 110

SP1 = Q(NN)

SP2 = Q(NN+1)

SP3 = Q(NN-1)

PP3 = (SP1+RD(MM)-RD(NR+1))/PP2

PP4 = (SP2+RD(MM)-RD(NR+1))/PP2

PP5 = (SP3+RD(MM)-RD(NR+1))/PP2

Figure 4.5.1.
U1 = SQRT(56.369*PP3/D1)
U2 = SQRT(56.369*PP4/D1)
U0 = SQRT(56.369*PP5/D1)

LOCAL MACH NO. CALCS

R2 = RD(NR+2) - (SP1+RD(MM))
R3 = 0.28571*ALOG(RR1/R2)
PP1 = 5.0*(EXP(R3)-1)
AM1 = SQRT(PP1)
IF(N.GT.NJ) Q(NN-1)=AM1
IF(N.GT.NJ) GO TO 23
P(NN-1)=AM1

R2 = U0*D1*32.18E6/V1
R1 = (R2*X1)/12
C1 = 0.142857*ALOG(R1)
C2 = EXP(C1)
C3 = 0.00127 * X1
P6 = C3/C2
GO TO 133

Y1 = U1 - U0
Y2 = U2 - U0
A1 = (Y2-(X2*Y1)/X1)/(X2*X2)-(X1*X2))
B2 = (Y1-(A1*(X1*X1))/X1
D2 = (2*A1*X1) + B2

DO 3
Mi = 1,1000
PI = (25 + M1*1000)/1E6
NODE = 1

P2 = (U1*P1*D1*32.18E6)/V1
C4 = 0.25 * ALOG(P2)
C5 = EXP(C4)
P3 = 0.0128/C5
IF (N.EQ.NJ.OR.N.EQ.(NJ+2)) GO TO 66
S1 = P3-(3.4*P1*D2*12/U1)
IF (N.EQ.3.0R.N.EQ.(NJ+3)) GO TO 77
GO TO 88

S1 = P3-(3.4*P1*D3*12/U1)
P4 = P5+(0.5*(S1+S2)*X1/12)
GO TO 99

P4 = (S1*X1/12) + P6
GO TO 99

P4 = P5+(0.5*(S1+S2)*X1/12)
IF (NODE.EQ.2) GO TO 155
IF (P4.LT.PI) GO TO 111
CONTINUE

IF (P4.GT.PI) GO TO 144
GO TO 144

P1 = P1 - 25E-6

DELTA(N) = 16.8 *PI
N2 = N-2

Figure 4.5.2.
IF(N.GT.NJ) N2 = N-6
DET(N2) = DELTA(N) - DS(N2)
NNJ = NN-2
IF(CL1.EQ.0.) GO TO 36
WRITE(5,25) NNJ,D2,AM1,DELTA(N),DET(N2)
FORMAT(5X,I2,4F12.4)
IF(N.EQ.(NJ2-2)) WRITE(5,35) REY,FPD
FORMAT('UNIT REYNOLDS
NO. = ',F8.1,' D* FPG = ',F8.4/)
P5 = P4
S2 = S1
IF (J-(NJ-1)) 2,44,55
IF (J-(NJ2-1)) 2,44,2
D3 = D2
GO TO 2
DELTA(N) = P6 * 16.8
REY = R2/12
FPD = DELTA(N)
CONTINUE
C WALL MACH NOS BETWEEN JACKS
C
IWALL=0
DO 300 J=25,33
SP1=P(J)
IF(IWALL.NE.0) SP1=Q(J)
R2=RD(NR+2)-(SP1+RD(5))
R3=0.28571*ALOG(RR1/R2)
PP1=5.0*CEXP(R3)-1)
IF(PP1.LE.0) GO TO 300
AM1=SQRT(PP1)
IF(IWALL.EQ.0) P(J)=AM1
IF(IWALL.NE.0) Q(J)=AM1
CONTINUE
IWALL=IWALL+1
IF(IWALL.EQ.1) GO TO 350
RETURN
END
SUBROUTINE SUME
COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(40),Q(40),RD(20)
COMMON/THREE/D(60),UTY(30),WB(30),WL(30),B(50),PD(40,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
DIMENSION UT(37),VT(37)
IWT=1
TOPI = 6.283185
CHD = 4.
IF(IWT.EQ.1) CHD=6.0
OR=24.56
AN1 = AN*.01745
AX = (CHD*COS(AN1)/8.0
AH = (CHD*SIN(AN1)/8.0
IPP = 7
IF(IWT.EQ.1) IPP = 4.33
OR = OR - ((IPP-3)*AX)
DT = 0
TOT = 0.0
NJ1 = NJ-4
Y4 = 0.0
Y3 = 3.0
MN = 0.0
EE = 0.0
F = 0.0
IJ = NJ-2
DO 160 I = 3,IJ
VI = W(I)+1
V2 = X(I)+1
WI = V1 * V1
X(I) = V2 * V2
EE = EE + ABS(XI(I)-X(I))
F = F + ABS(WI(I)-W(I))
CONTINUE
CET = EE/NJ1
CEB = F/NJ1
IF(CL1.EQ.0.)WRITE(5,170) CET,CEB,
WRITE(7,170) CET,CEB
FORMAT(I0X, WALL CP ERROR'/S X, TOP - 'F8.4,
5X, BOTTOM - 'F8.4)
DO 161 I = 1,37
X1 = OR + ((I-19)*1.0)
SUMU = 0.0
SUMV = 0.0
DO 162 J = 1,NJ1
X2 = X1 - (D(J+2))
Y1 = Y4+Y3-UTY(J)-DET(J)
Y2 = Y4-Y3-WBY(J)+DET(J)

Figure 4.6.1.
P1 = X2 * X2
P2 = P1 + (Y1*Y1)
P3 = P1 + (Y2*Y2)
R1 = X2/P2
R2 = Y1/P2
R3 = Y2/P3
U1 = (U(J+2)*R2)+(V(J+2)*R3)
U1 = (U1*WL(J))/TOPI
V1 = (U(J+2)+V(J+2))*R1
V1 = (V1*WL(J))/TOPI
SUMU = SUMU + U1
SUMV = SUMV + V1
CONTINUE
IF (MN.GT.0) GO TO 98
UT(I) = SUMU
VT(I) = SUMV
X3 = (I-19) * 1.0
IF (OT.GT.0) GO TO 161
WRITE(7,15) X3,UT(I),VT(I)
FORMAT(3F12.4)
CONTINUE
WRITE (7,20)
FORMAT(/5X,' MODEL ERRORS',25X,' CP'/)
SO = 0.0
SE = 0.0
SM = 0.0
DO 164 JJ = 1,9
X1 = OR - ((3-JJ)*AX)
Y4 = -(IPP-JJ)*AH
MN = 1
X3 = (1-19) * 1.0
IF (OT.GT.0) GO TO 161
WRITE(7,25)
X1,SUMU,SUMV,CP1
FORMAT(4F12.4)
IF (JJ.EQ.1) A1 = AA2
IF (JJ.EQ.9) A2 = AA2
AA2 = ATAN(SUMV/(1+SUMU))
CP1 = 1.0 - (((1+SUMU)*(1+SUMU))
TDT = TOT + CP1
IF (OT.GT.0) GO TO 104
X1 = X1 - OR
IF(CL1.EQ.0.) GO TO 104
WRITE(7,25) X1,SUMU,SUMV,CP1
FORMAT(4F12.4)
IF (JJ.EQ.1) A1 = AA2
IF (JJ.EQ.9) A2 = AA2
SP = AA2 * (1-CS(JJ))
MB = JJ/2
MB = MB*2
MB = MB*X2
IF (MB.EQ.JJ) GO TO 101
SO = SO + SP
GO TO 164
SE = SE + SP
GO TO 164
SM = SM + SP
CONTINUE
CPE = TOT/9.0

Figure 4.6.2.
P1 = SH(2*50)+(4*SE)
P2 = (AX/3)*P1
P3 = 2*P2
A3 = (A1-A2) * 59.29578
IF(CL1.NE.0) WRITE(7,35)
FORMAT(/10X,' EFFECT',25X,' DELTA CL')

CL = TOPI * A1
A1 = A1 * 59.29578
IF(CL1.EQ.0) GO TO 58
WRITE(7,30) A1,CL
FORMAT(/5X,' ALPHA ERROR = ',F8.4,' DEGREES',5X,F8.4)
WRITE(7,40) A3,P3
FORMAT(/5X,' INDUCED CAMBER = ',F8.4,' DEGREES',2X,F8.4)
WRITE(7,45) A3,P3
FORMAT(/5X,' INDUCED CAMBER = ',F8.4,' DEGREES',2X,F8.4)
WRITE(7,55) CP,E,-CP
FORMAT(5X,' AVERAGE',6X,'=',F8.4,13X,F8.4///)
IF(CL1.EQ.0.AND.PR1.EQ.1.) WRITE(5,57)
A1,A3,CPE
IF(CL1.EQ.0.AND.PR1.EQ.1.) WRITE(7,57)
A1,A3,CPE
FORMAT(' RESIDUALS = ',3F8.4)
ISC=0
IF(CET.LE.0.1)ISC=ISC+1
IF(CEB.LE.0.1)ISC=ISC+1
IF(ABS(A1).LE.0.15)ISC=ISC+1
IF(ABS(A3).LE.0.07)ISC=ISC+1
IF(ABS(CPE).LE.0.007)ISC=ISC+1
IF(ISC.EQ.5)PR1=1.

