ISAAC

Version 4.2

Users Guide

Joseph H. Morrison

Acknowledgements

The ISAAC code was originally developed under contract to the NASA Langley Research Center while the author was employed at Analytical Services and Materials, Inc., 107 Research Dr., Hampton, VA 23666.

Contents

Chapter 1

Introduction

ISAAC (*Integrated Solution Algorithm for Arbitrary Configurations*) is an Euler/Navier-Stokes computational fluid dynamics code. The code includes the capability of calculating the Euler equations for inviscid flow or the Navier-Stokes equations for viscous flows. ISAAC uses a domain decomposition structure to accommodate complex physical configurations. ISAAC provides options to calculate either steady-state or time dependent flow.

1.1 Turbulence Capabilities

ISAAC has been developed to test a large range of turbulence models. The turbulence models included in ISAAC include algebraic models, various k-ϵ eddy viscosity models, the k-ω eddy viscosity model, algebraic Reynolds stress models using both k-ϵ and k-ω formulations, and Reynolds stress transport models. Several of these models are developed for near wall integration and some are for high Reynolds number turbulence flows only. A wall function capability is included to integrate the high Reynolds number models on configurations with walls.

1.2 Unsteady Flows

ISAAC is capable of calculating unsteady flows using either an explicit Runge-Kutta time integration scheme or a second-order accurate iterative, implicit time integration scheme. The iterative, implicit time integration allows for the calculation of unsteady flows using a time step which is larger than the explicit scheme allows while retaining second-order temporal accuracy.

1.3 Geometric Generality

ISAAC uses a structured block approach to allow the discretization of complex geometries. This approach requires that grid distributions match on adjoining blocks. While this is not the most general approach for complex geometries, it allows for the discretization of many geometries. A separate grid generation capability is required to be used with ISAAC.

Additionally, ISAAC provides for options to calculate either two-dimensional or axisymmetric flows without the full computational time and storage overhead of a three-dimensional calculation.

Chapter 2

Governing Equations and Turbulence Models

2.1 Mean Equations

The compressible Reynolds averaged Navier-Stokes equations written in Favre variables are:

$$\frac{\partial}{\partial t}(\overline{\rho}) + \frac{\partial}{\partial x_j}(\overline{\rho}\tilde{u}_j) = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}(\overline{\rho}\tilde{u}_i) + \frac{\partial}{\partial x_j}(\overline{\rho}\tilde{u}_i\tilde{u}_j + \overline{p}\delta_{ij}) = \frac{\partial}{\partial x_j}(\overline{\sigma_{ij}}) - \frac{\partial}{\partial x_j}(\overline{\rho}\widetilde{u_i''u_j''}) \tag{2.2}$$

with

$$\overline{\sigma_{ij}} = \overline{\mu\left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right]} \simeq \overline{\mu}\left[\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) - \frac{2}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}\right] \tag{2.3}$$

and the Reynolds stress is defined as $\tau_{ij} \equiv \widetilde{u_i''u_j''}$

$$\begin{aligned} \frac{\partial}{\partial t}(\overline{\rho}\tilde{E}) + \frac{\partial}{\partial x_j}[\tilde{u}_j(\overline{\rho}\tilde{E} + \overline{p})] \quad = \quad & \frac{\partial}{\partial x_j}(\tilde{u}_i\overline{\sigma_{ij}} + \overline{\sigma_{ij}}\overline{u_i''} + \overline{\sigma'_{ij}u'_i}) - \\ & \frac{\partial}{\partial x_j}(\overline{q_j} + \overline{\rho}\tilde{u}_i\tau_{ij} + c_p\overline{\rho}\widetilde{u_j''T''} + \frac{\overline{\rho}\widetilde{u_i''u_i''u_j''}}{2}) \end{aligned} \tag{2.4}$$

and

$$\overline{q_j} = \overline{-k\frac{\partial T}{\partial x_j}} \simeq -\overline{k}\frac{\partial \tilde{T}}{\partial x_j} \tag{2.5}$$

2.2 Turbulence Equations

The resulting mean equations require closure for the unknown correlations that arise from the averaging procedures: the Reynolds stresses τ_{ij}, the turbulent heat flux $\overline{\rho}c_p\widetilde{u_j''T''}$, and the turbulent transport (triple velocity correlation) $\overline{\rho}\widetilde{u_i''u_i''u_j''}$.

2.2.1 Two-Equation Eddy Viscosity Turbulence Models

The Reynolds stresses are modeled through the Boussinesq assumption as

$$-\overline{\rho}\tau_{ij} = \mu_t \left[\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} \right] - \frac{2}{3}\overline{\rho}K\delta_{ij} \tag{2.6}$$

K-ε Eddy Viscosity Models

The K-ε eddy viscosity models can be written in the following general form:

$$\frac{\partial}{\partial t}(\overline{\rho}K) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{u}_k K) = \overline{\rho}\mathcal{P} - \overline{\rho}\varepsilon + \frac{\partial}{\partial x_k}\left[\left(\overline{\mu} + \frac{\mu_t}{\sigma_K} \right) \frac{\partial K}{\partial x_j} \right] \tag{2.7}$$

$$\frac{\partial}{\partial t}(\overline{\rho}\varepsilon) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{u}_k \varepsilon) = C_{\varepsilon 1}\frac{\overline{\rho}\varepsilon}{K}\mathcal{P} - C_{\varepsilon 2}\frac{\overline{\rho}\varepsilon^2}{K} + \frac{\partial}{\partial x_k}\left[\left(\overline{\mu} + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] - \mathcal{R} \tag{2.8}$$

where the eddy viscosity is calculated as

$$\mu_t = C_\mu \frac{\overline{\rho}K^2}{\varepsilon} \tag{2.9}$$

the production is given as

$$\mathcal{P} = -\tau_{ij}\frac{\partial \tilde{u}_i}{\partial x_j} \tag{2.10}$$

The coefficients for the K-ε models are given in Table 2.1 for the standard, high Reynolds number K-ε model [1], and the Renormalization Group model (RNG) [2], and the Anisotropic Dissipation Rate model (ADRM) [3].

The factor $f(\zeta)$ for the RNG K-ε model is

$$f(\zeta) = \frac{\zeta(1 - \zeta/4.38)}{(1 + 0.012\zeta^3)} \tag{2.11}$$

where $\zeta = (2\tilde{S}_{ij}\tilde{S}_{ij})^{1/2}K/\varepsilon$.

The coefficient $g(\eta, \xi)$ for the ADRM model

$$g(\eta, \xi) = \frac{2(1+\alpha)}{15C_\mu}\left[\frac{C_{\varepsilon 5} + 2C_\mu \eta^2 - 1}{(C_{\varepsilon 5} + 2C_\mu\eta^2 - 1)^2 - \frac{2}{3}\beta_2^2\eta^2 + 2\beta_1^2\xi^2} \right] \tag{2.12}$$

is a function of the invariants of the strain rate tensor and the rotation tensor

$$\eta = \frac{K}{\varepsilon}(\tilde{S}_{ij}\tilde{S}_{ij})^{1/2} \tag{2.13}$$

$$\xi = \frac{K}{\varepsilon}(\tilde{W}_{ij}\tilde{W}_{ij})^{1/2} \tag{2.14}$$

where the strain rate tensor and the rotation tensor are given as

$$\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \tag{2.15}$$

$$\tilde{W}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{u}_j}{\partial x_i} \right) \tag{2.16}$$

and $\alpha = (21\alpha_3 - 8)/22$, $\beta_1 = (7\alpha_3 + 1)/11$, $\beta_2 = (15\alpha_3 - 1)/11$, $\alpha_3 = 0.6$ and $C_{\varepsilon 5} = 5.8$.

	C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_K	σ_ε	$\mathcal{R}$
Standard K-ε	0.09	1.44	1.92	1.0	1.30	0
RNG K-ε	0.085	1.42	1.68	0.719	0.719	$f(\zeta)\mathcal{P}\overline{\rho}\varepsilon/K$
ADRM K-ε	0.094	$1.2+g(\eta,\xi)$	1.83	1.0	1.87	0

Table 2.1: Constants for K-ε models.

K-ω Eddy Viscosity Model

The K-ω eddy viscosity model of Wilcox [11] can be written as

$$\frac{\partial}{\partial t}(\overline{\rho}K) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{u}_k K) = \overline{\rho}\mathcal{P} - C_{K2}\overline{\rho}\omega K + \frac{\partial}{\partial x_k}\left[\left(\overline{\mu} + \frac{\mu_t}{\sigma_K}\right)\frac{\partial K}{\partial x_j}\right] \tag{2.17}$$

$$\frac{\partial}{\partial t}(\overline{\rho}\omega) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{u}_k \omega) = C_{\omega 1}\frac{\overline{\rho}\omega}{K}\mathcal{P} - C_{\omega 2}\overline{\rho}\omega^2 + \frac{\partial}{\partial x_k}\left[\left(\overline{\mu} + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] \tag{2.18}$$

with the eddy viscosity calculated as

$$\mu_t = C_\mu \frac{\overline{\rho}K}{\omega} \tag{2.19}$$

and the Reynolds stresses are calculated from the Boussinesq assumption as in the K-ε model. The coefficients for the K-ω model are given in Table 2.2.

C_μ	C_{K2}	$C_{\omega 1}$	$C_{\omega 2}$	σ_K	σ_ω
1.0	0.09	5/9	3/40	2.0	2.0

Table 2.2: K-ω model coefficients.

2.2.2 Explicit Algebraic Stress Models

The explicit algebraic stress model of Abid *et al.* [18]

$$\begin{aligned} -\overline{\rho}\tau_{ij} = {} & 2\mu_t^* \left[\left(\tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij}\right) + \alpha_4 \frac{K}{\varepsilon}\left(\tilde{S}_{ik}\tilde{W}_{kj} + \tilde{S}_{jk}\tilde{W}_{ki}\right)\right. \\ & \left. -\alpha_5 \frac{K}{\varepsilon}\left(\tilde{S}_{ik}\tilde{S}_{kj} - \frac{1}{3}\tilde{S}_{kl}\tilde{S}_{kl}\delta_{ij}\right)\right] - \frac{2}{3}\overline{\rho}K\delta_{ij} \end{aligned} \tag{2.20}$$

where

$$\mu_t^* = \alpha_1 \frac{3(1+\eta^2) + 0.2(\eta^6 + \xi^6)}{3 + \eta^2 + 6\eta^2\xi^2 + 6\xi^2 + \eta^6 + \xi^6} \tag{2.21}$$

and the constants are given as:

$$\begin{array}{lll} \alpha_1 = \left(\frac{4}{3} - C_2\right) g/2 & \alpha_2 = (2 - C_3)^2 g^2/4 & \alpha_3 = (2 - C_4)^2 g^2/4 \\ \alpha_4 = (2 - C_4)\, g/2 & \alpha_5 = (2 - C_3)\, g & g = \frac{1}{(C_1/2) + C_5 - 1} \end{array} \tag{2.22}$$

and the coefficients from the SSG [6] pressure-strain correlation are $C_1 = 6.8$, $C_2 = 0.36$, $C_3 = 1.25$, $C_4 = 0.40$, and $C_5 = 1.88$.

The coefficients for the K-ε EASM model are given in Table 2.3.

	C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_K	σ_ε	$\mathcal{R}$
EASM K-ε	0.081	1.44	1.83	1.0	1.51	0

Table 2.3: Constants for explicit algebraic stress model.

2.2.3 Reynolds Stress Turbulence Models

Reynolds stress models solve transport equations for the individual stress components directly rather than using an algebraic relationship for the stresses. These transport equations are coupled with a transport equation for the isotropic dissipation rate.

$$\begin{aligned}\frac{\partial}{\partial t}(\overline{\rho}\tau_{ij}) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{u}_k\tau_{ij}) &= \overline{\rho}\tilde{P}_{ij} + \overline{\rho}\Pi_{ij} - \overline{\rho}\varepsilon_{ij} \\ &\quad + \frac{\partial}{\partial x_k}(\overline{\rho}D^v_{ijk}) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{D}^t_{ijk}) \qquad (2.23)\\ \frac{\partial}{\partial t}(\overline{\rho}\varepsilon) + \frac{\partial}{\partial x_k}(\overline{\rho}\tilde{u}_k\varepsilon) &= C_{\varepsilon 1}\frac{\overline{\rho}\varepsilon}{K}\mathcal{P} - C_{\varepsilon 2}\frac{\overline{\rho}\varepsilon^2}{K} + \mathcal{R} \\ &\quad + \frac{\partial}{\partial x_k}\left(\overline{\mu}\frac{\partial\varepsilon}{\partial x_k}\right) + \frac{\partial}{\partial x_i}\left(\overline{\rho}\tilde{D}^t_{\varepsilon j}\right) \qquad (2.24)\end{aligned}$$

where $\overline{\rho}\tilde{P}_{ij} = -\overline{\rho}\tau_{ik}\partial\tilde{u}_j/\partial x_k - \overline{\rho}\tau_{jk}\partial\tilde{u}_i/\partial x_k$ is the exact production term, Π_{ij} is the modeled pressure-strain term, D^v_{ijk} is the molecular diffusion, and $\tilde{D}^t_{ijk}$ is the turbulent diffusion. The dissipation rate tensor is modeled isotropically for high Reynolds number turbulent flows ($\varepsilon = \varepsilon_{ii}/2$)

$$\varepsilon_{ij} = \frac{2}{3}\varepsilon\delta_{ij} \qquad (2.25)$$

There have been many pressure-strain models developed since the pioneering work of Launder, Reece and Rodi [4]. Several of these pressure-strain models are available. The pressure-strain may be written in a general form as

$$\begin{aligned}\Pi_{ij} &= -C_1\varepsilon b_{ij} + C_1'\varepsilon\left(b_{ik}b_{kj} - \frac{1}{3}b_{mn}b_{mn}\delta_{ij}\right) + C_2K\tilde{S}_{ij} \\ &\quad + C_3K\left(b_{ik}\tilde{S}_{jk} + b_{jk}\tilde{S}_{ik} - \frac{2}{3}b_{mn}\tilde{S}_{mn}\delta_{ij}\right) \\ &\quad + C_4K\left(b_{ik}\tilde{W}_{jk} + b_{jk}\tilde{W}_{ik}\right) \qquad (2.26)\end{aligned}$$

where

$$b_{ij} = \left(\frac{\tau_{ij}}{2K} - \frac{1}{3}\delta_{ij}\right) \qquad (2.27)$$

is the stress anisotropy tensor.

The coefficients for the Gibson-Launder [5] and Speziale-Sarkar-Gatski (SSG) [6] pressure-strain models are given in Table 2.4.

The molecular diffusion is modeled as

$$\overline{\rho}D^v_{ijk} = \overline{\mu}\left(\frac{\partial\tau_{jk}}{\partial x_i} + \frac{\partial\tau_{ki}}{\partial x_j} + \frac{\partial\tau_{ij}}{\partial x_k}\right) \qquad (2.28)$$

	C_1	C_1'	C_2	C_3	C_4
Gibson-Launder	3.6	0.	0.8	1.2	1.2
SSG	$3.4 + 1.8\mathcal{P}/\varepsilon$	4.2	$0.8 - 1.30(b_{ij}b_{ij})^{1/2}$	1.25	0.40

Table 2.4: Pressure-strain model constants.

