TAIR - a Transonic Airfoil Analysis Computer Code

F. Carroll Dougherty, Terry L. Holst, Karen L. Gundy and Scott D. Thomas

May 1981
TAIR - a Transonic Airfoil Analysis Computer Code

F. Carroll Dougherty
Terry L. Holst, Ames Research Center, Moffett Field, California
Karen L. Gundy
Scott D. Thomas, Informatics Inc., Palo Alto, California
TAIR - A TRANSONIC AIRFOIL ANALYSIS COMPUTER CODE

F. Carroll Dougherty, Terry L. Holst, Karen L. Gundy, and Scott D. Thomas

Ames Research Center

SUMMARY

The operation of the TAIR (Transonic AIRfoil) computer code, which uses a fast, fully implicit algorithm to solve the conservative full-potential equation for transonic flow fields about arbitrary airfoils, is described. The code description is given in two levels of sophistication: simplified operation and detailed operation. In addition, detailed descriptions of the program organization and theory are provided to simplify modification of TAIR for new applications. Examples with input and output are given for a wide range of cases, including incompressible, subcritical compressible, and transonic calculations.

1. INTRODUCTION

This report describes the operation of TAIR, a transonic airfoil analysis computer program written in FORTRAN. TAIR solves the conservative full-potential equation for the steady transonic flow field about an arbitrary airfoil immersed in a subsonic free stream. The full-potential formulation is considered exact under the assumptions of irrotational, isentropic, and inviscid flow. These assumptions are valid for a wide range of practical transonic flows typical of modern aircraft cruise conditions.

The primary features of TAIR include: (1) a new fully implicit iteration scheme (AF2) which is typically 4 to 10 times faster than classical successive-line overrelaxation (SLOR) algorithms; (2) a new, reliable artificial density spatial differencing scheme treating the conservative form of the full-potential equation; and (3) a numerical mapping procedure capable of generating curvilinear, body-fitted finite-difference grids about arbitrary airfoil geometries (see refs. 1-3 for theoretical details of these features). The coding in TAIR includes many useful options including: (1) several airfoil-geometry input options, (2) flexible user control over program output, (3) a "built-in" default solution, (4) numerous comments to facilitate program use or modification, and (5) a multiple solution capability.

The most important aspect of TAIR is algorithm reliability. Because this characteristic is generally regarded as the most important quality for a user-oriented CFD computer program, it was emphasized during the development of TAIR. The reliability of TAIR comes from two sources: the algorithm, including the AF2 iteration scheme and the artificial density differencing scheme designed to simulate rotated differencing (ref. 4), and a convergence monitoring section of logic (SUBROUTINE AUTO), which automatically updates solution parameters in an attempt to speed convergence or prevent divergence. Use of
the convergence monitoring logic in TAIR allows the use of "default mode" (ref. 3). In default mode, only three inputs are changed from case to case, free-stream Mach number, angle of attack, and airfoil coordinates. Other parameters, including relaxation factors, acceleration parameters, and temporal-damping coefficients are either held fixed or are automatically adjusted by internal computer code logic. This feature greatly simplifies code operation, especially for inexperienced users. Generally speaking, in the default mode less than 1% of all cases diverge — a considerably better performance than most CFD codes run in any mode of operation.

Besides reliability and simplicity, another important aspect of the TAIR code is speed. The average flow-solver run time for the default mesh (149x30) is about 9 sec. Most subcritical or weak-shock cases require less time and most strong-shock cases (local Mach numbers exceeding 1.3) require more time. The average grid generation time is about 2 sec. Both of these times are on the Ames CDC 7600 computer. The program requires only a modest amount of storage — 153,000 words octal (~55,000 decimal) — and is written to be as transportable as possible.

Instructions for the use of TAIR follow in subsequent chapters. Simplified usage, that is, essentially default mode usage, is described in chapter 2. Chapter 3 describes more detailed operation; it is necessary reading for any major deviation from default mode usage. Chapter 4 discusses program organization, and chapter 5 discusses theory; both of these chapters would be of interest to anyone contemplating a major modification of TAIR. Chapter 6 presents a discussion of results obtained with TAIR for a variety of different cases (CASE HISTORY). Appendix A discusses the appropriate changes required to increase program dimensions, and appendix B provides a table of program variable names with brief descriptions which are cross-referenced with variable names used in the theoretical development.

2. SIMPLIFIED OPERATION

The following sections of this chapter discuss the use of the TAIR computer program for simplified operation: the inputs and outputs, some numerical examples, and the error safeguards built into the program. This level of operation allows the calculation of standard airfoil cases in the transonic regime, using a set of optimized acceleration parameters stored in the code and an automatic updating routine.

Most of the parameter inputs to TAIR are divided between two NAMELISTs. The aerodynamic and flow-field convergence parameters are found in NAMELIST FLOWIN (defined in SUBROUTINE INITL), and the geometric and grid convergence parameters are found in NAMELIST GRIDIN (defined in SUBROUTINE GREGEN). Airfoil coordinates other than those generated in TAIR would be read from cards after the two NAMELISTs. More information regarding these inputs, including formats, individual parameter descriptions, and acceptable parameter ranges, is given in the INPUT/OUTPUT section at the end of the chapter. Default values for the NAMELIST parameters are initialized in INITL and GREGEN. With the NAMELIST format, the user needs to change only the values that differ from the
default values. If no parameters are changed, the predetermined default solution is executed, as discussed in example 1 of the INPUT/OUTPUT examples section.

The following sections define a partial list of input parameters a general user of TAIR might wish to change. For a more complete list of input parameters, or if the values of some of these input parameters vary widely from the default case or if certain options require more changes than are mentioned here, the user is directed to chapter 3.

In this chapter and in the following chapters, all program variable and subroutine names are capitalized. The first section describes the variables in the FLOWIN NAMELIST; the second identifies the parameters in the GRIDIN NAMELIST. The third section discusses three example cases, detailing the proper uses of the input parameters, the correct ordering of the data cards, and the formats required. The fourth section describes briefly the error safeguards built into the program, and the last section contains the actual input data decks and the output generated for the example cases.

Description of FLOWIN NAMELIST Parameters

ALPH Angle of attack, deg. Acceptable values: real numbers between -10.0 and +10.0. Default value: 1.0.

MESH Determines which grid generation option is used.

MESH=0...Grid is generated within the program (see IOPT parameter in NAMELIST GRIDIN for details about airfoil geometry representation). This option also causes the grid coordinates to be written to a scratch file, logical unit 48.

MESH=1...This option reads the grid from unit 48 and is useful when a series of calculations is to be made with the same mesh (see NCASE parameter discussion later in this section). When using the MESH=1 option, the GRIDIN NAMELIST data card should be omitted.

MESH=2...This option reads an initial solution grid from unit 48 and adapts it to a new airfoil. The user should read chapter 3 before selecting this option.

MESH=3...This option reads a mesh created by the program GRAPE (ref. 5). The user should see chapter 3 for further instructions.

Acceptable values: 0,1,2,3. Default value: 0.

MINF Free-stream Mach number. Acceptable values: real numbers between 0.0 and 1.0, exclusive. Default value: 0.75.
NCASE  Number of solutions per job. If multiple cases are run with the same airfoil, computing time can be saved by setting MESH=1 for the second and each succeeding case (see INPUT/OUTPUT example 3). For each new case, all NAMELIST parameters are reset to the default values. Thus, the second, third, etc., cases must each reset the appropriate solution parameters through the NAMELIST READ statements. Acceptable values: any positive integer. Default value: 1.

NI  The number of mesh points in the $\xi$ direction, numbered around the airfoil in a clockwise direction (see fig. 1). Note: if input value of NI varies widely from the default value, the user should see chapter 3. Acceptable values: 50 to 151. Default value: 149.

NJ  The number of mesh points in the $\eta$ direction, numbered from the outer boundary inward to the airfoil surface (see fig. 1). Note: if input value of NJ varies widely from the default value, the user should see chapter 3. An NI/NJ ratio of 4 or 5 works best with the flow solver in this code. Acceptable values: 12 to 31. Default value: 30.

NOUT  Primary output control parameter. NOUT=0 suppresses all output except for a few error messages. Table 1 shows each NOUT value, the type of output associated with it, and the amount of output generated (in number of pages). The page totals shown do not include error message printouts or update messages, so actual numbers may be slightly larger. For more information about a specific output format, see the INPUT/OUTPUT examples. Acceptable values: integers between 0 and 9. Default value: 5.

NSTEPS  Maximum number of iterations of the flow solver. Most cases should converge in less than the default number. Acceptable values: positive integers. Default value: 200.

Description of GRIDIN NAMELIST Parameters

IOPT  Determines which geometry option is used.

IOPT=1...This generates a NACA 00XX type airfoil. The input parameter TMAX determines the thickness (see below).

IOPT=2...This IOPT option is a circular-arc airfoil. (See chap. 3 for more detail.)

IOPT=3...A circular cylinder cross section is generated. (Read chap. 3 for more information.)

IOPT=4...This IOPT permits the user to read in an arbitrary airfoil geometry. The first card in the data deck should be the title of the airfoil, with a maximum of 28 characters in a 7A4 format (right justified with left "*" fill; see example at the end of the chapter). The second card contains the number of airfoil coordinate pairs in the data deck (NPTS)
in I5 format. This number is limited to 151 because of program dimensions. The next cards contain the coordinates XB and YB, in a 2F10.5 format. These coordinates should be nondimensionalized by chord, but will automatically be scaled if they are not. They are ordered starting with the lower trailing-edge point at \( X = 1.0 \), moving clockwise around the airfoil, and ending with the upper trailing edge point at \( X = 1.0 \).

**IOPT=5**...The Korn airfoil is used for this option (airfoil 75-06-12 in reference 6).

Acceptable values: 1,2,3,4,5. Default value: 1.

**TMAX** Airfoil thickness parameter, nondimensionalized by chord. This value is necessary for the IOPT=1 and IOPT=2 options. Acceptable values: real numbers between 0.0 and 1.0, exclusive. Default value: 0.12.

**Descriptions of INPUT/OUTPUT Examples**

This section discusses input/output examples for three cases: (1) the default case, (2) a user-supplied airfoil case (IOPT=4), and (3) a simple multiple-solution case (NCASE=4). Each example is shown with the required input data and the resulting output (at the end of this chapter). The first example shows the input data and output for the default case stored in TAIR. Note that no parameters of any sort need to be modified for this case, which involves a NACA 0012 airfoil at a free-stream Mach number of 0.75 and an angle of attack of 1°. The grid is generated within TAIR (MESH=0) and has dimensions of 149 by 30. The flow solver for this case requires 74 iterations (about 7 sec of CPU time on the CDC 7600 computer) and the grid generation requires 20 iterations (about 2 sec of CPU time).

Output for the default case includes a banner and three pages of input data: the FLOWIN and GRIDIN NAMELIST parameters and the initial airfoil coordinates, nondimensionalized by chord. The next page of output shows the airfoil coordinates, XB and YB, after they have been interpolated and clustered about the airfoil. The ARC LENGTH, normalized by chord, is calculated from the lower trailing-edge point to the upper trailing-edge point. DS is the first difference of the arc length. Note that the N1th point is not printed, since it is identical to the first point because of wrap-around periodicity. Following the interpolated airfoil coordinates are the messages printed by the automatic updating SUBROUTINE AUTO discussed in the next section of this chapter.

The last page of output gives the airfoil surface solution. The first line of output at the top of this page displays the airfoil and flow conditions used for this case. The second line lists the number of iterations needed for convergence; the maximum residual of the last iteration (RMAX); the number of supersonic points (NSP); and the coefficient of lift calculated using the circulation and the Kutta-Joukowski theorem \((CL(CLR))\). The surface solution is divided into three sets of columns numbered clockwise around the
airfoil starting at the lower surface trailing edge. These columns list the X
and Y coordinates normalized by chord, X/C and Y/C; the coefficient of pressure
at the airfoil surface; the density; and the surface Mach number. It should
be pointed out that the values shown are stored at the halfway points relative
to the original mesh, that is, (I+1/2, J+1/2). The last line of output on
this page gives the lift, wave-drag, and quarter-chord moment coefficients
determined from the final surface solution by numerical integration (trapezoid
rule).

The second INPUT/OUTPUT example employs the IOPT=4 option with the theo-
retical CAST 7 airfoil coordinates (ref. 7) and shows the order and format of
the airfoil coordinates as well as the appropriate changes for the parameters
in the FLOWIN and GRIDIN NAMELISTs. The chosen case runs the CAST 7 airfoil
with a Mach number of 0.7 and an angle of attack of 1.5°. The output control
parameter NOUT has been set to 3, limiting the total solution output to the
one page airfoil surface solution. Note that the title of the airfoil on the
first data card is right justified in the 28 space field and the remaining
spaces are filled with asterisks. The solution converged in 85 iterations
(about 9 sec of CPU time) and produced 191 supersonic points, a lift coeffi-
cient of 1.0008, and a wave-drag coefficient of 0.0042.

The third INPUT/OUTPUT example is a simple multiple-solution run (NCASE=4)
in which four solutions for the Korn airfoil (IOPT=5) are obtained. The Mach
number in each case is held fixed at 0.7 while the angle of attack varies from
0° to 2°. Since this example could be used for a lift versus alpha curve,
NOUT=2 is used to reduce the output. Note that the MESH=1 option is used, so
the GRIDIN card is omitted for the last three solutions. Also, parameter
values that differ from the default values must be reset on each FLOWIN card.
This is because all parameters are reset to default values at the beginning of
each solution. The NCASE change, however, need only be made on the first data
card. The NOUT=2 specification considerably reduces the amount of output.
Only the number of iterations, the maximum residual, the number of supersonic
points, the lift coefficient calculated from the circulation, and the lift, drag, and moment coefficients computed from the pressure integration are
printed for each of the four cases.

Error Safeguards

TAIR includes in its programming many checks and safeguards to ensure that
the code runs smoothly. Throughout the program, parameters and arrays are
checked for incorrect or out-of-range values. Two subroutines carry the bulk
of this safeguarding, SUBROUTINES CHECK and AUTO. If any value is corrected,
or if the program is stopped, there is always an error message generated
explaining the problem and the steps taken to resolve it. If NOUT is less
than 4, however, the messages printed by the AUTO updating procedure will be
suppressed. The remaining safeguards are contained in various subroutines and
are discussed in the individual subroutine descriptions in chapter 4.
INPUT/OUTPUT example 1- The following data cards were used for INPUT/OUTPUT example 1—the default case. The output for this case is on the next six pages.

$FLOWIN  $
$GRINDIN  $
APPLIED COMPUTATIONAL AERODYNAMICS BRANCH

NASA - AMES RESEARCH CENTER

MOFFETT FIELD, CA 94035

THIS COMPUTER PROGRAM SOLVES THE CONSERVATIVE FULL POTENTIAL EQUATION FOR THE
TRANSONIC FLOW AROUND AN ARBITRARY AIRFOIL, USING A FULLY IMPLICIT APPROXIMATE
FACTORIZATION SCHEME (AF2).

WRITTEN BY

DR. TERRY L. HOLST

AND

F. CARROLL DOUGHERTY
SFLOWIN
MINF = .75E+00,
NI = 149,
NJ = 38,
ALPH = .1E+01,
NSTEPS = 200,
NOUT = 5,
NOUT1 = 201,
NOUT2 = 8,
IINCR = 3,
JINCR = 2,
NCASE = 1,
ERR = -.1E-01,
OMEGA = .18E+01,
BETA = .45E+01,
ALOW = .7E-01,
AHIGH = .15E+01,
K = 8,
M = 8,
RGAM = .1E+01,
MESH = 8,
CON = .2E+01,
NRING = 3,
AFAC = .14E+02,
G = .14E+01,
NDIF = 8,
IAUTO = 1,
ICHECK = 1,
SEND
SGRIDIN
BINN  = -.1E+01,
RADMAX = .6E+01,
IOPT  = 1,
TMAX  = .12E+08,
IOPEN = 0,
XC    = .2E+08,
IOUT  = 1,
HTMAX = .4E+01,
WDTMX = .6E+01,
MAXIT = 50,
OMEG  = .2E+01,
KGRID = 0,
MGRID = 8,
AHGRID = .2E-01,
ALGRID = .1E-04,
ERGRID = .1E-01,
XCN   = .5E+08,
YCN   = 8.8,
SEND
### INITIAL AIRFOIL COORDINATES FROM INPUT ###