Figure 4.6.3.
SUBROUTINE FORCE
COMMON/ONE/NJ,M,P,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(30),Q(30),RD(20)
COMMON/THREE/B(60),WTY(30),WBY(30),WL(30),B(50),PD(48,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30)
DIMENSION A(50),BB(50),AJ(50),AG(50),AH(50)
REAL LS,L6,L7,J
INTEGER S
MT = MT+2
CALL ASSIGN(2,'WING.DAT',0,'OLD',,)
DEFINE FILE 2
READ(2'1) (AG(L),L=1,MT)
READ(2'2) (AJ(L),L=1,MT)
IHT = HT/2
WRITE(7,30)
FORMAT(///15X,' NACA SECTION ANALYSIS'/20X,' 0012-64'/
C /' RUN NO. = ')
WRITE(7,?)
WRITE(7,9)
FORMAT(/'$ ALPHA = '/)
WRITE(7,12)
FORMAT('+$F6.2)
DO 54 K = 2,(MT-1)
AH(K) = (AJ(K+1)-AJ(K-1))/2
AH(1) = AJ(2)/2
AH(IHT+1) = AJ(IHT+2)/2
AH(IHT) = (.0012 - AJ(IHT-1))/2
AH(MT) = (.0012 - AJ(MT-1))/2
L5 =0
DO 3 IQ = 1,MT
IF (IQ.EQ.1) GO TO 540
IF (IQ.EQ.IHT) GO TO 580
IF (IQ.EQ.(IHT+1)) GO TO 630
IF (IQ.EQ.MT) GO TO 530
IF (IQ.GT.(IHT+1)) GO TO 780
WF = (AG(IQ+1)-AG(IQ-1))/2
L6 = WF*B(IQ)
L5 = L5-L6
A(IQ) = -L6
GO TO 3
540 WF = AG(IQ+1)/2
L5 = -WF*B(IQ)
A(IQ) = L5
GO TO 3
580 WF = (1.0 - AG(IHT-1))/2
L6=WF*B(IQ)
L5=L5-L6
A(IQ)=-L6
CN = L5
GO TO 3
630 WF = AG(IQ+1)/2
L6 = WF*B(IQ)

Figure 4.7a.1.
0059     L5=L5+L6
0060     A(IQ) = L6
0061     GO TO 3
0062 530  WF = (1.0 - AG(MT-1))/2
0063     L6 = WF*B(MT)
0064     L5 =L5+L6
0065     A(MT)= L6
0066     CNL = L5-CN
0067     GO TO 3
0068 780  WF=(AG(IQ+1)-AG(IQ-1))/2
0069     L6 = WF*B(IQ)
0070     L5 = L5+L6
0071     A(IQ)= L6
0072 3  CONTINUE
0073 890  C3 = 0
0074     DO 40 IY = 1,MT
0075     IF (IY.GT.(IHT+1)) GO TO 980
0076     C5 = (B(IY)*AH(IY))
0077     AH(IY)=C5
0078     C3 = C3+C5
0079     IF (IY.EQ.IHT) GO TO 970
0080     GO TO 40
0081     DO 40 IY = 1,MT
0082     IF (IY.GT.(IHT+1)) GO TO 980
0083 970  CCP = C3
0084     GO TO 40
0085 980  C5=(B(IY)*AH(IY))
0086     AH(IY)= C5
0087     C3 = C3 + C5
0088 40  CONTINUE
0089     ST = C3-CCP
0090     IF(CL1.EQ.0.) GO TO 196
0091     WRITE (7,165)
0092 165  FORMAT(1111,10X,' UPPER SURFACE')
0093     WRITE (7,194)
0094 194  FORMAT(1111,10X,' LOWER SURFACE')
0095     WRITE(7,185)
0096 185  C9=0.
0097     CB=0.
0098     DO 5 S=1,MT
0099     IF (S.EQ.(IHT+1)) GO TO 1240
0100     IF (S.GT.(IHT+1)) GO TO 1270
0101     BB(S) = (-A(S)*AG(S))+(AH(S)*AJ(S))
0102     C9 = C9+BB(S)
0103     TIT = AG(S)*100
0104     IF(CL1.EQ.0.) GO TO 5
0105     WRITE(7,185) TIT,B(S),A(S),AH(S),BB(S)
0106     GO TO 5
0107 1240  IF(CL1.EQ.0.) GO TO 1270
0108     WRITE(7,195)
0109 195  FORMAT(1111,10X,' LOWER SURFACE')
0110     WRITE (7,194)
0111 194  FORMAT(1111,10X,' LOWER SURFACE')
0112     WRITE(7,185)
0113 185  C9=0.
0114     CB=0.
0115 1270  BB(S)=(A(S)*AG(S))+(AH(S)*AJ(S))
0116     CB = CB-BB(S)
0117     TIT = AG(S)*100
0118     IF(CL1.EQ.0.) GO TO 5

Figure 4.7a.2.
0120  WRITE (7,185) TIT,D(B(S),A(S),AH(S),-BB(S))
0121  185  FORMAT(4X,F4.1,4F12.4)
0122   5  CONTINUE
0123  ST1 = C9+C9
0124  WRITE(7,205)
0125  205  FORMAT(18X,'PRESSURE',2X,'SUCTION',2X,'TOTAL')
0126  WRITE(7,206)CN,CNL,LS
0127  206  FORMAT(12X,'CN',2X,3F9.4)
0128  WRITE(7,207)CCP,ST,C3
0129  207  FORMAT(12X,'CC',2X,3F9.4)
0130  WRITE(7,208)C9,CB,STI
0131  208  FORMAT(12X,'CH',2X,3F9.4)
0132  AT = AN * 0.017453
0133  CS1 = COS(AT)
0134  SN = SIN(AT)
0135  CL = (L5*CS1)-(C3*SN)
0136  CD = (C3*CS1) + (L5*SN)
0137  WRITE (7,225)
0138  225  FORMAT(18X,'WING PERFORMANCE'/18X,'CL',10X,'CD',10X,'CM')
0139  WRITE (7,235) CL,CD,STI
0140  235  FORMAT(10X,3F12.4)
0141  CALL CLOSE(2)
0142  END

FORCE

Figure 4.7a.3.
SUBROUTINE FORCE

COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN

COMMON/TWO/RN(30),RS(30),W(30),X(30),P(30),O(30),RD(20)

COMMON/THREE/D(60),WTY(30),WBY(30),WL(30),B(50),PD(48,4)

COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30),XI(30)

COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)

DIMENSION A(50),BB(50),AJ(50),AG(50),AH(50)

REAL L5,L6,L7,J

INTEGER S

CALL ASSIGN(2,'NPL.DAT',0,'OLD',,)

DEFINE FILE 2 (50,128,U,INR)

READ(2'1) (AG(L),L=1,MT)

READ(2'2) (AJ(L),L=1,MT)

IHT = 32

IF(CL1.EQ.0.) GO TO

WRITE(7,30)

FORMAT(11115X,'NPL SECTION ANALYSIS'/20X,' C

RUN NO. = ')

WRITE(7,7)

WRITE(7,9)

WRITE(7,12) AN

WRITE(7,15)

FORMAT('+',F6.2)

FORHAT('+',IS)

B(31)=(B(30)+B(32)/2.

DO

K = 2,(MT-1)

AH(K) = (AJ(K+1)-AJ(K-1))/2

AH(1) = AJ(2)/2

AH(IHT+1) = AJ(IHT+2)/2

AH(IHT) = (.0024)

AH(MT) = (.0024)

L5 = 0

DO 3 IQ = 1,MT

IF (IQ.EQ.1) GO TO 540

IF (IQ.EQ.IHT) GO TO 580

IF (IQ.EQ.(IHT+1)) GO TO 630

IF (IQ.EQ.MT) GO TO 530

IF (IQ.EQ.IHT+1)) GO TO 780

WF = (AG(IQ+1)-AG(IQ-1))/2

L6 = WF*B(IQ)

L5= L5-L6

A(IQ) = -L6

GO TO 3

540 WF = AG(IQ+1)/2

550 L5 = -WF*B(IQ)

560 A(IQ) = L5

570 GST = 3

580 WF = .5-(AG(IHT-1)/2)

590 L6=WF*B(IQ)

595 L5=L5-L6

596 A(IQ)=-L6

597 CN = L5

598 GO TO 3

630 WF = AG(IQ+1)/2

Figure 4.7b.1.
L6 = WF*B(IQ)

L5=L5+L6

A(IQ) = L6

GO TO 3

530 WF = .5-(AG(MT-1)/2)

L6 = WF*B(MT)

L5=L5+L6

A(MT)= L6

CNL = L5-CN

GO TO 3

780 WF=(AG(IQ+1)-AG(IQ-1))/2

L6 = WF*B(IQ)

L5=L5+L6

A(IQ)= L6

3 CONTINUE

890 C3 = 0

DO 40 IY = 1,MT

IF (IY.GT.(IHT+1)) GO TO 980

C5= (B(IY)*AH(IY))

AH(IY)=C5

C3 = C3+C5

IF (IY.EQ.IHT) GO TO 970

GO TO 40

970 CCP = C3

GO TO 40

980 C5=(B(IY)*AH(IY))

AH(IY) = C5

C3 = C3 + C5

40 CONTINUE

C3 = C3-CCP

IF(CL1.EQ.0.) GO TO 196

WRITE (7,165)

FORMAT(///,10X,' UPPER SURFACE')

WRITE (7,194)

165 FORMAT(IIII0X,' UPPER SURFACE')

194 FORMAT(IIII0X,' LOWER SURFACE')

16

C9=0.

C8=0.