There are three turbulent diffusion models that are available:
Daly and Harlow [7]

$$\tilde{D}^t_{ijk} = C_s \frac{K}{\varepsilon} \tau_{kl} \frac{\partial \tau_{ij}}{\partial x_l} \tag{2.29}$$

Hanjalic and Launder [8]

$$\tilde{D}^t_{ijk} = C_s \frac{K}{\varepsilon} \left(\tau_{il} \frac{\partial \tau_{jk}}{\partial x_l} + \tau_{jl} \frac{\partial \tau_{ki}}{\partial x_l} + \tau_{kl} \frac{\partial \tau_{ij}}{\partial x_l} \right) \tag{2.30}$$

Mellor and Herring [9]

$$\tilde{D}^t_{ijk} = C_s \frac{K^2}{\varepsilon} \left(\frac{\partial \tau_{jk}}{\partial x_i} + \frac{\partial \tau_{ki}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_k} \right) \tag{2.31}$$

The turbulent diffusion for the ε equation is modeled as

$$\overline{\rho} \tilde{D}^t_{\varepsilon j} = \overline{\rho} C_\varepsilon \left(\tau_{ij} \frac{K}{\varepsilon} \frac{\partial \varepsilon}{\partial x_j} \right) \tag{2.32}$$

The coefficients for the ε equation for the Gibson-Launder, SSG and SSG-ADRM models are given in Table 2.5.

For the anisotropic dissipation rate model (SSG-ADRM), the anisotropy of dissipation tensor, d_{ij}, which is defined as

$$d_{ij} = \left(\frac{\varepsilon_{ij}}{2\varepsilon} - \frac{1}{3} \delta_{ij} \right) \tag{2.33}$$

is modeled with the anisotropic model of Speziale and Gatski [3]

$$d_{ij} = -2C_{\mu\varepsilon} \left[\frac{K}{\varepsilon} \tilde{S}_{ij} + \beta_1^* \frac{K^2}{\varepsilon^2} (\tilde{S}_{ik} \tilde{W}_{kj} + \tilde{S}_{jk} \tilde{W}_{ki}) + \beta_2^* \frac{K^2}{\varepsilon^2} (\tilde{S}_{ik} \tilde{S}_{kj} - \frac{1}{3} \tilde{S}_{kl} \tilde{S}_{kl} \delta_{ij}) \right] \tag{2.34}$$

where

$$C_{\mu\varepsilon} = \frac{1}{15(C_{\varepsilon 5} + \frac{\mathcal{P}}{\varepsilon} - 1)} \left[1 + 2\beta_1^{*2} \xi^2 - \frac{2}{3} \beta_2^{*2} \eta^2 \right]^{-1} \tag{2.35}$$

and

$$\beta_1^* = \frac{7\alpha_3 + 1}{11(C_{\varepsilon 5} + \frac{\mathcal{P}}{\varepsilon} - 1)} \tag{2.36}$$

$$\beta_2^* = \frac{15\alpha_3 - 1}{11(C_{\varepsilon 5} + \frac{\mathcal{P}}{\varepsilon} - 1)} \tag{2.37}$$

	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	C_s	C_ε	$\mathcal{R}$
Gibson-Launder	1.44	1.92	0.11	0.16	0
SSG	1.44	1.83	0.11	0.184	0
SSG-ADRM	1.00	1.83	0.11	0.217	$-2(1+\alpha)\overline{\rho}\varepsilon d_{ij}\tilde{S}_{ij}$

Table 2.5: Reynolds stress model constants.

Chapter 3

Example Cases

3.1 NACA 0012 Inviscid Airfoil

3.1.1 Case Description

The NACA 0012 test case is an inviscid airfoil test case. This test case demonstrates the capability of ISAAC for inviscid airfoils and also demonstrates the grid sequencing and multigrid for convergence acceleration.

3.1.2 Flow Parameters

The flow parameters from the calculation are given in Table 3.1.

M_∞	0.8
α	1.25°

Table 3.1: Flow parameters for NACA 0012 airfoil.

3.1.3 Results

Figure 3.1 shows the 225 × 33 C-grid around the airfoil. The grid extends approximately 20 chords away from the airfoil. Figure 3.2 shows the pressure contours on the airfoil. Figure 3.3 shows the surface pressure coefficient for the airfoil. The strong shock is clearly evident on the upper surface at $x/c \approx 0.6$ and a weaker shock is apparent on the lower surface at $x/c \approx 0.35$.

Figure 3.4 shows the convergence history for the residual and the coefficient of lift for the NACA 0012 airfoil with and without multigrid. Multigrid improves the convergence dramatically for this case.

3.1.4 Example Input

Figure 3.5 shows the input for ISAAC to run the inviscid flow on the NACA 0012 airfoil. The calculation is run using mesh sequencing with 50 iterations computed on the 57 × 9

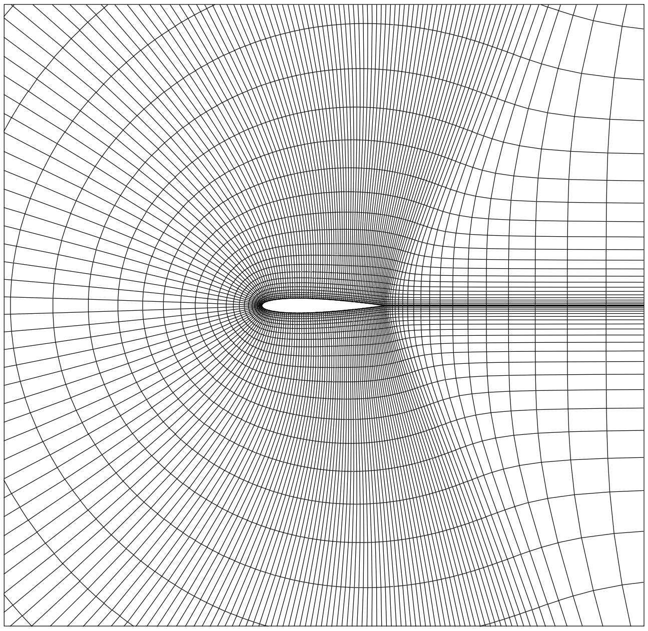

Figure 3.1: 225 × 33 C-grid for the NACA 0012 airfoil.

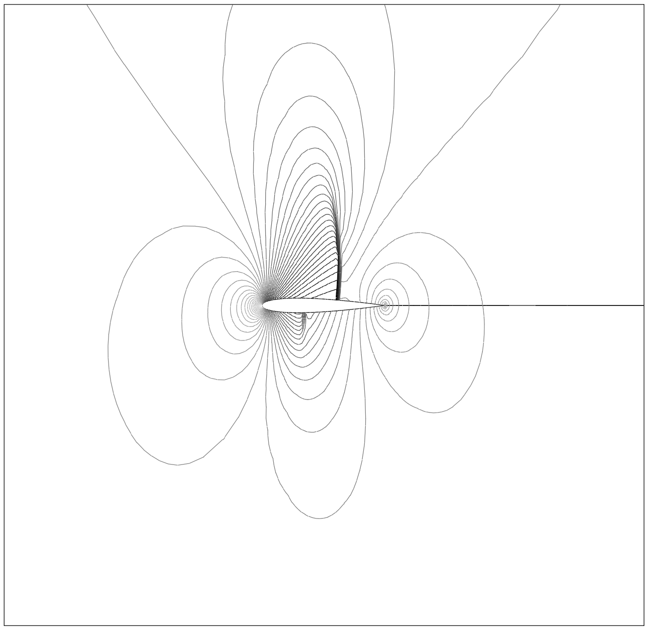

Figure 3.2: Pressure contours for NACA 0012.

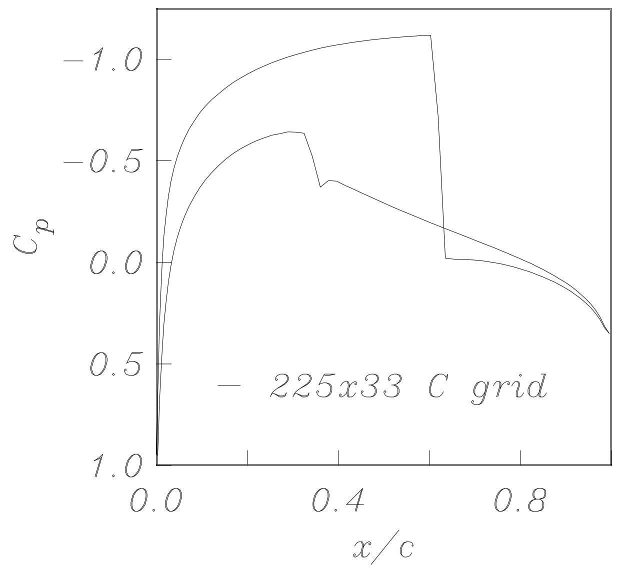

Figure 3.3: Surface pressure coefficient for NACA 0012 airfoil.

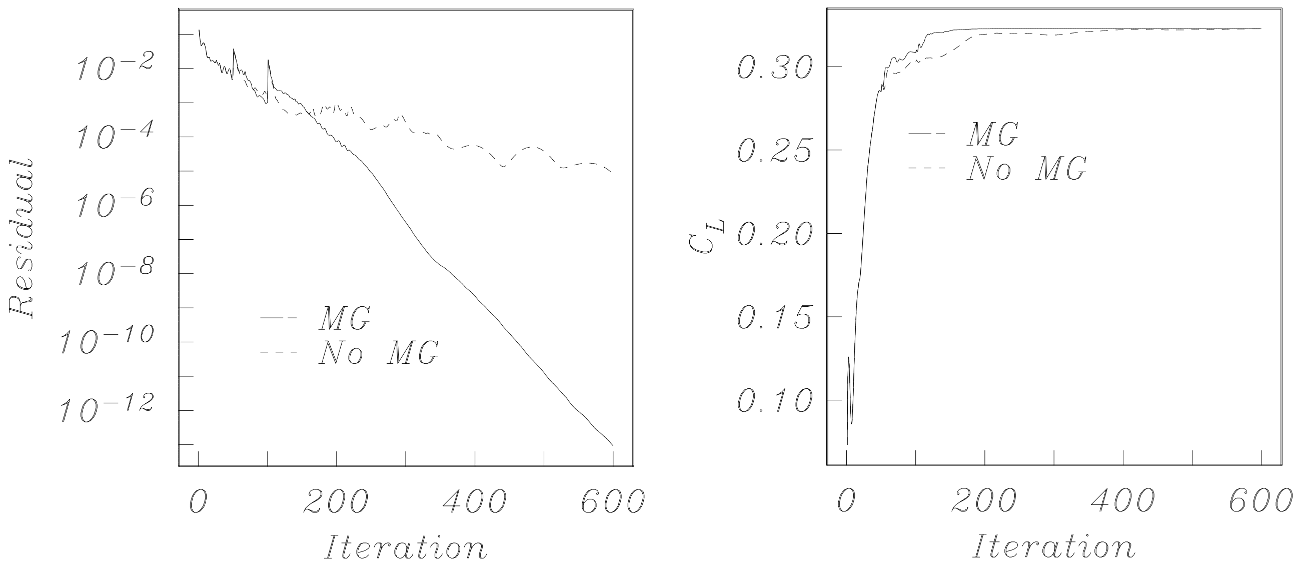

Figure 3.4: Residual and lift convergence history for NACA 0012 airfoil.

mesh followed by 50 iterations on the 113 × 17 mesh and a final 500 iterations on the fine 225 × 33 mesh. Three levels of multigrid are used on the finest mesh. The CFL is constant at 10.

3.1.5 Example Files

The data corresponding to the NACA 0012 test case is available in directory

```
ISAAC/example_cases/NACA0012/Grid/gr224n20
```

The original grid (provided by Dr. C. Swanson of NASA Langley) is a text file located in sub-directory `Grid/gr224n20`. A FORTRAN program is included to convert the text file to a binary PLOT3D file format to be read by ISAAC. This FORTRAN program is available at `Grid/cnvrtgrd.f`. The output from the above runs is given in sub-directory `Results`. This sub-directory includes the above example input file, the output, and the PLOT3D output for plotting.

```
'RESTART OUT'   1
     'naca0012.rst1'
'PLOT3D'  0
     'FORMATTED'
     'naca0012'
'MACH'       0.8e0
'ALPHA'      1.25
'KAPPA'      0.3333333333E0
'LIMIT'      3
'MULTIGRID LEVELS' 3
'MESH SEQUENCE'    2
     50
     50
'NITS'       500
'CFL'        10.0
'ITUPDT'     5
'TWOD'       1
'BLOCK'      1
     'DIMENSIONS'       1
           225      33        2
     'FLUX'    1
          'ROE'        1
     'GRID'       1
          'n12_225_33.grd'
     'BC'         6
          'BC'      33   1   1      193   1   2   'TANGENCY'
          'BC'       1  33   1      225  33   2   'FARFIELD'
          'BC'       1   1   1        1  33   2   'EXTRAPOLATE'
          'BC'     225   1   1      225  33   2   'EXTRAPOLATE'
          'BC'       1   1   1      225  33   1   'EXTRAPOLATE'
          'BC'       1   1   2      225  33   2   'EXTRAPOLATE'
'END BLOCK'    1
'CUT'   1.
     'ONE'     1       1   1   1        33   1   2
     'ONE'     1     225   1   1       193   1   2       1  2  3
'PRINT'    2
      1       1          1    1    2     226   1    2      1  1  1
      1       2         33    1    2     193   34   2     40  1  1
'END'       0
```

Figure 3.5: Input file for inviscid NACA 0012 airfoil.

3.2 Klebanoff Zero Pressure Gradient Flat Plate

3.2.1 Case Description

Klebanoff[19] measured an incompressible zero pressure gradient flat plate flow using a hot-wire anemometer. The boundary layer was tripped using a region roughened with sand. The free-stream velocity was 50 feet per second. The measurement location was 10.5 feet from the plate leading edge which corresponded to an apparent developmental length of 14.2 feet of smooth surface. The Reynolds number based on the distance from the virtual origin was 4.2×10^6. The boundary layer thickness was 3 inches at the measurement station. The measurements included the mean velocity and turbulence fluctuations. Additional measurements included spectra, skewness, flatness, intermittency, and probability density.

3.2.2 Flow Parameters

The flow parameters from the calculation are given in Table 3.2.

Re	4.2×10^6
δ	3 inches
U_∞	50 ft./sec.
L	14.2 ft.

Table 3.2: Flow parameters for Klebanoff flat plate.

3.2.3 Results

The grid for the flat plate extends $0.5L$ upstream of the leading edge of the flat plate. The outflow condition is applied $0.5L$ downstream of the measurement location which is located at $x = L$. The top of the grid is located approximately 10δ from the plate. The grid is 65×65 equally spaced in the streamwise direction and stretched to the wall. The first grid point is located at a $y^+ \approx 0.4$ with approximately 43 points located in the boundary layer.

Figure 3.6 shows the velocity calculated with the Speziale-Abid-Anderson [10] k-ε model, the algebraic stress model [18], the k-ω model [11], and the Zhang-So-Gatski-Speziale [12] Reynolds stress model.

Figure 3.7 shows the turbulence intensities and Reynolds shear stress calculated with the Speziale-Abid-Anderson k-ε model [10], the algebraic stress model [18], the k-ω model [11], and the Zhang-So-Gatski-Speziale Reynolds stress model [12].

3.2.4 Example Input

Figure 3.8 shows the input for ISAAC to run the algebraic stress model on the flat plate. The calculation is run using mesh sequencing with 500 iterations computed on the 17×17 mesh followed by 1000 iterations on the 33×33 mesh and a final 1000 iterations on the fine 65×65 mesh. The CFL ramps from 1 on the first iteration to 20 on the 50^{th} iteration on each of the meshes.