**---------- NACA 6612 AIRFOIL ----------**

<table>
<thead>
<tr>
<th>I</th>
<th>XB</th>
<th>YB</th>
<th>I</th>
<th>XB</th>
<th>YB</th>
<th>I</th>
<th>XB</th>
<th>YB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.6000</td>
<td>.0000</td>
<td>51</td>
<td>.23785</td>
<td>.85861</td>
<td>101</td>
<td>.27493</td>
<td>.85937</td>
</tr>
<tr>
<td>2</td>
<td>.5935</td>
<td>.0091</td>
<td>52</td>
<td>.23981</td>
<td>.85795</td>
<td>102</td>
<td>.29428</td>
<td>.85948</td>
</tr>
<tr>
<td>3</td>
<td>.58928</td>
<td>.01836</td>
<td>53</td>
<td>.24180</td>
<td>.85712</td>
<td>103</td>
<td>.31357</td>
<td>.85943</td>
</tr>
<tr>
<td>4</td>
<td>.59595</td>
<td>.02857</td>
<td>54</td>
<td>.18589</td>
<td>.85610</td>
<td>104</td>
<td>.33343</td>
<td>.85929</td>
</tr>
<tr>
<td>5</td>
<td>.59281</td>
<td>.03882</td>
<td>55</td>
<td>.16966</td>
<td>.85489</td>
<td>105</td>
<td>.35359</td>
<td>.85981</td>
</tr>
<tr>
<td>6</td>
<td>.58878</td>
<td>.04915</td>
<td>56</td>
<td>.15403</td>
<td>.85351</td>
<td>106</td>
<td>.37481</td>
<td>.85827</td>
</tr>
<tr>
<td>7</td>
<td>.58487</td>
<td>.05952</td>
<td>57</td>
<td>.13982</td>
<td>.85195</td>
<td>107</td>
<td>.39456</td>
<td>.85759</td>
</tr>
<tr>
<td>8</td>
<td>.58109</td>
<td>.07007</td>
<td>58</td>
<td>.12456</td>
<td>.85021</td>
<td>108</td>
<td>.41555</td>
<td>.85675</td>
</tr>
<tr>
<td>9</td>
<td>.57744</td>
<td>.08078</td>
<td>59</td>
<td>.10990</td>
<td>.84830</td>
<td>109</td>
<td>.43649</td>
<td>.85579</td>
</tr>
<tr>
<td>10</td>
<td>.57394</td>
<td>.09158</td>
<td>60</td>
<td>.90988</td>
<td>.84622</td>
<td>110</td>
<td>.45756</td>
<td>.85475</td>
</tr>
<tr>
<td>11</td>
<td>.56561</td>
<td>.10238</td>
<td>61</td>
<td>.58575</td>
<td>.84439</td>
<td>111</td>
<td>.47979</td>
<td>.85358</td>
</tr>
<tr>
<td>12</td>
<td>.55446</td>
<td>.11325</td>
<td>62</td>
<td>.47424</td>
<td>.84161</td>
<td>112</td>
<td>.50099</td>
<td>.85219</td>
</tr>
<tr>
<td>13</td>
<td>.54351</td>
<td>.12411</td>
<td>63</td>
<td>.36429</td>
<td>.83897</td>
<td>113</td>
<td>.52122</td>
<td>.85078</td>
</tr>
<tr>
<td>14</td>
<td>.53276</td>
<td>.13505</td>
<td>64</td>
<td>.25434</td>
<td>.83648</td>
<td>114</td>
<td>.54146</td>
<td>.84929</td>
</tr>
<tr>
<td>15</td>
<td>.52114</td>
<td>.14606</td>
<td>65</td>
<td>.14439</td>
<td>.83366</td>
<td>115</td>
<td>.56169</td>
<td>.84771</td>
</tr>
<tr>
<td>16</td>
<td>.50969</td>
<td>.15714</td>
<td>66</td>
<td>.03436</td>
<td>.83068</td>
<td>116</td>
<td>.58195</td>
<td>.84606</td>
</tr>
<tr>
<td>17</td>
<td>.49742</td>
<td>.16831</td>
<td>67</td>
<td>.5256</td>
<td>.82764</td>
<td>117</td>
<td>.60233</td>
<td>.84435</td>
</tr>
<tr>
<td>18</td>
<td>.48434</td>
<td>.17954</td>
<td>68</td>
<td>.41924</td>
<td>.82459</td>
<td>118</td>
<td>.62279</td>
<td>.84258</td>
</tr>
<tr>
<td>19</td>
<td>.47045</td>
<td>.19086</td>
<td>69</td>
<td>.3113</td>
<td>.82125</td>
<td>119</td>
<td>.64330</td>
<td>.84086</td>
</tr>
<tr>
<td>20</td>
<td>.45573</td>
<td>.20226</td>
<td>70</td>
<td>.20122</td>
<td>.81791</td>
<td>120</td>
<td>.66390</td>
<td>.83911</td>
</tr>
<tr>
<td>21</td>
<td>.44022</td>
<td>.21382</td>
<td>71</td>
<td>.09719</td>
<td>.81449</td>
<td>121</td>
<td>.68463</td>
<td>.83738</td>
</tr>
<tr>
<td>22</td>
<td>.42490</td>
<td>.22555</td>
<td>72</td>
<td>.00405</td>
<td>.81098</td>
<td>122</td>
<td>.70555</td>
<td>.83512</td>
</tr>
<tr>
<td>23</td>
<td>.40875</td>
<td>.23743</td>
<td>73</td>
<td>.00108</td>
<td>.80739</td>
<td>123</td>
<td>.72650</td>
<td>.83280</td>
</tr>
<tr>
<td>24</td>
<td>.39281</td>
<td>.24947</td>
<td>74</td>
<td>.00005</td>
<td>.80373</td>
<td>124</td>
<td>.74753</td>
<td>.83047</td>
</tr>
<tr>
<td>25</td>
<td>.37704</td>
<td>.26166</td>
<td>75</td>
<td>.00000</td>
<td>.80000</td>
<td>125</td>
<td>.76861</td>
<td>.82804</td>
</tr>
<tr>
<td>26</td>
<td>.36144</td>
<td>.27397</td>
<td>76</td>
<td>.00000</td>
<td>.79637</td>
<td>126</td>
<td>.78979</td>
<td>.82554</td>
</tr>
<tr>
<td>27</td>
<td>.34600</td>
<td>.28641</td>
<td>77</td>
<td>.00000</td>
<td>.79279</td>
<td>127</td>
<td>.81102</td>
<td>.82301</td>
</tr>
<tr>
<td>28</td>
<td>.33075</td>
<td>.29901</td>
<td>78</td>
<td>.00000</td>
<td>.78919</td>
<td>128</td>
<td>.83235</td>
<td>.82056</td>
</tr>
<tr>
<td>29</td>
<td>.31568</td>
<td>.31176</td>
<td>79</td>
<td>.00000</td>
<td>.78560</td>
<td>129</td>
<td>.85370</td>
<td>.81818</td>
</tr>
<tr>
<td>30</td>
<td>.30081</td>
<td>.32463</td>
<td>80</td>
<td>.00000</td>
<td>.78201</td>
<td>130</td>
<td>.87515</td>
<td>.81580</td>
</tr>
<tr>
<td>31</td>
<td>.28612</td>
<td>.33763</td>
<td>81</td>
<td>.00000</td>
<td>.77843</td>
<td>131</td>
<td>.89660</td>
<td>.81349</td>
</tr>
<tr>
<td>32</td>
<td>.27162</td>
<td>.35078</td>
<td>82</td>
<td>.00000</td>
<td>.77486</td>
<td>132</td>
<td>.91816</td>
<td>.81118</td>
</tr>
<tr>
<td>33</td>
<td>.25731</td>
<td>.36408</td>
<td>83</td>
<td>.00000</td>
<td>.77130</td>
<td>133</td>
<td>.93976</td>
<td>.80886</td>
</tr>
<tr>
<td>34</td>
<td>.24318</td>
<td>.37753</td>
<td>84</td>
<td>.00000</td>
<td>.76776</td>
<td>134</td>
<td>.96140</td>
<td>.80655</td>
</tr>
<tr>
<td>35</td>
<td>.22922</td>
<td>.39113</td>
<td>85</td>
<td>.00000</td>
<td>.76423</td>
<td>135</td>
<td>.98306</td>
<td>.80425</td>
</tr>
<tr>
<td>36</td>
<td>.21536</td>
<td>.40486</td>
<td>86</td>
<td>.00000</td>
<td>.76071</td>
<td>136</td>
<td>.99474</td>
<td>.80195</td>
</tr>
<tr>
<td>37</td>
<td>.20158</td>
<td>.41872</td>
<td>87</td>
<td>.00000</td>
<td>.75721</td>
<td>137</td>
<td>.96561</td>
<td>.79972</td>
</tr>
<tr>
<td>38</td>
<td>.18789</td>
<td>.43270</td>
<td>88</td>
<td>.00000</td>
<td>.75372</td>
<td>138</td>
<td>.94658</td>
<td>.79750</td>
</tr>
<tr>
<td>39</td>
<td>.17428</td>
<td>.44680</td>
<td>89</td>
<td>.00000</td>
<td>.74926</td>
<td>139</td>
<td>.92755</td>
<td>.79529</td>
</tr>
<tr>
<td>40</td>
<td>.16075</td>
<td>.46103</td>
<td>90</td>
<td>.00000</td>
<td>.74481</td>
<td>140</td>
<td>.90853</td>
<td>.79309</td>
</tr>
<tr>
<td>41</td>
<td>.14730</td>
<td>.47540</td>
<td>91</td>
<td>.00000</td>
<td>.74038</td>
<td>141</td>
<td>.88952</td>
<td>.79089</td>
</tr>
<tr>
<td>42</td>
<td>.13392</td>
<td>.48990</td>
<td>92</td>
<td>.00000</td>
<td>.73596</td>
<td>142</td>
<td>.87053</td>
<td>.78868</td>
</tr>
<tr>
<td>43</td>
<td>.12060</td>
<td>.50452</td>
<td>93</td>
<td>.00000</td>
<td>.73156</td>
<td>143</td>
<td>.85155</td>
<td>.78647</td>
</tr>
<tr>
<td>44</td>
<td>.10734</td>
<td>.51929</td>
<td>94</td>
<td>.00000</td>
<td>.72719</td>
<td>144</td>
<td>.83259</td>
<td>.78427</td>
</tr>
<tr>
<td>45</td>
<td>.09414</td>
<td>.53411</td>
<td>95</td>
<td>.00000</td>
<td>.72286</td>
<td>145</td>
<td>.81364</td>
<td>.78208</td>
</tr>
<tr>
<td>46</td>
<td>.08099</td>
<td>.54906</td>
<td>96</td>
<td>.00000</td>
<td>.71854</td>
<td>146</td>
<td>.79470</td>
<td>.77989</td>
</tr>
<tr>
<td>47</td>
<td>.06790</td>
<td>.56405</td>
<td>97</td>
<td>.00000</td>
<td>.71424</td>
<td>147</td>
<td>.77577</td>
<td>.77770</td>
</tr>
<tr>
<td>48</td>
<td>.05485</td>
<td>.57910</td>
<td>98</td>
<td>.00000</td>
<td>.70994</td>
<td>148</td>
<td>.75685</td>
<td>.77551</td>
</tr>
<tr>
<td>49</td>
<td>.04184</td>
<td>.59420</td>
<td>99</td>
<td>.00000</td>
<td>.70565</td>
<td>149</td>
<td>.73794</td>
<td>.77332</td>
</tr>
<tr>
<td>50</td>
<td>.02889</td>
<td>.60934</td>
<td>100</td>
<td>.00000</td>
<td>.70136</td>
<td>150</td>
<td>.71904</td>
<td>.77113</td>
</tr>
</tbody>
</table>

11
### Final Airfoil Coordinates After Cubic Spline Interpolation

<table>
<thead>
<tr>
<th>I</th>
<th>XB</th>
<th>YB</th>
<th>ARC LENGTH</th>
<th>DS</th>
<th>I</th>
<th>XB</th>
<th>YB</th>
<th>ARC LENGTH</th>
<th>DS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.38688</td>
<td>-0.86888</td>
<td>0.88888</td>
<td>0.88853</td>
<td>51</td>
<td>21.8188</td>
<td>-0.5787</td>
<td>0.75851</td>
<td>0.61897</td>
</tr>
<tr>
<td>2</td>
<td>2.39586</td>
<td>-0.80815</td>
<td>0.81015</td>
<td>0.81142</td>
<td>62</td>
<td>20.1931</td>
<td>-0.56784</td>
<td>0.8173</td>
<td>0.62164</td>
</tr>
<tr>
<td>3</td>
<td>3.39529</td>
<td>-1.09719</td>
<td>1.09719</td>
<td>1.09719</td>
<td>73</td>
<td>20.5045</td>
<td>-0.55608</td>
<td>0.8698</td>
<td>0.62654</td>
</tr>
<tr>
<td>4</td>
<td>4.39368</td>
<td>-0.8978</td>
<td>0.8978</td>
<td>0.8978</td>
<td>84</td>
<td>21.9552</td>
<td>-0.54489</td>
<td>0.92395</td>
<td>0.6353</td>
</tr>
<tr>
<td>5</td>
<td>5.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>95</td>
<td>23.4448</td>
<td>-0.53365</td>
<td>0.9789</td>
<td>0.64469</td>
</tr>
<tr>
<td>6</td>
<td>6.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>106</td>
<td>24.9328</td>
<td>-0.52232</td>
<td>1.0323</td>
<td>1.02143</td>
</tr>
<tr>
<td>7</td>
<td>7.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>117</td>
<td>26.4212</td>
<td>-0.51101</td>
<td>1.08715</td>
<td>1.06353</td>
</tr>
<tr>
<td>8</td>
<td>8.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>128</td>
<td>27.9091</td>
<td>-0.50008</td>
<td>1.13675</td>
<td>1.10761</td>
</tr>
<tr>
<td>9</td>
<td>9.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>139</td>
<td>29.3963</td>
<td>-0.49005</td>
<td>1.18647</td>
<td>1.1525</td>
</tr>
<tr>
<td>10</td>
<td>10.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>150</td>
<td>30.8775</td>
<td>-0.48002</td>
<td>1.23634</td>
<td>1.19837</td>
</tr>
<tr>
<td>11</td>
<td>11.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>161</td>
<td>32.3583</td>
<td>-0.4700</td>
<td>1.28637</td>
<td>1.24528</td>
</tr>
<tr>
<td>12</td>
<td>12.39286</td>
<td>-0.88829</td>
<td>0.88829</td>
<td>0.88829</td>
<td>172</td>
<td>33.8391</td>
<td>-0.46008</td>
<td>1.33649</td>
<td>1.29329</td>
</tr>
</tbody>
</table>

*** Parameter Update by Subroutine AUTO... Beta = 3.3 ***

*** Parameter Update by Subroutine AUTO... Con = -1.2 ***
<table>
<thead>
<tr>
<th>I</th>
<th>X/C</th>
<th>Y/C</th>
<th>CP</th>
<th>RHO</th>
<th>M</th>
<th>I</th>
<th>X/C</th>
<th>Y/C</th>
<th>CP</th>
<th>RHO</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.9959</td>
<td>.8811</td>
<td>.6882</td>
<td>.8726</td>
<td>.5292</td>
<td>51</td>
<td>.2897</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>2</td>
<td>.9969</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>52</td>
<td>.2899</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>3</td>
<td>.9939</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>53</td>
<td>.2891</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>4</td>
<td>.9872</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>54</td>
<td>.2888</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>5</td>
<td>.9885</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>55</td>
<td>.2886</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>6</td>
<td>.9835</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>56</td>
<td>.2883</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>7</td>
<td>.9774</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>57</td>
<td>.2879</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>8</td>
<td>.9784</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>58</td>
<td>.2877</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>9</td>
<td>.9725</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>59</td>
<td>.2871</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
<tr>
<td>10</td>
<td>.9536</td>
<td>.8811</td>
<td>.6888</td>
<td>.8726</td>
<td>.5292</td>
<td>60</td>
<td>.2863</td>
<td>.8576</td>
<td>.5686</td>
<td>.6539</td>
<td>.9648</td>
</tr>
</tbody>
</table>

**Solution after 74 iterations, $R_{max} = .3456e-04$, $N_S = 11W$, $C_{c(D)} = .2441**
**INPUT/OUTPUT example 2**—These data cards are the input for INPUT/OUTPUT example 2—an IOPT=4 case using the CAST 7 airfoil. Please note the format of the airfoil coordinates. The output for this case is shown on the next page.

```
$FLOWIN MINF=0.7,ALPH=1.5,NOUT=3, $
$GRIDIN IOPT=4, $
*************** CAST 7 AIRFOIL
61
  1.00000  -0.00215
  0.97750  -0.00123
  0.95000  -0.00040
  0.92000   0.00079
  0.87500   0.00104
  0.81500  -0.00209
  0.75500  -0.00810
  0.69500   0.01566
  0.63500  -0.02372
  0.57500  -0.03193
  0.51500  -0.03854
  0.45500  -0.04430
  0.39500  -0.04836
  0.33500  -0.05027
  0.27500  -0.04965
  0.21500  -0.04632
  0.15500  -0.04040
  0.11500  -0.03518
  0.08750  -0.03182
  0.06500  -0.02713
  0.04750  -0.02363
  0.03500  -0.02085
  0.02500  -0.01853
  0.01750  -0.01663
  0.01250  -0.01505
  0.00750  -0.01270
  0.00500  -0.01082
  0.00350  -0.00927
  0.00200  -0.00713
  0.00100  -0.00506
  0.00040  -0.00319
  0.00000   0.00000
  0.00040   0.00337
  0.00100   0.00531
  0.00200   0.00754
  0.00350   0.01012
  0.00500   0.01229
  0.00750   0.01534
  0.01250   0.02011
  0.03500   0.03273
  0.04750   0.03770
  0.06500   0.04308
```

14
<table>
<thead>
<tr>
<th></th>
<th>I / X</th>
<th>Y / C</th>
<th>CP</th>
<th>RHO</th>
<th>M</th>
<th>I / X</th>
<th>Y / C</th>
<th>CP</th>
<th>RHO</th>
<th>M</th>
<th>I / X</th>
<th>Y / C</th>
<th>CP</th>
<th>RHO</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.9990</td>
<td>.9980</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
<tr>
<td>2</td>
<td>.9980</td>
<td>.9979</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
<tr>
<td>3</td>
<td>.9980</td>
<td>.9979</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
<tr>
<td>4</td>
<td>.9974</td>
<td>.9965</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
<tr>
<td>5</td>
<td>.9974</td>
<td>.9965</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
<tr>
<td>6</td>
<td>.9974</td>
<td>.9965</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
<tr>
<td>7</td>
<td>.9974</td>
<td>.9965</td>
<td>.459</td>
<td>.8779</td>
<td>.8779</td>
<td>51</td>
<td>.2938</td>
<td>.9459</td>
<td>.8569</td>
<td>.788</td>
<td>.7862</td>
<td>28</td>
<td>101</td>
<td>.9715</td>
<td>.9568</td>
</tr>
</tbody>
</table>

*** LIFT, WAVE DRAG, AND QUARTER-CORD MOMENT COEFFICIENTS...CL = 1.0000, CD = .0042, CM = .1479 ***

---

*CAST 7 AIRFOIL SURFACE SOLUTION....MINF = .788, ALPH = 1.600 DEG*
**INPUT/OUTPUT example 3**—The cards below form the data deck for the multiple case run for INPUT/OUTPUT example 3. Note that due to the use of the MESH=1 option, no GRIDIN card is necessary for the last three solutions. The output for this example is shown on the next page.

```
$FLOWIN  MINF=0.7, ALPH=0.0, NOUT=2, NCASE=4,  $
$GRIDIN  IOPT=5,   $
$FLOWIN  MINF=0.7, ALPH=0.5, NOUT=2, MESH=1,  $
$FLOWIN  MINF=0.7, NOUT=2, MESH=1,  $
$FLOWIN  MINF=0.7, ALPH=2.0, NOUT=2, MESH=1,  $
```


***** SOLUTION AFTER 29 ITERATIONS, RMAX = .2784E-04, NSP = 13, CL(CIR) = .5293 *****
*** LIFT, WAVE DRAG, AND QUARTER-CHORD MOMENT COEFFICIENTS.......CL = .5263, CD = -.5895, CM = -.1324 ***

***** SOLUTION AFTER 53 ITERATIONS, RMAX = .3347E-04, NSP = 37, CL(CIR) = .6286 *****
*** LIFT, WAVE DRAG, AND QUARTER-CHORD MOMENT COEFFICIENTS.......CL = .6258, CD = -.5884, CM = -.1328 ***

***** SOLUTION AFTER 67 ITERATIONS, RMAX = .4122E-04, NSP = 73, CL(CIR) = .7331 *****
*** LIFT, WAVE DRAG, AND QUARTER-CHORD MOMENT COEFFICIENTS.......CL = .7257, CD = .5891, CM = -.1301 ***

***** SOLUTION AFTER 102 ITERATIONS, RMAX = .3877E-04, NSP = 187, CL(CIR) = .9834 *****
*** LIFT, WAVE DRAG, AND QUARTER-CHORD MOMENT COEFFICIENTS.......CL = .9768, CD = .8341, CM = -.1294 ***
3. DETAILED OPERATION

This chapter discusses additional input/output options available in TAIR and provides more information about parameters defined in chapter 2. Additional INPUT/OUTPUT examples are included to embellish this more detailed discussion. The default values of the acceleration parameters listed in this section have been optimized for the standard transonic case. The use of some options requires that these acceleration parameters be changed to aid convergence or to prevent divergence. Some of the necessary changes are written into SUBROUTINE CHECK discussed in chapter 4 and, therefore, are implemented automatically. Other changes can be made by the user; they are discussed in this chapter. The examples show some of the different ways in which TAIR can be used, with appropriate changes in various parameters.

The input parameters discussed in this chapter include not only new variables, but also more complete information for some of the parameters described in chapter 2. If a particular parameter must be changed for a certain option, a note will be made in the description of that option to that effect. All program variable and subroutine names are capitalized. In addition, any program variable mentioned but not defined below is defined in appendix B.

Description of FLOWIN NAMELIST Parameters

**AFAC**

ALPHA multiplier parameter. The value of ALPHA (see definition in appendix B and more detailed discussion in chapter 5) near the airfoil boundary, that is, within NRING & lines of the airfoil boundary, is restricted to never fall below AFAC*ALOW. This allows ALOW to be made smaller, that is, larger effective pseudotime steps over most of the flow field, and, therefore, it provides for much faster convergence. NRING and ALOW are inputs and are defined later in this section. Acceptable values: real numbers between 1.0 and 20.0. Default value: 14.0.