DO 5 S=1,MT

IF (S.EQ.(IHT+1)) GO TO 1240

IF (S.GT.(IHT+1)) GO TO 1270

BB(S) = (-A(S)*AG(S)+(AH(S)*AJ(S))

C9 = C9+BB(S)

TIT = AG(S)*100

IF(CL1.EQ.0.) GO TO 5

WRITE(7,185) TIT,B(S),A(S),AH(S),BB(S)

GO TO 5

1240 IF(CL1.EQ.0.) GO TO 1270

WRITE(7,195)

195 FORMAT(///10X,' LOWER SURFACE')

WRITE (7,194)

1270 BB(S)=(A(S)*AG(S)+(AH(S)*AJ(S))

C8= C8-BB(S)

TIT = AG(S)*100

Figure 4.7b.2.
IF(CL1.EQ.0.) GO TO 5
WRITE (7,185) TIT,B(S),A(S),AH(S),-BB(S)
185 FORMAT(4X,F4.1,4F12.4)
CONTINUE
5
ST1 = C8+C9
IF(CL1.EQ.0) GO TO 209
WRITE(7,205)
205 FORMAT(//18X,'PRESSURE',2X,'SUCTION',2X,'TOTAL'/)
WRITE(7,206)CN,CNL,L5
206 FORMAT(12X,'CN',2X,3F9.4)
WRITE(7,207)CCP,ST,C3
207 FORMAT(12X,'CC',2X,3F9.4)
WRITE(7,208)C9,CS,ST1
208 FORMAT(12X,'CM',2X,3F9.4)
AT = AN * 0.017453
CS1 = COS(AT)
SN = SIN(AT)
CL = (L5*CS1)-(C3*SN)
CD = (C3*CS1) + (L5*SN)
WRITE (7,225)
225 FORMAT(22X,' WING PERFORMANCE'/18X,'CL',10X,'CD',10X,'CM')
WRITE (7,235) CL,CD,ST1
235 FORMAT(10X,3F12.4)
CALL CLOSE(2)
END

Figure 4.7b.3.
SUBROUTINE SET

DATA DIMENSION DA(100)

COMMON/NONE/NJ,HT,SR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN

COMMON/TWO/RT(30),U(30),X(30),P(40),Q(40),R(20)

COMMON/THREE/J(60),W(30),B(30),W(30),X(30),Y(30)

COMMON/FIVE/E(30),H(30),Y(30),G(30),W(20)

COMMON/SEVEN/PC(50)

COMMON/EIGHT/TAB,AMP,IFN,IAM

DATA Y(1),Y(2),G(1),G(2),Y(23),G(23),G(24)/0.0,0.0,0.0,0.0,

CALL ASSIGN(3,'PAD.DAT',0,'OLD',)

DEFINE FILE 3 (50,512,U,INR)

CALL ASSIGN(2,'NPL.DAT',0,'OLD',)

DEFINE FILE 2 (50,128,U,INR)

DATA OUTPUT

NJ1 = NJ-4

IWT=1

IF(C11,EQ.0.) GO TO 12

WRITE(C7,65)

FORMAT(/" TRANSDUCER OUTPUT"/" CP VALUES CHANNELS 2-4/

C 7X, '1', 7X, '2', 7X, '3', 7X, '4')

WRITE(C7,70)

FORMAT(7I3,4F9.4)

CONTINUE

WRITE(7,300)ITRN

FORMAT(/" RUN ',I4,' OUTPUT'/12X,' EXT VEL.',7X,' MOVEMENT',

C '8X, 'Y CO-ORD')

DO 45 J = 1,NJ

IF(J.LT.3.OR.J.GT.22) GO TO 63

WMT = -WTY(J-2)

WMB = -WBY(J-2)

GO TO 62

WMT = 0.0

WMB = 0.0

WRITE(7,50) D(J),E(J),H(J),Y(J),G(J),WMT,WMB

FORMAT(7F9.4)

CONTINUE

WRITE(7,300)ITRN

FORMAT(/" RUN ',I4,' DATA'/15X,' TOP WALL')

WRITE(7,320)

FORMAT(/"5X,' JACK',5X,' MACH NO.'

THN=0

BMN=0

CALL ASSIGN(6,'WALM.DAT',0,'OLD',)

DEFINE FILE 6 (20,256,U,INR)

WRITE(7,21)

FORMAT(/"$ MACH NO. RECORD (1-20) = ")

READ(7,22)INR

WRITE(7,22)INR

CONTINUE

WRITE(7,320)

FORMAT(/"5X,' JACK',5X,' MACH NO.'

THN=0

BMN=0

CALL ASSIGN(6,'WALM.DAT',0,'OLD',)

DEFINE FILE 6 (20,256,U,INR)

WRITE(7,21)

FORMAT(/"$ MACH NO. RECORD (1-20) = ")

READ(7,22)INR

WRITE(7,22)INR

CONTINUE

WRITE(7,320)

FORMAT(/"5X,' JACK',5X,' MACH NO.'

THN=0

BMN=0

CALL ASSIGN(6,'WALM.DAT',0,'OLD',)

DEFINE FILE 6 (20,256,U,INR)

WRITE(7,21)

FORMAT(/"$ MACH NO. RECORD (1-20) = ")

READ(7,22)INR

WRITE(7,22)INR

CONTINUE

WRITE(7,320)

FORMAT(/"5X,' JACK',5X,' MACH NO.'
CONTINUE
WRITE(6,'(M)')(X(J),J=1,42)
CALL CLOSE(6)
GMN=0
DO 170 J = 1,NJ1
   IS1 = RS(J) + ((Y(J+2))*PC(J))
   DA(J)=IS1
   IP = RS(J)
   YM=P(J+1)
   WM = -WTY(J)-DET(J)
   IF (DA(16).LT.123.) DA(16)=123.
   IF (DA(17).LT.178.) DA(17)=178.
   IF (DA(18).LT.161.) DA(18)=161.
   IF (DA(19).LT.140.) DA(19)=140.
   IF (DA(20).LT.216.) DA(20)=216.
   IF(CL1.EQ.0.)GO TO 170
   WRITE(7,840)J,WM,YM,Y(J+2),IP,IS1
   FORMAT(7X,I2,11X,3F11.4,2(7X,I3))
   IF(J.LT.5.OR.J.GT.13)GO TO 170
   IBJ=J+20
   WRITE(7,810)P(IBJ)
   FORMAT(13X,FI1.4)
CONTINUE
IF(CL1.GT.0.) WRITE(7,20)
FORMAT(/'BOTTOM WALL'/)
DO 180 J = 1,NJ1
   IS1 = RN(J) - ((G(J+2))*PC(J+NJ1))
   DA(J+NJ1)=IS1
   IP = RN(J)
   YM=Q(J+16)
   IF (DA(36).LT.209.) DA(36)=209.
   IF (DA(37).LT.143.) DA(37)=143.
   IF (DA(38).LT.136.) DA(38)=136.
   IF (DA(39).LT.183.) DA(39)=183.
   IF (DA(40).LT.270.) DA(40)=270.
   IF(CL1.EQ.0.)GO TO 180
   WRITE(7,840)J,WM,YM,G(J+2),IP,IS1
   GO TO 180
   IF(J.GT.19)GO TO 180
   AMN=(P(J+1)-Q(J+1))/2
   GMN=GMN+(AMN*AMN)
   TMN=TMN+(P(J+1)*P(J+1))
   BMN=BMN+(Q(J+1)*Q(J+1))
   IF(J.LT.5.OR.J.GT.13)GO TO 180
   IBJ=J+20
   WRITE(7,810)Q(IBJ)
CONTINUE
KT ::= 2*NJ1
DO 190 J = 1,NJ
   DO 190 J = 1,NJ
   DA(J+KT) = E(J)
   KS = KT + NJ
   DO 200 J = 1,NJ
   Figure 4.8.2.
0121 200  DA(J+KS) = H(J)
0122  IF(CLI.EQ.0.)WRITE(5,30) IFN
0124  WRITE(7,30) IFN
0125  30  FORMAT(/' WALL & MODEL OUTPUT RECORD NO. = 'I4//50('=')//)
0126  35  FORMAT(I2)
0127  ND = 2 * (NJ+NJ1)
0128  WRITE(3,'IFN) (DA(J),J=1,ND)
0129  MT2 = MT + 2
0130  IF(IWT.EQ.1) MT2=MT
0132  WRITE(2,'IFN) (B(J),J=1,MT2)
0133  CALL CLOSE(2)
0134  CALL CLOSE(3)
0135  RETURN
0136  END

SET

Figure 4.8.3.
SUBROUTINE WALL

COMMON/EIGHT/TAB,AMP,IFN,IAM
NWJ=40
DIMENSION PC(50),IM(50),ADC(50),PV(50)
DIMENSION PL(10)
DATA PL/130.,178.,161.,140.,216.,209.,143.,136.,183.,270./
CALL ASSIGNC3,'PAD.DAT',0,'OLD',)
DEFINE FILE 3 (50,512,U,INR)
READ(3',IFN) (PV(J),J=1,NWJ)
CALL CLOSE(3)

CALCULATE MOVEMENT
IF=1
IP=5
CALL INIT
50
IU=0
ICNT=0
IS=0
ITOL=3
DO 25 L=1,NWJ
IAI=L+23
IJP=CVSWG(IADC(IAI))
IPV=IJP*24414
IMOVE=ADC(L)-IPV
ADC(L)=IPV
IF(IPV.GE.980) WRITE(5,200)
200 FORMAT(/' POT LIMIT REACHED ON JACK ',14)
IF(IPV.GE.980) STOP
IF(L.LE.15) GO TO 60
IF(L.GT.20.AND.L.LE.35) GO TO 60
IF(L.EQ.36) IW=15
NDJ=L-(15tIW)
IF(PV(L).LT.PL(NDJ)) PV(L)=PL(NDJ)
IPD=PV(L)-IP
IMV=IPV-IPD
IMV=IPV-IPD
IJ=L
IF(IP.EQ.0.AND.IMV.GT.0) IMV=0
IF(L.GT.20) IMV=-IMV
IF(L.GT.20) GO TO 90
IEV=IJ/2
IEV=IEVtIEV
IF(IEV.NE.IJ) IMV=-IMV
IF(IEV.NE.IJ) IMV=-IMV
190 IF(IF.EQ.1)GO TO 300
IF(L.EQ.1.OR.L.EQ.21) GO TO 110
IF(IABS(IM(L))GT.50.AND.IABS(IMOVE).LE.10) GO TO 400
IF(IS.GE.39)ICNT=ICNT+1
IF(IS.GE.39.AND.ICNT.EQ.6) GO TO 500
110 IF(IABS(IMV)-IABS(IM(L)).GT.50) GO TO 500
300 IM(L)=IMV
300 IM(L)=IMV
40 IF(IABS(IMV).LE.ITOL) GO TO 40