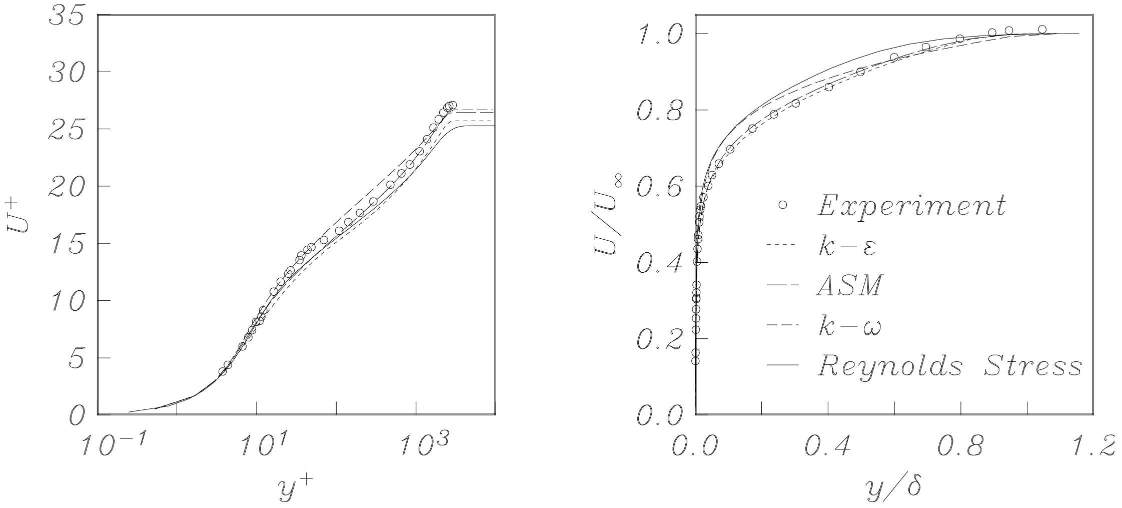

Figure 3.6: Velocity for incompressible zero pressure gradient flat plate.

3.2.5 Example Files

The example files for the incompressible flat plate are in directory

```
ISAAC/example_cases/Klebanoff
```

Sub-directory `Experimental_Data` contains the experimental data digitized from the report. The FORTRAN program to generate the grid used for the flat plate is included in sub-directory `Grid/flatplate.f`. The input files and output are included in the sub-directory `Results`.

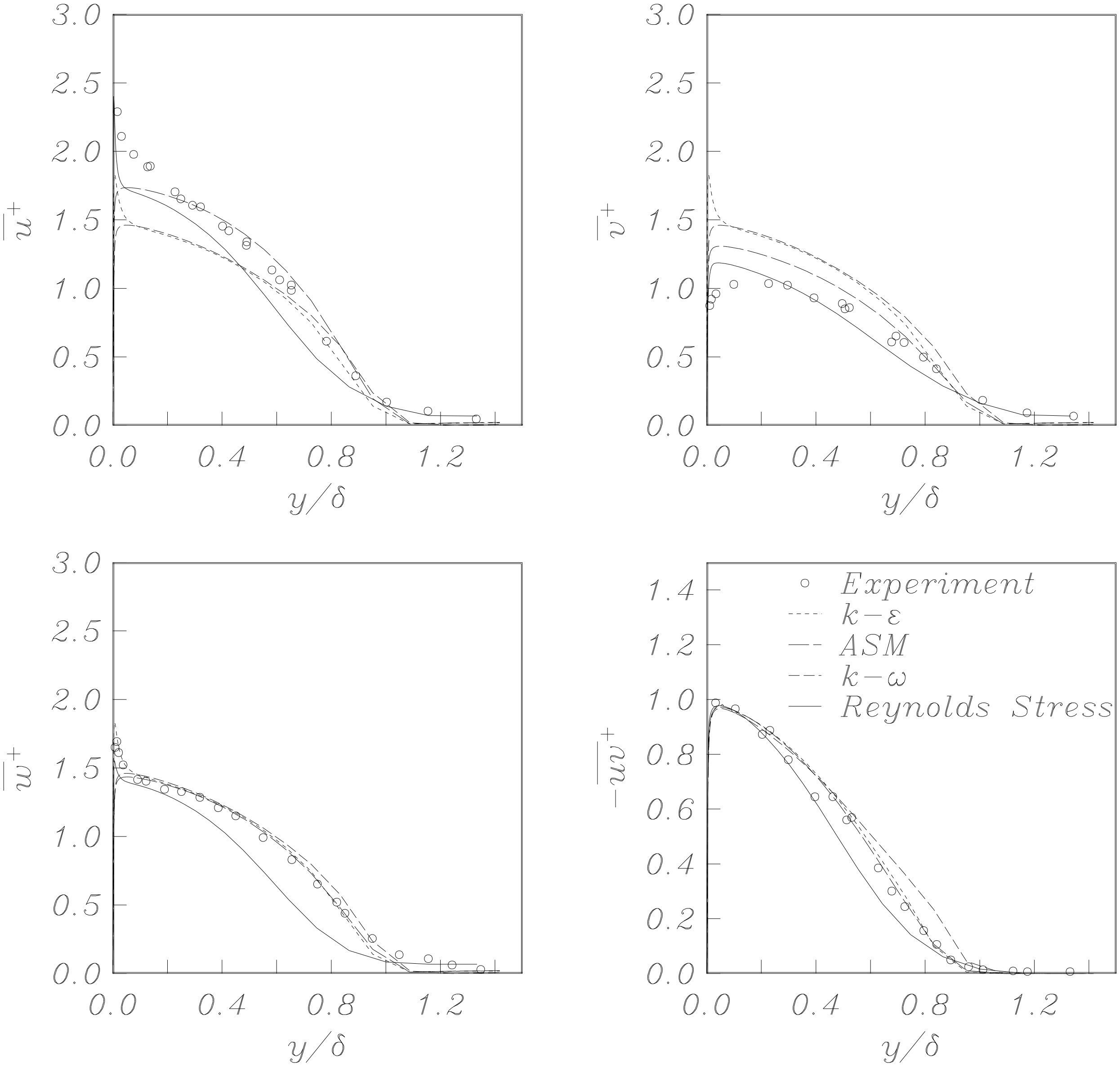

Figure 3.7: Turbulence intensities and Reynolds shear stress for incompressible zero pressure gradient flat plate.

```
'RESTART OUT' 1
     'asm.rst1'
'RE'         4.2e6
'PR'         0.7200000E+00
'SOLVER'     1
     'DIAGONAL AF3F'      1
'SIUNIT'     1
'TINF'      300.e0
'MACH'       0.2e0
'ALPHA'      0.0
'BETA'       0.0
'KAPPA'      0.3333333333E0
'LIMIT'      3
'MESH SEQUENCE'  2
     500
    1000
'NITS'       1000
'CFL'        1.00
'CFLFNL'     15.00
'ITDFNL'     50
'ITUPDT'     5
'TWOD'       1
'THIN LAYER' 1
'K EPSILON'  1
   'ASM'     0
'GRID'       1
          'flatplate.grd'
'BLOCK'      1
     'DIMENSIONS'       1
           65      65        2
     'FLUX'    1
          'ROE'        1
     'VISCOUS'    1
           2
     'BC'         7
          'BC'       1   1   1      65  65  1   'EXTRAPOLATE'
          'BC'       1   1   2      65  65  2   'EXTRAPOLATE'
          'BC'       1   1   1      17   1  2   'SYMMETRY XZ'
          'BC'      17   1   1      65   1  2   'WALL'
          'BC'       1  65   1      65  65  2   'EXTRAPOLATE'
          'BC'       1   1   1       1  65  2   'SUBSONIC INFLOW'
          'BC'      65   1   1      65  65  2   'SUBSONIC OUTFLOW'
     'END BLOCK'    1
'PRINT'    1
       1       2    49   1   2      49  65  2       1  1  1
'MONITOR'  0
      20000
'END'       0
```

Figure 3.8: Input file for algebraic stress model for Klebanoff flat plate.

3.3 Supersonic Flat Plate

3.3.1 Case Description

A supersonic, turbulent flat plate boundary layer was studied by Spina and Smits [20]. The zero pressure gradient boundary layer case is the inflow condition for a series of supersonic tests studying shock boundary layer interactions and compression corners. Results of calculations on the compression corner cases are reported in Morrison *et al.* [16].

3.3.2 Flow Parameters

The flow parameters from the calculation are given in Table 3.3.

M_∞	2.87
Re	$6.48 \times 10^7/m$
T_∞	$94.425K$
T_w	$260.34K$

Table 3.3: Flow parameters for supersonic flat plate.

3.3.3 Results

Figure 3.9 shows the velocity calculated with the k-ω model [11], and the Zhang-So-Gatski-Speziale [12] k-ε and Reynolds stress models. The grid used is a stretched 101 × 101 grid with the first grid point located at $y^+ \approx 0.26$ and approximately 77 grid points in the boundary layer. Figure 3.10 shows the temperature calculated with the models.

3.3.4 Example Input

Figure 3.11 shows the input for ISAAC to run the k-ω model for the supersonic flatplate with the marching scheme. The initial CFL is 0.5 and is ramped to 25 over 50 iterations for each solution plane. The time step is updated every 5 iterations (`'ITUPDT'` 5) and the implicit Jacobians are also updated every 5 iterations (`'ITUPJ'` 5). Each solution plane is converged more than 7 orders of magnitude in the residual for the mass conservation equation. The `PRINT` statement is used to generate output to plot the velocity profiles at the 85th plane.

3.3.5 Example Files

The grid file, solutions, and experimental data for the supersonic flatplate test case are available in directory

```
ISAAC/example_cases/Settles/Flat_Plate
```

The grid is generated with the FORTRAN code `Grid/rampgrd.f`. The results for this case are in sub-directory

```
ISAAC/example_cases/Settles/Flat_Plate/Results.
```

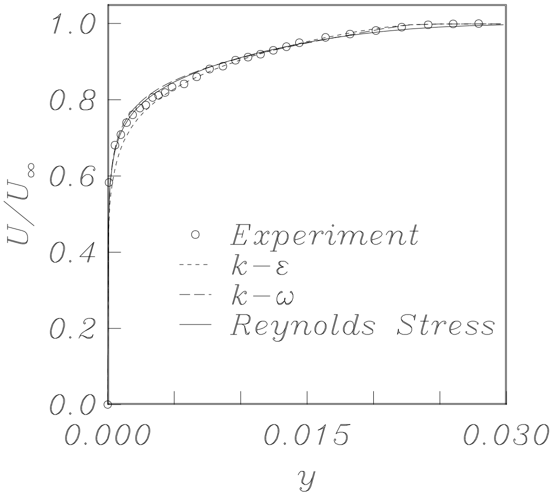

Figure 3.9: Velocity for supersonic flat plate.

3.4 NACA 4412 Airfoil

3.4.1 Case Description

The NACA 4412 airfoil was tested at several conditions by The angle of attack was 13.87° at a Mach number of 0.2. The grid used is a 257 × 81 C-mesh with 177 grid points on the surface of the airfoil. The grid extends approximately 17 chords from the airfoil and characteristic based boundary conditions are applied at the outer boundary.

3.4.2 Flow Parameters

The flow parameters from the calculation are given in Table 3.4.

M_∞	0.2
α	13.87°
Re	1.52×10^6

Table 3.4: Flow parameters for NACA 4412.

3.4.3 Results

Figure 3.12 shows the surface pressure coefficient for the airfoil. The shock is clearly evident on the upper surface at $x/c \approx 0.6$. Figure 3.13 shows the calculated velocity profiles at six streamwise locations along the airfoil surface for the four models. The k-ε models best

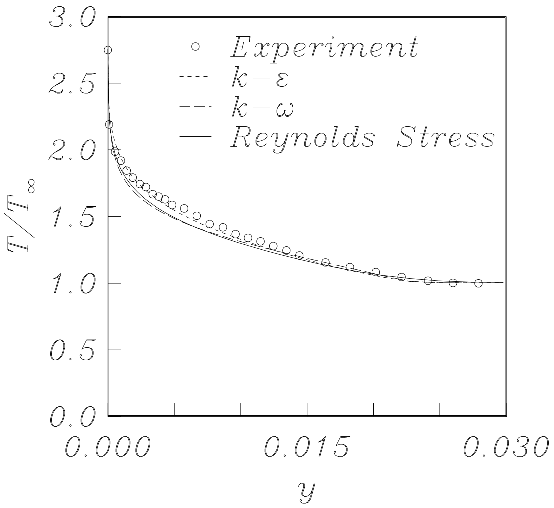

Figure 3.10: Temperature for supersonic flat plate.

```
'RESTART OUT' 1
     'kw.rst1'
'RE'         64.806890e06
'PR'         0.7200000E+00
'SOLVER'     1
     'MARCH'      1
         02    101
'VIGNERON'   1
'TOLER'      0.5E-07
'SIUNIT'     1
'TINF'       94.425e0
'TWALL'     260.340e0
'MACH'       2.87e0
'ALPHA'      0.0
'KAPPA'      0.3333333333E0
'LIMIT'      3
'NITS'       1500
'CFL'        0.5
'CFLFNL'    25.
'ITDFNL'     50
'ITUPDT'     5
'ITUPJ'      5
'TWOD'       1
'THIN LAYER' 1

'K OMEGA'    0

'BLOCK'      1
     'DIMENSIONS'       1
           101      101       2
     'FLUX'    1
          'ROE'        1
     'VISCOUS'    1
           2
     'GRID'       1
          'flat.grd'
     'BC'         4
          'BC'        1    1   1      101    1  2   'WALL'
          'BC'        1  101   1      101  101  2   'EXTRAPOLATE'
          'BC'        1    1   1        1  101  2   'FIX'
          'BC'      101    1   1      101  101  2   'EXTRAPOLATE'
     'END BLOCK'    1
'PRINT'    1
       1       2     85    1   2       85  101  2    20  1  1
'MONITOR'  0
      20000
'END'       0
```

Figure 3.11: Input file for k-ω model for supersonic flatplate.

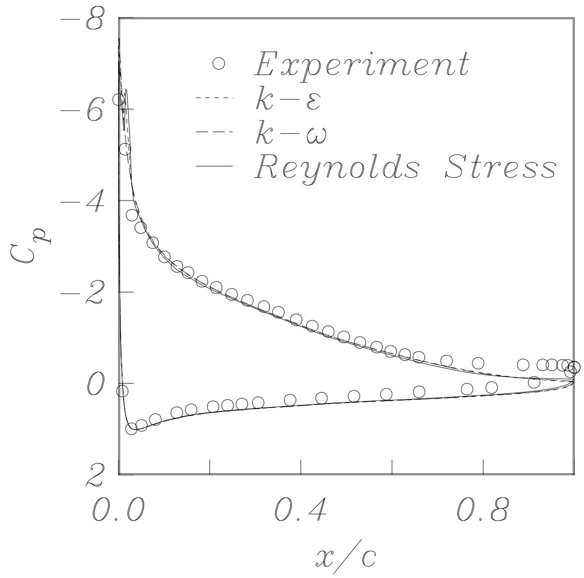

Figure 3.12: Surface pressure coefficient for NACA 4412 airfoil.

predict the velocity profile upstream of the shock while the Reynolds stress model best predicts the velocity inflection downstream of the shock. Figures 3.16, 3.14, and 3.15 show the calculated Reynolds stresses at seven streamwise locations. The k-ω model best predicts the wake profiles while the Reynolds stress model predicts too large of a velocity defect.

3.4.4 Example Input

Figure 3.17 shows the input for ISAAC to run the k-ω model for NACA 4412 airfoil. The calculation is run using mesh sequencing with 250 iterations computed on the 33×13 mesh, 250 iterations on the 65×25 mesh, 250 iterations on the 129×49 mesh, and a final 500 iterations on the fine 257×97 mesh. The initial CFL is 1 and is ramped to 5 over 100 iterations for each mesh level. The `PRINT` statement is used to generate output to plot the velocity profiles. The velocity plots are made after an additional 4000 iterations are run to ensure a converged solution.