**AHIGH**

Largest element in the ALPHA acceleration parameter sequence. This element represents the inverse of the minimum pseudotime step in a sequence of M elements. The larger values of ALPHA damp out the high-frequency errors in the solution. The combination of the two default ALOW and AHIGH parameters leads to the fastest convergence for the most cases. Acceptable values: real numbers greater than 0.5. Default value: 15.

**ALOW**

Smallest element in the ALPHA acceleration parameter sequence. This element represents the inverse of the maximum pseudotime step in a sequence of M elements. The smaller values in this sequence damp out the low-frequency errors. The default ALOW value (0.07) is very close to the stability bound — for instance ALOW=0.06 is usually unstable. It is insensitive to small changes to the grid (changing airfoils, etc.), but very sensitive to major changes to the number of points in the grid or to the spacing. The running of TAIR with a GRAPE grid is
one of these changes; ALOW=0.07 is unstable and must be increased to 0.08. This change will be made within CHECK if that subroutine is called. A large change in RADMAX or BINN in the GRIDIN NAMELIST might also necessitate a change in ALOW. Acceptable values: real numbers between 0.05 and 1.0. Default value: 0.07.

**BETA**

The βt coefficient. Proper specification of this parameter is very important for maintaining algorithm stability. SUBROUTINE AUTO contains a fairly complex modification or updating procedure for BETA, based on the changes in the average and maximum residuals and the NSP and circulation (GNP) buildups. If the user were to switch off SUBROUTINE AUTO, the default BETA would probably be too high for optimal convergence, but would maintain stability for most cases. Generally, small values of BETA (-1.0) are required for small regions of supersonic flow, moderate values of BETA (-3.0 to 4.0) are required for moderate regions of supersonic flow, etc. (See chap. 5 for a detailed description of the BETA update logic.) Acceptable values: positive real numbers. Default value: 4.5.

**CON**

Parameter controlling amount of upwind bias on the density. The default value is set for strong-shock cases and should be lowered by the user for easier cases if AUTO (which decreases CON automatically in these cases) has been turned off. The higher the CON value, the more the shock profile is smeared. Acceptable values: real numbers between 0.5 and 4.0. Default value: 2.0.

**ERR**

Convergence criteria. There are two different convergence tests available in TATR. The sign of ERR designates which test is used, and the magnitude specifies the convergence tolerance.

ERR<0.0...The solution is declared converged after N iterations if the following three conditions are met:

\[
(1) \frac{RMAX^N}{RMAX} < |ERR|
\]

\[
(2) \frac{|NSP^N - NSP^{N-M}|}{NSP^N} \leq \frac{|ERR|}{2}
\]

\[
(3) \frac{|\Gamma^N - \Gamma^{N-M}|}{\Gamma^N} \leq \frac{|ERR|}{2}
\]

where NSP is the number of supersonic points, Γ is the circulation (GNP), and M is the number of elements in the ALPHA acceleration parameter sequence (see subsequent discussion of M). Suggested value: -0.01.
The solution is declared converged after \( N \) iterations if:

\[
\frac{R_{\text{MAX}}^N}{R_{\text{MAX}}} \leq \text{ERR}
\]

Suggested value: 0.001.

If \( R_{\text{MAX}}^N \) ever exceeds 10 times \( R_{\text{MAX}}^1 \) then the iteration process is terminated and a message is printed indicating solution divergence. Acceptable values: \( |\text{ERR}| \leq 0.05 \). Default value: -0.01.

**G**

Ratio of specific heats. TAIR can be used for computations involving different fluids by changing \( G \). Acceptable values: real numbers between 1.0 and 2.0. Default value: 1.4 (air).

**IAUTO**

Switch for SUBROUTINE AUTO.

- **IAUTO=1**...SUBROUTINE AUTO is turned on; solution parameters (BETA, CON, NDIF, and RGAM) are automatically updated in an attempt to optimize convergence.
- **IAUTO=0**...SUBROUTINE AUTO bypassed; default or user-supplied values remain unchanged.

Acceptable values: 0,1. Default value: 1.

**ICHECK**

Switch for SUBROUTINE CHECK.

- **ICHECK=1**...Input data checked for consistency. Both of the NAMELISTs and the airfoil coordinates (if read in) are checked. If any out-of-range or incorrect value is found then CHECK will either print a warning, correct the value, or cause the solution to be terminated.
- **ICHECK=0**...Input data not checked.

Acceptable values: 0,1. Default value: 1.

**IINCR**

Grid solution output increment. If the final grid coordinates are printed (NOUT=9), the \( \xi \)-direction values will be printed with an IINCR increment. Acceptable values: positive integers. Default value: 3.

**JINCR**

Grid solution output increment. If the final grid coordinates are printed (NOUT=9), the \( \eta \)-direction values will be printed with a JINCR increment. Acceptable values: positive integers. Default value: 2.

**K**

Starting element in the ALPHA sequence. \( K \) is increased by one before the sequence starts. Acceptable values: positive integers from 0 to \( M-1 \). Default value: 0.
M Number of elements in the ALPHA sequence. Acceptable values: integers greater than 1, greater than K, and less than 20. Default value: 8.

MESH Determines which grid generation option is used.

MESH=0...The grid is generated within the program (discussed in chap. 2).

MESH=1...This option reads the grid from unit 48. In addition to being useful for multiple-case runs, this option can be used to read the coordinates of a grid generated elsewhere. The required unformatted inputs are: first record — title of either the airfoil or the mesh used (28 characters); second record — the grid dimensions, NX and NY, and the third record — the grid coordinates, X and Y, read in using (X(I,J),Y(I,J),I=1,NX),J=1,NY). NX and NY should be the same as NI and NJ — if not, the program will automatically change NI and NJ to NX and NY. It is the user's responsibility to make sure that the program dimensions are not exceeded. See appendix A for directions on how to change the present program dimensions. The GRIDIN NAMELIST data card is not required for MESH=1.

MESH=2...This option reads an initial solution grid from logical unit 48 (same format as above), and adapts it to a new airfoil. This is accomplished by subtracting the difference between the original and new airfoils from each grid point, including the outer boundary grid points. This option is designed for small perturbations between the two airfoils (specifically numerical optimization). If two radically different airfoils are used, the program may generate a flow-field solution with slight inaccuracies. The GRIDIN data card with the airfoil data must be included with this option.

MESH=3...Implementation of this option causes TAIR to read a mesh created by a computer program called GRAPE (ref. 5). The airfoil selection is made within GRAPE, which generates the Poisson grid and writes the information to logical unit 7. TAIR reads this information from logical unit 48, which includes NX and NY, the number of \(\xi\) and \(\eta\) coordinates, respectively; TEOPEN, the vertical distance across the trailing edge (important if running an open trailing-edge case); and X and Y, the grid coordinates. The READ statement used for this option in TAIR is compatible with the WRITE statement in GRAPE. This option also performs two tasks necessary to make the GRAPE grid compatible with the TAIR flow solver. First, the \(\xi\) lines in GRAPE are numbered from the inner boundary (body surface) to the outer boundary and must be reordered to be consistent with TAIR. The second task concerns the trailing edge of the airfoil. TAIR has its first and N1th points in the
The number of solutions per job. If several cases using GRAPE-generated grids are run on one job, the mesh parameter must always be set to 3, neither the MESH=1 nor the MESH=2 READ format is compatible with the GRAPE WRITE format. If these GRAPE-generated grid solutions are combined with TAIR-generated grid solutions, the GRAPE cases must be run first, as the MESH=0 option will overwrite unit 48 with the TAIR-generated grid coordinates. INPUT/OUTPUT example 4 shows a multiple-case run that addresses this situation. Acceptable values: any positive integer. Default value: 1.

Rotated differencing parameter. NDIF must be turned on for cases with strong shocks at the trailing edge to maintain stability. SUBROUTINE AUTO automatically turns the rotated differencing on when it detects the shock moving to the trailing edge.

NDIF=0...No rotated differencing. The spatial differencing scheme is upwind biased along only the $\xi$ direction.

NDIF=1...Rotated differencing. The spatial differencing scheme is upwind biased along both the $\xi$ and $\eta$ directions.

Acceptable values: 0,1. Default value: 0.

Solution output frequency. An intermediate solution controlled by NOUT (CP's, densities, aerodynamic coefficients, etc.) will be printed every NOUT1 iterations. Note: when the iteration terminates either because the iteration limit (N=NSTEPS) is encountered or because of convergence, a solution is automatically printed. Acceptable values: positive integers. Default value: 201.
NOUT2 Convergence parameter output frequency. The convergence parameters, including RMAX, CMAX, NSP, ALPHA, BETA, and GNP, will be printed every iteration for the first $M$ iterations, and then every NOUT2 iterations. Acceptable values: positive integers. Default value: 8.

NRING Number of inner $\xi$ lines ($\eta = \text{constant lines}$) for which AFAC logic is applied. Acceptable values: integers between 2 and 10. Default value: 3.

OMEGA Relaxation parameter for the flow-solver algorithm. Linear stability of the AF2 algorithm is maintained for values of OMEGA between 0 and 2. In the present nonlinear case values slightly less than 2 are required for stability with the optimum value being near 1.8. Acceptable values: 0.0 to 2.0. Default value: 1.8.

RGAM Circulation relaxation parameter. If RGAM is too low, the circulation buildup and solution convergence will be slow. If RGAM is too high, the circulation will develop a divergent oscillation. The input value of RGAM is modified or updated as the solution continues by internal logic (MAIN and AUTO). (See chap. 5 for the theory behind this control.) RGAM's effect on convergence is relatively minor except in the case of a shock near the trailing edge. Under this circumstance a lower value of RGAM is automatically implemented when AUTO logic is activated (IAUTO=1). Acceptable values: real numbers between 0.2 and 1.5. Default value: 1.0.

Description of GRIDIN NAELIST Parameters

AHGRID Largest value in the ALPHA acceleration parameter sequence used for grid generation. Acceptable values: real numbers between 0.01 and 0.05. Default value: 0.02.

ALGRID Smallest value in the ALPHA acceleration parameter sequence used for grid generation. Acceptable values: numbers greater than 0.0 and smaller than 0.01. Default value: 0.00001.

BINN Stretching parameter affecting the grid-point distribution on the inner boundary (airfoil surface).

BINN=0.0...The grid points are distributed evenly. This option is not suitable for airfoils. A warning message will be printed by CHECK if this option is run with any geometry except the circular cylinder geometry (IOPT=3).

BINN=-2.0...The surface grid-point distribution in terms of arc length is read from cards. The first card has the number of points in the array, NS (format I5). The next cards contain the distribution points, starting with 0 at the lower trailing-edge point moving clockwise around the airfoil to 1 at the upper trailing-edge point (format 8F10.6). If NS and NI differ, the
arc-length distribution read from cards will be automatically scaled by cubic spline interpolation to match the NI dimension exactly.

BINN=-1.0...The grid-point distribution is established from a data statement, which was developed by using a circle plane conformal mapping about the NACA 0012 airfoil. If the number of points in the data statement (151) and the number in NI are different, then the arc-length distribution given by the data statement will be automatically scaled in number by cubic spline interpolation to match the NI dimension.

BINN.GT.1.0...(and .LT.10.0) The surface distribution is established from a clustering formula. Values closer to 1 will cluster more at the airfoil leading and trailing edges. Values of BINN much larger than 1 will cause the grid point distribution to approach equal spacing. The suggested airfoil value for this option is 1.05.

Acceptable values: -2.0, -1.0, 0.0, real numbers between 1.0 and 10.0, exclusive. Default value: -1.0.

ERGRID Convergence tolerance for grid generation. Acceptable values: real numbers less than or equal to 0.01. Default value: 0.01.

HTMAX Height of rectangular outer boundary (IOUT=3). Acceptable values: real numbers between 2.0 and 8.0. Default value: 4.0.

IOPEN Open or closed trailing-edge option (IOPT=1 only; IOPEN is ignored for all other values of IOPT).

IOPEN=0...Trailing edge is closed. The standard NACA 00XX airfoil profile is extended to a point at the trailing edge and renormalized.

IOPEN=1...Trailing edge is open. The program will generate a standard NACA 00XX airfoil with the associated opening at the trailing edge.

Acceptable values: 0,1. Default value: 0.

IOPT Determines which airfoil option is used.

IOPT=1...This generates a NACA-00XX-type airfoil with thickness determined by TMAX.

IOPT=2...This IOPT option produces a circular-arc airfoil. The thickness is determined by the TMAX parameter and the leading-edge bluntness by the XC parameter (discussed below). The clustering parameter BINN is automatically changed in CHECK (BINN=1.03) for this option to provide
more grid point clustering at the airfoil leading edge, the smallest element of the ALPHA sequence ALOW is increased to 0.1, and AFAC is increased to 20.

IOPT=3...This option produces a circular cylinder cross-section geometry. BINN is automatically set to zero in SUBROUTINE CHECK, providing equal spacing around the circular perimeter.

IOPT=4...This IOPT permits the user to read in an arbitrary airfoil geometry (see chap. 2).

IOPT=5...The Korn airfoil is used for this option (airfoil 75-06-12 in ref. 6).

Acceptable values: 1,2,3,4,5. Default value: 1.

IOUT Determines which outer boundary is used.

IOUT=1...This generates a circular outer boundary with radius RADMAX centered at XCN and YCN.

IOUT=2...The outer boundary is read from cards. First all X coordinates are read from 1 to NI-1 in the first record, followed by all the Y coordinates in the second record (format 8F10.0).

IOUT=3...This option generates a base rectangular outer boundary with semicircular upstream and downstream ends. The dimensions of the base rectangle are controlled by HTMAX and WDTHMX. When the rectangle becomes too elongated, more grid points in the η-direction will be required to maintain accuracy. In addition, the acceleration parameters may need to be reevaluated for some of these cases.

Acceptable values: 1,2,3. Default value: 1.

KGRID Starting element in the ALPHA sequence used for grid generation. Acceptable values: integers between 0 and MGRID-1. Default value: 0.

MAXIT Maximum number of iterations for grid generation. Acceptable values: integers between 10 and 100. Default value: 50.

MGRID Number of elements in ALPHA sequence used for grid generation. Acceptable values: integers greater than 1 and greater than KGRID. Default value: 8.

OMEG Relaxation factor for grid generation. Acceptable values: real numbers between 1.0 and 2.0. Default value: 2.0.

RADMAX Circular outer boundary radius (IOUT=1). The center of the circle is placed at XCN and YCN, and the radial distance is nondimensionalized.
by the chord. RADMAX is related to NJ; if RADMAX is increased dramatically, then NJ should be increased. Acceptable values: real numbers between 3.0 and 18.0. Default value: 6.0.

**WDTHMX** Width of rectangular outer boundary (IOUT=3). Suggested WDTHMX/HTMAX ratio should not greatly exceed 2. Acceptable values: real numbers between 4.0 and 12.0. Default value: 6.0.

**XC** Leading-edge bluntness parameter for IOPT=2 case. The leading edge of the circular-arc airfoil will be blunted according to the parameter XC. A parabola will be matched to the airfoil leading edge (both position and slope) at X/C=XC (see fig. 3). Smaller values of XC will produce smaller amounts of blunting. Once blunted, the airfoil is renormalized, causing the maximum thickness to be slightly larger than TMAX. Acceptable values: real numbers between 0.2 and 0.4. Default value: 0.2.

**XCN** The X coordinate for the center of the circular outer boundary (IOUT=1). XCN is normalized by chord and measured with respect to the airfoil leading edge. Acceptable values: real numbers between -1.0 and 1.0. Default value: 0.5.

**YCN** The Y coordinate for the center of the circular outer boundary (IOUT=1). YCN is normalized by chord and measured with respect to the airfoil leading edge. Acceptable values: real numbers between -1.0 and 1.0. Default value: 0.0.

**INPUT/OUTPUT Examples**

This section contains two INPUT/OUTPUT examples. The first (example 4) describes most of the output available beyond the default output (INPUT/OUTPUT example 1 in chap. 2 described all the default output, NOUT ≤ 5). The second example (example 5) compares two TAIR solutions, one with an internally generated grid and the other with a GRAPE-generated grid. Discussion is included for each example.

Example 4 is the default case, that is, NACA 0012 airfoil, MINF=0.75, and ALPH=1.0, with the NOUT value changed from the default value of 5 to 8. The input consists of the two NAMELIST data cards, with the NOUT change on the FLOWIN card. The output shown in example 4 is only the additional output generated beyond the default output previously discussed in chapter 2. The last NOUT option, NOUT=9, generates all of the above output plus the grid coordinates. Because of its length, this grid output will not be shown; however, it will be discussed at the end of this section.

The first page of output after the interpolated coordinates contains the convergence history for the ADI grid generation routine. From left to right, the columns are: N, the iteration number; CXMAX, the maximum correction for the X equation; I and J, the position of CXMAX; CYMAX, the maximum correction for the Y equation; I and J, the position of CYMAX; RXMAX, the maximum residual for the X equation; I and J, the position of RXMAX; RYMAX, the maximum...
residual for the Y equation; I and J, the position of RYMAX; and ALPHA, the acceleration parameter sequence used for the ADI method. This output will be printed if NOUT≥8. For a description of the internal grid generation algorithm, see chapter 5.

The next three pages of output in example 4 list the transformation metrics (A1, A2, A3, and XJ) for the ξ lines nearest the airfoil. The XJ quantity is the Jacobian of the transformation and is approximately proportional to the negative inverse of the cell area. The A1, A2, and A3 metric arrays used in TAIR are computed from

\[ A1 = \frac{[\nabla_x \cdot \nabla_x]}{XJ} \]
\[ A2 = \frac{[\nabla_x \cdot \nabla_y]}{[4 \times XJ]} \]
\[ A3 = \frac{[\nabla_y \cdot \nabla_y]}{XJ} \]

and differ slightly from the A1, A2, and A3 metric quantities discussed in the theory (see chap. 5). The A2 quantity indicates the level of cell skewness: if A2 is zero, the cell is orthogonal. A1 is approximately equal to the ratio of the normal side of the cell to the tangential side. A3 is approximately the inverse of A1. A1, A3, and XJ must always be negative due to the left-handed coordinate system used in the code. A change in sign in any of these quantities indicates a bad grid (overlapping coordinate lines) and will cause the solution to diverge. The A2 quantity will show both signs. For symmetric airfoils the sign of A2 above the airfoil will be the opposite of the sign below the airfoil. The A1 values are actually stored at (I+1/2, J), the A2 values at (I, J), the A3 values at (I, J+1/2), and the XJ values at (I+1/2, J+1/2). This output will be printed if NOUT≥8.

The convergence history for the flow solver is printed after the transformation metrics. The first column is the iteration number, N. CMAX and the following I and J are the maximum correction and its position. RMAX and the I and J to its right are the maximum residual and its position. RAVG is the residual averaged over the entire mesh. The BETA parameter used in TAIR varies from iteration to iteration providing IAUTO=1 (see chap. 5 for more information regarding the BETA update logic); therefore, the next column on the convergence history output page shows the value of BETA used for a given iteration. The next column displays the ALPHA sequence used by the flow solver, followed by the buildup of the number of supersonic points (NSP). The circulation and the coefficient of lift calculated from the circulation are shown in the last two columns. This information is automatically printed for the first M iterations, then printed only when N is a multiple of NOUT2. The update messages from SUBROUTINE AUTO, for example, the messages indicating changes in BETA and CON from example 4, are printed in the convergence history output as they occur. One final message is printed after the flow-solver convergence history output indicating that (1) the solution has converged (if the solution has passed the internal convergence test), (2) the solution iteration limit (N=NSTEPS) was encountered before the convergence test was passed, or (3) the solution is diverging. If the convergence history terminates in either of the first two modes, a flow-field solution, controlled by NOUT, is
printed. If the solution diverges no final solution will be printed. The flow-field convergence history output will be printed if NOUT>6.