Figure 4.9.1.
GO TO 35
IS=IS+1
IMV=0
CALL MOVE(IJ,IMV)
CONTINUE
IF=0
IF(IS.EQ.40) GO TO 70
CALL START(PV,IM)
IF(IS.LT.40) GO TO 50
IF(IF.EQ.0) GO TO 100
IF=0
GO TO 50
GO TO 50
IW=0
DO 85 J=16,40
IF(J.GT.20.AND.J.LT.36) GO TO 85
ILIM=J-(15+IW)
IF(IABS(IE).LE.ITOL) WRITE(5,80) J
FORMAT(' LOWER POT LIMIT REACHED ON JACK ',I4)
CONTINUE
GO TO 250
WRITE(5,150) L
FORMAT(/I' JACK ',I4,' JAMMED!!!!!!!!!!!')
GO TO 250
WRITE(5,160) L
FORMAT(/WALL OUT OF CONTROL AT JACK ',I4/40('*')
RETURN
END

SUBROUTINE INIT
INITIALISE TSWT CONTROL SYSTEM
NWJ=40
IDI=128
DO 10 J=1,NWJ
ICOM=J*IDI+0
CALL DID(ICOM,IDI,INPUT)
IF(INPUT.NE.0) WRITE(5,30) J
FORMAT(' JACK ',I4,' I/O ERROR')
CONTINUE
RETURN
END
INIT

Figure 4.9.2.
SUBROUTINE START(PV,IM)
DIMENSION PV(50),IM(50)
ISUM=0
DO 10 J=21,40
ISUM=ISUM+ABS(IM(J))
IAM=ISUM/20
RCT=(IAM/2)*25.
IF(RCT.GT.600.) RCT=600.
IF(RCT.LT.25.) RCT=25.
IF(RCT.GT.50.) CALL DIO(129,128,INPUT)
IF(RCT.GT.50.) CALL DIO(149,128,INPUT)
START
IPSA=45
IDI=0
ICOM=IPSA+IDI+4096
CALL DIO(ICOM,IDI,INPUT)
C
WAIT FOR WALL TO MOVE
IEND=0
CALL SETR(5,0,RCT,IEND)
CALL LWAIT(0,IEND)
STOP ALL MOTORS
CALL DIO(45,IDI,INPUT)
RETURN
END
SUBROUTINE MOVE(IJ,IMV)
IF(IMV=0) 15,10,5
IDR=3072
GO TO 20
IDR=0
GO TO 20
IDR=2560
IDI=128
ICOM=IJ+IDI+IDR
LOAD COMMAND
CALL DIO(ICOM,IDI,INPUT)
RETURN
END
SUBROUTINE DIO(ICOM,IDI,INPUT)
CALL IPOKE("167774",45)
CALL IPOKE("167774",ICOM)
IF(IP=0) GO TO 10
CALL IPOKE("167774",ICOM)
IF(IDI.EQ.0) GO TO 20
IF(IP.EQ.0) GO TO 20
IF(IDI.EQ.128) GO TO 30
INPUT=IP=167772)
CALL IPOKE("167770",0)
RETURN
END
MOVE
SUBROUTINE DIO(ICOM,IDI,INPUT)
CALL IPOKE("167774",45)
CALL IPOKE("167774",ICOM)
IF(IP.EQ.0) GO TO 10
CALL IPOKE("167774",ICOM)
IF(IDI.EQ.0) GO TO 20
IF(IP.EQ.0) GO TO 20
IF(IDI.EQ.128) GO TO 30
INPUT=IP=167772)
CALL IPOKE("167770",0)
RETURN
END
hay
DIO

Figure 4.9.3.
PROGRAM REAN

MAIN CONTROL PROGRAM

COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,PR3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(40),Q(40),RD(20)
COMMON/THREE/B(60),WBY(30),WLY(30),WLY(30),B(50),PD(48,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DE(60),CS(20)
COMMON/SEVEN/PC(50)
COMMON/EIGHT/TAB,AMP,IFN,IAM

CL1 = 500.
NR = 8
NJ = 24
WRITE(7,5)

5 FORMAT(16X,' UNIVERSITY OF SOUTHAMPTON'/5X,
C ' TRANSONIC SELF-STREAMLINING WIND TUNNEL'/20X,8('**'))

WRITE(7,10)

10 FORMAT(17X,' DATA REANALYSIS'/17X,15('**'))

CALL IDATE(I,J,K)
WRITE(7,1S) J,I,K

15 FORMAT(19X,I2,2('-',I2))

WRITE(7,40)

40 FORMAT(/' t RUN NO. = ')

READ (7,45) ITRN
FORMA(14)

22 FORMAT(7,110)

24 FORMAT(/' RUN NO. = ')

READ(7,45) IFN

WRITE(7,50)

50 FORMAT(/' MODEL ALPHA (DEG) = ')

READ(7,60) AN

WRITE(7,70)

70 FORMAT(/' NO. OF MODEL TAPS ? ')

READ(7,45) MT

WRITE(7,80)

80 FORMAT(/' INPUT 1 FOR AUTO IN-FILE SELECTION - ')

READ(7,45) IAM

WRITE(7,85)

85 FORMAT(/' INPUT AMBIENT CONDITIONS'/
C ' TEMP (DEG.C) = ')

READ(7,25) TAB

WRITE(7,90)

90 FORMAT(/' PRES. (CM HG) = ')

READ(7,25) AMP

CALL DATA

CALL REDUCE

CALL WAS

CALL STAR

CALL SUME

IF(MT.EQ.0) GO TO 100

CALL FORCE

CALL SET

END

Figure 4.10.1.
0001 PROGRAM RUN
0002 DIMENSION TUN(5)
0003 CALL ASSIGN(2,'RUN.DAT',0,'OLD',)
0004 DEFINE FILE 2 (6,256,U,1JR)
0005 READ(2,1)(TUN(J),J=1,5)
0006 WRITE(5,25)
0007 25 FORMAT('S INPUT 1 FOR RECORD NO. INCREMENT ')
0008 READ(5,35)INC
0009 35 FORMAT(I6)
0010 IAD=CVSWG(IADC(21))
0011 TUN(3)=(IAD-2045)/8.15
0012 IF(INC.EQ.1) GO TO 20
0014 WRITE(5,5)
0015 5 FORMAT(// ' GOOD MORNING'// INPUT PRES.,')
0016 READ(5,45)TUN(4)
0017 45 FORMAT(F8.4)
0018 WRITE(5,15)
0019 15 FORMAT(//' INPUT NEXT RUN NO. & FILE NO.,')
0020 READ(5,10)IR,IF
0021 10 FORMAT(2I6)
0022 TUN(1)=IR
0023 TUN(2)=IF
0024 GO TO 30
0025 20 TUN(2)=TUN(2)+1
0026 30 WRITE(2,1)(TUN(J),J=1,5)
0027 WRITE(5,55)(TUN(J),J=1,5)
0028 55 FORMAT(//' RUN DATA STORED '/5F8.2)
0029 CALL CLOSE(2)
0030 END
RUN

Figure 4.11.1.
Figure 5  A sample output of the control software OFLEX
UNIVERSITY OF SOUTHAMPTON
TRANSONIC SELF-STREAMLINING WIND TUNNEL

AUTO MODE
22-1-82

RUN NO. = 407

MODEL ALPHA (DEG) = 4.0

AMBIENT CONDITIONS
TEMP = 22.25  PRES(CM HG) = 76.65

ITERATION RECORD NO. = 31

MACH NO. = 0.5945

REYNOLDS NO. = 0.118538E+07

RECORD = 30

WALL CP ERROR
TOP - 0.0061  BOTTOM - 0.0055
RESIDUALS = 0.0287 -0.0042  0.0014

WALL & MODEL OUTPUT RECORD NO. = 31

=================================================================

ITERATION RECORD NO. = 32

MACH NO. = 0.5948

REYNOLDS NO. = 0.118590E+07

RECORD = 31

WALL CP ERROR
TOP - 0.0026  BOTTOM - 0.0038
RESIDUALS = -0.0163  0.0051  0.0000

WALL & MODEL OUTPUT RECORD NO. = 32

=================================================================

Figure 5.1.
ITERATION RECORD NO. = 33

MACH NO. = 0.5966

REYNOLDS NO. = 0.118846E+07

RECORD = 32

WALL CP ERROR
TOP = 0.0029  BOTTOM = 0.0028
RESIDUALS = -0.0066  0.0101 -0.0017

WALL & MODEL OUTPUT RECORD NO. = 33

__________________________________________________________

Figure 5.2.
Figure 6. A sample output of the re-analysis software ORLEX
RUN ORLEX
UNIVERSITY OF SOUTHAMPTON
TRANSONIC SELF-STREAMLINING WIND TUNNEL
********

DATA REANALYSIS
***************
11-2-82

RUN NO. = 389
FILE NO. = 21

MODEL ALPHA (DEG) = 3.0
NO. OF MODEL TAPS ? 50

INPUT 1 FOR AUTO IN-FILE SELECTION -1

INPUT AMBIENT CONDITIONS
TEMP (DEG.C) = 22.0
PRES. (CM HG) = 76.32

DATA INPUT FILE = *ADC.DAT
*******************************************************************************
ITERATION RECORD NO. = 21
*******************************************************************************

MACH NO. = 0.8038

REYNOLDS NO. = 0.143046E+07

FILE NO. ?
1 FOR M=<.725
2 FOR .725<M>.825
3 FOR M=>.825 ANS = 2

WALL CONTOURS RECORD = 20

WAS COMPUTING NOW !!
DELTA STAR CALCS.