3.4.5 Example Files

The grid file, solutions, and experimental data for the NACA 4412 test case is available in directory

```
ISAAC/example_cases/NACA4412
```

The formatted grid file is available in the sub-directory `Grid` as `wig4412257.grd`. A routine to convert this to the appropriate PLOT3D format for ISAAC is `cnvt.f`. The results for this case are in sub-directory

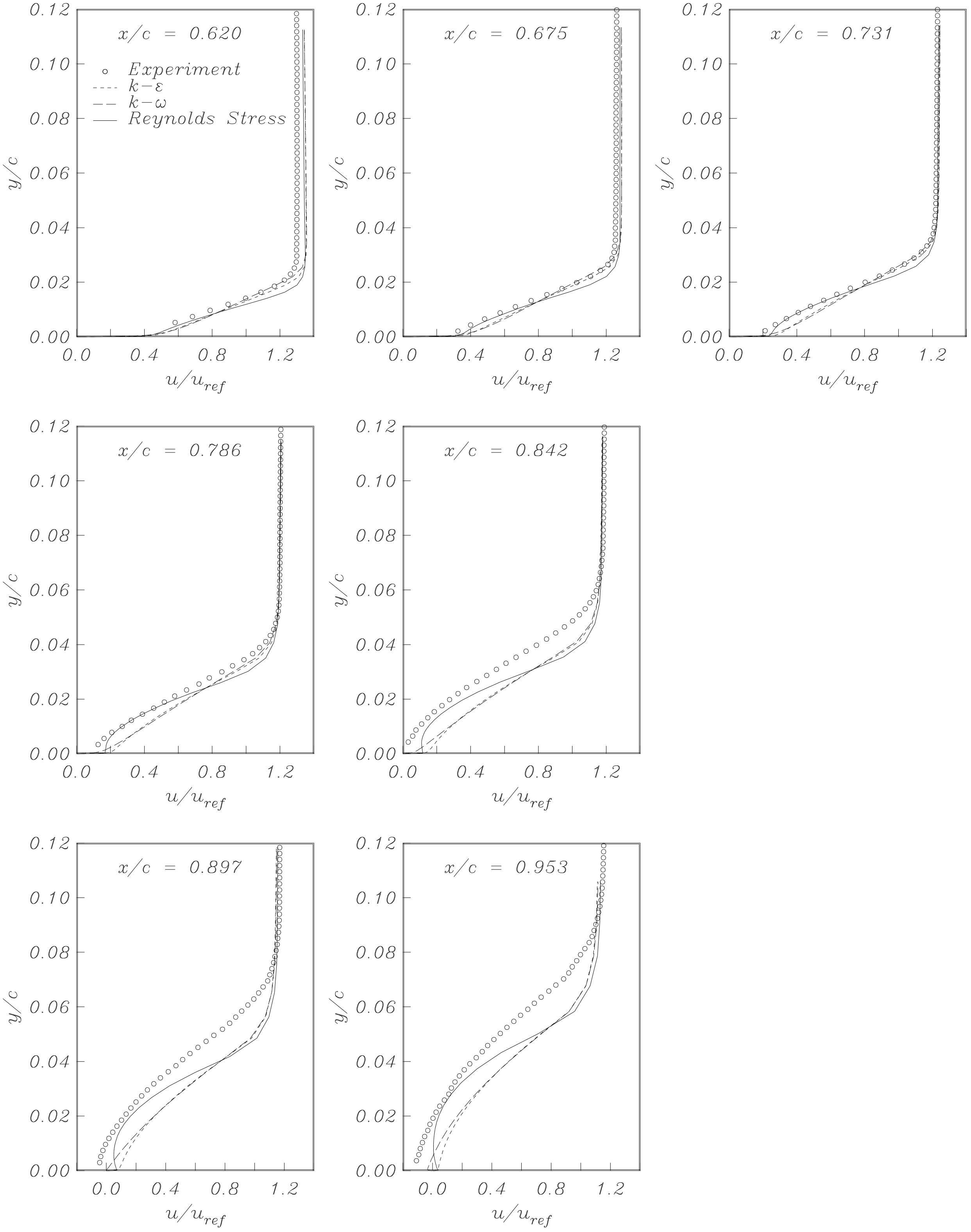

Figure 3.13: Velocity for NACA 4412 airfoil.

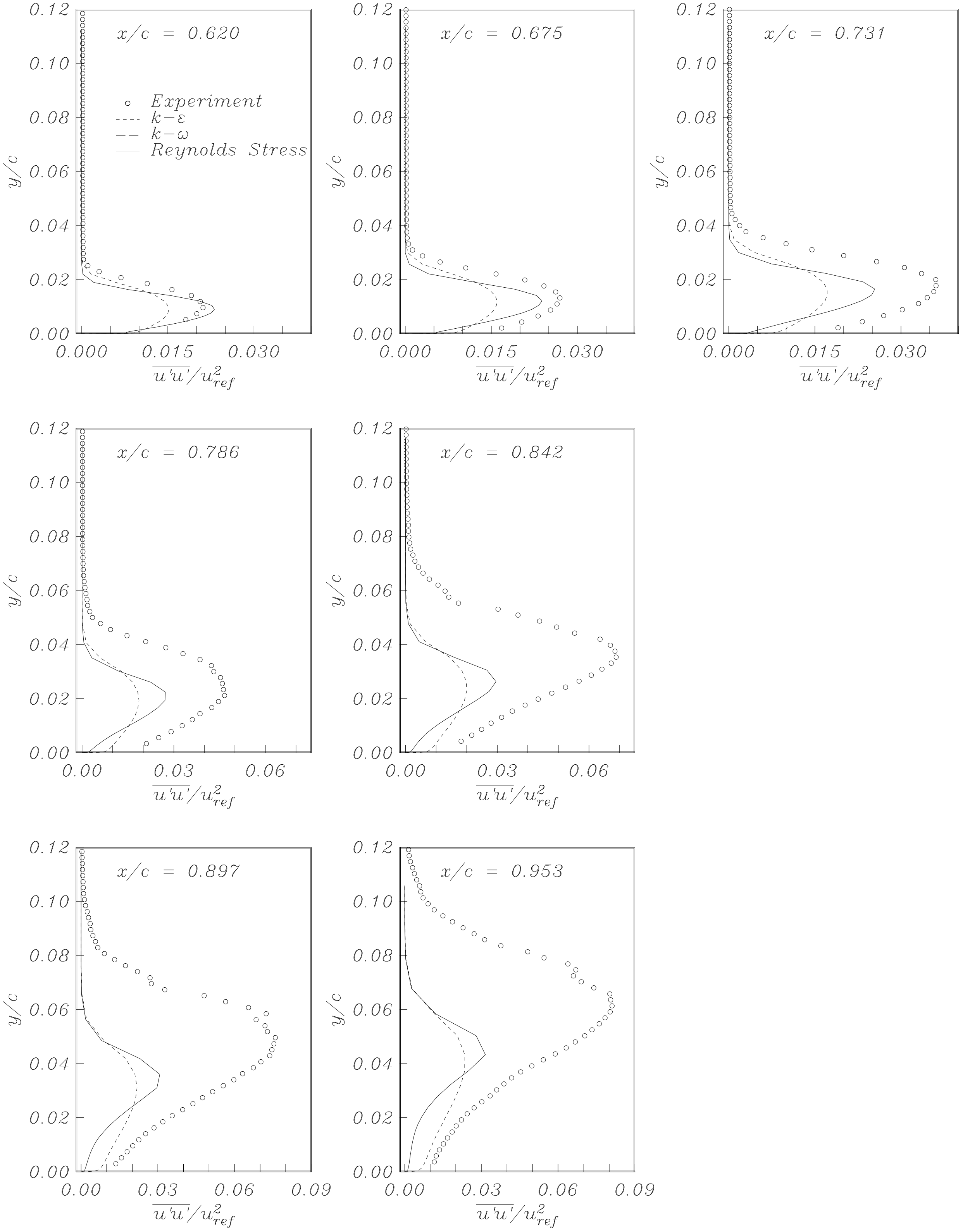

Figure 3.14: Measured and predicted normal stress for NACA 4412 airfoil.

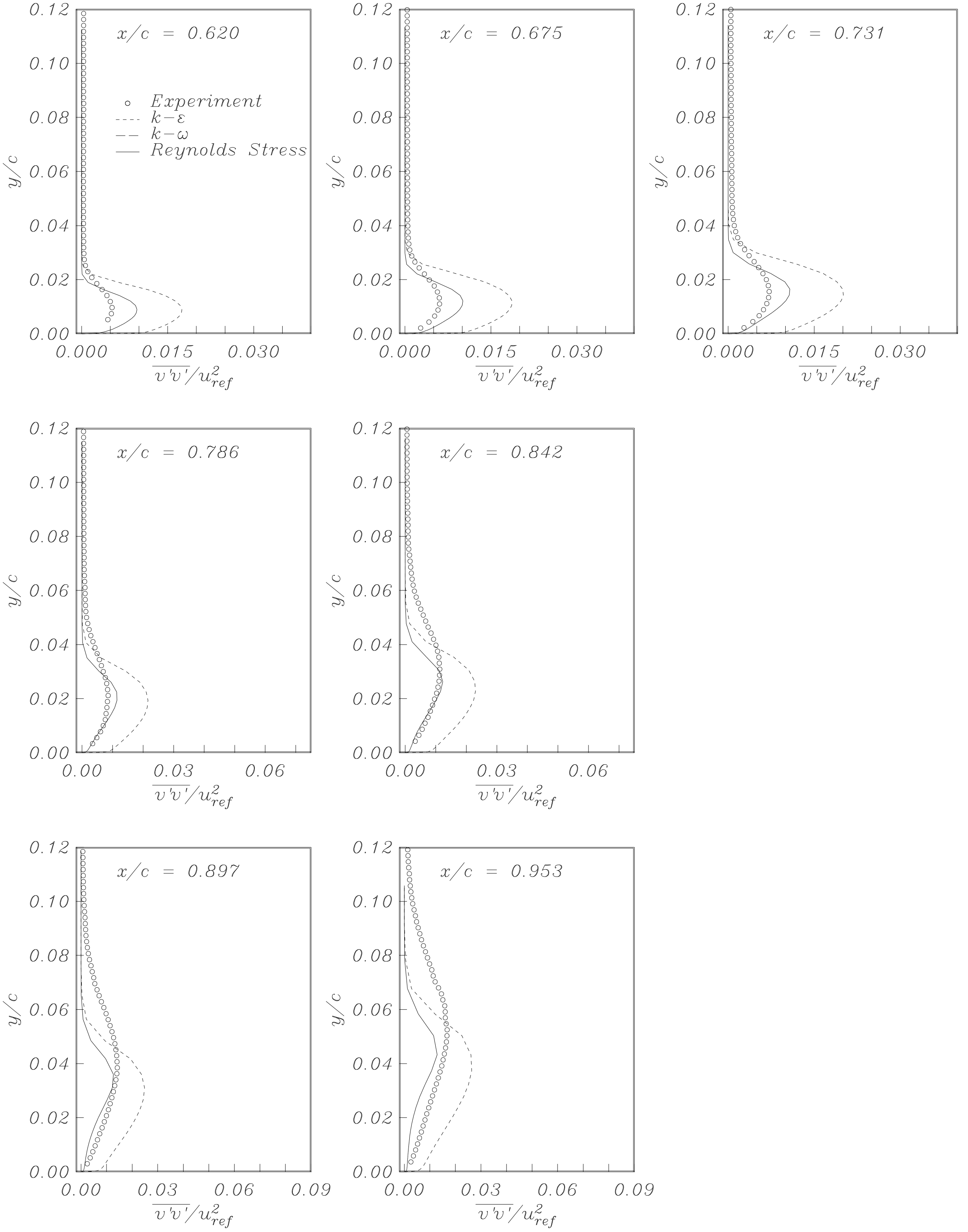

Figure 3.15: Measured and predicted normal stress for NACA 4412 airfoil.

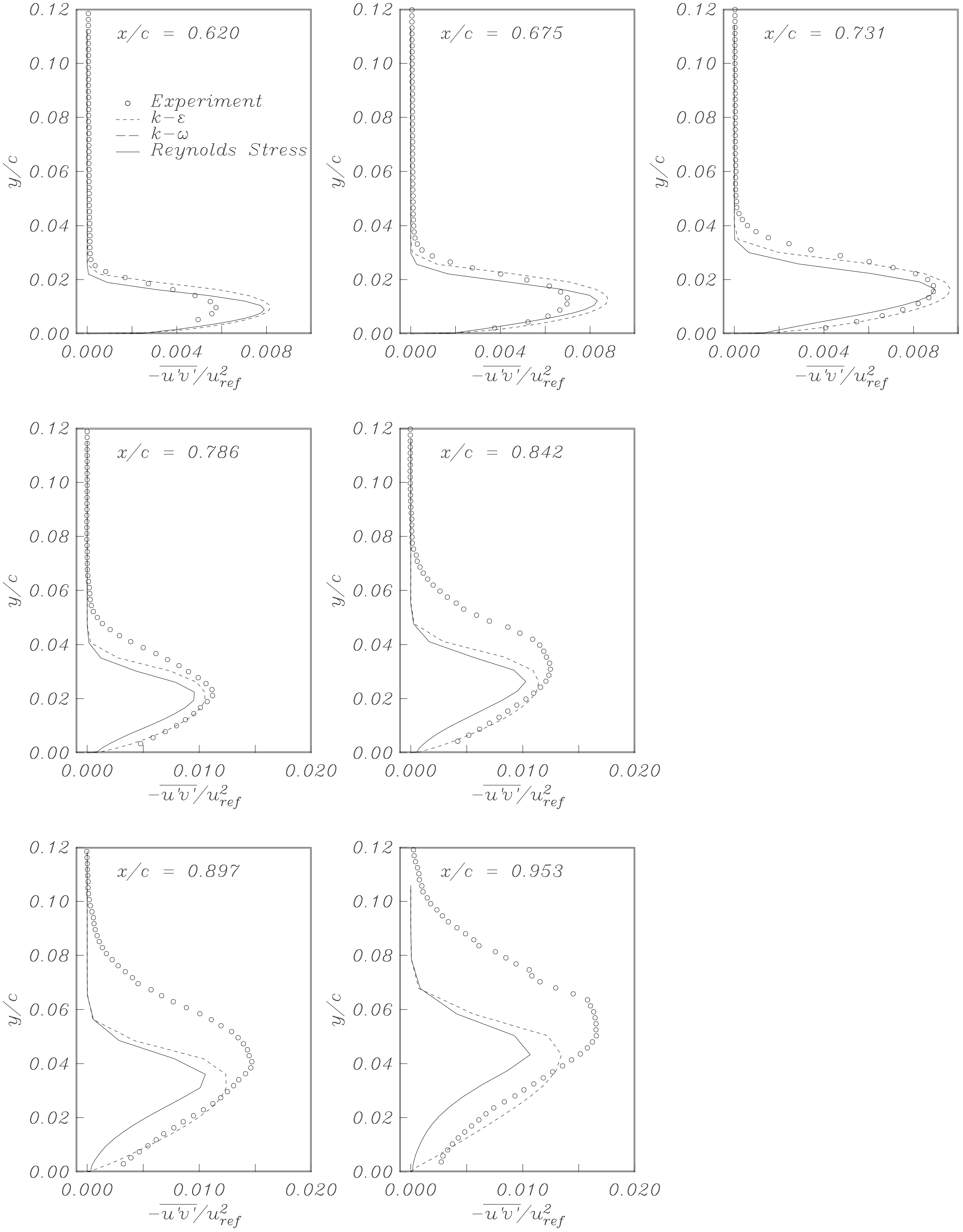

Figure 3.16: Measured and predicted shear stress for NACA 4412 airfoil.

```
'RESTART OUT' 1
     'kw.rst1'
'RE'         1.52e06
'PR'         0.7200000E+00
'SOLVER'     1
     'DIAGONAL AF3F'      1
'SIUNIT'     1
'TINF'      300.e0
'MACH'       0.20e0
'ALPHA'      13.87
'BETA'       0.0
'KAPPA'      0.0
'LIMIT'      3
'NITS'       500
'CFL'        5.00
'CFLFNL'     5.00
'ITDFNL'     100
'ITUPDT'     5
'ITUPJ'      5
'TWOD'       1
'THIN LAYER' 1
'MESH SEQUENCE'   2
     100
     200
'MULTIGRID LEVELS'  3
'K OMEGA'    2
     'MINIMUM TKE'   1.e-24
     'PRDLIM'        20.
'BLOCK'      1
     'DIMENSIONS'       1
           257      81       2
     'FLUX'    1
          'ROE'        1
     'VISCOUS'    1
           2
     'GRID'       1
          'naca4412_257.grd'
     'BC'         6
          'BC'      41   1   1     217   1  2   'WALL'
          'BC'       1  81   1     257  81  2   'FARFIELD'
          'BC'       1   1   1       1  81  2   'EXTRAPOLATE'
          'BC'     257   1   1     257  81  2   'EXTRAPOLATE'
          'BC'       1   1   1     257  81  1   'EXTRAPOLATE'
          'BC'       1   1   2     257  81  2   'EXTRAPOLATE'
     'END BLOCK'    1
'CUT'      1
     'ONE'     1     1   1   1      41   1  2
     'ONE'     1   257   1   1     217   1  2       1  2  3
'PRINT'    3
       1       2   188   1   2     196  81  2       2  1  1
       1       2   199   1   2     199  81  2       1  1  1
       1       2   204   1   2     204  81  2       1  1  1
'MONITOR'  0
      20000
'END'       0
```

Figure 3.17: Input file for k-ω model for NACA 4412 airfoil.