The last two pages of output in example 4 show printer contour plots of the density and Mach number arrays plotted in the computational domain. These maps display a one-digit number (0-9) for each grid point in the flow field. For the density map this number (IVAR) is computed by

\[
IVAR = \text{IFIX} \left( \frac{9.999 \times (\text{RHO}(I,J) - \text{RHO}_{\text{min}})}{\text{RHO}_{\text{max}} - \text{RHO}_{\text{min}}} \right)
\]

For the Mach number map this number is computed by

\[
IVAR = \begin{cases} 
\text{IFIX} (9.999 \times M_{i,j}) & \text{if } M_{i,j} < 1.0 \\
\text{IFIX} [9.999 \times (M_{i,j} - 1.0)] & \text{if } M_{i,j} \geq 1.0
\end{cases}
\]

where \(M_{i,j}\) is the local Mach number. These maps start on the left side with the value I=ISTART and end on the right side with I=NI (upper vortex sheet boundary). The ISTART parameter is computed as the maximum of 1 and NI-132+1. Therefore, if NI>132, the left-hand portion of the flow field will not be displayed. The J index starts with one at the top of the map (outer boundary) and increases downward until it reaches NJ at the bottom of the map (airfoil boundary). Thus with this type of output the whole flow field can be qualitatively surveyed with only a single page of output.

The NOUT=9 option prints the finite-difference grid. The output printed for each value of I and J (incremented by IINC and JINC, respectively) includes the X and Y coordinates, the A1, A2, and A3 transformation metrics, and the Jacobian, XJ. For IINC=3 and JINC=2, this option produces eight pages of output.

The last INPUT/OUTPUT example (example 5) is a comparison of a case run with a GRAPE-generated grid and the same case run with the grid generated in TAIR. The airfoil is the NACA 0012, and the conditions are subcritical: MINF=0.63, ALPH=2.0. The input data for this example are shown at the beginning of example 5. As discussed earlier in this chapter, the GRAPE case must be run first. The final airfoil surface solution is all that is desired, so NOUT is set to 3. ALOW is updated for the GRAPE case (see the ALOW parameter discussion in the INPUT section). Note that the GRIDIN card is not used with the MESH=3 option, but must be included for the second case because of the grid generation within TAIR.

The two solutions show very minor differences. Perhaps the biggest difference lies in the wave-drag coefficient, which for this subcritical calculation should be zero. The TAIR-generated grid case predicts a wave-drag coefficient of -0.0007, an obvious error; the GRAPE-generated grid case produces the correct answer (within four digits). The primary speculation for this improved accuracy is associated with the airfoil boundary grid control available from GRAPE (ref. 8). The \(n\) mesh lines (\(\xi = \) constant lines) are forced to
approach the airfoil orthogonally (or as nearly orthogonally as possible), and as such, the accuracy of the wave-drag coefficient calculation improves.
INPUT/OUTPUT example 4- The following cards form the data deck for the first INPUT/OUTPUT example in chapter 3. This case uses the default parameters, changing only the amount of output. The additional output (over the default amount) is shown on the following pages.

$FLOWIN NOUT=8, $
$GRIDIN$

*** ADI CONVERGENCE HISTORY FOR NUMERICAL GRID GENERATION ***

<table>
<thead>
<tr>
<th>N</th>
<th>CXMAX</th>
<th>I</th>
<th>J</th>
<th>CYMAX</th>
<th>I</th>
<th>J</th>
<th>RXMAX</th>
<th>I</th>
<th>J</th>
<th>RVMAX</th>
<th>I</th>
<th>J</th>
<th>ALPHA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>83438E-01</td>
<td>1</td>
<td>3</td>
<td>86679E-01</td>
<td>35</td>
<td>5</td>
<td>21655E-02</td>
<td>1</td>
<td>2</td>
<td>13484E-02</td>
<td>186</td>
<td>2</td>
<td>20880E-01</td>
</tr>
<tr>
<td>2</td>
<td>69806E-01</td>
<td>75</td>
<td>5</td>
<td>15195E+00</td>
<td>36</td>
<td>7</td>
<td>48887E-03</td>
<td>75</td>
<td>4</td>
<td>77454E-03</td>
<td>37</td>
<td>5</td>
<td>67533E-02</td>
</tr>
<tr>
<td>3</td>
<td>60981E-01</td>
<td>124</td>
<td>8</td>
<td>18646E+00</td>
<td>37</td>
<td>8</td>
<td>47880E-03</td>
<td>75</td>
<td>2</td>
<td>61729E-03</td>
<td>36</td>
<td>2</td>
<td>22797E-02</td>
</tr>
<tr>
<td>4</td>
<td>69160E-01</td>
<td>126</td>
<td>9</td>
<td>16294E+00</td>
<td>37</td>
<td>10</td>
<td>17998E-03</td>
<td>128</td>
<td>4</td>
<td>27473E-03</td>
<td>37</td>
<td>4</td>
<td>76967E-03</td>
</tr>
<tr>
<td>5</td>
<td>64404E-01</td>
<td>54</td>
<td>10</td>
<td>18247E+00</td>
<td>112</td>
<td>11</td>
<td>26652E-03</td>
<td>128</td>
<td>2</td>
<td>48656E-03</td>
<td>114</td>
<td>2</td>
<td>25905E-03</td>
</tr>
<tr>
<td>6</td>
<td>44499E-01</td>
<td>54</td>
<td>12</td>
<td>47708E-01</td>
<td>112</td>
<td>10</td>
<td>32499E-03</td>
<td>128</td>
<td>2</td>
<td>43565E-03</td>
<td>113</td>
<td>2</td>
<td>87731E-04</td>
</tr>
<tr>
<td>7</td>
<td>29526E-01</td>
<td>1</td>
<td>17</td>
<td>34329E-01</td>
<td>136</td>
<td>22</td>
<td>36372E-03</td>
<td>129</td>
<td>2</td>
<td>45956E-03</td>
<td>113</td>
<td>2</td>
<td>29619E-04</td>
</tr>
<tr>
<td>8</td>
<td>33651E-01</td>
<td>1</td>
<td>19</td>
<td>15847E-01</td>
<td>132</td>
<td>18</td>
<td>37344E-03</td>
<td>129</td>
<td>2</td>
<td>46015E-03</td>
<td>113</td>
<td>2</td>
<td>18088E-04</td>
</tr>
<tr>
<td>9</td>
<td>14540E-01</td>
<td>128</td>
<td>3</td>
<td>17591E-01</td>
<td>112</td>
<td>3</td>
<td>36916E-03</td>
<td>128</td>
<td>2</td>
<td>45216E-03</td>
<td>113</td>
<td>2</td>
<td>28880E-01</td>
</tr>
<tr>
<td>10</td>
<td>14852E-01</td>
<td>127</td>
<td>5</td>
<td>16987E-01</td>
<td>112</td>
<td>5</td>
<td>10281E-03</td>
<td>128</td>
<td>4</td>
<td>12177E-03</td>
<td>112</td>
<td>4</td>
<td>67523E-02</td>
</tr>
<tr>
<td>11</td>
<td>11184E-01</td>
<td>126</td>
<td>7</td>
<td>13196E-01</td>
<td>111</td>
<td>5</td>
<td>84514E-04</td>
<td>138</td>
<td>2</td>
<td>10988E-03</td>
<td>112</td>
<td>2</td>
<td>22797E-02</td>
</tr>
<tr>
<td>12</td>
<td>56624E-02</td>
<td>1</td>
<td>13</td>
<td>65583E-02</td>
<td>136</td>
<td>12</td>
<td>46803E-04</td>
<td>12</td>
<td>1</td>
<td>51889E-04</td>
<td>138</td>
<td>2</td>
<td>76967E-03</td>
</tr>
<tr>
<td>13</td>
<td>91053E-02</td>
<td>1</td>
<td>12</td>
<td>76484E-02</td>
<td>132</td>
<td>13</td>
<td>46502E-04</td>
<td>12</td>
<td>1</td>
<td>49412E-04</td>
<td>138</td>
<td>2</td>
<td>25905E-03</td>
</tr>
<tr>
<td>14</td>
<td>90926E-02</td>
<td>1</td>
<td>13</td>
<td>77232E-02</td>
<td>129</td>
<td>14</td>
<td>55886E-04</td>
<td>12</td>
<td>1</td>
<td>56679E-04</td>
<td>138</td>
<td>2</td>
<td>87731E-04</td>
</tr>
<tr>
<td>15</td>
<td>61849E-02</td>
<td>1</td>
<td>13</td>
<td>56815E-02</td>
<td>127</td>
<td>15</td>
<td>72508E-04</td>
<td>12</td>
<td>1</td>
<td>64241E-04</td>
<td>137</td>
<td>2</td>
<td>29619E-04</td>
</tr>
<tr>
<td>16</td>
<td>14253E-02</td>
<td>1</td>
<td>19</td>
<td>23997E-02</td>
<td>125</td>
<td>13</td>
<td>78876E-04</td>
<td>12</td>
<td>1</td>
<td>69537E-04</td>
<td>137</td>
<td>2</td>
<td>18088E-04</td>
</tr>
<tr>
<td>17</td>
<td>32332E-02</td>
<td>1</td>
<td>3</td>
<td>25583E-02</td>
<td>135</td>
<td>3</td>
<td>83876E-04</td>
<td>12</td>
<td>1</td>
<td>71982E-04</td>
<td>137</td>
<td>2</td>
<td>28880E-01</td>
</tr>
<tr>
<td>18</td>
<td>36721E-02</td>
<td>1</td>
<td>5</td>
<td>27769E-02</td>
<td>133</td>
<td>5</td>
<td>24891E-04</td>
<td>147</td>
<td>4</td>
<td>18888E-04</td>
<td>134</td>
<td>4</td>
<td>67523E-02</td>
</tr>
<tr>
<td>19</td>
<td>76592E-02</td>
<td>1</td>
<td>7</td>
<td>36592E-02</td>
<td>138</td>
<td>7</td>
<td>18589E-04</td>
<td>12</td>
<td>1</td>
<td>14894E-04</td>
<td>135</td>
<td>2</td>
<td>22797E-02</td>
</tr>
<tr>
<td>20</td>
<td>49347E-02</td>
<td>1</td>
<td>8</td>
<td>31876E-02</td>
<td>126</td>
<td>8</td>
<td>14347E-04</td>
<td>148</td>
<td>2</td>
<td>11298E-04</td>
<td>135</td>
<td>2</td>
<td>76967E-03</td>
</tr>
<tr>
<td>---</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-22656E+01</td>
<td>-2617E+01</td>
<td>.1715E-11</td>
<td>-2394E+00</td>
<td>.3694E+00</td>
<td>-958E+00</td>
<td>-2387E+00</td>
<td>-7854E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-2295E+01</td>
<td>-2617E+01</td>
<td>.1483E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.9966E-02</td>
<td>-2396E+00</td>
<td>.3055E+00</td>
<td>-4829E+00</td>
<td>-1715E+00</td>
<td>-7556E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.9966E-02</td>
<td>-2396E+00</td>
<td>.3055E+00</td>
<td>-4829E+00</td>
<td>-1715E+00</td>
<td>-7556E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>-2242E+01</td>
<td>-2574E+01</td>
<td>.1234E-01</td>
<td>-2365E+00</td>
<td>.3965E+00</td>
<td>-6894E+00</td>
<td>-2815E+00</td>
<td>-7965E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### AFZ CONVERGENCE HISTORY FOR THE NUMERICAL SOLUTION OF THE CONSERVATIVE, TRANSONIC FULL POTENTIAL EQUATION ###

<table>
<thead>
<tr>
<th>N</th>
<th>CMAX</th>
<th>I</th>
<th>J</th>
<th>RMAX</th>
<th>I</th>
<th>J</th>
<th>RAVG</th>
<th>BETA</th>
<th>ALPHA</th>
<th>NSP</th>
<th>CIRCULATION</th>
<th>CLICIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.18659E-01</td>
<td>00</td>
<td>24</td>
<td>.79994E-03</td>
<td>76</td>
<td>27</td>
<td>.43794E-04</td>
<td>3.0E00</td>
<td>.79888E-01</td>
<td>65</td>
<td>.36876E-03</td>
<td>1411</td>
</tr>
<tr>
<td>6</td>
<td>.61615E-02</td>
<td>189</td>
<td>38</td>
<td>.37281E-03</td>
<td>193</td>
<td>27</td>
<td>.12971E-04</td>
<td>2.5E03</td>
<td>.78888E-01</td>
<td>181</td>
<td>.57732E-03</td>
<td>2289</td>
</tr>
<tr>
<td>24</td>
<td>.33776E-02</td>
<td>189</td>
<td>38</td>
<td>.24228E-03</td>
<td>111</td>
<td>38</td>
<td>.61214E-05</td>
<td>2.1E04</td>
<td>.78888E-01</td>
<td>184</td>
<td>.58528E-03</td>
<td>2331</td>
</tr>
<tr>
<td>32</td>
<td>.12115E-02</td>
<td>189</td>
<td>38</td>
<td>.21134E-03</td>
<td>190</td>
<td>38</td>
<td>.22294E-06</td>
<td>1.8E03</td>
<td>.78888E-01</td>
<td>186</td>
<td>.61717E-03</td>
<td>2361</td>
</tr>
<tr>
<td>48</td>
<td>.89831E-03</td>
<td>189</td>
<td>27</td>
<td>.21556E-03</td>
<td>190</td>
<td>38</td>
<td>.18155E-05</td>
<td>1.5E05</td>
<td>.78888E-01</td>
<td>186</td>
<td>.62481E-03</td>
<td>2399</td>
</tr>
<tr>
<td>64</td>
<td>.76857E-03</td>
<td>189</td>
<td>27</td>
<td>.43365E-03</td>
<td>189</td>
<td>38</td>
<td>.4847E-06</td>
<td>1.7E05</td>
<td>.78888E-01</td>
<td>189</td>
<td>.63248E-03</td>
<td>2428</td>
</tr>
<tr>
<td>56</td>
<td>.53113E-03</td>
<td>189</td>
<td>27</td>
<td>.29614E-03</td>
<td>190</td>
<td>28</td>
<td>.7628E-06</td>
<td>2.5E06</td>
<td>.78888E-01</td>
<td>189</td>
<td>.63728E-03</td>
<td>2428</td>
</tr>
<tr>
<td>64</td>
<td>.13160E-03</td>
<td>189</td>
<td>27</td>
<td>.17001E-03</td>
<td>118</td>
<td>38</td>
<td>.4828E-06</td>
<td>2.1E06</td>
<td>.78888E-01</td>
<td>118</td>
<td>.63899E-03</td>
<td>2445</td>
</tr>
</tbody>
</table>

### FINAL CONVERGED SOLUTION ***
INPUT/OUTPUT example 5- The cards below form the data deck for INPUT/OUTPUT example 5 in chapter 3. This is a multiple-case run comparing a GRAPE generated grid solution to a TAIR generated grid solution. The output for this case is shown on the next two pages.

$FLOWIN  
MINF= .63, ALPH= 2.0, NI= 141, NJ= 31, MESH= 3, ALLOW=.08, NCASE= 2, NOUT= 3,  $
$FLOWIN  
MINF= .63, ALPH= 2.0, NI= 141, NJ= 31, NOUT= 3,  $
$GRIDIN  $

38
4. PROGRAM INFORMATION

This chapter contains individual descriptions of most of the program modules including flowcharts for the larger, more complicated routines. This information should provide the reader with deeper insight into the details of the TAIR program logic and, therefore, allow for more efficient code operation. In addition, this chapter (along with chap. 5 on theoretical development) should provide assistance to the user who wishes to modify the TAIR computer program for new applications.

The TAIR computer program consists of 16 program modules: a main program which contains the AF2 flow-solver logic (MAIN); several major subroutines which provide for solution initialization, grid generation, geometry specifications, solution output, etc. (INITL, GREGN, GEMPAC, INNER, OUTER, ADI, CHECK, AUTO, and OUTPUT); and several minor subroutines which perform a variety of simpler yet very necessary functions (TRIB, TRIP, CSPLIN, FORCE, PRNPLT, and XYOUT). Each of these routines along with a brief description of its function is listed in table 2. A subroutine tree showing the relation of each routine to its called and calling routines is displayed in figure 4.

In addition, the TAIR program utilizes nine COMMON blocks including a blank, or unlabeled COMMON block used to store the velocity potential (PHI) and density (RHO) arrays, and eight labeled COMMON blocks. These blocks contain NAMELIST variables, boundary-condition arrays, and control variables (COM1 and GRGN), indexing arrays and metric arrays (COM2 and COM4), airfoil coordinates and titles (COM3 and COM5), and miscellaneous solution parameters (CONV and SCRACH). Table 3 displays these nine COMMON blocks along with brief descriptions and the names of the routines which reference each block.

Tables 2 and 3, figure 4, and the flowcharts to be discussed shortly provide the user with a quick reference source for most of the information to follow. Also, prominent program variables (cross-referenced with theoretical names) are listed alphabetically with brief descriptions in appendix B. These measures should enable the user to assimilate quickly much of the information contained in this chapter and should be referred to as needed during the following more detailed discussion.

Description of MAIN Program

Program execution is initiated at the beginning of MAIN with a call to SUBROUTINE INITL, as indicated in the flow chart of figure 5. After receiving the initial solution and transformation metrics from INITL, the multiple-solution loop in MAIN (DO 1001 ICASE=1,NC) is entered. The value of NC is established via the first call to INITL from the NCASE parameter (NAMELIST FLOWIN). Inside the multiple-solution loop, for the second and each succeeding solution, SUBROUTINE INITL is called again for reinitialization of solution parameters. Next MAIN starts the main AF2 iteration loop (DO 1000 N=1,NSTEPS). In the first sweep (bidirectional along the η-direction), the density coefficients RI and RJ are calculated. If the rotated differencing is turned on (NDIF=1), the RJ coefficient is upwinded. Then the residual R and the first sweep
intermediate result F are calculated. Due to the AF2 algorithm construction, this sweep starts at the airfoil boundary (J=NJ) and proceeds backwards in J to the outer boundary (J=1). In the second sweep of the AF2 flow solver (periodic tridiagonal along the x direction), the matrix coefficients A, D, B, and C are calculated, and the correction (COR) is obtained from the tridiagonal solver TRIP. This sweep starts at the outer boundary (J=1) and proceeds forward in J to the airfoil boundary (J=NJ). Then the velocity potential (PHI), the circulation (GNP), and the density (RHO) are obtained for the new iteration level. The density values at the trailing edge are averaged to remove any oscillation that might be caused by the trailing-edge mapping singularity. At this point the solution is checked for convergence (see chap. 3, NAMELIST FLOWIN, for a discussion of ERR, the flow-solver convergence parameter). If the solution is diverging, the iteration procedure is stopped. If the solution is converging, but not yet converged, it goes through another iteration after a call to SUBROUTINE AUTO (providing IAUTO=1). If the solution is declared converged, iteration terminates, the solution is printed according to NOUT, and the program either begins the next case or ends.