TOP WALL

<table>
<thead>
<tr>
<th>TAP NO.</th>
<th>DU/DX</th>
<th>MACH NO.</th>
<th>D*</th>
<th>DD*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3870</td>
<td>0.8238</td>
<td>0.0155</td>
<td>-0.0005</td>
</tr>
<tr>
<td>2</td>
<td>-3.0340</td>
<td>0.8057</td>
<td>0.0227</td>
<td>-0.0004</td>
</tr>
<tr>
<td>3</td>
<td>0.6563</td>
<td>0.7998</td>
<td>0.0290</td>
<td>-0.0000</td>
</tr>
<tr>
<td>4</td>
<td>0.1942</td>
<td>0.8092</td>
<td>0.0344</td>
<td>-0.0009</td>
</tr>
<tr>
<td>5</td>
<td>1.9132</td>
<td>0.8012</td>
<td>0.0395</td>
<td>-0.0012</td>
</tr>
<tr>
<td>6</td>
<td>14.1533</td>
<td>0.8127</td>
<td>0.0403</td>
<td>-0.0042</td>
</tr>
<tr>
<td>7</td>
<td>32.5273</td>
<td>0.8366</td>
<td>0.0386</td>
<td>-0.0076</td>
</tr>
<tr>
<td>8</td>
<td>54.5744</td>
<td>0.8975</td>
<td>0.0349</td>
<td>-0.0130</td>
</tr>
<tr>
<td>9</td>
<td>55.2920</td>
<td>0.8884</td>
<td>0.0302</td>
<td>-0.0189</td>
</tr>
<tr>
<td>10</td>
<td>-2.2115</td>
<td>1.0633</td>
<td>0.0294</td>
<td>-0.0218</td>
</tr>
<tr>
<td>11</td>
<td>-53.4752</td>
<td>0.9817</td>
<td>0.0344</td>
<td>-0.0189</td>
</tr>
<tr>
<td>12</td>
<td>-37.5962</td>
<td>0.9032</td>
<td>0.0424</td>
<td>-0.0130</td>
</tr>
<tr>
<td>13</td>
<td>-17.6038</td>
<td>0.8758</td>
<td>0.0483</td>
<td>-0.0088</td>
</tr>
<tr>
<td>14</td>
<td>-10.6314</td>
<td>0.8436</td>
<td>0.0571</td>
<td>-0.0038</td>
</tr>
<tr>
<td>15</td>
<td>-4.2687</td>
<td>0.8159</td>
<td>0.0672</td>
<td>0.0013</td>
</tr>
<tr>
<td>16</td>
<td>-0.3833</td>
<td>0.8144</td>
<td>0.0735</td>
<td>0.0029</td>
</tr>
<tr>
<td>17</td>
<td>0.1915</td>
<td>0.8130</td>
<td>0.0781</td>
<td>0.0033</td>
</tr>
<tr>
<td>18</td>
<td>0.5746</td>
<td>0.8159</td>
<td>0.0823</td>
<td>0.0029</td>
</tr>
<tr>
<td>19</td>
<td>0.1913</td>
<td>0.8173</td>
<td>0.0861</td>
<td>0.0021</td>
</tr>
<tr>
<td>20</td>
<td>0.0000</td>
<td>0.8173</td>
<td>0.0907</td>
<td>0.0021</td>
</tr>
</tbody>
</table>

BOTTOM WALL

<table>
<thead>
<tr>
<th>TAP NO.</th>
<th>DU/DX</th>
<th>MACH NO.</th>
<th>D*</th>
<th>DD*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4216</td>
<td>0.8077</td>
<td>0.0155</td>
<td>-0.0009</td>
</tr>
<tr>
<td>2</td>
<td>-1.1125</td>
<td>0.8060</td>
<td>0.0223</td>
<td>-0.0004</td>
</tr>
<tr>
<td>3</td>
<td>-1.5025</td>
<td>0.7981</td>
<td>0.0290</td>
<td>-0.0000</td>
</tr>
<tr>
<td>4</td>
<td>-3.5534</td>
<td>0.7935</td>
<td>0.0353</td>
<td>0.0004</td>
</tr>
<tr>
<td>5</td>
<td>-7.0914</td>
<td>0.7721</td>
<td>0.0437</td>
<td>0.0025</td>
</tr>
<tr>
<td>6</td>
<td>-12.0694</td>
<td>0.7533</td>
<td>0.0508</td>
<td>0.0063</td>
</tr>
<tr>
<td>7</td>
<td>-3.1566</td>
<td>0.7367</td>
<td>0.0542</td>
<td>0.0084</td>
</tr>
<tr>
<td>8</td>
<td>8.0421</td>
<td>0.7459</td>
<td>0.0550</td>
<td>0.0075</td>
</tr>
<tr>
<td>9</td>
<td>7.1233</td>
<td>0.7563</td>
<td>0.0550</td>
<td>0.0059</td>
</tr>
<tr>
<td>10</td>
<td>6.3412</td>
<td>0.7634</td>
<td>0.0554</td>
<td>0.0046</td>
</tr>
<tr>
<td>11</td>
<td>-3.0033</td>
<td>0.7714</td>
<td>0.0567</td>
<td>0.0042</td>
</tr>
<tr>
<td>12</td>
<td>-8.1537</td>
<td>0.7563</td>
<td>0.0596</td>
<td>0.0058</td>
</tr>
<tr>
<td>13</td>
<td>0.6964</td>
<td>0.7520</td>
<td>0.0622</td>
<td>0.0063</td>
</tr>
<tr>
<td>14</td>
<td>8.0705</td>
<td>0.7742</td>
<td>0.0630</td>
<td>0.0025</td>
</tr>
<tr>
<td>15</td>
<td>3.0736</td>
<td>0.8001</td>
<td>0.0634</td>
<td>-0.0017</td>
</tr>
<tr>
<td>16</td>
<td>1.9875</td>
<td>0.8001</td>
<td>0.0664</td>
<td>-0.0037</td>
</tr>
<tr>
<td>17</td>
<td>1.6074</td>
<td>0.8149</td>
<td>0.0697</td>
<td>-0.0055</td>
</tr>
<tr>
<td>18</td>
<td>-2.0260</td>
<td>0.8121</td>
<td>0.0748</td>
<td>-0.0054</td>
</tr>
<tr>
<td>19</td>
<td>-1.6459</td>
<td>0.7998</td>
<td>0.0806</td>
<td>-0.0051</td>
</tr>
<tr>
<td>20</td>
<td>0.0000</td>
<td>0.7998</td>
<td>0.0861</td>
<td>-0.0046</td>
</tr>
</tbody>
</table>

UNIT REYNOLDS NO. = 357276.9  D* FPG = 0.0088
WALL CP ERROR
TOP - 0.0072
RESIDUAL ERRORS
BOTTOM - 0.0064

<table>
<thead>
<tr>
<th>X</th>
<th>U/UFS</th>
<th>V/UFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-19.000</td>
<td>0.0009</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-17.000</td>
<td>0.0009</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-16.000</td>
<td>0.0008</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-15.000</td>
<td>0.0007</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-14.000</td>
<td>0.0006</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-13.000</td>
<td>0.0004</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-12.000</td>
<td>0.0003</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-11.000</td>
<td>0.0002</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-10.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-9.000</td>
<td>0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-8.000</td>
<td>0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-7.000</td>
<td>0.0004</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-6.000</td>
<td>0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-5.000</td>
<td>0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-4.000</td>
<td>0.0002</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-3.000</td>
<td>0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-2.000</td>
<td>0.0013</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-1.000</td>
<td>0.0021</td>
<td>-0.0000</td>
</tr>
<tr>
<td>0.000</td>
<td>0.0024</td>
<td>-0.0000</td>
</tr>
<tr>
<td>1.000</td>
<td>0.0021</td>
<td>-0.0000</td>
</tr>
<tr>
<td>2.000</td>
<td>0.0014</td>
<td>-0.0000</td>
</tr>
<tr>
<td>3.000</td>
<td>0.0008</td>
<td>-0.0000</td>
</tr>
<tr>
<td>4.000</td>
<td>0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>5.000</td>
<td>0.0003</td>
<td>-0.0000</td>
</tr>
<tr>
<td>6.000</td>
<td>0.0002</td>
<td>-0.0000</td>
</tr>
<tr>
<td>7.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>8.000</td>
<td>0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>9.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>10.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>11.000</td>
<td>0.0002</td>
<td>-0.0000</td>
</tr>
<tr>
<td>12.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>13.000</td>
<td>0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>14.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>15.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>16.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>17.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>18.000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
</tbody>
</table>

MODEL ERRORS

<table>
<thead>
<tr>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.4979</td>
</tr>
<tr>
<td>-0.7490</td>
</tr>
<tr>
<td>0.0000</td>
</tr>
<tr>
<td>0.7490</td>
</tr>
<tr>
<td>1.4979</td>
</tr>
<tr>
<td>2.2469</td>
</tr>
<tr>
<td>2.9959</td>
</tr>
<tr>
<td>3.7449</td>
</tr>
<tr>
<td>4.4938</td>
</tr>
</tbody>
</table>

EFFECT
DELTA CL

ALPHA ERROR = -0.0045 DEGREES -0.0005
INDUCED CAMBER = 0.0007 DEGREES -0.0013
VEL. ERROR CP = -0.0048
AVERAGE = -0.0029 0.0048