ISAAC/example_cases/NACA4412/Results.

3.5 RAE 2822 Airfoil

3.5.1 Case Description

The RAE 2822 airfoil was tested at several conditions by Cooke, McDonald and Firmin [21]. The test case presented here corresponds to Case 9. The airfoil is 12.1 percent thick. The measured angle of attack was 3.19° with a tunnel corrected value of 2.8°. The grid used is a 257 × 97 C-mesh with 177 grid points on the surface of the airfoil. The grid extends approximately 18 chords from the airfoil and characteristic based boundary conditions are applied there.

3.5.2 Flow Parameters

The flow parameters from the calculation are given in Table 3.5.

M_∞	0.73
α	2.8°
Re	6.5×10^6

Table 3.5: Flow parameters for RAE2822 Case 9.

3.5.3 Results

Figure 3.18 shows the 257 × 97 C-grid around the airfoil. Figure 3.19 shows the surface pressure coefficient for the airfoil. The shock is clearly evident on the upper surface at $x/c \approx 0.6$. Figure 3.20 shows the skin friction calculated with the four models. The k-ε model best predicts the skin friction up to the shock and the k-ω model best predicts the post-shock skin friction. All of the models relax too quickly after the shock. The Reynolds stress model predicts too high skin friction throughout. Figure 3.21 shows the calculated velocity profiles at six streamwise locations along the airfoil surface for the four models. The k-ε models best predict the velocity profile upstream of the shock while the Reynolds stress model best predicts the velocity inflection downstream of the shock. Figure 3.22 shows the calculated velocity profiles at three streamwise locations in the airfoil wake for the four models. The k-ω model best predicts the wake profiles while the Reynolds stress model predicts too large of a velocity defect.

3.5.4 Example Input

Figure 3.23 shows the input for ISAAC to run the k-ω model for RAE2822 airfoil. The calculation is run using mesh sequencing with 250 iterations computed on the 33 × 13 mesh, 250 iterations on the 65 × 25 mesh, 250 iterations on the 129 × 49 mesh, and a final 500 iterations on the fine 257 × 97 mesh. The initial CFL is 1 and is ramped to 5 over 100 iterations for each mesh level. The `PRINT` statement is used to generate output to plot the velocity profiles. The velocity plots are made after an additional 4000 iterations are run to ensure a converged solution.

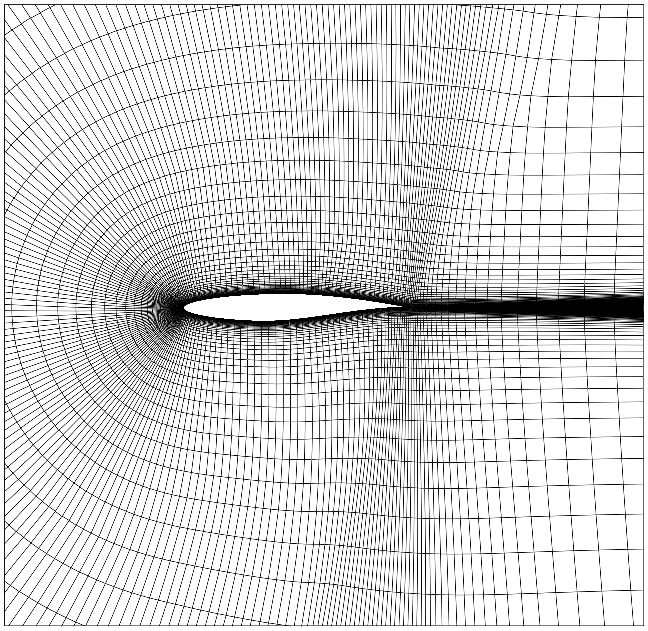

Figure 3.18: 257 × 97 C-grid for the RAE2822 airfoil.

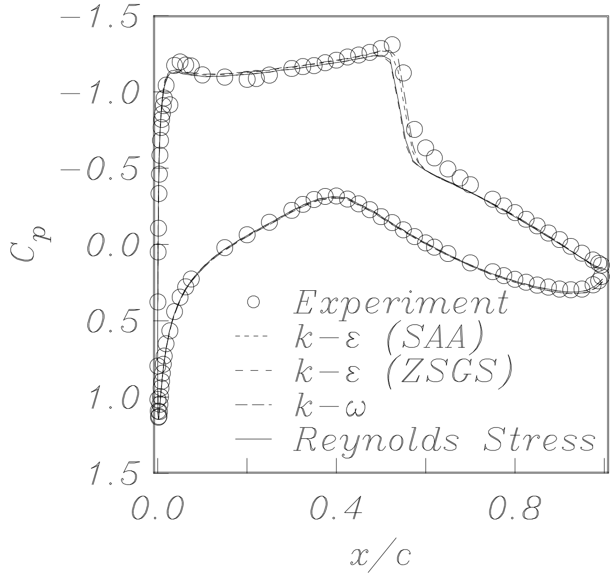

Figure 3.19: Surface pressure coefficient for RAE2822 airfoil.

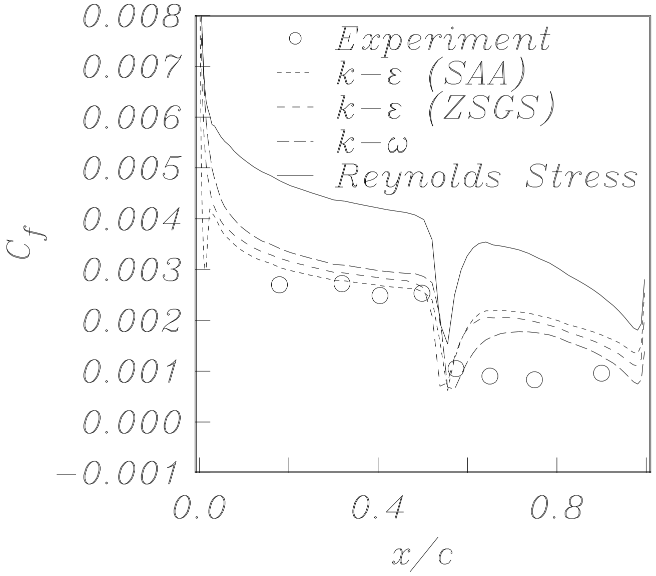

Figure 3.20: Skin friction coefficient for RAE2822 airfoil.

3.5.5 Example Files

The grid file, solutions, and experimental data for the RAE2822 test case is available in directory

```
ISAAC/example_cases/RAE2822
```

The grid file is available in the sub-directory `Grid` in formatted PLOT3D form as `rae_257x97.fmt` in binary PLOT3D form for a Cray computer as `rae9_257.grd`. The results for this case are in sub-directory

```
ISAAC/example_cases/RAE2822/Case_9/Results.
```

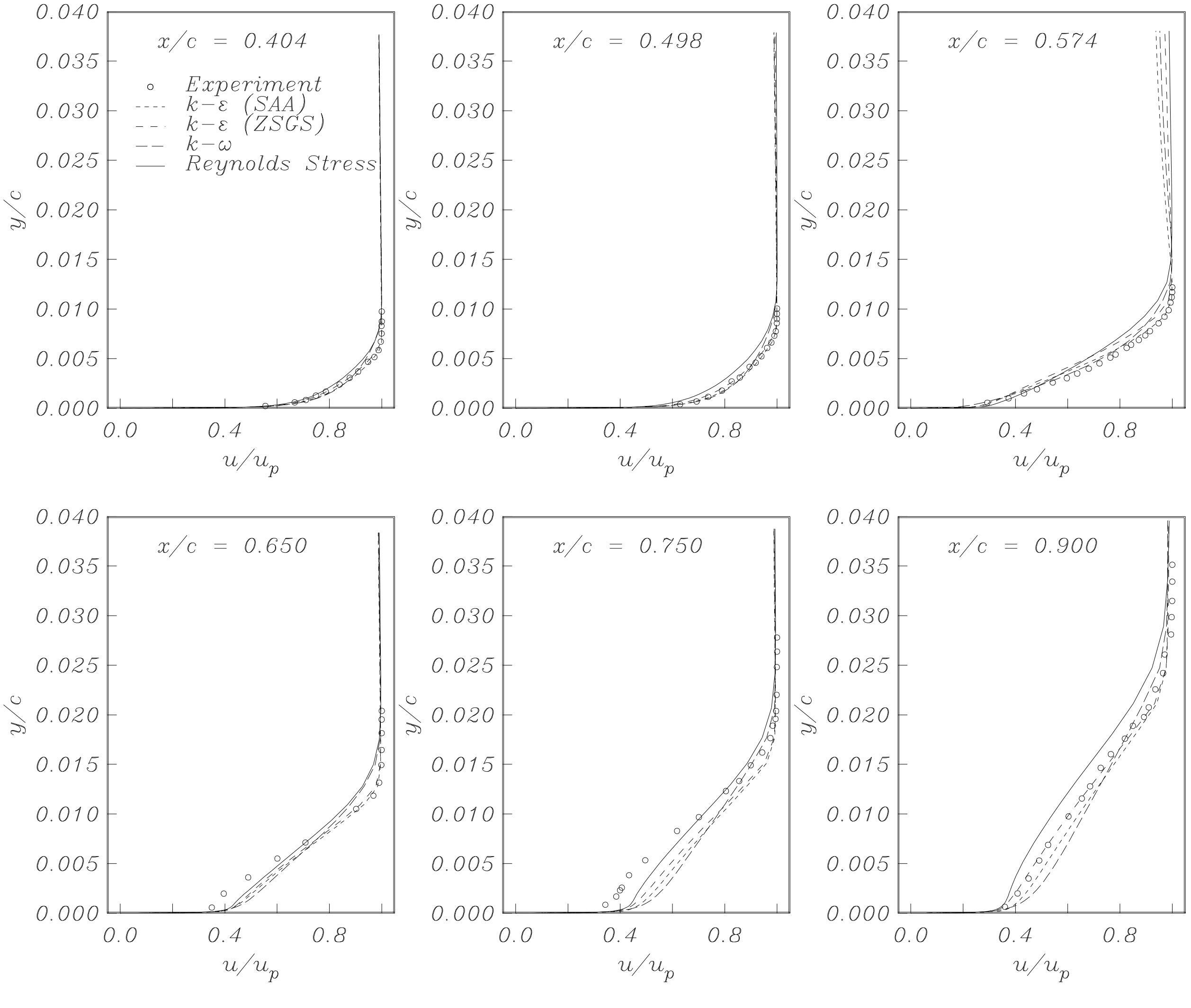

Figure 3.21: Velocity for RAE2822 airfoil.

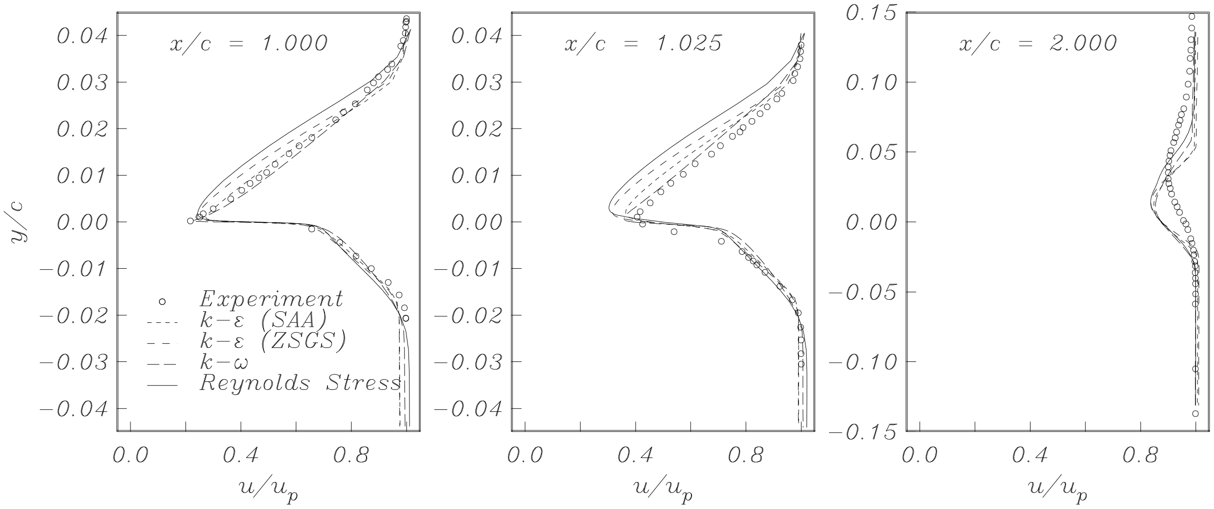

Figure 3.22: Wake velocity for RAE2822 airfoil.

```
'RESTART OUT' 1
    'kw.rst1'
'GRID FORMATTED'  0
'GRID'       1
    '../../Grid/rae_257x97.fmt'
'RE'         6.5e06
'PR'         0.7200000E+00
'SOLVER'     1
     'DIAGONAL AF3F'      1
'ENGLISH'     1
'TINF'      460.e0
'MACH'       0.73e0
'ALPHA'      2.8
'BETA'       0.0
'KAPPA'      0.0
'LIMIT'      3
'MESH SEQUENCE'   3
    250
    250
    250
'NITS'       500
'CFL'        1.0
'CFLFNL'     5.0
'ITDFNL'     100
'ITUPDT'     5
'TWOD'       1
'THIN LAYER' 1
'K OMEGA'    0
'BLOCK'      1
     'DIMENSIONS'       1
           257      97       2
     'FLUX'    1
          'ROE'        1
     'VISCOUS'    1
           2
     'BC'         6
          'BC'      41   1   1     217   1  2   'WALL'
          'BC'       1  97   1     257  97  2   'FARFIELD'
          'BC'       1   1   1       1  97  2   'EXTRAPOLATE'
          'BC'     257   1   1     257  97  2   'EXTRAPOLATE'
          'BC'       1   1   1     257  97  1   'EXTRAPOLATE'
          'BC'       1   1   2     257  97  2   'EXTRAPOLATE'
     'END BLOCK'    1
'CUT'      1
     'ONE'     1     1   1   1      41   1  2
     'ONE'     1   257   1   1     217   1  2       1  2  3
'MONITOR'  0
      20000
'PRINT'    2
       1       2   178   1   2     178  98  2      1  1  1
       1       2   181   1   2     181  98  2      1  1  1
'END'       0
```

Figure 3.23: Input file for k-ω model for RAE2822 airfoil.