Description of SUBROUTINE INITL

This subroutine reads the option and acceleration parameters (NAMELIST FLOWIN), contains a set of default values for all NAMELIST FLOWIN parameters, and computes the mapping metrics, the initial flow-field solution, and most of the solution control parameters. INITL first defines the default values, then reads the NAMELIST FLOWIN data cards and updates the default parameters with user-supplied values. If NOUT=4, this updated list is printed. If ICHECK=1, SUBROUTINE CHECK is called to ensure that all input parameters are within range. The different MESH functions then follow. If MESH=1 or 2, the number of x and y points (NX, NY) and the coordinates of the grid (X, Y) are read from logical unit 48. Then the two body-surface arrays (XB, YB) are defined, TEOPEN is calculated (if MESH=1), and a check is made to ensure that NI and NJ match NX and NY. If the NX and NY values do not match NI and NJ, the NI and NJ values will be changed to NX and NY. An error message announcing this change will always be printed, regardless of the NOUT value. If MESH=3, the program reads the mesh dimensions NX and NY, TEOPEN, and the grid coordinates from unit 48. This information must have been generated by the GRAPE grid generation program (ref. 5). In order to make the GRAPE mesh compatible with TAIR, NI must be one greater than the number of x points in the GRAPE grid (called JMAX in GRAPE): A check on grid dimension compatibility similar to the MESH=1 or 2 option check is performed for this option. For TAIR, the y coordinates must be renumbered to read from the outer boundary (J=1) to the airfoil surface (J=NJ), instead of the airfoil outward numbering system employed by GRAPE. Since TAIR uses a double-stored row of points along the vortex sheet, and the GRAPE code does not, a second set of points at I=NI must be added to the GRAPE grid. The last task performed in the MESH=3 section is to assign the body-surface arrays, XB and YB.

Next, INITL computes a series of counters, including a periodic indexing array (IA), based on the mesh dimensions. GRGEN is then called to either compute a new mesh (MESH=0) or to add finishing touches to the mesh which was read in from unit 48 (MESH=1,2,3). After INITL has received the grid
coordinates, the metric quantities and the Jacobian are computed. Some smoothing around the trailing edge is done to these quantities to reduce the effect of the trailing-edge mapping singularity on the flow-field solution. If NOUT=29, SUBROUTINE XYOUT is called to print the grid coordinates and metric quantities. INITL then calculates the free-stream properties and the initial solutions for the velocity potential PHI and the density RHO. Figure 6 shows a brief logical flowchart for this subroutine.

Description of SUBROUTINE GRGEN

SUBROUTINE GRGEN sets up the finite-difference grid, using the SUBROUTINES GEMPAC, INNER, OUTER, and ADI. If a mesh has already been read in and does not need to be changed (MESH=1 or 3), the main body of the coding is skipped. If a new mesh is generated (MESH=0) or an old mesh is modified (MESH=2), the following steps occur. After the default variables have been set for the GRIDIN NAMELIST parameters, the NAMELIST is read for user updates. If NOUT=4, this updated list is printed. If the check option is on, GRGEN calls CHECK to scan the NAMELIST parameters. It then calls GEMPAC to determine the airfoil coordinates. If airfoil coordinates are read in (IOPT=4), and the check option is on, the coordinates will be checked for mispunched values. SUBROUTINE INNER is called next to cluster the points about the airfoil surface. This clustering operation is controlled by the BINN parameter. If MESH=0, GRGEN calls OUTER to determine the outer boundary point distribution, fills in the middle of the mesh using an exponential formula to obtain the mesh initial conditions, and then calls SUBROUTINE ADI (the alternating-direction-implicit solver) to solve for the final finite-difference grid. Finally, SUBROUTINE GRGEN creates the double-stored, Nith row of grid points along the vortex sheet. If MESH=2, GRGEN establishes the final mesh by modifying a previously generated mesh stored on unit 48 to adapt to the new airfoil. This simple operation involves a shearing of the old mesh and therefore should only be attempted for relatively small airfoil changes. The value of the clustering control parameter (BINN) used to generate the new airfoil clustering distribution should be identical with the value used to establish the old mesh stored on unit 48. Use of this option eliminates the CALL to ADI and therefore greatly reduces the mesh-generation execution time. This option was intended for use with a numerical optimization application. Finally, GRGEN places the XB and YB arrays at halfpoints using calls to SUBROUTINE CSPLIN (cubic spline interpolation) for all MESH options (0, 1, 2, and 3). If MESH is greater than zero, the program will return to INITL. For MESH=0, GRGEN writes the grid solution to a logical unit 48 before returning to INITL. This grid solution can then be used for second, third, etc. calculations during the same run using the MESH=1 or 2 options, or it can be made permanent by using the appropriate JCL control statements. See figure 7 for a brief logical flowchart of SUBROUTINE GRGEN.

Description of SUBROUTINE GEMPAC

This subroutine establishes the inner boundary shape (J=NJ, airfoil coordinates). The two arrays, XB and YB, are the nondimensionalized inner boundary coordinates starting at the lower trailing-edge point and moving clockwise around the airfoil to the upper trailing-edge point. SUBROUTINE GEMPAC
contains five basic geometry options controlled by the parameter IOPT. For
IOPT=1, a NACA-00XX-type airfoil is generated from the standard analytic
expression. This airfoil can have an open or closed trailing edge. IOPT=2
generates a circular-arc airfoil with a blunted nose controlled by the XC
parameter. IOPT=3 yields the cross section of a circular cylinder. The
read-in option for other airfoils is IOPT=4, and the fifth option, IOPT=5, uses
the Korn airfoil coordinates stored in a data statement. (See chap. 3 for a
complete description of all these options.) After the XB and YB inner boundary
arrays are filled, the appropriate airfoil title is assigned to TITLE, the
coordinates are normalized so that the X coordinates of the leading and trail-
ing edges are equal to 0.0 and 1.0, respectively, and the trailing-edge thick-
ness (TEOPEN) is calculated. If NOUT>4, the coordinates will be printed out
before the logic returns to GRGEN.

Descriptions of SUBROUTINES INNER and OUTER

SUBROUTINE INNER distributes the points about the airfoil surface. First
it calculates the existing arc-length distribution (S) from the XB and YB
body-surface coordinates. Next, the desired arc-length distribution (S2) is
computed according to the BINN parameter. If BINN is greater than one, the
new arc-length distribution is computed from a stretching formula. The closer
the BINN value is to one, the more the points are clustered at the leading and
trailing edges. If BINN=0.0, the desired arc-length distribution is equally
spaced (note that this distribution option for S2 is unacceptable for standard
airfoil calculations). If BINN=-1.0, the new arc-length distribution is taken
from a data statement which was calculated using conformal mapping. If
BINN=-2.0, the user-supplied distribution is read from cards. (See chap. 3
for a more complete description of these clustering options.) If the desired
inner boundary arc-length distribution (S2) is established via the BINN=-1.0
or -2.0 options, the number of points in the distribution will generally not
agree with the desired final number of points (NI) input in INITL from
NAMELIST FLOWIN. This discrepancy is removed by interpolating the S2 distri-
bution up or down so as to involve NI points. The distribution of points
inherent in the S2 array is not changed by this interpolation, only the number
of points is changed. The existing and desired arc-length arrays (S and S2)
are then used to interpolate new values of XB and YB. The new values of XB
and YB are thereby distributed according to the desired arc-length distribu-
tion (S2). The interpolation process used is that of cubic spline interpola-
tion (SUBROUTINE CSPLIN). If NOUT>5, the final XB and YB coordinates will be
printed. After setting the X(I,NJ) and Y(I,NJ) coordinate arrays equal to XB
and YB, control returns to GRGEN.

SUBROUTINE OUTER sets up the point distribution on the outer boundary of
the finite-difference grid. OUTER is divided into three sections; the IOUT
parameter determines which section is used. IOUT=1 computes a circular outer
boundary with radius RADMAX centered at XCN and YCN, then returns to GRGEN.
If IOUT=2, the outer boundary distribution is read from cards, and the logic
returns to GRGEN. If IOUT=3, a rectangular mesh with semicircular ends is
calculated. The nondimensionalized height and width of the base rectangle are
given by HTMAX and WDTIMAX, respectively. This latter option should prove use-
ful for wind-tunnel wall applications. However, options for different
flow-solver boundary conditions that model solid-wall or porous-wall conditions have not been included. (See chap. 3 for more details about implementing the IOUT options.)

Description of SUBROUTINE ADI

SUBROUTINE ADI determines the final grid coordinates (X and Y) by requiring that they be solutions to appropriately formulated elliptic partial differential equations. An alternating-direction-implicit iteration scheme is used to determine the final numerical values of X and Y. Pertinent aspects about this algorithm are discussed in chapter 5. SUBROUTINE ADI starts almost immediately with the main iteration loop, DO 1000 N=1,MAXIT. Just inside this loop the ALPHA acceleration parameter, which effectively behaves like the inverse of a time-step, is computed from user-specified values of ALGRID and AHGRID. Inside the first sweep (tridiagonal inversions along the $\xi$ or wrap-around direction), the metrics, the X and Y equation residuals, and the tridiagonal matrix coefficients are all computed. SUBROUTINE ADI then calls SUBROUTINE TRIP (tridiagonal solver with periodic boundary conditions) for the X-equation inversion and SUBROUTINE TRIB (tridiagonal solver with fixed boundary conditions) for the Y-equation inversion. The two intermediate results are stored in the FX and FY two-dimensional arrays. The second sweep (tridiagonal inversions along the $\eta$ or normal-like direction) uses these values to obtain the new values of X and Y. SUBROUTINE TRIB is used for inverting the tridiagonal matrix equations for both the X and Y equations of the second sweep. If NOUT$\geq$8, the ADI grid-generation convergence history is printed. Finally, a check for convergence is made. If the grid solution has converged, control returns to VRGEN; if not, it returns to the start of the routine for another iteration.

Descriptions of SUBROUTINEs CHECK and AUTO

SUBROUTINE CHECK scans the user inputs to see that there are no out-of-range or incorrect values. The call to CHECK is suppressed if ICHECK is set to zero. This subroutine is divided into three sections, and the variable ICH (set in INITL and GREN) designates which section of the routine to use. The first section (ICH=1) checks the parameters in the FLOWIN NAMELIST. The second section (ICH=2) checks the airfoil coordinates for mispunched or erratic points. The last section (ICH=3) checks the parameters in the GRIDIN NAMELIST. If an out-of-range value is detected, one of the following three things will occur. One, the error is serious and the logical ICHECK flag is set to true. The solution is then terminated when the logic returns to MAIN. In a multiple-solution run, execution of the remaining cases will be unaffected, providing the error detected in the preceding section does not persist. Two, the parameter in error is reset to some predetermined value and the solution continues. Or three, the subroutine issues a warning that an out-of-range or inaccurate parameter has been found but has not been changed. In all of these responses, a message is generated, regardless of the NOUT value.

SUBROUTINE AUTO scans the convergence history and updates the solution parameters (BETA, CON, RGAM, and NDIF) to speed convergence or to prevent
divergence. The call to SUBROUTINE AUTO is suppressed if the IAUTO parameter of NAMELIST FLOWIN is set equal to zero, and none of the solution parameters are changed from their initial values. The first section of logic in this subroutine increases or decreases BETA, the time-like dissipation parameter, based on changes in the average and maximum residuals. (See chap. 5 for more detailed information about this process.) The next section decreases BETA if the NSF builds up slowly, or increases it if the NSF is building rapidly. This reflects the need for more time-like dissipation when the supersonic region is large and less when the supersonic region is small. The section following this activates the rotated differencing (NDIF=1) and lowers the circulation relaxation factor (RGAM) when the supersonic flow approaches the airfoil trailing edge. Next the artificial viscosity parameter (CON) is lowered if the circulation and NSF are building too slowly. For weak-shock calculations this reduced value of CON is superior because it allows for sharper, less-smeared shock profiles. The second-to-last section of logic lowers BETA to remove an additional amount of temporal damping. This level of BETA produces a stable iteration only for weak-shock cases in which the extent of supersonic flow is small. When flow conditions permit this small level of BETA, very fast convergence is possible. If any of these changes is made, a message describing the change will be printed, providing NOUT>4. The last section looks at only the airfoil surface flow-field solution and, if NOUT>4, prints a warning when the Mach number exceeds 1.3. This Mach number value, occurring immediately upstream of a normal shock, is generally accepted as the limit where significant discrepancies exist between the full-potential equation and the more exact Euler equation formulations. Solutions which exceed this limit should be interpreted with caution.

**Description of SUBROUTINE OUTPUT**

SUBROUTINE OUTPUT contains most of the solution output and is called exclusively by MAIN. OUTPUT is divided into four sections and is controlled by the variable II set in MAIN and passed to OUTPUT through the calling argument list. In addition, all output is controlled by the output control parameter NOUT. The first section (II=1) prints the transformation metrics near the airfoil surface (NOUT>8) and sets up the headings for the solution convergence history (NOUT>6). The second section of OUTPUT (II=2) prints the AF2 convergence history data, providing all print criteria are satisfied. The parameters NOUT and NOUT2 control this operation. If N is a multiple of the NOUT1 parameter, an intermediate airfoil solution will be printed. (Caution: small values for NOUT1 will cause excessive amounts of output.) Included in the intermediate airfoil solution output is the following: if NOUT27, contour maps of the density and Mach number will be printed. If NOUT23, the title of the airfoil or mesh and the flow-field conditions (MINF and ALPH) will be printed. If NOUT22, the iteration number (N), the maximum residual (RMAX), the number of supersonic points (NSP), and the coefficient of lift calculated from the circulation (CL) will be printed. The airfoil surface solution for the Nth iteration is printed next (if NOUT23). Finally, SUBROUTINE OUTPUT calls SUBROUTINE FORCE to calculate and print out the lift, wave-drag, and quarter-chord moment coefficients.
When the main iteration loop has ended, either by converging or reaching the maximum number of iterations (N=NSTEPS), SUBROUTINE OUTPUT is again called. If the iteration limit is reached (II=3), or the solution has converged (II=4), a statement indicating the nature of the iteration termination is printed, providing NOUTz6. After either of these two statements has been printed, the control logic is sent back to the second section of SUBROUTINE OUTPUT to print the final solution. This final solution depends, as before, on NOUT and includes: the printer contour maps, the title line, the last line of convergence information, the airfoil surface solution, and the force coefficients.

Description of Minor Subroutines

The TAIR program contains several smaller subroutines which are classified as utility routines. These routines include CSPLIN, TRIB, TRIP, PRNPLT, XYOUT, and FORCE.

SUBROUTINE CSPLIN performs a cubic spline interpolation. Inputs via the formal parameter argument list include the independent variable array (XX) which defines the new function interpolated, the independent and dependent arrays (X and Y) which define the function to be interpolated, the index limits N1 and N2 for the XX and YY arrays, and the index limits J1 and J2 for the X and Y arrays. The YY array contains the found interpolates resulting from the interpolation process and is passed back to the original calling routine as a formal parameter in the argument list. The A, B, C, D, F, and H arrays are only dummy arrays defined as formal parameters to save storage. If any element of XX is outside the range of X (with a fudge factor added), or if the independent array X is not monotonic, the X, Y, XX, and YY values are printed along with a message explaining the problem, and the program is stopped.

SUBROUTINE FORCE is called by OUTPUT to calculate and print out the lift, wave-drag, and quarter-chord moment coefficients. SUBROUTINE FORCE numerically integrates the airfoil surface pressure coefficient distribution, using a trapezoid rule integration algorithm. The inputs to SUBROUTINE FORCE are made through the formal argument list and include: the number of points on the airfoil surface (NI), the airfoil surface coordinates (X and Y), the pressure coefficient distribution (CP), the angle of attack (ALPH), and the distance across the trailing edge (TEOPEN). The coefficients are printed when NOUTz1.

SUBROUTINES TRIB and TRIP are scalar tridiagonal matrix inversion routines with fixed and periodic boundary conditions, respectively. The formal parameter argument lists consist of A, B, and C, which are the below-diagonal, diagonal, and above-diagonal matrix elements, respectively; F, the RHS column vector; X, Q, and S, dummy arrays of scratch storage; and index limits, NL, NU, and JINC, for TRIB and J1, J2, and JINC for TRIP. Inclusion of the JINC parameter allows the inversion of a tridiagonal matrix using every (JINC)th entry in the A, B, C, and F arrays. For all cases in TAIR JINC is equal to one. Each subroutine passes back the result array in the array originally holding the RHS values (F). Neither subroutine prints any output. An object-code version of both TRIB and TRIP suitable for execution on the CDC 7600 computer is available. The object-code version executes approximately twice
as fast as the original FORTRAN version, which allows a 10% reduction in overall computer time for most TAIR calculations. For all computer times listed in this report the faster object-code versions of TRIB and TRIP have been used.

SUBROUTINE PRNPLT (called by SUBROUTINE OUTPUT) produces a two-dimensional printer contour map for the array VAR. Each element of this array is assigned an integer from 0 to 9 according to its size relative to the maximum and minimum variations in the array. The resulting integers are printed across the page with a 13211 format to create a single page "snapshot" of the entire flow field. If an attempt is made to print a map with no variation, SUBROUTINE PRNPLT will print a warning message and ignore the request. Arrays exceeding 132 in the first dimension are truncated so that the last 132 entries in the array receive a position in the printed contour map. (See chap. 3 for more discussion and an example.)

SUBROUTINE XYOUT is called by SUBROUTINE INITL to print the finite-difference grid metrics (A1, A2, A3, and XJ) and coordinates (X and Y) providing NOUT>9. The formal parameter argument list includes the grid dimensions NI and NJ and the printout increments in the I and J directions IINCR and JINCR.

5. FULL-POTENTIAL EQUATION ALGORITHM

Theoretical details of the algorithm used in TAIR are now discussed. Emphasis in this section is on relating details of code operation and organization to different aspects of the numerical algorithm. Many of the variables used in this theoretical chapter are listed, briefly defined, and cross-referenced with the corresponding TAIR names in appendix B. This information along with that of the previous chapter is intended to provide enough detail to allow modification of TAIR for new applications. For more information about the algorithm, references 1-3 are suggested.