WALLS STREAMLINED

Figure 6.3.
## UPPER SURFACE

<table>
<thead>
<tr>
<th>ZCHORD</th>
<th>CP LOCAL</th>
<th>CN LOCAL</th>
<th>CC LOCAL</th>
<th>CH LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0227</td>
<td>-0.0038</td>
<td>0.0079</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.7</td>
<td>-0.7242</td>
<td>0.0036</td>
<td>-0.0062</td>
<td>-0.0001</td>
</tr>
<tr>
<td>1.0</td>
<td>-1.0257</td>
<td>0.0046</td>
<td>-0.0024</td>
<td>-0.0001</td>
</tr>
<tr>
<td>1.6</td>
<td>-1.2861</td>
<td>0.0064</td>
<td>-0.0027</td>
<td>-0.0002</td>
</tr>
<tr>
<td>2.0</td>
<td>-1.2979</td>
<td>0.0066</td>
<td>-0.0019</td>
<td>-0.0002</td>
</tr>
<tr>
<td>2.6</td>
<td>-1.3172</td>
<td>0.0070</td>
<td>-0.0018</td>
<td>-0.0002</td>
</tr>
<tr>
<td>3.1</td>
<td>-1.3824</td>
<td>0.0068</td>
<td>-0.0015</td>
<td>-0.0002</td>
</tr>
<tr>
<td>3.6</td>
<td>-1.3838</td>
<td>0.0146</td>
<td>-0.0026</td>
<td>-0.0006</td>
</tr>
<tr>
<td>5.2</td>
<td>-1.3318</td>
<td>0.0270</td>
<td>-0.0041</td>
<td>-0.0015</td>
</tr>
<tr>
<td>7.7</td>
<td>-1.2563</td>
<td>0.0314</td>
<td>-0.0038</td>
<td>-0.0025</td>
</tr>
<tr>
<td>10.2</td>
<td>-1.1720</td>
<td>0.0439</td>
<td>-0.0040</td>
<td>-0.0046</td>
</tr>
<tr>
<td>15.2</td>
<td>-1.1809</td>
<td>0.0591</td>
<td>-0.0044</td>
<td>-0.0091</td>
</tr>
<tr>
<td>20.2</td>
<td>-1.1601</td>
<td>0.0580</td>
<td>-0.0035</td>
<td>-0.0119</td>
</tr>
<tr>
<td>25.2</td>
<td>-1.1279</td>
<td>0.0563</td>
<td>-0.0028</td>
<td>-0.0143</td>
</tr>
<tr>
<td>30.2</td>
<td>-1.1249</td>
<td>0.0844</td>
<td>-0.0031</td>
<td>-0.0256</td>
</tr>
<tr>
<td>40.2</td>
<td>-1.0382</td>
<td>0.0779</td>
<td>-0.0021</td>
<td>-0.0314</td>
</tr>
<tr>
<td>45.2</td>
<td>-0.7681</td>
<td>0.0383</td>
<td>-0.0004</td>
<td>-0.0173</td>
</tr>
<tr>
<td>50.2</td>
<td>-0.5044</td>
<td>0.0189</td>
<td>-0.0001</td>
<td>-0.0095</td>
</tr>
<tr>
<td>52.7</td>
<td>-0.4348</td>
<td>0.0109</td>
<td>0.0000</td>
<td>-0.0057</td>
</tr>
<tr>
<td>55.2</td>
<td>-0.3941</td>
<td>0.0099</td>
<td>0.0001</td>
<td>-0.0054</td>
</tr>
<tr>
<td>57.7</td>
<td>-0.3548</td>
<td>0.0089</td>
<td>0.0001</td>
<td>-0.0051</td>
</tr>
<tr>
<td>60.2</td>
<td>-0.3606</td>
<td>0.0090</td>
<td>0.0002</td>
<td>-0.0054</td>
</tr>
<tr>
<td>62.7</td>
<td>-0.3504</td>
<td>0.0087</td>
<td>0.0003</td>
<td>-0.0055</td>
</tr>
<tr>
<td>65.2</td>
<td>-0.2664</td>
<td>0.0067</td>
<td>0.0003</td>
<td>-0.0043</td>
</tr>
<tr>
<td>67.7</td>
<td>-0.2778</td>
<td>0.0069</td>
<td>0.0003</td>
<td>-0.0047</td>
</tr>
<tr>
<td>70.2</td>
<td>-0.2736</td>
<td>0.0102</td>
<td>0.0007</td>
<td>-0.0071</td>
</tr>
<tr>
<td>75.2</td>
<td>-0.3361</td>
<td>0.0119</td>
<td>0.0010</td>
<td>-0.0088</td>
</tr>
<tr>
<td>80.2</td>
<td>-0.1737</td>
<td>0.0068</td>
<td>0.0010</td>
<td>-0.0039</td>
</tr>
<tr>
<td>85.3</td>
<td>-0.0976</td>
<td>0.0049</td>
<td>0.0007</td>
<td>-0.0041</td>
</tr>
<tr>
<td>90.1</td>
<td>-0.0143</td>
<td>0.0007</td>
<td>0.0001</td>
<td>-0.0006</td>
</tr>
<tr>
<td>95.2</td>
<td>0.0626</td>
<td>-0.0031</td>
<td>-0.0007</td>
<td>0.0029</td>
</tr>
<tr>
<td>****</td>
<td>0.1394</td>
<td>-0.0034</td>
<td>0.0003</td>
<td>0.0034</td>
</tr>
</tbody>
</table>

## LOWER SURFACE

<table>
<thead>
<tr>
<th>ZCHORD</th>
<th>CP LOCAL</th>
<th>CN LOCAL</th>
<th>CC LOCAL</th>
<th>CH LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0684</td>
<td>0.0053</td>
<td>0.0098</td>
<td>0.0000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8331</td>
<td>0.0074</td>
<td>0.0098</td>
<td>-0.0003</td>
</tr>
<tr>
<td>1.8</td>
<td>0.5949</td>
<td>0.0127</td>
<td>0.0059</td>
<td>-0.0004</td>
</tr>
<tr>
<td>5.3</td>
<td>0.1998</td>
<td>0.0085</td>
<td>0.0026</td>
<td>-0.0005</td>
</tr>
<tr>
<td>10.2</td>
<td>-0.0488</td>
<td>-0.0024</td>
<td>-0.0005</td>
<td>0.0003</td>
</tr>
<tr>
<td>15.3</td>
<td>-0.1493</td>
<td>-0.0074</td>
<td>-0.0009</td>
<td>0.0012</td>
</tr>
<tr>
<td>20.2</td>
<td>-0.2291</td>
<td>-0.0142</td>
<td>-0.0009</td>
<td>0.0029</td>
</tr>
<tr>
<td>27.7</td>
<td>-0.3163</td>
<td>-0.0275</td>
<td>-0.0000</td>
<td>0.0076</td>
</tr>
<tr>
<td>37.6</td>
<td>-0.3310</td>
<td>-0.0288</td>
<td>0.0017</td>
<td>0.0107</td>
</tr>
<tr>
<td>45.1</td>
<td>-0.2138</td>
<td>-0.0162</td>
<td>0.0020</td>
<td>0.0072</td>
</tr>
<tr>
<td>52.6</td>
<td>-0.0488</td>
<td>-0.0037</td>
<td>0.0006</td>
<td>0.0019</td>
</tr>
<tr>
<td>60.1</td>
<td>0.0888</td>
<td>0.0067</td>
<td>-0.0011</td>
<td>-0.0040</td>
</tr>
<tr>
<td>67.5</td>
<td>0.3073</td>
<td>0.0179</td>
<td>-0.0030</td>
<td>-0.0121</td>
</tr>
<tr>
<td>71.7</td>
<td>0.3629</td>
<td>0.0318</td>
<td>-0.0039</td>
<td>-0.0228</td>
</tr>
<tr>
<td>85.1</td>
<td>0.4490</td>
<td>0.0412</td>
<td>-0.0036</td>
<td>-0.0351</td>
</tr>
<tr>
<td>90.1</td>
<td>0.4430</td>
<td>0.0221</td>
<td>-0.0000</td>
<td>-0.0200</td>
</tr>
<tr>
<td>95.1</td>
<td>0.3825</td>
<td>0.0190</td>
<td>0.0005</td>
<td>-0.0180</td>
</tr>
<tr>
<td>****</td>
<td>0.1311</td>
<td>0.0032</td>
<td>0.0003</td>
<td>-0.0032</td>
</tr>
</tbody>
</table>