3.6 Multi-Element Airfoil

3.6.1 Case Description

The multi-element airfoil test case is the NHLP-2D airfoil reported as AGARD Case A-2 [22]. This three element airfoil has been studied by Fejtek [23]. The grid (provided by Dr. T. Nelson of de Havilland, Inc.) consists of four C-grids; one C-grid around the flap, a second around the main element, a third around the slat, and the fourth around the overall configuration. The flap grid has a mesh of 245 × 49, the main element grid has a mesh of 549 × 89, the slat grid has a mesh of 577 × 57, and the outer grid has a mesh of 641 × 33. Figure 3.24 shows every other grid line around the airfoil.

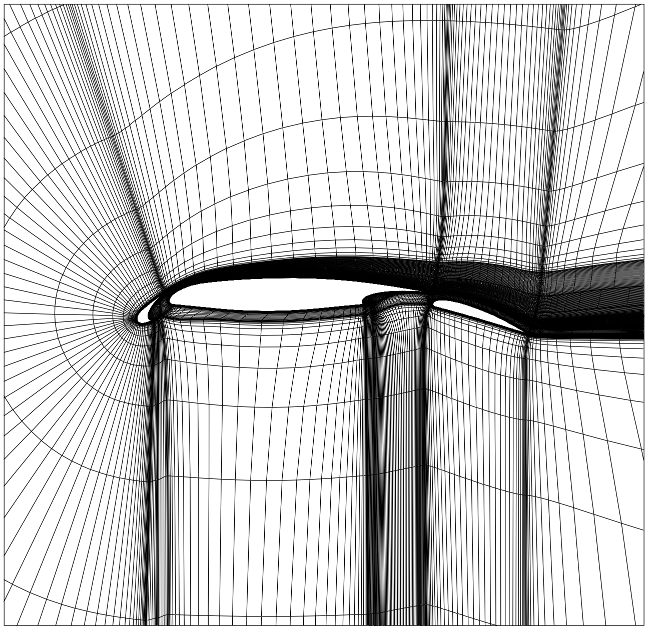

Figure 3.24: Grid for the NHLP-2D three-element airfoil.

Transition is specified at $x/c = 0.125$ on the upper and lower surfaces of the main element and is left free on both surfaces of the slat and flap.

3.6.2 Flow Parameters

The flow parameters from the calculation are given in Table 3.6.

M_∞	0.195
Re	3.52×10^6
α	$4.01°$

Table 3.6: Flow parameters for multi-element airfoil.

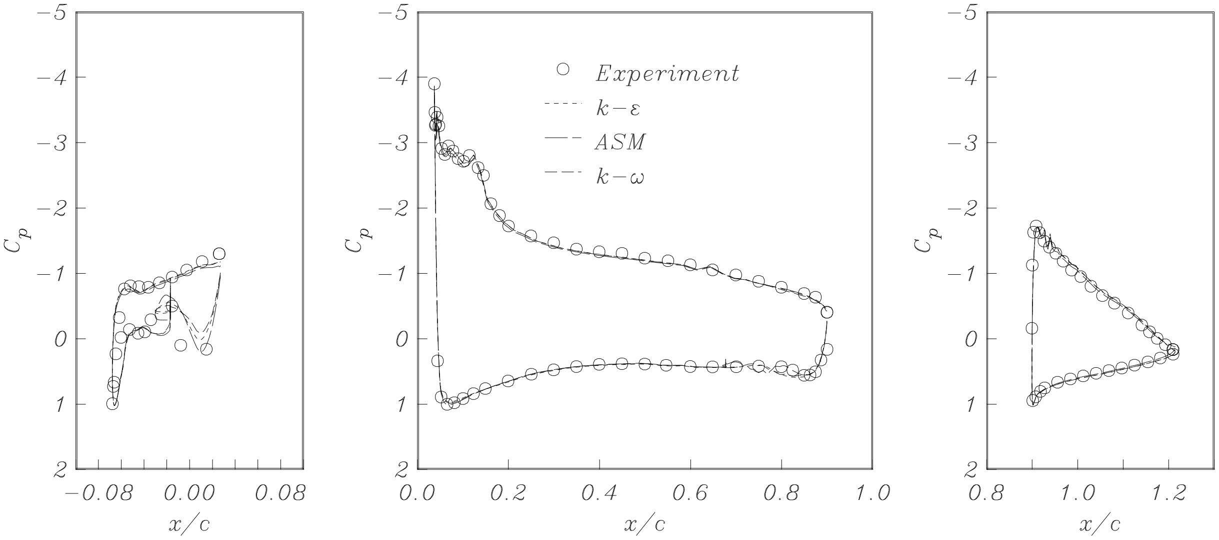

Figure 3.25: Wall pressure coefficient for NHLP-2D mulit-element airfoil.

3.6.3 Results

Figure 3.25 shows the pressure calculated with the three models. The k-ε model, the k-ω model, and the algebraic stress model all predict very similar wall pressures. Figure 3.26 shows the total pressure coefficient profiles calculated with the three models. The k-ε model, the k-ω model, and the algebraic stress model all predict very similar wall pressures.

3.6.4 Example Input

Figure 3.27 shows the input for ISAAC to run the k-ω model for AGARD A2 multi-element airfoil. The calculation is run using mesh sequencing with 200 iterations computed on the twice coarsened mesh, 200 iterations on the once coarsened mesh, and 1000 iterations on the fine mesh. The CFL is 3 for each mesh level.

Transition is specified on the main element on block 2 using the `'ENFORCE LAMINAR FLOW'` command. Laminar flow is specified for the region with $299 < i < 387$ for all j.

3.6.5 Example Files

The grid file, solutions, and experimental data for the multielement airfoil is available in the sub-directory

```
ISAAC/example_cases/NHLP_2D
```

The grid file is available in the sub-directory `Grid` in formatted PLOT3D form as `new.g`. The results for this case are in sub-directory

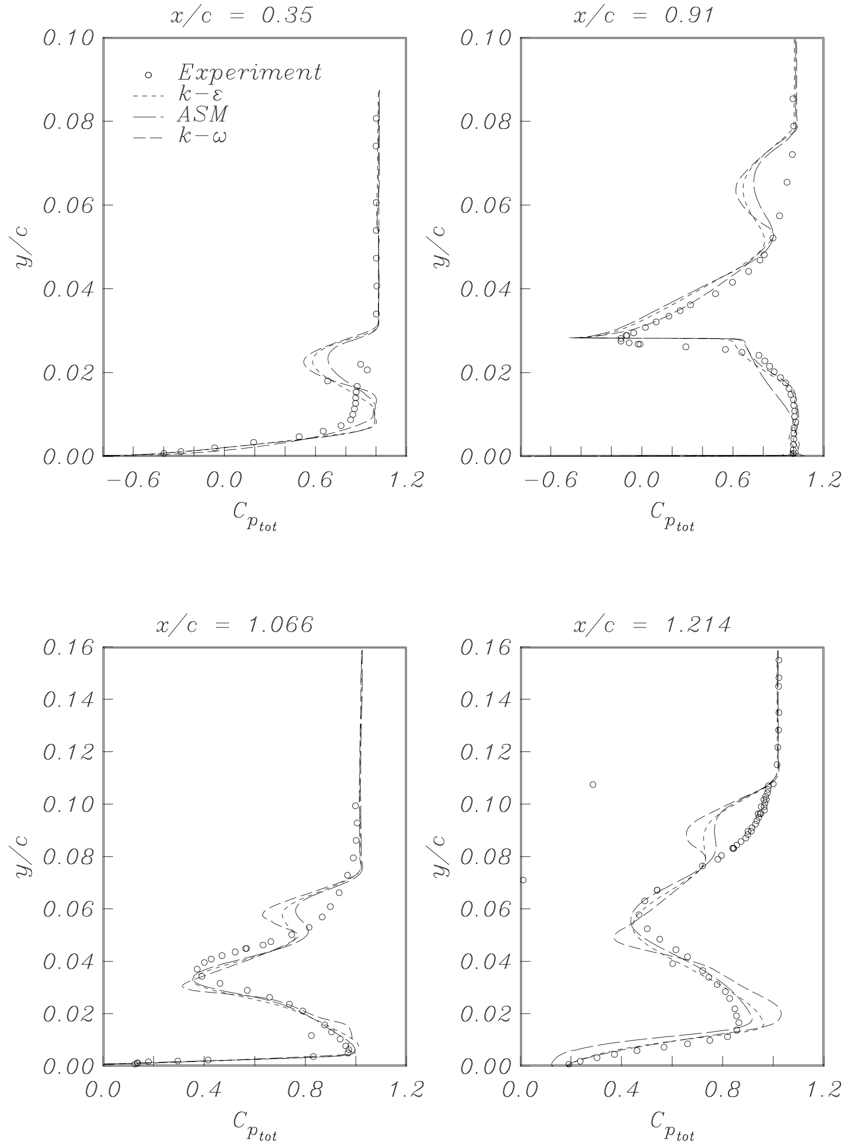

Figure 3.26: Total pressure coefficient profiles for NHLP-2D mulit-element airfoil.

```
'RESTART OUT' 1
    'kw.rst1'
'GRID FORMATTED'  0
'GRID'       1
    'new.g'
'RE'         3.52e06
'PR'         0.7200000E+00
'SOLVER'     1
     'DIAGONAL AF3F'      1
'ENGLISH'     1
'TINF'      460.e0
'MACH'       0.195e0
'ALPHA'      4.01
'BETA'       0.0
'KAPPA'      0.3333333333333e0
'LIMIT'      3

'MESH SEQUENCE'      2
    200
    200

'MULTIGRID LEVELS'   3

'NITS'       1000
'CFL'        3.0
'CFLFNL'     3.0
'ITDFNL'     100
'ITUPDT'     5
'TWOD'       1
'THIN LAYER' 1

'ENFORCE LAMINAR FLOW'    1
     2    299   1   1     387   89   2

'K OMEGA'   4
     'MINIMUM TKE'        1.e-24
     'MINIMUM OMEGA'      1.e-24
     'FULL OMEGA PROD'    0.
     'PRDLIM'             20.
```

Figure 3.27: Input file for k-ω model for AGARD Case A2 Multi-element airfoil.

```
'BLOCK'      1
     'DIMENSIONS'       1
           245      49       2
     'FLUX'    1
          'ROE'         1
     'VISCOUS'    2
           1    2
     'BC'         3
          'BC'      49   1   1     197   1  2   'WALL'
          'BC'       1   1   1       1  49  2   'EXTRAPOLATE'
          'BC'     245   1   1     245  49  2   'EXTRAPOLATE'
     'END BLOCK'    1

'BLOCK'      2
     'DIMENSIONS'       1
           549      89       2
     'FLUX'    1
          'ROE'         1
     'VISCOUS'    2
           1    2
     'BC'         3
          'BC'     113   1   1     437   1  2   'WALL'
          'BC'       1   1   1       1  89  2   'EXTRAPOLATE'
          'BC'     549   1   1     549  89  2   'EXTRAPOLATE'
     'END BLOCK'    2

'BLOCK'      3
     'DIMENSIONS'       1
           577      57       2
     'FLUX'    1
          'ROE'         1
     'VISCOUS'    2
           1    2
     'BC'         3
          'BC'     185   1   1     393   1  2   'WALL'
          'BC'       1   1   1       1  57  2   'EXTRAPOLATE'
          'BC'     577   1   1     577  57  2   'EXTRAPOLATE'
     'END BLOCK'    3

'BLOCK'      4
     'DIMENSIONS'       1
           641      33       2
     'FLUX'    1
          'ROE'         1
     'VISCOUS'    2
           1    2
     'BC'         3
          'BC'       1  33   1     641  33  2   '2D FARFIELD VORTEX'
          'BC'       1   1   1       1  33  2   'EXTRAPOLATE'
          'BC'     641   1   1     641  33  2   'EXTRAPOLATE'
     'END BLOCK'    4
```

Figure 3.27: Continued. Input file for k-ω model for AGARD Case A2 Multi-element airfoil.

```
'CUT'      8
     'B1WAKE'     1     1     1   1      49    1  2
     'B1WAKE'     1   245     1   1     197    1  2       1  2  3

     'B1TOB2'     1   113    49   1     245   49  2
     'B1TOB2'     2   133    89   1       1   89  2       1  2  3

     'B1TOB4'     1     1    49   1     113   49  2
     'B1TOB4'     4     1     1   1     113    1  2       1  2  3

     'B2WAKE'     2     1     1   1     113    1  2
     'B2WAKE'     2   549     1   1     437    1  2       1  2  3

     'B2TOB3'     2   317    89   1     549   89  2
     'B2TOB3'     3   233    57   1       1   57  2       1  2  3

     'B2TOB4'     2   133    89   1     317   89  2
     'B2TOB4'     4   113     1   1     297    1  2       1  2  3

     'B3WAKE'     3     1     1   1     185    1  2
     'B3WAKE'     3   577     1   1     393    1  2       1  2  3

     'B3TOB4'     3   233    57   1     577   57  2
     'B3TOB4'     4   297     1   1     641    1  2       1  2  3

'MONITOR'  0
      20000

'PRINT'    0

'END'       0
```

Figure 3.27: Concluded. Input file for k-ω model for AGARD Case A2 Multi-element airfoil.

ISAAC/example_cases/NHLP_2D/Results.

Chapter 4

Input Description

The input to ISAAC is an ASCII file containing input lines to reset default values of the flow parameters and file names. The input is designed so that only parameters that need to be modified from the default value needs to be input. The input is designed so that most of the input lines can be input in an arbitrary order. The general form of a line of input is

```
’COMMAND’ value
```

Note that the single quotes are required as the input routine uses list directed i/o, and the upper case of `COMMAND` is required as no case checking is implemented.

The *value* takes on appropriate meaning for the `COMMAND`. For example, for commands such as

```
’MACH’    3.0
```

value would be the Mach number (3.0) of the case. While for the command

```
’SOURCE’    1
```

value (1) would mean ‘true’ or ‘on’ where (0) would have meant ‘false’ or ‘off’ to control the evaluation of the source terms in the turbulence model. For other commands such as

```
’PRINT’    9
```

value (9) would mean to read in 9 cards following this card which would describe which data to print to the output file. The system is not completely consistent, but is generally self-evident. Any questions about what input commands to use and how to set them and what the defaults are will be arbitrated by the routine RDDATA.

4.1 General Inputs

```
’END’ value
```

Terminates the input. Lines after the `END` are ignored. This is a required input.