Governing Equations

The full-potential equation in strong conservation-law form is given by

\[(\rho \phi_x)_x + (\rho \phi_y)_y = 0\]  

(1a)

\[\rho = \left[ 1 - \frac{\gamma - 1}{\gamma + 1} (\phi_x^2 + \phi_y^2) \right]^{1/\gamma - 1}\]  

(1b)

where \(\phi\) is the full or exact velocity potential, \(\rho\) is the fluid density, \(x\) and \(y\) are Cartesian coordinates in the streamwise and vertical directions, and \(\gamma\) is the ratio of specific heats. The density \(\rho\) and velocity components \((\phi_x \text{ and } \phi_y)\) are nondimensionalized by the stagnation density \(\rho_s\) and the critical speed of sound \(a^*\), respectively.
Equation (1) is transformed from the physical domain (Cartesian coordinates) into the computational domain by using a general independent variable transformation. This general transformation, indicated by (see fig. 1)

\[ \xi = \xi(x,y) \]
\[ \eta = \eta(x,y) \]

maintains the strong conservation law form of equation (1). The full-potential equation written in the computational domain (\(\xi - \eta\) coordinate system) is given by

\[ \left( \frac{\partial U}{\partial \xi} \right)_\xi + \left( \frac{\partial V}{\partial \eta} \right)_\eta = 0 \]  
\[ \rho = \left[ 1 - \frac{\gamma - 1}{\gamma + 1} (A_1 \phi^2_\xi + 2A_2 \phi_\xi \phi_\eta + A_3 \phi^2_\eta) \right]^{1/\gamma-1} \]

where

\[ U = A_1 \phi_\xi + A_2 \phi_\eta \]
\[ V = A_2 \phi_\xi + A_3 \phi_\eta \]
\[ A_1 = \xi_x^2 + \xi_y^2 \]
\[ A_2 = \xi_x \eta_x + \xi_y \eta_y \]
\[ A_3 = \eta_x^2 + \eta_y^2 \]

and

\[ J = \xi_x \eta_y - \xi_y \eta_x \]

The \(U\) and \(V\) quantities are the contravariant velocity components along the \(\xi\) and \(\eta\) directions, respectively; \(A_1, A_2,\) and \(A_3\) are metric quantities; and \(J\) is the Jacobian of the transformation. To evaluate the expressions of equation (5), the following metric identities are necessary:

\[ J = 1/(x_\xi y_\eta - x_\eta y_\xi) \]
\[ \xi_x = Jy_\eta, \quad \eta_x = -Jy_\xi \]
\[ \xi_y = -Jx_\eta, \quad \eta_y = Jx_\xi \]

The transformed full-potential equation (eqs. (3), (4)) is only slightly more complicated than the original Cartesian form (eq. (1)) and offers several
significant advantages. The main advantage is that boundaries associated with
the physical domain are transformed to boundaries of the computational domain.
This aspect is illustrated in figure 1, in which the physical and computational
domains for a typical transformation are shown. The inner airfoil boundary
becomes the $\eta = \eta_{max}$ computational boundary and the outer physical boundary
becomes the $\eta = \eta_{min}$ computational boundary. Note that no restrictions have
been placed on the shape of the outer boundary. Arbitrarily shaped outer
boundaries, including wind-tunnel walls, may be used.

Another advantage of this approach is the ability to adjust arbitrarily
the mesh spacing on the airfoil surface or in the mesh interior, with the pro-
vision that the smoothness of the mesh is not disrupted. This, in theory,
could be used to cluster mesh points around any gradients in the flow field
(e.g., shock waves or the leading-edge stagnation point). The next section
introduces the method of generating the finite-difference meshes used in the
TAIR program.

Grid Generation

The automatic grid generation scheme used by Thompson et al. (ref. 9) has
been adapted for use in the TAIR computer code. Basically, this grid-
generation scheme uses numerically generated solutions of Poisson's equation
(or, in the present application, Laplace's equation) to establish regular and
smooth finite-difference meshes around arbitrary bodies. These equations are
transformed to (and solved in) the computational domain (i.e., $\xi$ and $\eta$ are
the independent variables and $x$ and $y$ are the dependent variables). The
transformed equations are given by

$$
Ax_{\xi\xi} - 2Bx_{\xi\eta} + Cx_{\eta\eta} = 0
$$

$$
Ay_{\xi\xi} - 2By_{\xi\eta} + Cy_{\eta\eta} = 0
$$

where

$$
A = x_{\eta}^2 + y_{\eta}^2, \quad B = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}, \quad C = x_{\xi}^2 + y_{\xi}^2
$$

The numerical solution of equation (7) is achieved by first replacing all
derivatives in equations (7) and (8) by standard second-order-accurate finite
differences. A residual operator can be defined and is given by

$$
L( )_{i,j} = [A_{i,j}\delta_{\xi\xi} - 2B_{i,j}\delta_{\xi\eta} + C_{i,j}\delta_{\eta\eta}]( )_{i,j}
$$

where the $i$ and $j$ subscripts indicate position in the finite-difference mesh.
The operators used in equation (9) are defined by
\[ \delta_{\xi} \xi_{i,j} = (i+1,j) - 2(i,j) + (i-1,j) \]
\[ \delta_{\eta} \eta_{i,j} = \frac{1}{4} [(i+1,j+1) - (i+1,j-1) - (i-1,j+1) + (i-1,j-1)] \]
\[ \delta_{nn} (i,j) = (i+1,j) - 2(i,j) + (i,j-1) \]

The spatial increments (\(\Delta\xi\) and \(\Delta\eta\)) are equal to 1 and therefore have been omitted. Once boundary values and an initial solution for \(x_{i,j}\) and \(y_{i,j}\) have been established, the final interior values can be computed by relaxation. Usually, either successive overrelaxation (SOR) or successive-line overrelaxation (SLOR) is used for this purpose. However, in the TAIR program a faster alternating-direction-implicit (ADI) relaxation algorithm is applied. This algorithm can be expressed by

**Step 1:**

\[(a - A_{i,j}^n \delta_{\xi}) f_{i,j}^n = a_0 x_i^n \]
\[(a - A_{i,j}^n \delta_{\eta}) g_{i,j}^n = a_0 y_i^n \]

**Step 2:**

\[(a - C_{i,j}^n \delta_{nn}) (x_{i,j}^{n+1} - x_i^n) = f_{i,j}^n \]
\[(a - C_{i,j}^n \delta_{nn}) (y_{i,j}^{n+1} - y_i^n) = g_{i,j}^n \]

where \(f_{i,j}^n\) and \(g_{i,j}^n\) are intermediate results stored at each point in the finite-difference mesh. In step 1, the \(f\) and \(g\) arrays are obtained by solving two tridiagonal matrix equations for each \(\eta = \) constant line. The corrected values of \(x\) and \(y\) are then obtained in the second step from the \(f\) and \(g\) arrays, respectively, by solving two tridiagonal matrix equations for each \(\xi = \) constant line. Because of the implicit construction of this scheme, each point in the finite-difference mesh influences every other point during each iteration. As a result, evolution of the solution proceeds at a much faster rate.

**Spatial Differencing**

The finite-difference approximations used to discretize equations (3) to (5) in the TAIR computer program are described in this section. These spatial difference approximations, which are valid for both subsonic and supersonic regions of flow, are given by

\[ \frac{\delta}{\delta_{\xi}} \left( \frac{\partial u}{\partial j} \right)_{i+1/2,j} + \frac{\delta}{\delta_{\eta}} \left( \frac{\partial v}{\partial j} \right)_{i,j+1/2} = 0 \]
where the \( \delta_\xi \) and \( \delta_\eta \) backward-difference operators are defined by
\[
\delta_\xi (i,j) = (i,j) - (i-1,j) \\
\delta_\eta (i,j) = (i,j) - (i,j-1)
\] (14)

The quantities \((U/J)_{i+1/2,j}\) and \((V/J)_{i,j+1/2}\) used in equation (13), are computed using standard, second-order-accurate, finite-difference formulas. An example is given by
\[
(U)_{i+1/2,j} = \left( \frac{A_1}{J} \right)_{i+1/2,j} (\phi_{i+1,j} - \phi_{i,j}) \\
+ \frac{1}{4} \left( \frac{A_2}{J} \right)_{i+1/2,j} (\phi_{i+1,j+1} - \phi_{i+1,j-1} + \phi_{i,j+1} - \phi_{i,j-1})
\] (15)

The Jacobian, \( J \), and the metric quantities, \( A_1, A_2, \) and \( A_3 \), used both in equation (15) and in the density calculation (to be discussed shortly) are first computed at integer points \((i,j)\) using standard, fourth-order-accurate, finite-difference formulas. These quantities are regrouped and permanently stored at different locations given by
\[
A_1(I,J) = \left( \frac{A_1}{J} \right)_{i+1/2,j} \\
A_2(I,J) = \frac{1}{4} \left( \frac{A_2}{J} \right)_{i+1/2,j} \\
A_3(I,J) = \left( \frac{A_3}{J} \right)_{i+1/2,j+1/2} \\
XJ(I,J) = J_{i+1/2,j+1/2}
\] (16)

where the coded variable names are on the left and the analytical expressions are on the right. Because the interpolation process by which the metric quantities are moved is fourth-order accurate, the final expressions \( A_1, A_2, A_3, \) and \( XJ \) are themselves fourth-order accurate. During the iteration process, values of \( A_1, A_2, A_3, \) and \( XJ \) which are required at points other than where they are stored are obtained by second-order accurate averaging.

The density coefficients of equation (13), \( \bar{\rho}_{i+1/2,j} \) and \( \bar{\rho}_{i,j+1/2} \), are defined by
\[
\bar{\rho}_{i+1/2,j} = [(1 - \nu)\rho]_{i+1/2,j} + \nu_{i+1/2,j}^{\rho_{i+1/2,k+1/2,j}} \\
\bar{\rho}_{i,j+1/2} = [(1 - \nu)\rho]_{i,j+1/2} + \nu_{i,j+1/2}^{\rho_{i,j+1/2,k+1/2,j}}
\] (17a-b)

52
where

\[ k = 1 \quad \text{when} \quad U_{i+1/2,j} \geq 0 \]  
\[ \xi = 1 \quad \text{when} \quad V_{i,j+1/2} < 0 \]  

The density values required in equation (17) are computed using a binomial series expansion of equation (4). To improve the computational efficiency, only the first four terms are retained. The final density expression, after some rearranging, is given by

\[ \rho = 1 + Q2[C1 + Q2(C2 + Q2 \cdot C3)] \]  

where

\[ C1 = -\frac{1}{\gamma + 1}, \quad C2 = \frac{2 - \gamma}{2(\gamma + 1)^2}, \quad C3 = \frac{(2 - \gamma)(3 - 2\gamma)}{6(\gamma + 1)^3} \]

and

\[ Q2 = A1\phi_\xi^2 + 2A2\phi_\xi\phi_\eta + A3\phi_\eta^2 \]

Examination of the error produced by this approximation clearly shows that even under the most adverse conditions (e.g., large local values of \( Q2 \)), the resulting error in \( \rho \) will not exceed a few tenths of a percent. Use of this series expansion to compute \( \rho \) eliminates the need for the exponentiation operation, which is computationally very expensive. This single simplification saves about 30% of the CPU time required for a flow-field computation.

Values of the density computed from equation (19) are computed and stored at cell centers, that is, at \( i + 1/2, j + 1/2 \), using values of \( \phi_\xi \) and \( \phi_\eta \) computed from

\[ \phi_{\xi i+1/2,j+1/2} = \frac{1}{2} (\phi_{i+1,j+1} - \phi_{i,j+1} + \phi_{i+1,j} - \phi_{i,j}) \]  
\[ \phi_{\eta i+1/2,j+1/2} = \frac{1}{2} (\phi_{i+1,j+1} - \phi_{i+1,j} + \phi_{i,j+1} - \phi_{i,j}) \]

Values of the density required at \( i + 1/2, j \) or \( i, j + 1/2 \) are obtained using simple averages. The switching functions, \( v_{i+1/2,j} \) and \( v_{i,j+1/2} \), used in equation (17), control the amount of upwinding in the finite-difference scheme, and are defined by (for example)

\[ v_{i+1/2,j} = \begin{cases} 
\max[(M_i^2, j - 1)\text{CON}, 0] & \text{for} \quad U_{i+1/2,j} > 0 \\
\max[(M_{i+1,j}^2, j - 1)\text{CON}, 0] & \text{for} \quad U_{i+1/2,j} < 0 
\end{cases} \]
where $M_{i,j}$ is the local Mach number and $CON$ is a user-specified constant. In TAIR, equation (22) is actually used to compute a quantity called $SIGMA$, which equals $1 - \nu$. This $SIGMA$ parameter is used in place of $\nu$ as the switching function.

Use of the density coefficients given by equation (17) is equivalent to the addition of an appropriately differenced artificial viscosity term. This effectively maintains an upwind influence in the differencing scheme for supersonic regions anywhere in the finite-difference mesh for any orientation of the velocity vector, thus approximating a rotated-differencing scheme. Other variations of this rotated-differencing scheme achieved by this upwind evaluation of the density are discussed in references 10-12. If rotated differencing is not required, that is, if the supersonic flow region is small, the spatial differencing algorithm can be simplified somewhat by upwinding along only the $\xi$ direction. The switching function in equation (17b), $\psi_{i,j+1/2}$, is effectively set to zero for this option. Thus, no tests for supersonic flow or flow direction must be made for the $\tilde{\phi}_{i,j+1/2}$ calculation. This logic is coded into TAIR and controlled by the parameter NDIF. If NDIF=1, upwinding along both the $\xi$ and $\eta$ directions is used. If NDIF=0 (default), only upwinding along the $\xi$ direction is used. If IAUTO=1 and if the region of supersonic flow moves beyond about 90% of chord on the airfoil surface, the default value of NDIF will automatically be changed to one. This mode of operation has not only proven to be efficient, but is also extremely reliable.

**AF2 Iteration Scheme**

The AF2 fully implicit approximate factorization scheme is given by

**Step 1:**

$$(\alpha - \frac{\xi}{\eta}B_j)\phi_i^n = \omega L \phi_i^n$$

**Step 2:**

$$(\alpha \delta_\eta + \omega \delta_\xi - \frac{\xi}{\eta}B_i \delta_\xi)C_i^n = \phi_i^n$$

where

$$B_i = \left(\frac{\delta A_1}{J}\right)^{i+1/2}_{j}, \quad B_j = \left(\frac{\delta A_3}{J}\right)^{i}_{j+1/2}$$

In equation (23), the $n$ superscript is an iteration index; $\alpha$ is an acceleration parameter (to be discussed shortly), $\omega$ is a relaxation parameter called $OMEGA$ in TAIR; $L\phi_i^n$ is the $n$th iteration residual (defined by eq. (13)), and $\phi_i^n$ is an intermediate result stored at each point in the finite-difference mesh. In step 1, the $f$ array is obtained by solving a simple bidiagonal matrix equation for each $\xi = constant$ line. The correction array is then obtained in the second step from the $f$ array by solving a tridiagonal matrix equation for each $\eta = constant$ line. Note that with the AF2 scheme,
the $\eta$ direction difference approximation is split between the two steps. This generates a $\phi_{\eta n}$-type term, which is useful to the iteration scheme as time-like dissipation. (The iterative process is considered as an iteration in pseudotime. Thus the time derivative is introduced by $(\eta)^{n+1} - (\eta)^n - \Delta t(\eta)$.) The split $\eta$ term also places a sweep direction restriction on both steps, namely, outward (away from the airfoil) for the first step and inward (toward the airfoil) for the second step. No sweep restrictions are placed on either of the two sweeps due to flow direction.

A $\phi_{\xi t}$-type term has been added inside the parentheses of step 2 (see eq. (23b)) to provide time-dependent dissipation in the $\xi$ direction. The double arrow notation in equation (23b) on the $\delta_{\xi}$-difference operator indicates that the difference is always upwind, which on the upper surface is a backward difference and on the lower surface is a forward difference. The sign is chosen in such a way that the addition of $\phi_{\xi t}$ increases the magnitude of the second sweep diagonal. The parameter $\beta$ is fixed at a value of 0.3 in subsonic regions. In supersonic regions, $\beta$ (called BETA in TAIIR) is initialized via user specification and then updated using logic similar to that presented in reference 13. The updating procedure exists in SUBROUTINE AUTO and is given by

$$\begin{align*}
\text{If } \text{RATIO} < 2.0 & \quad \text{then } \beta^n = 0.98 \beta^{n-1} \\
\text{If } \text{RATIO} > 2.1 & \quad \text{then } \beta^n = 1.1 \beta^{n-1} \\
\text{If } \beta^n > \text{BHIG} & \quad \text{then } \beta^n = \text{BHIG} \\
\text{If } \beta^n < \text{BLOW} & \quad \text{then } \beta^n = \text{BLOW}
\end{align*}$$

(25)

where

$$\text{RATIO} = \frac{\text{RAVG}^n}{\text{RAVG}^{n-m}} + \frac{\text{RMAX}^n}{\text{RMAX}^{n-m}}$$

(26)

and

$$\text{BHIG} = \beta^1 + 1, \quad \text{BLOW} = \beta^1 - 1$$

(27)

In equation (26), $m$ is the number of elements in the $a$ sequence (see the sequence definition given by eq. (28)), $\text{RAVG}^n$ is the $n$th iteration average residual and $\text{RMAX}^n$ is the $n$th iteration maximum residual. The logic defined by equations (25) to (27) monitors solution convergence through the parameter RATIO. If convergence is progressing satisfactorily, $\beta$ is reduced; if not, $\beta$ is increased. BHIG and BLOW are upper and lower bounds which limit the amount of $\beta$ variation. In addition to the above logic, other larger increases or decreases in $\beta$ are possible. During the iteration process, the base value of $\beta$ including the values of BHIG and BLOW can be increased or decreased, if the developing solution requires more or less time-like dissipation. This logic, which has largely been developed empirically, automatically keys on the growth rates of NSP (number of supersonic points) and GNP (the amount of circulation). If these quantities grow rapidly, then
\[ \beta, \text{BHIGH, and BLOW are all increased; if they grow slowly, then these quantities are decreased.} \]

The quantity \( \alpha \) appearing in equation (23) (called ALPH\(A\) in TAIR\(R\)) can be considered as \( \Delta t^{-1} \). This direct analogy to time provides one strategy for obtaining fast convergence, namely, advance time as fast as possible with large time steps (i.e., small values of \( \alpha \)). This is effective for attacking the low-frequency errors but not the high-frequency errors. The best overall approach is to use an \( \alpha \) sequence containing several values of \( \alpha \). The small values are particularly effective for reducing the low-frequency errors, and the large values are particularly effective for reducing the high-frequency errors. The \( \alpha \) sequence used in the TAIR computer program is given by

\[ \alpha_k = \alpha_H \left( \frac{\alpha_L}{\alpha_H} \right)^{k-1/m-1} \quad k = 1, 2, \ldots, m \]  

(28)

where the sequence endpoints are given by \( \alpha_L \) and \( \alpha_H \) (called A\(L\)OW and A\(H\)IGH in TAIR\(R\)), and \( m \) is the number of elements in the sequence.

Circulation Update and Boundary Conditions

The airfoil surface boundary condition is that of flow tangency (i.e., no flow through the airfoil surface); it requires that the \( \eta \) contravariant velocity component at the airfoil surface be zero (i.e., \( V = 0 \)). This boundary condition is implemented by applying

\[ \left( \frac{\partial V}{\partial j} \right)_{j=1,NJ-1/2} = -\left( \frac{\partial V}{\partial j} \right)_{j=1,NJ+1/2} \]  

(29)

where \( j = NJ \) is the airfoil surface. In other expressions (eqs. (15) and (21)) where \( \phi_\eta \) is required at the airfoil surface, the \( V = 0 \) boundary condition is used again to obtain

\[ \phi_\eta \bigg|_{\text{surface}} = -\frac{A_2}{A_3} \phi_\xi \bigg|_{\text{surface}} \]  

(30)

At the outer boundary of the computational mesh, the velocity potential is held fixed at the initial free-stream value.

Special boundary conditions inherent in the AF2 iteration algorithm arise at the airfoil surface. During each iteration at the beginning of the first sweep, a boundary condition on \( f \) (see eq. (23a)) must be applied at the airfoil surface. Because \( f \) is a complicated function with little physical meaning, specification of its value is difficult. As the iteration process drives the solution to a steady state, the value of \( f \) approaches zero. Therefore, boundary specification of \( f \) should be consistent with this fact. Even if the \( f \) boundary condition is consistent with the steady-state solution, a poor choice can slow convergence or even cause instability. For lack
of a better boundary condition, \( f_n = 0 \) is used and seems to produce acceptable results.