## PRESSURE SUCTION TOTAL

| CN | 0.6319 | 0.0755 | 0.7075 |
| CC | -0.0350 | 0.0193 | -0.0157 |
| CM | -0.1870 | -0.0844 | -0.2715 |

WING PERFORMANCE

| CL | 0.7073 | 0.0214 | -0.2715 |
| CD |       |       |         |
| CM |       |       |         |

Figure 6.4.
<table>
<thead>
<tr>
<th>Transducer Output CP Values Channels 2-4</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 26.5666</td>
<td>-0.0463</td>
<td>0.0891</td>
<td>-0.0107</td>
<td></td>
</tr>
<tr>
<td>2 3.6152</td>
<td>-0.0063</td>
<td>0.1375</td>
<td>-0.0069</td>
<td></td>
</tr>
<tr>
<td>3 42.8183</td>
<td>0.0038</td>
<td>0.1788</td>
<td>0.0075</td>
<td></td>
</tr>
<tr>
<td>4 49.5844</td>
<td>-0.0171</td>
<td>0.1154</td>
<td>0.0176</td>
<td></td>
</tr>
<tr>
<td>5 55.2891</td>
<td>9.0006</td>
<td>0.1331</td>
<td>-0.0649</td>
<td></td>
</tr>
<tr>
<td>6 55.5544</td>
<td>-0.0248</td>
<td>0.1037</td>
<td>0.1066</td>
<td></td>
</tr>
<tr>
<td>7 26.5003</td>
<td>-0.0777</td>
<td>0.1504</td>
<td>0.1431</td>
<td></td>
</tr>
<tr>
<td>8 55.9856</td>
<td>-0.2113</td>
<td>0.1702</td>
<td>0.1230</td>
<td></td>
</tr>
<tr>
<td>9 57.4449</td>
<td>-0.4048</td>
<td>0.1594</td>
<td>0.1015</td>
<td></td>
</tr>
<tr>
<td>10 57.4781</td>
<td>-0.5595</td>
<td>0.8655</td>
<td>0.0858</td>
<td></td>
</tr>
<tr>
<td>11 56.3836</td>
<td>-0.3903</td>
<td>0.0904</td>
<td>0.0680</td>
<td></td>
</tr>
<tr>
<td>12 54.6921</td>
<td>-0.2219</td>
<td>0.0778</td>
<td>0.1014</td>
<td></td>
</tr>
<tr>
<td>13 26.5334</td>
<td>-0.1621</td>
<td>0.1490</td>
<td>0.1108</td>
<td></td>
</tr>
<tr>
<td>14 52.8015</td>
<td>-0.0915</td>
<td>0.1700</td>
<td>0.0617</td>
<td></td>
</tr>
<tr>
<td>15 53.0005</td>
<td>-0.0229</td>
<td>0.1365</td>
<td>0.0120</td>
<td></td>
</tr>
<tr>
<td>16 52.5362</td>
<td>-0.0197</td>
<td>-0.2664</td>
<td>0.0120</td>
<td></td>
</tr>
<tr>
<td>17 52.1382</td>
<td>-0.0164</td>
<td>-0.2778</td>
<td>-0.0207</td>
<td></td>
</tr>
<tr>
<td>18 52.0719</td>
<td>-0.0227</td>
<td>-0.2736</td>
<td>-0.0144</td>
<td></td>
</tr>
<tr>
<td>19 26.6993</td>
<td>-0.0259</td>
<td>-0.2361</td>
<td>0.0125</td>
<td></td>
</tr>
<tr>
<td>20 50.1150</td>
<td>0.0341</td>
<td>-0.1737</td>
<td>0.1953</td>
<td></td>
</tr>
<tr>
<td>21 43.9792</td>
<td>1.0152</td>
<td>-0.0976</td>
<td>-0.0094</td>
<td></td>
</tr>
<tr>
<td>22 38.0755</td>
<td>0.0032</td>
<td>-0.0143</td>
<td>0.0507</td>
<td></td>
</tr>
<tr>
<td>23 36.9146</td>
<td>-0.0306</td>
<td>1.1970</td>
<td>0.1238</td>
<td></td>
</tr>
<tr>
<td>24 35.9860</td>
<td>-0.1368</td>
<td>0.1394</td>
<td>0.1330</td>
<td></td>
</tr>
<tr>
<td>25 26.9978</td>
<td>1.0057</td>
<td>0.0876</td>
<td>0.0514</td>
<td></td>
</tr>
<tr>
<td>26 35.0905</td>
<td>-0.2630</td>
<td>0.1088</td>
<td>0.1102</td>
<td></td>
</tr>
<tr>
<td>27 35.2231</td>
<td>-0.5160</td>
<td>0.1518</td>
<td>0.0619</td>
<td></td>
</tr>
<tr>
<td>28 34.9910</td>
<td>-0.7459</td>
<td>0.1370</td>
<td>1.0440</td>
<td></td>
</tr>
<tr>
<td>29 2.5870</td>
<td>-0.2625</td>
<td>0.1574</td>
<td>0.0932</td>
<td></td>
</tr>
<tr>
<td>30 7.8605</td>
<td>-0.2373</td>
<td>0.1383</td>
<td>0.0945</td>
<td></td>
</tr>
<tr>
<td>31 26.5334</td>
<td>-0.2352</td>
<td>0.1550</td>
<td>0.0680</td>
<td></td>
</tr>
<tr>
<td>32 13.2004</td>
<td>-0.3198</td>
<td>0.1407</td>
<td>0.0623</td>
<td></td>
</tr>
<tr>
<td>33 22.0359</td>
<td>-0.1557</td>
<td>0.1401</td>
<td>0.1152</td>
<td></td>
</tr>
<tr>
<td>34 27.6279</td>
<td>-0.1036</td>
<td>0.1227</td>
<td>0.0365</td>
<td></td>
</tr>
<tr>
<td>35 29.9165</td>
<td>-0.0356</td>
<td>0.1471</td>
<td>0.0107</td>
<td></td>
</tr>
<tr>
<td>36 31.7075</td>
<td>-0.0254</td>
<td>0.1513</td>
<td>0.1195</td>
<td></td>
</tr>
<tr>
<td>37 26.5666</td>
<td>-0.0146</td>
<td>0.1070</td>
<td>-0.0069</td>
<td></td>
</tr>
<tr>
<td>38 33.6643</td>
<td>-0.0140</td>
<td>0.3073</td>
<td>-0.0075</td>
<td></td>
</tr>
<tr>
<td>39 33.9960</td>
<td>-0.0317</td>
<td>0.3629</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>40 31.4090</td>
<td>-0.0343</td>
<td>0.4490</td>
<td>-0.0289</td>
<td></td>
</tr>
<tr>
<td>41 27.6279</td>
<td>-0.0153</td>
<td>0.4430</td>
<td>0.0964</td>
<td></td>
</tr>
<tr>
<td>42 24.5434</td>
<td>-0.2015</td>
<td>0.3825</td>
<td>0.1291</td>
<td></td>
</tr>
<tr>
<td>43 26.5334</td>
<td>-0.6707</td>
<td>0.1311</td>
<td>0.0768</td>
<td></td>
</tr>
<tr>
<td>44 11.3099</td>
<td>-0.2734</td>
<td>0.1413</td>
<td>0.0611</td>
<td></td>
</tr>
<tr>
<td>45 1.6252</td>
<td>0.0057</td>
<td>0.1131</td>
<td>0.0504</td>
<td></td>
</tr>
<tr>
<td>46 9.7842</td>
<td>-0.0922</td>
<td>0.0778</td>
<td>0.1121</td>
<td></td>
</tr>
<tr>
<td>47 1.6252</td>
<td>-0.0165</td>
<td>0.0892</td>
<td>0.0787</td>
<td></td>
</tr>
<tr>
<td>48 0.2653</td>
<td>-0.0540</td>
<td>-0.0742</td>
<td>0.0964</td>
<td></td>
</tr>
<tr>
<td>RUN 389 OUTPUT</td>
<td>EXT VEL.</td>
<td>MOVEMENT</td>
<td>Y CO-ORD</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>3.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>6.0000</td>
<td>0.0117</td>
<td>0.0013</td>
<td>0.0007</td>
<td></td>
</tr>
<tr>
<td>9.0000</td>
<td>0.0008</td>
<td>0.0011</td>
<td>0.0005</td>
<td></td>
</tr>
<tr>
<td>12.0000</td>
<td>-0.0007</td>
<td>-0.0021</td>
<td>-0.0001</td>
<td></td>
</tr>
<tr>
<td>15.0000</td>
<td>0.0048</td>
<td>-0.0056</td>
<td>-0.0000</td>
<td></td>
</tr>
<tr>
<td>18.0000</td>
<td>0.0018</td>
<td>-0.0170</td>
<td>-0.0003</td>
<td></td>
</tr>
<tr>
<td>20.0000</td>
<td>0.0119</td>
<td>-0.0279</td>
<td>-0.0004</td>
<td></td>
</tr>
<tr>
<td>21.0000</td>
<td>0.0255</td>
<td>-0.0408</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>22.0000</td>
<td>0.0602</td>
<td>-0.0386</td>
<td>0.0014</td>
<td></td>
</tr>
<tr>
<td>23.0000</td>
<td>0.1059</td>
<td>-0.0394</td>
<td>0.0029</td>
<td></td>
</tr>
<tr>
<td>24.0000</td>
<td>0.1359</td>
<td>-0.0456</td>
<td>0.0041</td>
<td></td>
</tr>
<tr>
<td>25.0000</td>
<td>0.1034</td>
<td>-0.0247</td>
<td>0.0034</td>
<td></td>
</tr>
<tr>
<td>26.0000</td>
<td>0.0681</td>
<td>-0.0262</td>
<td>0.0014</td>
<td></td>
</tr>
<tr>
<td>27.0000</td>
<td>0.0506</td>
<td>-0.0301</td>
<td>0.0006</td>
<td></td>
</tr>
<tr>
<td>29.0000</td>
<td>0.0267</td>
<td>-0.0189</td>
<td>0.0006</td>
<td></td>
</tr>
<tr>
<td>32.0000</td>
<td>0.0067</td>
<td>-0.0033</td>
<td>0.0005</td>
<td></td>
</tr>
<tr>
<td>35.0000</td>
<td>0.0069</td>
<td>-0.0025</td>
<td>0.0001</td>
<td></td>
</tr>
<tr>
<td>38.0000</td>
<td>0.0042</td>
<td>0.0057</td>
<td>0.0004</td>
<td></td>
</tr>
<tr>
<td>41.0000</td>
<td>0.0071</td>
<td>0.0045</td>
<td>0.0002</td>
<td></td>
</tr>
<tr>
<td>44.0000</td>
<td>0.0070</td>
<td>-0.0039</td>
<td>0.0007</td>
<td></td>
</tr>
<tr>
<td>47.0000</td>
<td>0.0070</td>
<td>-0.0039</td>
<td>0.0007</td>
<td></td>
</tr>
<tr>
<td>50.0000</td>
<td>0.0074</td>
<td>-0.0023</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>53.0000</td>
<td>0.0074</td>
<td>-0.0023</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>MACH NO.</td>
<td>RECORD (1-20)</td>
<td>WALL &amp; MODEL OUTPUT RECORD NO. =</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td>----------------------------------</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.0158</td>
<td>0.7634</td>
<td>0.0007</td>
<td>467</td>
</tr>
<tr>
<td>2</td>
<td>0.0183</td>
<td>0.7714</td>
<td>0.0005</td>
<td>566</td>
</tr>
<tr>
<td>3</td>
<td>0.0546</td>
<td>0.7563</td>
<td>-0.0001</td>
<td>551</td>
</tr>
<tr>
<td>4</td>
<td>0.0875</td>
<td>0.7520</td>
<td>-0.0000</td>
<td>611</td>
</tr>
<tr>
<td>5</td>
<td>0.1456</td>
<td>0.7742</td>
<td>-0.0003</td>
<td>639</td>
</tr>
<tr>
<td>6</td>
<td>0.8101</td>
<td>0.8001</td>
<td>-0.0004</td>
<td>692</td>
</tr>
<tr>
<td>7</td>
<td>0.8256</td>
<td>0.8001</td>
<td>0.0004</td>
<td>746</td>
</tr>
<tr>
<td>8</td>
<td>0.8748</td>
<td>0.8149</td>
<td>0.0014</td>
<td>890</td>
</tr>
<tr>
<td>9</td>
<td>0.9339</td>
<td>0.8121</td>
<td>0.0029</td>
<td>922</td>
</tr>
<tr>
<td>10</td>
<td>1.0569</td>
<td>0.7998</td>
<td>0.0041</td>
<td>941</td>
</tr>
<tr>
<td>11</td>
<td>1.1771</td>
<td>0.7998</td>
<td>0.0034</td>
<td>933</td>
</tr>
<tr>
<td>12</td>
<td>0.9315</td>
<td>0.9032</td>
<td>0.0014</td>
<td>896</td>
</tr>
<tr>
<td>13</td>
<td>0.9189</td>
<td>0.8758</td>
<td>0.0006</td>
<td>869</td>
</tr>
<tr>
<td>14</td>
<td>0.3407</td>
<td>0.8436</td>
<td>0.0006</td>
<td>870</td>
</tr>
<tr>
<td>15</td>
<td>0.2582</td>
<td>0.8159</td>
<td>0.0005</td>
<td>775</td>
</tr>
<tr>
<td>16</td>
<td>0.2401</td>
<td>0.8144</td>
<td>0.0001</td>
<td>318</td>
</tr>
<tr>
<td>17</td>
<td>0.2027</td>
<td>0.8130</td>
<td>0.0004</td>
<td>362</td>
</tr>
<tr>
<td>18</td>
<td>0.1696</td>
<td>0.8159</td>
<td>0.0002</td>
<td>339</td>
</tr>
<tr>
<td>19</td>
<td>0.1368</td>
<td>0.8173</td>
<td>0.0007</td>
<td>300</td>
</tr>
<tr>
<td>20</td>
<td>0.0842</td>
<td>0.8173</td>
<td>0.0007</td>
<td>365</td>
</tr>
</tbody>
</table>