`'AXISYMMETRIC'` *value*
`'TWOD'` *value*

Sets the case to axisymmetric (`AXISYMMETRIC`), two-dimensional (`TWOD`), or three-dimensional calculation (neither specified). Either the axisymmetric or two-dimensional option reduces the computational cost by avoiding unnecessary flux evaluations in the third dimension, and reads in a planar grid. Either the axisymmetric or two-dimensional option requires that the input value of KDIM = 2 and that the K = 1 and K = 2 boundary conditions are set to the new `'AXISYMMETRIC'` boundary condition type for axisymmetric flow or to `'EXTRAPOLATE'` for two-dimensional flow. Additionally, the boundary condition `'AXISYMMETRIC CENTERLINE'` is used for the singular line at $y = 0$ for axisymmetric cases run with a singular line. The default obtained by omitting this input is a three-dimensional calculation.

`'RECALCULATE YNORMAL'` *value*

Specifies that the y_n array is to be recalculated if *value* = 1. Any other value of *value* does not recalculate y_n. The value of y_n is initially calculated when a case is first initialized. Since this calculation is expensive (it currently searches over the entire flow domain for every wall point) the value of y_n is stored in the restart file and this calculation is avoided on restart runs. However, there are certain cases where it is necessary to be able to recompute this array, e.g. switching boundary conditions and turning on/off a wall or wall function boundary condition that was not initially set or modifying the grid. The default is to calculate y_n only when the flowfield is first initialized.

`'THIN LAYER'` *value*
`'NAVIER STOKES'` *value*

Use thin-layer or full Navier-Stokes approximation for viscous terms. The default (obtained if neither option is specified) is to use the full Navier-Stokes evaluation.

4.2 Flow Parameters

`'SIUNIT'` *value*
`'ENGLISH'` *value*

Sets input (`TINF`, `TWALL`, ...) to SI units or English units. While *value* is required in the input, it is not used. The default is English units.

`'TINF'` *value*

Set the freestream temperature to *value*. `TINF` is input in Kelvin if `SIUNIT` is set or Rankine if `ENGLISH` is set. The default freestream temperature is $450°R$.

`'GAMMA'` *value*

Set the ratio of specific heats to *value*. The default is $\gamma = 1.4$.

'RE' *value*

Set the Reynolds number to *value*. The default Reynolds number is 1×10^6.

'PR' *value*

Set the Prandtl number to *value*. The default Prandtl number is 0.7.

'MACH' *value*

Set the freestream Mach number to *value*. The default Mach number is 0.8.

'ALPHA' *value*

Set angle of attack (in degrees). For the TWOD option the angle of attack is defined as a rotation about the z-axis. For three-dimensional calculations the angle of attack is defined as a rotation about the y-axis. The default angle of attack is 0°.

'BETA' *value*

Set sideslip angle - *Not currently implemented.*

4.3 Boundary Condition Parameters

'TWALL' *value*

Set fixed wall temperature to *value*. Omitting the TWALL input specifies adiabatic wall conditions. TWALL is input in Kelvin if SIUNIT is set or Rankine if ENGLISH is set. The default is adiabatic wall conditions. Used with the WALL boundary condition.

'INFLOW PTOTAL' *value*

Set the total pressure for the SUBSONIC INFLOW boundary condition as $p/p_\infty = value$. *Used only with the* SUBSONIC INFLOW *boundary condition.*

'INFLOW TTOTAL' *value*

Set the total temperature for the SUBSONIC INFLOW boundary condition as $T_t = value$. The total temperature is set to Kelvins if SIUNIT is specified or to degrees Rankine if ENGLISH units are specified. *Used only with the* SUBSONIC INFLOW *boundary condition.*

'BACK PRESSURE' *value*

Set the back pressure for the SUBSONIC OUTFLOW boundary condition as $p_{back}/p_\infty = value$. The default is that the back pressure is equal to the reference pressure. *Used only with the* SUBSONIC OUTFLOW *boundary condition.*

'JET CONDITIONS' *value*
'END JET' *value*

Read in the jet conditions to be used with the JET boundary conditions. The jet conditions are contained between these two inputs and are

'U' *value*

Set the jet velocity to $u/a_\infty = value$. The default is 0.

'V' *value*

Set the jet velocity to $v/a_\infty = value$. The default is 0.

'W' *value*

Set the jet velocity to $w/a_\infty = value$. The default is 0.

'P' *value*

Set the jet pressure $p_{\text{jet}}/p_\infty = value$. The default is $p_{\text{jet}}/p_\infty = 1$.

'T' *value*

Set the jet temperature $T_{\text{jet}}/T_\infty = value$. The default is $T_{\text{jet}}/T_\infty = 1$.

'K' *value*

Set the jet turbulent kinetic energy $k_{\text{jet}}/a_\infty^2 = value$. The default is $k_{\text{jet}}/a_\infty^2 = 1 \times 10^{-4}$.

'EPSILON' *value*

Set the jet turbulent dissipation rate $\varepsilon_{\text{jet}}/(a_\infty^3/L) = value$. The default is $\varepsilon_{\text{jet}}/(a_\infty^3/L) = 1 \times 10^{-4}$.

'OMEGA' *value*

Set the jet specific dissipation rate $\omega_{\text{jet}}/(a_\infty/L) = value$. The default is $\omega_{\text{jet}}/(a_\infty/L) = 1 \times 10^{-4}$.

'TAUXX' *value*

Set the jet normal stress $\tau_{xx}/a_\infty^2 = value$. The default is $\tau_{xx}/a_\infty^2 = 2k_{\text{jet}}/3$.

'TAUYY' *value*

Set the jet normal stress $\tau_{yy}/a_\infty^2 = value$. The default is $\tau_{yy}/a_\infty^2 = 2k_{\text{jet}}/3$.

```
     'TAUZZ' value
```

Set the jet normal stress $\tau_{zz}/a_\infty^2 = value$. The default is $\tau_{zz}/a_\infty^2 = 2k_{\text{jet}}/3$.

```
     'TAUXY' value
```

Set the jet shear stress $\tau_{xy}/a_\infty^2 = value$. The default is $\tau_{xy}/a_\infty^2 = 0$.

```
     'TAUXZ' value
```

Set the jet shear stress $\tau_{xz}/a_\infty^2 = value$. The default is $\tau_{xz}/a_\infty^2 = 0$.

```
     'TAUYZ' value
```

Set the jet shear stress $\tau_{yz}/a_\infty^2 = value$. The default is $\tau_{yz}/a_\infty^2 = 0$.

4.4 File I/O

```
'GRID BINARY' value
'GRID FORMATTED' value
```

This sets the input of the grid file to either binary (`GRID BINARY`) or formatted (`GRID FORMATTED`) form. The default is to read the grid in binary form.

```
'GRID' value
     'gridfile'
```

Set the grid file to read for the all zones to be *gridfile*. Alternately, grid files can be specified in the `BLOCK` section so that separate files are used for each zone. The grid file is read in PLOT3D single or multiblock format. The single block PLOT3D format is used if a single zone is specified. Otherwise, this specification of the grid file results in the use of the PLOT3D multiblock format.

```
'RESTART IN' value
     'flat.rst1'
```

This sets the run up as a restart to read in the restart file `flat.rst1`. Omitting this sequence results in starting the solution from initial conditions.

```
'RESTART OUT' value
     'flat.rst2'
```

This sets the output restart file as `flat.rst2`. The default output restart file is `isaac.rst`.

```
'SAVE RESTART' value
```

This outputs the restart file at intermediate timesteps after every *value* iterations. The restart file is set by the `'RESTART OUT'` input and is over-written when it is written at intermediate timesteps. The default is to write the restart file only at the end of the run.

```
'PLOT3D' value
     'BINARY'
     'test'
```

This turns on the output of a PLOT3D file for plotting. No PLOT3D plot files are created if this option is omitted. The PLOT3D plot output creates three plot files: the grid file, the solution file, and function file. The PLOT3D files are output at the cell centers of the solution. The solution file contains the mean flow variables in PLOT3D variables (ρ, ρu, ρv, ρw, ρE). The PLOT3D function file contains the properties array and turbulence quantities. N.B. The PLOT3D function file can be read in several plotting packages including Fieldview but *cannot* be read by Fast (this is a restriction in Fast for the function file format). The input for the PLOT3D specification requires the input of the file format (`BINARY` or `FORMATTED`) followed by a single filename. The individual file-names for the grid file, the solution file, and the function file are built from the base filename and the file type e.g. for the above input the grid filename would be **`test.g.unf`**, the solution filename would be **`test.q.unf`**, and the function file containing the turbulence quantities and the property array would be **`test.qt.unf`**. Had the file type been specified as `FORMATTED`, then the file extension would have been '.fmt'.

`'PRINT'` ***value***

Read in *value* lines to output the solution. The *value* print lines follow the following format:

block	*dir*	*istrt*	*jstrt*	*kstrt*	*iend*	*jend*	*kend*	*iinc*	*jinc*	*kinc*
1	2	2	1	2	50	51	2	24	1	1
1	1	1	1	2	51	3	2	1	1	1

where *dir* is the coordinate direction to print along (1 specifies *i*, 2 specifies *j*, and 3 specifies *k*), and print begins at *istrt*, *jstrt*, *kstrt* ends at *iend*, *jend*, *kend* and increments by *iinc*, *jinc*, *kinc*. The output is performed at cell centers with the grid averaged to cell centers.

`'PRINT MAX RESIDUAL'` ***value***

Print the location of the maximum residual of the mean flow equations in the computational field to aid in debugging. This examples prints the maximum residual every 100 iterations. The default is to not output the maximum residuals.

`'PRINT FORCES'` ***value***

Control printing of the lift and drag coefficients. Printing of lift and drag is enabled by default and can be turned on or off with this command. `'PRINT FORCES' 0` turns printing off and `'PRINT FORCES' 1` turns printing on. The calculation of forces is forced if the `2D FARFIELD VORTEX` boundary condition is specified.

```
'REFERENCE AREA' value
```

Set the reference area for force coefficients calculation to *value*. The default value of the reference area is 1.

```
'MONITOR' value
     20000
```

Set up call to user supplied monitor routine. The next line tells the number of iterations to call this routine. The default is to call the monitor routine every 999, 999 iterations (never).

4.5 Numerical Parameters

```
'SOLVER' value
     'DIAGONAL AF3F' 1
```

Set the time integration routine. There are four options available: `DIAGONAL AF3F`, `AF3F`, `MARCH`, and `RK`. `DIAGONAL AF3F` sets the diagonalized spatially split approximate factorization scheme. This is the default. `AF3F` sets the block inversion spatially split approximate factorization scheme. `MARCH` sets the marching (PNS) scheme. `RK` sets the explicit Runge-Kutta scheme.

```
'KAPPA' value
```

Sets the coefficient in the κ scheme in the MUSCL interpolation to *value*. The preferred value is $\kappa = 1/3$. The default value is $\kappa = 0$.

```
'LIMIT' value
```

Sets the limiter in MUSCL scheme. 0 sets no limiter (unlimited). 1 sets van Albada's limiter. 2 sets the minmod limiter. 3 sets Venkatakrishnan's limiter. The preferred limiter is Venkatakrishnan's limiter (3). The default limiter is the minmod limiter.

```
'NITS' value
```

Sets the number of iterations to perform on this run to *value*. The default is 200 iterations.

```
'NITFO' value
```

Sets the number of iterations to perform on this run with first order upwind advection for all equations to *value*. The default is to perform no iterations with the first order scheme. Initializing the flowfield on coarse meshes or in early iterations with the first order scheme can improve robustness.

```
'CFL' value
'DT' value
```

Set the initial time step (`DT`) or the initial Courant number (`CFL`) to *value*. The default is to use a time step based on a constant Courant number of 1 (`CFL`).

```
'CFLFNL' value
```

Sets the final time step or Courant number to *value*. The time step or Courant number is set to the value of `CFLFNL` after `ITDFNL` iterations.

```
'ITDFNL' value
```

Sets the number of iterations to ramp from `CFL` to `CFLFNL` to *value*. The default is 200.

```
'ITUPDT' value
```

Sets the frequency of recalculating the time step. Every *value* iterations the CFL is recalculated as

$$cfl = iter\frac{(\mathtt{CFLFNL} - \mathtt{CFL})}{\mathtt{ITDFNL}}$$

where cfl is the new time step at each point and $iter$ is the current iteration. The default is to update the time step every iteration `ITUPDT` $= 1$.

```
'MESH SEQUENCE' value
     100
     200
```

Set the number of coarser meshes to *value*. The mesh sequencing option initializes the solution procedure on a sequence of coarser meshes to improve the convergence to a steady state. The solution proceeds with 100 iterations on the twice coarsened mesh followed by 200 iterations on the once coarsened mesh. This solution is then interpolated to the finest mesh as the initial condition on the finest mesh. The default is to calculate only on the finest mesh.

```
'MULTIGRID LEVELS' value
```

Set the maximum number of multigrid levels to *value*. This also sets the maximum number of multigrid levels for the coarse meshes used in mesh sequencing. Specifying *value* greater than one automatically activates the multigrid solution procedure.

```
'IMPLICIT SMOOTHING' value
```

Sets the coefficient used in the constant coefficient implicit smoothing of coarse grid corrections in the multigrid prolongation to *value*. The default is 0.25.

```
'IMPLICIT SOURCE' value
     'DIAGONAL'
```

Control of implicit source term treatment. There are three options available: `DIAGONAL`, `BLOCK`, and `NONE`. `DIAGONAL` is the default and approximates the Jacobian as $|S_\phi|/(\rho\phi)$ for each of the turbulence equations (from van Dromme and HaMinh). `BLOCK` uses an approximate form of the Jacobian for the turbulence equations coupling all turbulence equations. `NONE` uses no implicit source term treatment and is **not** recommended.

'ENTROPY FIX' *value*

Set the coefficient for the entropy fix to the acoustic eigenvalues on Roe's approximate Riemann solver to *value*. The input coefficient, *value*, multiplies the maximum eigenvalue to set the minimum that the $U + a$ or $U - a$ eigenvalues can be blended using Harten's entropy fix. The default is no entropy fix.

4.6 Additional Inputs for Marching (MARCH Scheme)

The following inputs are only used for the marching (MARCH) scheme.

'VIGNERON' *value*

Turns on Vigneron approximation to march. Used only with marching scheme.

'TOLER' *value*

Sets convergence tolerance for marching to *value*.

'SIGMA' *value*

Sets the Vigneron safety factor for marching to *value*. The default is 0.95.

'ITUPJ' *value*

Sets the number of iterations before updating the implicit Jacobians for marching scheme to *value*. The default is to update the Jacobians every time step.

4.7 Time Accurate Calculations

Time accurate calculations can be achieved either by using the RK time integration and modifying the Runge-Kutta coefficients for time accuracy or by using the iterative, implicit time accurate option. The iterative, implicit time accurate option allows for a second-order accurate time integration to be performed using a larger time step than the Runge-Kutta scheme. The iterative procedure integrates in pseudo-time to remove linearization errors from the implicit procedure. This procedure allows significantly larger time steps to be taken than an explicit scheme would allow.