To facilitate the circulation calculation process, the velocity potential function is written as the sum of two parts

\[
\phi = \phi + \Gamma \xi
\]

(31)

where the first part is the nonlinear contribution and the second part essentially consists of a vortex with circulation strength, \( 2\pi \Gamma \) (\( \Gamma \) is referred to as GNP in TAIR). The idea of breaking the velocity potential into different parts was used in reference 14. The velocity potential function stored in the PHI(I,J) array in TAIR consists of only the nonlinear contribution given by \( \phi \). The vortex solution contribution is added in explicitly by modifying each \( \phi_\xi \) term (in eqs. (15) and (21)) to include

\[
\phi_\xi = \phi_\xi + \Gamma
\]

(32)

The \( \phi_n \) terms are, of course, not affected by this modification. Use of this substitution provides for a significant improvement in the circulation convergence rate because each grid point, through equation (32), feels the influence of the most recent value of the circulation during each iteration. At the end of each iteration, the circulation is recomputed from

\[
\Gamma^n = \frac{1}{2} (\phi^n_{N1-1,NJ} - \phi^n_{2,NJ})
\]

(33)

where \( N1 - 1,NJ \) and \( 2,NJ \) are one grid point upstream of the trailing edge on the upper and lower surfaces, respectively. To maintain stability for some situations, the new value of the circulation must be underrelaxed. The \( n \)th iteration relaxation factor used for the circulation, \( RG^n \), is computed using

\[
RG^n = RGAM \cdot \exp \left[ -\left| \frac{2(\Gamma^n - 2\Gamma^{n-1} + \Gamma^{n-2})}{\Gamma^n} \right| \cdot \frac{n}{100} \right]
\]

if \( RG^n < \frac{1}{3} RGAM \), then \( RG^n = \frac{1}{3} RGAM \)

(34)

where \( RGAM \) is a user-specified constant. Existence of the second difference on the circulation in the exponential term causes \( RG^n \) to automatically be reduced if the circulation growth history begins to oscillate. Use of the \( n/100 \) term causes an additional smoothing effect in the circulation growth history curve as the iteration process continues.

When calculations are performed on airfoils with open trailing edges, special logic along the vortex sheet boundary must be considered. The amount of openness or difference in \( y \) coordinates at the airfoil trailing edge, defined as \( TEOPEN \) in TAIR, is continued downstream of the airfoil with constant thickness all the way to the outer boundary. One purpose of this configuration is to simulate the existence of a wake leaving the airfoil trailing edge. The
velocity potential \( \phi \) for this situation has different values along \( i = 1 \) and \( i = NI \), when the angle of attack, \( \text{ALPH} \), is nonzero. This difference is given by

\[
\phi_{NI,j} - \phi_{1,j} = \text{QYINF} \cdot \text{TEOPEN}
\]  

(35)

where \( \text{QYINF} \) is the \( y \) component of the free-stream velocity. As a result, all differences of \( \phi \) across the vortex sheet must include the amount of this jump to maintain consistent results. In addition, the circulation computation must also be modified to include this jump. Equation (33) becomes

\[
\rho^n = \frac{1}{2} (\psi^n_{NI-1,NJ} - \psi^n_{2,NJ} - \text{QYINF} \cdot \text{TEOPEN})
\]  

(36)

6. CASE HISTORY

This chapter describes several different cases run with the TAIR computer code and discusses the results. First, a typical finite-difference grid generated by TAIR is shown in figure 8. This grid was generated for a NACA 0012 airfoil and consists of 4470 points (149 x 30, the default grid). Figure 8(a) shows the entire grid, including the circular outer boundary; figure 8(b) shows a close-up of the grid about the airfoil. The clustering of lines at the leading and trailing edges is shown in more detail in figures 8(c) and 8(d). This grid was generated with the center of the circular outer boundary at the default airfoil midchord position and required 20 iterations of the ADI grid generation routine for convergence and about 1.6 sec of computer time on the CDC 7600 computer.

The first flow-solver solution involves the supercritical Korn airfoil at a free-stream Mach number of 0.74 and an angle of attack of 0\(^\circ\). The pressure coefficient distribution for this slightly off-design calculation is shown in figure 9. The present TAIR version (cycle 8) is compared with a past version (cycle 7, ref. 15) and a result from the GRUMFOIL computer code (ref. 16). All three cases are in excellent agreement. The GRUMFOIL computer code is similar to TAIR in that both codes solve the conservative full-potential equation, but differs in that TAIR uses the AF2 iteration scheme and GRUMFOIL uses a hybrid direct-solver/SJOR iteration scheme (ref. 14). This hybrid iteration scheme is composed of one direct-solver iteration (very effective for reducing low-frequency errors but unstable for supersonic regions) followed by several SJOR iterations. The purpose of the SJOR iterations is to smooth high-frequency errors generated by the direct-solver step in regions of supersonic flow.

The second figure associated with this case (fig. 10) presents the rms error \( E_{\text{rms}} \) convergence-history curves for each of the three iteration schemes. The \( E_{\text{rms}} \) at iteration \( n \) \( (E_{\text{rms}}^n) \) is defined by
\[ E_{\text{rms}}^n = \left[ \frac{\sum_{i=1}^{NI} (C_{p_i}^n - C_{p_i}^0)^2}{\sum_{i=1}^{NI} C_{p_i}^2} \right]^{1/2} \]

where \( C_{p_i}^n \) is the surface pressure coefficient at the \( i \)th grid point and the \( n \)th iteration; \( C_{p_i}^0 \) is the surface pressure coefficient at the \( i \)th grid point taken from the converged solution, and \( NI \) is the total number of surface grid points. Using \( E_{\text{rms}} \) to compare convergence performance is a much more quantitatively correct procedure than using the standard maximum residual quantity. (More discussion on this point can be found in refs. 1, 17.)

The four curves shown in figure 10 correspond to the following iteration schemes: (1) AF2 (cycle 8), (2) AF2 (cycle 7), (3) hybrid, and (4) SLOR. There are two major differences between TAIR cycles 7 and 8. One, the circulation updating algorithm of cycle 8 is considerably improved relative to that of cycle 7. This improvement enhances the convergence speed of the TAIR code for cases with reasonably large amounts of lift. Details of the new circulation algorithm are presented in chapter 5. And two, the automatic updating routine used in cycle 8 (SUBROUTINE AUTO, discussed in chap. 4) is more sophisticated than the one used in cycle 7. This change improves the code reliability and in some cases the computational efficiency. The SLOR iteration scheme is simply the CRUMFOIL hybrid iteration scheme without the benefit of the direct-solver step. Each convergence-history curve is constructed by plotting \( E_{\text{rms}} \) versus CPU time (Ames CDC 7600 computer). The hybrid case has been computed with default values for all relaxation parameters. Convergence for the SLOR scheme has been approximately optimized by a trial-and-error adjustment of the relaxation parameter. The TAIR runs were computed with default values for all relaxation parameters except for the artificial viscosity parameter CON, which was set to 1.2 for a sharper shock profile. Set-up times, that is, the CPU time required for grid generation, solution initialization, and coarse- and medium-mesh calculations, are included in each convergence-history curve. The cycle 7 and 8 AF2 curves include 6.0 and 1.6 sec, respectively, for grid generation and initialization. The difference in these times is partially due to the coding efficiency changes but primarily due to a reduction in the unnecessarily tight grid convergence tolerance first used with cycle 7. The hybrid and SLOR curves use coarse-medium-fine mesh sequences. Converged results from the coarse mesh are interpolated onto the medium mesh, then from the medium mesh onto the fine mesh, thus providing a good initial guess for the fine-mesh calculation. The set-up times for these cases are 23 sec for the hybrid case and 28 sec for the SLOR case. For this calculation a two-order-of-magnitude reduction in \( E_{\text{rms}} \) produced essentially converged results. At this level of convergence the cycle 8 version of the AF2 scheme is about 2 times faster than the cycle 7 version, about 5 times faster than the hybrid scheme, and about 10 times faster than SLOR.

The second case presents an interesting airfoil shock-wave pattern that develops as the free-stream Mach number approaches 1. This case, a NACA 0012 airfoil with Mach number of 0.95 and angle of attack of 4°, shows with Mach
number contours a "fishtail" shock system (fig. 11). Supersonic-to-supersonic oblique shocks emanate from the trailing edge and merge with a normal shock downstream of the airfoil. The oblique shock emanating from the trailing-edge upper surface has been strengthened by the addition of circulation, while the oblique shock emanating from the trailing-edge lower surface has been weakened and is almost nonexistent. The normal shock above the airfoil plane is much stronger than the normal shock below the plane. This shock-wave pattern is characteristic of solutions with free-stream Mach numbers near unity and has been observed experimentally as well as computationally. This case required only about 100 iterations or about 10 sec of computer time on the CDC 7600 computer for convergence. The rapid convergence of this difficult case demonstrates the reliability and efficiency of the present transonic flow solution procedure. For more information about this case and others like it, see references 3 and 4.

The next two figures compare incompressible theory with two results from TAIR. These two figures plot the surface pressure coefficient distributions versus the X/C coordinate direction. Figure 12 shows that the result of a circular cylinder calculation (Mach number of 0.00001 and angle of attack of 0°) is in excellent agreement with classical incompressible theory (ref. 18). Figure 13 presents the result of the NACA 0012 airfoil at Mach number of 0.00001 and angle of attack of 0°. This, too, is in excellent agreement with theory (ref. 19) everywhere except at the trailing edge. Here the TAIR result differs because the density extrapolation used at the airfoil trailing edge has been designed to more closely approximate experimental results rather than the trailing-edge stagnation predicted in the theoretical results of reference 19. These incompressible calculations generally require about 2 sec of CDC 7600 computer time for convergence (20-40 iterations).

The last case discussed in this chapter shows a comparison of computed and experimental results. The three figures (figs. 14-16) show the surface pressure coefficient versus X/C for three supercritical CAST 7 (ref. 7) airfoil calculations. These calculations were run at a Mach number of 0.7 and three different angles of attack. To account for viscous effects, the TAIR angles of attack were found by approximately matching computed and experimental lifts. Generally speaking, the three plots show good agreement between TAIR and experiment. Slight differences occur at the trailing edges, probably due to trailing-edge separation. The two cases with lower angles of attack, -0.15° and 0.55°, show basically good agreement at the shock, but the largest angle of attack case, 1.35°, produces more disagreement because of the relatively strong shock. This is probably due to the lack of shock/boundary layer viscous modeling.

These cases and the cases in the INPUT/OUTPUT sections of chapters 2 and 3 substantiate the claims made at the beginning of this manual about the transonic airfoil analysis computer program TAIR. They show that the code is reliable over a large range of transonic flows and is simple to operate—all of the cases presented can be run with a few changes of the input parameters. TAIR results agree with incompressible theory and with transonic experimental results, and the present algorithm also shows substantial improvement in computational efficiency when compared with other standard relaxation schemes for the conservative full-potential equation.
APPENDIX A

PROGRAM CHANGES TO INCREASE DIMENSIONS

This appendix lists the changes that must be made in the TAIR program to increase (or decrease) the mesh dimensions. The following statements were changed for a CDC 7600 and, due to the overflow of small core memory (SCM), two LEVEL 2 statements were included to assign certain arrays to large core memory (LCM). For other computer systems these LEVEL 2 statements should be omitted. This example illustrates a dimension change from (151,31) to (209,43).

These changes must be made in the COMMON blocks. The default statement is listed first, followed by the appropriate changes.

```
COMMON /COM2/ A1(151,31),A2(151,31),A3(151,31),XJ(151,31)
COMMON /COM2/ A1(209,43),A2(209,43),A3(209,43),XJ(209,43)
LEVEL 2,A1,A2,A3,XJ
(/COM2/ appears in MAIN, INITL, OUTPUT, and XYOUT)
COMMON /COM3/ XB(151),YB(151)
COMMON /COM3/ XB(209),YB(209)
(/COM3/ appears in INITL, OUTPUT, GRGEN, INNER, and CHECK)
COMMON /COM4/ IA(153),A4(151),BODYBC(151)
COMMON /COM4/ IA(211),A4(209),BODYBC(209)
(/COM4/ appears in MAIN, INITL, OUTPUT, XYOUT, and ADI)
COMMON PHI(151,31),RHO(151,31)
COMMON PHI(209,43),RHO(209,43)
(/ appears in MAIN, INITL, OUTPUT, and AUTO)
COMMON /SCRACH/ X(151,31),Y(151,31),ARDUM(151,12)
COMMON /SCRACH/ X(209,43),Y(209,43),ARDUM(209,12)
(/SCRACH/ appears in MAIN, INITL, OUTPUT, ADI, XYOUT, GRGEN, INNER, and OUTER)
```

The following additional changes must be made in the various routines as indicated. First the default statements are listed, followed by the appropriate changes.
MAIN

DIMENSION A(151),B(151),C(151),D(151),
1 SIGMA(151,31),F(151,31),RI(151,31),COR(151),
2 RI(151),FIM1(151),FLUXJM(151),RJA3(151)
DIMENSION RJPH(151),RJMH(151),PXCJM1(151)
DIMENSION QTTRP(151),STRP(151),GNPCY(20),NSPCY(20)

DIMENSION A(209),B(209),C(209),D(209),
1 SIGMA(209,43),F(209,43),RI(209,43),COR(209),
2 RI(209),FIM1(209),FLUXJM(209),RJA3(209)
DIMENSION RJPH(209),RJMH(209),PXCJM1(209)
DIMENSION QTTRP(209),STRP(209),GNPCY(20),NSPCY(20)

SUBROUTINE INITL

DIMENSION XJDUM(151),A1DUM(151),A3DUM(151),XJ1(151),XJNJM(151),
1 A31(151),A3NJM(151)

DIMENSION XJDUM(209),A1DUM(209),A3DUM(209),XJ1(209),XJNJM(209),
1 A31(209),A3NJM(209)

SUBROUTINE OUTPUT

DIMENSION RA1(151,31)
DIMENSION IVAR(151),CP(151)
IMAX=151

DIMENSION RA1(209,43)
DIMENSION IVAR(209),CP(209)
IMAX=209

SUBROUTINE GRGEN

DIMENSION S(151),S1(151),S2(151),DUM(151),XDUM(151),YDUM(151),
1 A(151),B(151),C(151),D(151),F(151),H(151)

DIMENSION S(209),S1(209),S2(209),DUM(209),XDUM(209),YDUM(209),
1 A(209),B(209),C(209),D(209),F(209),H(209)

SUBROUTINE INNER

DIMENSION S(151),S2(151),DUM(151),XDUM(151),YDUM(151),
1 A(151),B(151),C(151),D(151),F(151),H(151)
DIMENSION SS(151),S3(151),S4(151)

DIMENSION S(209),S2(209),DUM(209),XDUM(209),YDUM(209),
1 A(209),B(209),C(209),D(209),F(209),H(209)
DIMENSION SS(209),S3(209),S4(209)
SUBROUTINE ADI

COMMON /COM2/ FX(151,31), FY(151,31), A2(151,31)
DIMENSION A(151), B(151), C(151), D(151), F(151), G(151), WORK(151)
DIMENSION QTRP(151), STRP(151)

COMMON /COM2/ FX(209,43), FY(209,43), A2(209,43)
LEVEL 2, FX, FY, A2
DIMENSION A(209), B(209), C(209), D(209), F(209), G(209), WORK(209)
DIMENSION QTRP(209), STRP(209)

SUBROUTINE CHECK

IF (NI.LE.151) GO TO 37
NI=151
IF (NJ.LE.31) GO TO 41
NJ=31

IF (NI.LE.209) GO TO 37
NI=209
IF (NJ.LE.43) GO TO 41
NJ=43
APPENDIX B

PROGRAM VARIABLES

This appendix describes the prominent variables in TAIR and, as appropriate, cross-references them with the theoretical variables they represent. The TAIR variable is listed on the left, the theoretical variable (if there is one) is listed beside it in parentheses, and the description is on the right. If the parameter is included in a COMMON block or NAMELIST, a special note is included. More complete descriptions of all NAMELIST parameters can be found in chapters 2 and 3.

AFAC  ALPHA multiplier for small ALPHAs and inner rings.
      NAMELIST: FLOWIN. COMMON: /COM1/.

AHGRID (α_H) Largest value in the grid generation ALPHA acceleration parameter sequence. NAMELIST: GRIDIN. COMMON: /GRGN/.

AHIGH (α_H) Largest value in the AF2 ALPHA acceleration parameter sequence. NAMELIST: FLOWIN. COMMON: /COM1/.

ALGRID (α_L) Smallest value in the grid generation ALPHA acceleration parameter sequence. NAMELIST: GRIDIN. COMMON: /GRGN/.

ALLOW (α_L) Smallest value in the AF2 ALPHA acceleration parameter sequence. NAMELIST: FLOWIN. COMMON: /COM1/.


ALPHA (α) AF2 acceleration parameter used in MAIN. This parameter is effectively the inverse of a pseudotime step and sequentially takes on M values ranging from AHIGH to ALLOW. (See chap. 5 for more discussion.) This parameter is also used in SUBROUTINE ADI as an ADI acceleration parameter which sequentially takes on MGRID values ranging from AHGRID to ALGRID. COMMON: /CONV/.

ARDUM  Array used for scratch storage. COMMON: /SCRACH/.

A1 (A1) Metric quantity. A1 is approximately the ratio of the normal side of a grid cell to the tangential side. (For more information about this and other metric quantities, see chaps. 3 and 5.) A1 is computed in INITL. COMMON: /COM2/.

A2 (A2) Metric quantity. A2 is proportional to the skewness of the grid cell and is computed in INITL. COMMON: /COM2/.

A3 (A3) Metric quantity. A3 is approximately the inverse of A1 and is computed in INITL. COMMON: /COM2/.

64
A4 Special metric grouping computed in INITL and used to implement the airfoil tangency boundary conditions during the residual calculation. COMMON: /COM4/.

BETA (β) $\phi_{rt}$ coefficient (acceleration parameter). (See chap. 5 for more information.) NAMELIST: FLOWIN. COMMON: /COM1/.

BHIGH The upper bound on the BETA parameter discussed in chapter 5 (SUBROUTINE AUTO).

BNN Stretching parameter for inner boundary grid point distribution. NAMELIST: GRIDIN. COMMON: /GRGN/.

BLOW The lower bound on the BETA parameter discussed in chapter 5 (SUBROUTINE AUTO).

BODYBC Special metric grouping computed in INITL and used to implement the airfoil tangency boundary conditions during the density calculation. COMMON: /COM4/.

CMA Maximum correction for each iteration. COMMON: /CONV/.

CON Parameter control for upwind bias on density. NAMELIST: FLOWIN. COMMON: /COM1/.

COR Correction array (MAIN module).

CPSTAR ($C_p$) Pressure coefficient at the sonic condition defined in SUBROUTINE INITL. COMMON: /COM1/.

CXMAX Maximum correction for the X equation calculated in the second sweep of the ADI grid generation routine (SUBROUTINE ADI).

CYMAX Maximum correction for the Y equation calculated in the second sweep of the ADI grid generation routine (SUBROUTINE ADI).

DYPRES Nondimensionalized dynamic pressure defined in INITL. COMMON: /COM1/.

ERGRID Convergence tolerance for ADI grid generation. NAMELIST: GRIDIN. COMMON: /GRGN/.

ERR Convergence tolerance for AF2 flow solver. NAMELIST: FLOWIN. COMMON: /COM1/.