**RUN 389 DATA**

**TOP WALL**

**JACK MACH NO.**

**MACH NO. RECORD (1-20) = 20**

**BOTTOM WALL**

**FIGURE 6.7.**
Assumed origin of wall boundary layer

Wall measuring points
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Wall jack numbers

Flexible wall

Dummy straight wall extension

△ Wall anchor point
● Wall and diffuser sliding joint

FIG. 7 SOFTWARE REPRESENTATION OF EACH FLEXIBLE WALL
APPENDIX A. DIGITAL INPUT/OUTPUT PROTOCOL
FOR TSWT CONTROL SYSTEM

The transfer of digital information between the computer and wind tunnel hardware is an important part of the on-line control system. Digital I/O involves a complex interaction of system software and hardware. A command code has been devised to simplify the operation of the control system. The protocol of digital I/O are described in the following sections.

Data Format

Each packet of information sent to the wind tunnel consists of the data destination and operation plus the data itself. Information from the wind tunnel consists of data only. All sets of information are contained in a single 16-bit word in the following manner:

Digital Output Word (Equivalent to a six digit octal number in binary code)

<table>
<thead>
<tr>
<th>Digits</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address (0-63 Devices)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add£'ium</td>
</tr>
<tr>
<td>Data 3</td>
</tr>
<tr>
<td>Data 2</td>
</tr>
<tr>
<td>Data 1</td>
</tr>
</tbody>
</table>

Digital Input Word

<table>
<thead>
<tr>
<th>Digits</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- A1 -
Device Addresses

Each address consists of six bits of binary information which corresponds to a decimal number referred to as the software address. The following scheme has been chosen:

<table>
<thead>
<tr>
<th>Device</th>
<th>Software Address</th>
<th>Binary Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stepper Motors</td>
<td>1 → 44</td>
<td>0 0 0 0 1 0</td>
</tr>
<tr>
<td>Pulse Sequence Generator</td>
<td>45</td>
<td>1 0 1 1 0 1</td>
</tr>
<tr>
<td>Scanivalves (1) (2)</td>
<td>46 48</td>
<td>1 0 1 1 0 0 - 1 1 0 0 0 0</td>
</tr>
<tr>
<td>Encoders (1) (2)</td>
<td>47 49</td>
<td>1 0 1 1 1 1 - 1 1 0 0 0 1</td>
</tr>
<tr>
<td>Power Supplies etc.</td>
<td>50 → 63</td>
<td>1 1 0 0 1 0 - 1 1 1 1 1</td>
</tr>
</tbody>
</table>

Data Operation

The transfer of data takes three forms - read only, write only and write before read. These operations determine the type of data to be sent, if any. For example, the read only function requires no data from the computer other than the device address.

Information on data operation is sent to the wind tunnel by adding a chosen software value to the decimal equivalent of the digital output word. This effectively sets the required bits for correct information transfer. The software values are chosen thus:

<table>
<thead>
<tr>
<th>Data Information</th>
<th>Binary Representation</th>
<th>Software Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read only</td>
<td>15 8 7 6</td>
<td>-32768 + 128 = -32640</td>
</tr>
<tr>
<td>Write only</td>
<td>0</td>
<td>0 + 0 = 0</td>
</tr>
<tr>
<td>Write before read</td>
<td>0</td>
<td>0 + 128 = 128</td>
</tr>
</tbody>
</table>

Digit 6 3
Data

Actions at the wind tunnel are determined by the corresponding data 1 or 2 bits of the output word. Data 3 bits of the input word allow checks to be made on system devices.

The output data is transferred by the setting of data 1 and 2 bits, achieved by the modification of the digital output word as for the data operating information.

The software values were chosen thus:

<table>
<thead>
<tr>
<th>Function</th>
<th>Binary Representation</th>
<th>Software Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Motor direction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Forward-go</td>
<td>1 0 1</td>
<td>2560</td>
</tr>
<tr>
<td>Reverse-go</td>
<td>1 1 0</td>
<td>3072</td>
</tr>
<tr>
<td>Digit 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) Scanivalve Move</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Home</td>
<td>1 0 1</td>
<td>20480</td>
</tr>
<tr>
<td>-Step on one</td>
<td>1 0 0</td>
<td>16384</td>
</tr>
<tr>
<td>Digit 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) Pulse sequence generator</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Start</td>
<td>0 0 1</td>
<td>4096</td>
</tr>
<tr>
<td>d) Motor Power Supply</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Off</td>
<td>0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>-On</td>
<td>0 1 0</td>
<td>8192</td>
</tr>
<tr>
<td>Digit 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- A3 -
### Function Binary Representation Software Value

e) Pulse sequence generator 14 13 12 11 10 9 8 0 \rightarrow 32512

Increment step size (currently non operational).

<table>
<thead>
<tr>
<th>No.</th>
<th>10s</th>
<th>100s</th>
<th>1000s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Input data will be in binary code to allow simple software manipulation. The complete input word will always be read regardless of the quantity of information being transferred.

The types of data are as follows:

<table>
<thead>
<tr>
<th>Device</th>
<th>No. of bits of Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanivalve - encoder output</td>
<td>7</td>
</tr>
<tr>
<td>- at home position</td>
<td>1</td>
</tr>
<tr>
<td>Pulse sequence generator</td>
<td></td>
</tr>
<tr>
<td>- finish pulse</td>
<td>1</td>
</tr>
<tr>
<td>- step value</td>
<td>7</td>
</tr>
<tr>
<td>System monitor</td>
<td>16</td>
</tr>
<tr>
<td>Motor direction</td>
<td>3</td>
</tr>
</tbody>
</table>

### Command Coding

Each command sent to the wind tunnel must be unambiguous and provide information on data destination and operation and the data itself. This is achieved by placing a decimal code number in binary on the 16 output lines. This command number N is then decoded by the wind tunnel hardware and some operation performed.

The command number for each operation is determined by the code

\[
N = \text{Device Address} + \text{Data Operation Value} + \text{Data Value}
\]
using software values only. For example, if the scanivalve (1) is required to step on one port, the following command number would be sent to the wind tunnel

\[ N = 46 + 0 + 16384 = 16430 \]

Summary

The interface between wind tunnel hardware and the computer is by means of 16 output lines and 16 input lines plus control lines. The voltage levels on the output lines are controlled by the software generated command numbers described above. The voltage levels on the input lines are controlled by software selected tunnel hardware. All transfer of data is accompanied by a handshaking procedure to ensure correct sequencing of communication operations.
**Title and Subtitle**

Control Software for Two Dimensional Airfoil Tests Using a Self-Streamlining Flexible Walled Transonic Test Section

**Abstract**

The current operation of the Transonic Self-Streamlining Wind Tunnel (TSWT) involves on-line data acquisition with automatic wall adjustment. A tunnel run consists of streamlining the walls from known starting contours in iterative steps and acquiring model data. Each run performs what is described as a streamlining cycle. The associated control software is presented here.

**Key Words (Suggested by Author(s))**

Facilities, research and support
Flexible wall wind tunnel
End of Document