ETAX ($\eta_x$) First difference of $\eta$ with respect to $x$ (SUBROUTINE INITL).

ETAY ($\eta_y$) First difference of $\eta$ with respect to $y$ (SUBROUTINE INITL).
F (f)  First sweep intermediate result. In MAIN, F is equivalenced to X and SIGMA to save storage. (See chap. 5 for more discussion about this quantity.)

FX (f)  Intermediate X result at end of first sweep of the ADI grid generation routine (SUBROUTINE ADI).

FY (g)  Intermediate Y result at end of first sweep of the ADI grid generation routine (SUBROUTINE ADI).

G (γ)  Ratio of specific heats. NAMELIST: FLOWIN. COMMON: /COM1/.

GM (γ - 1)  G - 1.0. GM is defined in INITL. COMMON: /COM1/.

GN (r^n)  Nth iteration value of circulation; initialized in INITL and used to relax the (N+1)st value of circulation at the end of the main iteration loop. COMMON: /COM1/.

GNM (r^n-1)  (N-1)st iteration value of circulation.

GNP (r^n+1)  (N+1)st value of circulation. GNP is initialized in INITL and updated to the (N+1)st value at the end of each iteration. COMMON: /COM1/.

GNPD  GNP*2.0. GNPD is initialized in INITL and updated in MAIN. COMMON: /COM1/.

GNTE  Circulation quantity involving the velocity potential jump across the airfoil trailing edge. GNTE is initialized in INITL and updated in MAIN. COMMON: /COM1/.

GP (γ + 1)  G + 1.0. GP is defined in INITL. COMMON: /COM1/.

HTMAX  Height of rectangular outer boundary (IOUT=3). NAMELIST: GRIDIN. COMMON: /GRGN/.

I  ξ coordinate index.

IA  Periodic counter used for differencing around the trailing edge. IA is set up in INITL. COMMON: /COM4/.

IAUTO  Switch for SUBROUTINE AUTO. NAMELIST: FLOWIN. COMMON: /COM1/.

ICASE  DO control variable for multicase loop. ICASE runs from one to NC (MAIN module).

ICHECK  Switch for SUBROUTINE CHECK. NAMELIST: FLOWIN. COMMON: /COM1/.
ICMAX  $\xi$ coordinate position of maximum correction (CMAK).  
COMM:  /CONV/.

IHALF  Midpoint of $\xi$-direction coordinates [(NI/2) + 1].  IHALF 
is defined in INITL.  COMM:  /COM1/.

IMH  IHALF - 1.  COMM:  /COM1/.

IHP  IHALF + 1.  COMM:  /COM1/.

JINC  Grid solution output increment.  NAMELIST: FLOWIN.  COMM: 
/COM1/.

IOPEN  Open trailing-edge option for IOPT=1.  NAMELIST: GRIDIN. 
COMM:  /GRGN/.

IOPT  Airfoil option parameter.  NAMELIST: GRIDIN.  COMM: 
/GRGN/.

IOUT  Outer boundary option parameter.  NAMELIST: GRIDIN. 
COMM:  /GRGN/.

IRMAX  $\xi$ coordinate position of maximum residual (RMAK).  COMM: 
/CONV/.

J  $\eta$ coordinate index.

JCMAX  $\eta$ coordinate position of maximum correction (CMAK). 
COMM:  /CONV/.

JINC  Grid solution output increment.  NAMELIST: FLOWIN.  COMM: 
/COM1/.

JRMAX  $\eta$ coordinate position of maximum residual (RMAK).  COMM: 
/CONV/.

K  Starting element in the AF2 ALPHA sequence.  NAMELIST: 
FLOWIN.  COMM:  /COM1/.

KGRID  Starting element in the ALPHA sequence used for grid gener-
ation.  NAMELIST: GRIDIN.  COMM:  /GRGN/.

KK  (k)  Counter for the $M$ elements in the AF2 ALPHA sequence. 
COMM:  /CONV/.

KKGRI  (k)  Counter for the $M$GRID elements in the grid generation ALPHA 
sequence (SUBROUTINE ADI).

LCHECK  Logical variable.  If an input error in a major variable is 
detected in CHECK, LCHECK will be set to true and the solu-
tion will be stopped.  COMM:  /COM1/.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Number of elements in the AF2 ALPHA sequence. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>MAXIT</td>
<td>Maximum number of iterations for grid generation. NAMELIST: GRIDIN. COMMON: /GRID/.</td>
</tr>
<tr>
<td>MESH</td>
<td>Mesh option parameter. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>MGRID</td>
<td>Number of elements in the ALPHA sequence for grid generation. NAMELIST: GRIDIN. COMMON: /GRID/.</td>
</tr>
<tr>
<td>MINF</td>
<td>Free-stream Mach number. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>N</td>
<td>DO control variable for the main iteration loop.</td>
</tr>
<tr>
<td>NC</td>
<td>Number of solutions per job (multicase run). The value of NC is established from NCASE after the first call to INITL.</td>
</tr>
<tr>
<td>NCASE</td>
<td>Number of solutions in a job. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NDIF</td>
<td>Rotated differencing parameter. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NI</td>
<td>Number of points in the $\xi$ direction (around airfoil). NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NI-1</td>
<td>NI - 1. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NI-2</td>
<td>NI - 2. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NJ</td>
<td>Number of points in the $\eta$ direction ($J=\text{NJ}$ is airfoil surface, $J=1$ is outer boundary). NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NJM</td>
<td>NJ - 1. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NJM2</td>
<td>NJ - 2. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NOUT</td>
<td>Primary output control parameter. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NOUT1</td>
<td>Solution output frequency. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NOUT2</td>
<td>Convergence parameter output frequency. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
<tr>
<td>NRING</td>
<td>Number of inner $\xi$ rings for which AFAC logic is applied. NAMELIST: FLOWIN. COMMON: /COM1/.</td>
</tr>
</tbody>
</table>
NSP  Number of supersonic points in the solution. COMMON: /CONV/.

NSTEPS  Maximum number of iterations in main flow-solver. NAMELIST: FLOWIN. COMMON: /COM1/.

NX  Number of points in the \( \xi \) direction of the mesh to be read in. This parameter should be compatible with NI (SUBROUTINE INITL).

NY  Number of points in the \( \eta \) direction of the mesh to be read in. This parameter should be compatible with NJ (SUBROUTINE INITL).

OMEG (\( \omega \))  Relaxation parameter for the ADI grid generation algorithm. NAMELIST: GRIDIN. COMMON: /GRGN/.

OMEGA (\( \omega \))  Relaxation parameter for the AF2 flow solver algorithm. NAMELIST: FLOWIN. COMMON: /COM1/.

PHI (\( \phi \))  Velocity potential array. PHI is initialized in INITL, and updated every iteration in MAIN. COMMON: / /.

PINF  Nondimensionalized free-stream pressure. PINF is defined in INITL. COMMON: /COM1/.

PSTAR  Nondimensionalized pressure at the sonic condition. PSTAR is defined in INITL.

QINF  Nondimensionalized free-stream velocity. QINF is defined in INITL. COMMON: /COM1/.

QXINF  The X component of the free-stream velocity, defined in INITL.

QYINF  The Y component of the free-stream velocity, defined in INITL. COMMON: /COM1/.

R  Residual calculated in MAIN.

RADMAX  Circular outer boundary radius (IOUT=1). NAMELIST: GRIDIN. COMMON: /GRGN/.

RATIO  The sum of the ratios of the average and maximum residuals at iteration \( N \) to the average and maximum residuals at iteration \( N-N \). (See chap. 5 for more details -- (SUBROUTINE AUTO).)

RAVG (\( R_{avg} \))  The average residual calculated in MAIN. COMMON: /CONV/.

69
Circulation relaxation parameter. RG is the working parameter which is gradually lowered as the circulation builds. (See chap. 5 for the details of this process.) Its initial value is RGAM (MAIN module).

Circulation relaxation parameter. NAMELIST: FLOWIN.

1.0-RGAM. RGAM1 is initialized in INITL, and then redefined as 1.0-RG in MAIN during the iteration process. COMMON: /COM1/.

The nondimensionalized density array. RHO is stored at (I + 1/2, J + 1/2). It is initialized in INITL and updated every iteration in MAIN. COMMON: / /.

Nondimensionalized free-stream density (SUBROUTINE INITL).

Density value at Mach = 1.3. RHO13 is used as a check to see if the local Mach number exceeds 1.3. It is defined in INITL. COMMON: /COM1/.

The density coefficient in the \( \xi \) direction. (See chap. 5 for more discussion of this parameter.)

The density coefficient in the \( \eta \) direction. (See chap. 5 for more discussion of this parameter.)

The maximum residual calculated in MAIN. COMMON: /CONV/.

Density at the sonic condition. RSTAR is defined in INITL. COMMON: /COM1/.

The X grid generation equation residual (SUBROUTINE ADI).

The maximum X grid generation equation residual (SUBROUTINE ADI).

The Y grid generation equation residual (SUBROUTINE ADI).

The maximum Y grid generation equation residual (SUBROUTINE ADI).

Airfoil arc-length distribution calculated from the initial airfoil coordinates (XB and YB) (SUBROUTINE INNER).

Switching function for upwinding of the density. (See chap. 5 for details of the \( v \) calculation). SIGMA is equivalenced to \( F \) and \( X \) to save storage.
The desired airfoil arc-length distribution, dependent on the clustering parameter BINN (SUBROUTINE INNER).

TEOPEN
Distance across an open trailing edge. TEOPEN is calculated in GEMPAC, if MESH=0, or in INITL, if MESH > 0. COMMON: /COM1/.

TITLE
The title of the airfoil or mesh used in a given solution (28-character string). COMMON: /COM5/.

TMAX
Airfoil thickness parameter. NAMELIST: GRIDIN. COMMON: /GRGN/.

WIDTHMX
Width of rectangular outer boundary (IOUT=3). NAMELIST: GRIDIN. COMMON: /GRGN/.

X (x)
The x coordinate of the finite-difference grid. In MAIN, X is equivalenced to SIGMA and F. COMMON: /SCRACH/.

XB
Airfoil surface x coordinates stored at (I + 1/2, NJ). COMMON: /COM3/.

XC
Leading-edge bluntness parameter for IOPT=2 option. NAMELIST: GRIDIN. COMMON: /GRGN/.

XCN
The X coordinate for the center of the circular outer boundary (IOUT=1). NAMELIST: GRIDIN. COMMON: /GRGN/.

XETA (x_η)
First difference of x with respect to η (SUBROUTINE INITL).

XIX (ξ_x)
First difference of ξ with respect to x (SUBROUTINE INITL).

XIX (ξ_y)
First difference of ξ with respect to y (SUBROUTINE INITL).

XJ (J)
The Jacobian of the numerical mapping transformation. XJ is computed in INITL. (See chaps. 3 and 5 for more information.) COMMON: /COM2/.

XXI (x_ξ)
First difference of x with respect to ξ (SUBROUTINE INITL).

Y (y)
The y coordinate of the finite-difference grid. In MAIN, Y is equivalenced to RI. COMMON: /SCRACH/.

YB
Airfoil surface y coordinates stored at (I + 1/2, NJ). COMMON: /COM3/.

YCN
The Y coordinate for the center of the circular outer boundary (IOUT=1). NAMELIST: GRIDIN. COMMON: /GRGN/.

YETA (y_η)
First difference of y with respect to η (SUBROUTINE INITL).

YXI (y_ξ)
First difference of y with respect to ξ (SUBROUTINE INITL).
REFERENCES


<table>
<thead>
<tr>
<th>NOUT</th>
<th>Output information</th>
<th>Total number of pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x x x x x x x x x x Aerodynamic coefficients</td>
<td>2 lines</td>
</tr>
<tr>
<td>2</td>
<td>x x x x x x x x Last line of convergence history</td>
<td>5 lines</td>
</tr>
<tr>
<td>3</td>
<td>x x x x x x x x Airfoil surface solution</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>x x x x x x x x Input data (NAMELISTs, airfoil</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>coordinates)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>x x x x x x x x Interpolated airfoil coordinates</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>x x x x x x x x Complete convergence history</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>x x x x x x x x Mach and RHO contour maps</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>x x x x x x x x Grid convergence history and trans-</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>formation metrics near airfoil</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>x x x x x x x x Final grid coordinates</td>
<td>21</td>
</tr>
</tbody>
</table>

74
<table>
<thead>
<tr>
<th>Routine name</th>
<th>Calling routine</th>
<th>Routines called</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADI</td>
<td>GREGEN</td>
<td>TRIB, TRIP</td>
<td>Grid-generation algorithm</td>
</tr>
<tr>
<td>AUTO</td>
<td>MAIN</td>
<td></td>
<td>Solution parameter updating</td>
</tr>
<tr>
<td>CHECK</td>
<td>INITL, GREGEN</td>
<td></td>
<td>Checks input data</td>
</tr>
<tr>
<td>CSPLIN</td>
<td>INNER, GREGEN</td>
<td>TRIB</td>
<td>Cubic spline interpolation</td>
</tr>
<tr>
<td>FORCE</td>
<td>OUTPUT</td>
<td></td>
<td>Aerodynamic coefficient calculation</td>
</tr>
<tr>
<td>GEMPAC</td>
<td>GREGEN</td>
<td></td>
<td>Aerodynamic geometry input</td>
</tr>
<tr>
<td>GREGEN</td>
<td>INITL, CHECK, GEMPAC, INNER, OUTER ADI, CSPLIN</td>
<td>Grid generation input (NAMELIST GRIDIN) and grid initialization</td>
<td></td>
</tr>
<tr>
<td>INITL</td>
<td>MAIN</td>
<td>CHECK, GREGEN, XYOUT</td>
<td>Solution input (NAMELIST FLOWIN) and flow-field initialization</td>
</tr>
<tr>
<td>INNER</td>
<td>GREGEN</td>
<td>CSPLIN</td>
<td>Distributes grid points on inner (airfoil) boundary</td>
</tr>
<tr>
<td>MAIN</td>
<td>INITL, OUTPUT, AUTO, TRIP</td>
<td></td>
<td>Main flow-solver algorithm</td>
</tr>
<tr>
<td>OUTER</td>
<td>GREGEN</td>
<td></td>
<td>Outer boundary shape and point distribution</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>MAIN</td>
<td>FORCE, PRNPLT</td>
<td>Main flow-solver output</td>
</tr>
<tr>
<td>PRNPLT</td>
<td>OUTPUT</td>
<td></td>
<td>Printer contour plot</td>
</tr>
<tr>
<td>TRIB</td>
<td>CSPLIN, ADI</td>
<td></td>
<td>Tridiagonal inversion algorithm</td>
</tr>
<tr>
<td>TRIP</td>
<td>ADI, MAIN</td>
<td></td>
<td>Tridiagonal periodic inversion algorithm</td>
</tr>
<tr>
<td>XYOUT</td>
<td>INITL</td>
<td></td>
<td>Grid-generation output</td>
</tr>
<tr>
<td>COMMON block</td>
<td>Calling routine</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>/ /</td>
<td>MAIN, INITL, GRGEN, AUTO</td>
<td>Blank COMMON...Velocity potential and density arrays</td>
<td></td>
</tr>
<tr>
<td>/COM1/</td>
<td>MAIN, INITL, OUTPUT, GRGEN, CHECK, AUTO</td>
<td>General solution parameters and NAMELIST FLOWIN parameters</td>
<td></td>
</tr>
<tr>
<td>/COM2/</td>
<td>MAIN, INITL, OUTPUT, XYOUT, ADI</td>
<td>Metric quantities</td>
<td></td>
</tr>
<tr>
<td>/COM3/</td>
<td>INITL, OUTPUT, GRGEN, INNER, CHECK</td>
<td>Airfoil coordinates</td>
<td></td>
</tr>
<tr>
<td>/COM4/</td>
<td>MAIN, INITL, OUTPUT, XYOUT, ADI</td>
<td>Periodic indexing array and airfoil boundary condition arrays</td>
<td></td>
</tr>
<tr>
<td>/COM5/</td>
<td>INITL, OUTPUT, GRGEN, GEMPAC</td>
<td>Airfoil titles</td>
<td></td>
</tr>
<tr>
<td>/CONV/</td>
<td>MAIN, OUTPUT, AUTO</td>
<td>Solution convergence parameters</td>
<td></td>
</tr>
<tr>
<td>/GRGN/</td>
<td>GRGEN, ADI, CHECK, MAIN</td>
<td>Grid-generation parameters and NAMELIST GRIDIN parameters</td>
<td></td>
</tr>
<tr>
<td>/SCRACH/</td>
<td>MAIN, INITL, OUTPUT, ADI, XYOUT, GRGEN, INNER, OUTER</td>
<td>Scratch storage</td>
<td></td>
</tr>
</tbody>
</table>
Figure 1.- Numerically generated finite-difference mesh.
Figure 2.— Comparison of GRAPE and TAIR trailing edges.

Figure 3.— Bluntness of circular-arc airfoil.
Figure 4.- TAIR SUBROUTINE tree.
Figure 5.- Flowchart for MAIN module.
Figure 6.— Logic flow for SUBROUTINE INITL.
Figure 7.- Logic flowchart for SUBROUTINE GRGEN.
Figure 8.— TAIR-generated finite-difference grid (the default case).
Figure 9.—Pressure coefficient comparison (Korn airfoil, $M_\infty = 0.74$, $\alpha = 0^\circ$).
AF2 (CYCLE 8, PRESENT)
· AF2 (CYCLE 7, REF. 15)
□ HYBRID
△ SLOR

Figure 10. - Convergence-history comparisons (Korn airfoil, $M_\infty = 0.74$, $\alpha = 0^\circ$).
Figure 11.— Mach number contours (NACA 0012 airfoil, $M_\infty = 0.95$, $\alpha = 4^\circ$).

Figure 12.— Pressure coefficient comparison (circular cylinder cross section, $M_\infty = 0.00001$, $\alpha = 0^\circ$).
Figure 13.- Pressure coefficient comparison (NACA 0012 airfoil, $M_{\infty} = 0.00001, \alpha = 0^\circ$).

Figure 14.- Pressure coefficient comparison (CAST 7 airfoil, $M_{\infty} = 0.7$).
Figure 15. - Pressure coefficient comparison (CAST 7 airfoil, $M_\infty = 0.7$).

Figure 16. - Pressure coefficient comparison (CAST 7 airfoil, $M_\infty = 0.7$).
The operation of the TAIR (Transonic AIRfoil) computer code, which uses a fast, fully implicit algorithm to solve the conservative full-potential equation for transonic flow fields about arbitrary airfoils, is described. The code description is given in two levels of sophistication: simplified operation and detailed operation. In addition, detailed descriptions of the program organization and theory are provided to simplify modification of TAIR for new applications. Examples with input and output are given for a wide range of cases, including incompressible, subcritical compressible, and transonic calculations.

**Abstract**

Transonic flow
Numerical methods

---

17. Key Words (Suggested by Author(s))

19. Security Classif. (of this report)

Unclassified

21. No. of Pages

90

22. Price

$9.50

*For sale by the National Technical Information Service, Springfield, Virginia 22161*
**DO NOT REMOVE SLIP FROM MATERIAL**

Delete your name from this slip when returning material to the library.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DATE</th>
<th>MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M. Mueller</td>
<td>6/04/93</td>
<td>12</td>
</tr>
<tr>
<td>John Batista</td>
<td>10/22</td>
<td>499</td>
</tr>
</tbody>
</table>

NASA Langley (Rev. Dec. 1991)