'INITIALIZE TIME DEPENDENT' *value*

This option reads in a steady solution and uses it to initialize the unsteady calculation procedure. This option should then be omitted when the unsteady calculation is continued on the next run. This is not used if the flowfield is being initialized from freestream conditions for an iterative, implicit time accurate calculation.

```
'IMPLICIT TIME DEPENDENT' value
```

Specify the iterative, implicit time accurate option. Read *value* options to control the iterative, implicit algorithm. Specifying *value* = 0 sets all of the options to their default values. If neither the DT nor the TIME STEP option is set for the implicit, iterative time accurate option, the code reports an error and exits. The allowed options are:

```
    'SUB ITERATIONS' value
```

Set the number of subiterations per time step to *value*. The default is for 2 subiterations.

```
    'TIME STEP' value
```

Set the physical time step to *value*. The time step defaults to be equal to DT if it is omitted and the DT option is used.

```
    'T-TS' value
    'TAU-TS' value
    'TAU-TS 2ND ORDER' value
```

All of these provide second order temporal accuracy. There are two options for the iterative time dependent: physical time iteration (t-ts) or pseudo time iteration (τ-ts) (see Rumsey et al., Computers & Fluids, 1996 for details). The physical time iteration option T-TS requires that DT and TIME STEP are equal and iterates at this time step. This limits the maximum time step that can be taken due to the stability limit of the iteration scheme. The alternative is the pseudo time iteration, TAU-TS, which takes a physical time step of TIME STEP and iterates each time advancement step at the local value of CFL or DT. This removes the time step restriction as the iterative process advances within its stability limits. The TAU-TS 2ND ORDER requires an additional level of storage for the Q^{m-1} sub-iteration level.

4.8 Zone Description

Each zone of the multizone grid is specified using a BLOCK specification. Between the beginning and ending of the block specification there are several inputs that are used to specify the block grid size, the boundary conditions, and the numerical controls.

```
'BLOCK' value
'END BLOCK' value
```

Delimits the beginning and the ending of each zone. *value* is the zone number beginning with 1 and must match in the 'BLOCK' and 'END BLOCK' statements.

The following inputs are valid in the BLOCK section.

```
'DIMENSIONS' value
       idim         jdim         kdim
```

Set the dimensions of the current block to *idim, jdim, kdim.* The value of *kdim* must be input as 2 if the `TWOD` option is set.

```
'FLUX' value
      'ROE' 1
```

Set the flux evaluation scheme to Roe's flux-difference split scheme - currently this is the only choice.

```
'VISCOUS' value
       dir1         dir2         dir3
```

This sets the coordinate directions on which viscous terms are evaluated. The second line specifies the coordinate directions where viscous terms are evaluated by reading in *value* integers. The integers input correspond to the coordinate direction with 1 corresponding to the `I` coordinate direction, 2 to `J`, and 3 to `K`.

```
'GRID' value
      'gridfile'
```

Set the grid file to read for the zone to be *gridfile.* The grid file can be specified in the `BLOCK` section so that separate files are used for each zone. Alternately, the grid file can be specified outside of the `BLOCK` section and a single grid file is read for all zones. The grid file is read in PLOT3D single block format in either formatted or binary form.

4.8.1 Boundary Conditions

The boundary conditions are specified for each block in the `BLOCK` section of input. Boundary conditions are specified on a rectangular sub-region of each face of the computational cube. The indices specifying the beginning and ending of each boundary condition subregion correspond to the grid points of the input grid. Multiple, different boundary conditions can be specified on any block face.

```
'BC' value
```

Specify the *value* boundary conditions for the zone. The next *value* lines of input specify the boundary condition data for the zone. Note that zone-to-zone communication is handled in a separate input that is outside of the `BLOCK` section.

The boundary condition specification lines follow the following format:

```
      'BC'        istrt   jstrt   kstrt        iend   jend   kend        'bctype'
```

istrt, *jstrt*, *kstrt*, *iend*, *jend*, and *kend* specify the beginning and ending indices for the boundary condition. The boundary condition can run on any rectangular segment of a zone face. The zone face that the boundary condition applies to is set by specifying matching indices; i.e. if *kstrt* = *kend* = 1 then the $k = 1$ face boundary condition is set. Any face can have multiple boundary conditions on it with the only restrictions being that the boundary condition is specified as a rectangular domain and that the total number of boundary conditions for any block is restricted to a value set in the main routine (currently this limit is 10; ISAAC can be recompiled with a larger value of `MXBCS` in the main routine). The indices are specified as the grid point values (not cell centers) so that two segments on a single face would have one index in common.

Legitimate boundary conditions to specify for *'bctype'* are:

`'WALL'`

Set boundary condition for a viscous wall. This boundary condition is used for laminar or turbulent flow when integrating directly to the wall. The velocities on the wall are set to zero. The pressure on the wall is calculated from $\partial p/\partial n = 0$. The wall temperature is set as either adiabatic wall $\partial T/\partial n = 0$ or as a specified wall temperature. The input `TWALL` specifies that a specified wall temperature is to be used and sets the value of the wall temperature. Omitting the `TWALL` input specifies the adiabatic wall condition. Turbulent kinetic energy and Reynolds stresses are set to zero on a wall and a derived condition is specified for either of the dissipation rates.

The `WALL` *boundary condition requires that the grid be refined such that the first grid point away from the wall occurs at a normal spacing such that $y^+ < 1$.*

`'WALL FUNCTIONS'`

Set boundary condition using wall functions. This boundary condition does not integrate directly to the wall and is only appropriate *away* from the wall. The velocity components are matched to a log-law at the first interior solution point. This boundary condition is only valid for the k-ε or Reynolds stress models.

The `WALL FUNCTIONS` *boundary condition requires that the grid be refined such that the first grid point away from the wall occurs at a normal spacing such that $y^+ > 30$.* Additionally, the second grid point should be located close to the first grid point away from the wall, $y_3 - y_2 \approx (y_2 - y_1)/10$, to resolve the gradients in the turbulence quantities.

`'TANGENCY'`

This boundary condition is used to set the flow tangent to the surface. This is used for inviscid calculations.

`'FARFIELD'`

This boundary condition is used to set the flow from 1D Riemann invariants. This is used as the boundary condition at the outer domain for airfoil flows.

`'2D FARFIELD VORTEX'`

This boundary condition is used to set the flow from 1D Riemann invariants with a farfield point vortex correction. *This boundary condition is restricted to two-dimensional calculations.*

'SYMMETRY XY'

This boundary condition sets the flow symmetric about XY plane.

'SYMMETRY XZ'

This boundary condition sets the flow symmetric about XZ plane.

'SYMMETRY YZ'

This boundary condition sets the flow symmetric about YZ plane.

'EXTRAPOLATE'

This boundary condition is used to set the boundary points equal to the interior points.

'FIX'

This boundary condition sets boundary points equal to freestream conditions.

'JET'

This boundary condition is used to set the boundry points to the jet conditions. The jet conditions are specified on input using the `JET CONDITIONS` input. This boundary condition can be combined with the `FIX` boundary condition to set up either a supersonic mixing layer or a supersonic jet. The `FIX` boundary condition would set part of the boundary to the reference condition and the `JET` boundary condition would set the other portion resulting in a top hat profile or a step profile.

'PERIODIC'

This boundary condition specifies periodicity in the coordinate direction.

'SUBSONIC INFLOW'

This boundary condition is used for subsonic inflow with a specified total pressure and total temperature at inflow. Default values of the total temperature and total pressure are specified. These parameters can be modified by using the inputs: `INFLOW PTOTAL` to specify the total pressure and `INFLOW TTOTAL` to specify the total temperature.

`SUBSONIC INFLOW` can not currently be used on a face adjoining a wall or wall function face. This limitation may be removed later.

'SUBSONIC OUTFLOW'

This boundary condition is used for subsonic outflow with a specified back pressure. The default back pressure is set equal to the reference pressure, $p/p_\infty = 1$. The back pressure may be specified on input with the `BACK PRESSURE` input.

'PROFILE CELL CENTER'

This boundary condition is used to read in a file containing data at cell centers for this boundary. The next line after the PROFILE CELL CENTER must contain a filename for the file that contains the data.

'AXISYMMETRIC'

This boundary condition is used for the $k = 1$ and $k = 2$ planes for axisymmetric flow.

'AXISYMMETRIC CENTERLINE'

This boundary condition is used for the singular line for axisymmetric flow.

4.9 Inter-Zone Data Transfer

Zones communicate with each other by inputting a CUT condition that specifies which grid points on one zone correspond to which grid points of a second zone. A single CUT input is required for each of these interfaces.

'CUT' *value*

Specifies *value* cuts to pass data from zone to zone. Two data lines are read in for each cut in the following format:

```
    'NAME1'   block1   istrt1 jstrt1 kstrt1   iend1 jend1 kend1
    'NAME2'   block2   istrt2 jstrt2 kstrt2   iend2 jend2 kend2   iord jord
kord
```

The strings NAME1 and NAME2 must match and are provided for easy identification of each cut condition. *block1* and *block2* identify the two zones/blocks that share a common interface; these block numbers correspond to the values input in the BLOCK section. The *istrt1*, *jstrt1*, *kstrt1*, *iend1*, *jend1*, *kend1* values specify the corners of the rectangular region in *block1* which corresponds to *istrt2*, *jstrt2*, *kstrt2*, *iend2*, *jend2*, *kend2* in *block2*. Note that the ending indices may be smaller than the beginning indices but the right hand rule applies. The order (*iord*,*jord*,*kord*) variables are only input on the second line and specify which coordinate direction of the *block2* zone corresponds to the i, j, and k coordinate directions in *block1*, i.e. *iord* defines the coordinate direction in *block2* that corresponds to the i direction of *block1*, *jord* corresponds to the j coordinate direction and *kord* corresponds to the k coordinate direction. *iord*, *jord* and *kord* may take the value of 1, 2, or 3 corresponding to i, j, or k. The following is a sample cut that is used for the trailing edge cut of a C-grid for an airfoil. The $I = 1$ grid point corresponds to the $I = 257$ grid point for $J = 1$ and the $I = 41$ grid point (which is at the airfoil trailing edge on the lower surface) corresponds to the $I = 217$ grid point on the airfoil upper surface at the trailing edge. The i, j, and k directions match for the two pieces of this cut.

```
’CUT’    1
    ’ONE’     1        1    1    1       41    1   2
    ’ONE’     1      257    1    1      217    1   2        1  2  3
```

4.10 Turbulence Modeling

This section of input controls the choice of turbulence model that is used.

```
’BALDWIN LOMAX’ value
’K EPSILON’ value
’K OMEGA’ value
’REYNOLDS STRESS’ value
```

Set the turbulence model to the Baldwin Lomax algebraic model, a K-ε model, a K-ω model or a Reynolds Stress model. The value of *value* sets the number of model constants to be read in to reset values set to default values. Omitting this line turns no turbulence model on; this is omitted for either inviscid or laminar flow.

4.10.1 Baldwin-Lomax Model

Specify the algebraic Baldwin-Lomax model. *The implementation of the Baldwin-Lomax model is currently only valid for flatplate boundary layers.* This restriction *may* be eased in the future, but this code is not written to test algebraic models.

4.10.2 K-ε Model

The default K-ε model is the Zhang, So, Gatski, and Speziale model. Additional `K EPSILON` models are specified as:

`’SAA’`	*value*	Speziale, Abid, Anderson
`’ZSGS’`	*value*	Zhang, So, Gatski, Speziale
`’ZSSL’`	*value*	Zhang, So, Speziale, Lai
`’HIGH RE’`	*value*	High Reynolds no.
`’RNG’`	*value*	RNG of Yakhot, Orszag, Thangam, Gatski, and Speziale
`’ASM’`	*value*	Explicit algebraic Reynolds stress model
`’ADRM’`	*value*	Anisotropic dissipation rate model

The Speziale-Abid-Anderson (`’SAA’`), Zhang-So-Gatski-Speziale (`’ZSGS’`), Zhang-So-Speziale-Lai (`’ZSSL’`), and explicit algebraic Reynolds stress models (`’ASM’`) are valid for integration to the wall. The high Reynolds number (`’HIGH RE’`), Renormalization Group (`’RNG’`), and anisotropic dissipation rate models (`’ADRM’`) models are not valid for integration to the wall and must use wall functions.

Other inputs that are specified under the `K EPSILON` command can be used to modify model constants or initial conditions. These are:

```
’CMU’ value
```

Specify C_μ to be *value.*

'CEPS1' *value*

Specify C_{ε_1} to be *value*.

'CEPS2' *value*

Specify C_{ε_2} to be *value*.

'SIGMAK' *value*

Specify σ_k to be *value*.

'SIGMAE' *value*

Specify σ_ε to be *value*.

'SIGMARHO' *value*

Set the coefficient for the pressure velocity correlation term to *value*.

'A2KE' *value*

Set the damping coefficient for the near wall term in the Speziale-Abid-Anderson model to *value*. The default value is 4.9.

'ALF1' *value*

Set the coefficient in Sarkar's compressibility correction to *value*. The default value is 0.

'ALF2' *value*

Set the coefficient in Sarkar's pressure dilatation model for the production term to *value*. The default value is 0.

'ALF3' *value*

Set the coefficient in Sarkar's pressure dilatation model for the dissipation term to *value*. The default value is 0.

'PRT' *value*

Specify turbulent Prandtl number, Pr_t, to be *value*.

'PRDLIM' *value*

Set the value of the production limiter to *value*. The production limiter limits the maximum value that the production can be to *value* $\times \rho\varepsilon$. The code outputs a counter for the number of times that this limit is applied in the flowfield. The default value is 100. The minimum value is ≈ 10.

```
'FULL EPSILON PROD' value
```

Use the full production term in the ε transport equation instead of the production limited by the value of `PRDLIM`. This can sometimes improve the robustness of a run at the expense of slightly slower convergence.

```
'TKEINF' value
```

Specify the initial (freestream) value of the turbulent kinetic energy to *value*. The initial (freestream) value of ε is calculated from the values of `TKEINF` and `RMUTNF`.

```
'RMUTNF' value
```

Specify the initial (freestream) value of the turbulent eddy viscosity to *value*. The initial (freestream) value of ε is calculated from the values of `TKEINF` and `RMUTNF`.

```
'MINIMUM TKE' value
```

Specify the minimum value of the turbulent kinetic energy allowed by the `FIXQ` option to *value*.

```
'MINIMUM EPSILON' value
```

Specify the minimum value of the dissipation rate allowed by the `FIXQ` option to *value*.

```
'COMPRESSIBLE DISS' value
      'SARKAR'
```

Specify Sarkar's compressible dissipation model. The default is no compressible dissipation model. The compressible dissipation model can be turned off by specifying `'NONE'`.

```
'PRESSURE DILATATION' value
      'SARKAR'
```

Specify Sarkar's pressure dilatation model. The default is no pressure dilatation model. The pressure dilatation model can be turned off by specifying `'NONE'`.

4.10.3 K-ω Model

The default K-ω model is Wilcox's model. Additional `K OMEGA` models are:

`'ASM'`	Explicit algebraic Reynolds stress model

Both the default K-ω model and the explicit algebraic Reynolds stress (`'ASM'`) model are valid for integration to the wall.

Other inputs that are specified under the `K OMEGA` command can be used to modify model constants or initial conditions. These are:

`'CMU'` ***value***

Specify C_μ to be *value*.

`'BSTRKW'` ***value***

Specify β^* (C_{K_2}) to be *value*.

`'BKW'` ***value***

Specify β (C_{ω_2}) to be *value*.

`'GKW'` ***value***

Specify γ (C_{ω_1}) to be *value*.

`'SIGMAK'` ***value***

Specify σ_k to be *value*.

`'SIGMAW'` ***value***

Specify σ_ω to be *value*.

`'PRT'` ***value***

Specify turbulent Prandtl number, Pr_t, to be *value*.

`'PRDLIM'` ***value***

Set the value of the production limiter to *value*. The production limiter limits the maximum value that the production can be to *value* $\times\rho\varepsilon$. The code outputs a counter for the number of times that this limit is applied in the flowfield. The default value is 100. The minimum value is ≈ 10.

`'FULL OMEGA PROD'` ***value***

Use the full production term in the ω transport equation instead of the production limited by the value of `PRDLIM`. This can sometimes improve the robustness of a run at the expense of slightly slower convergence.

`'TKEINF'` ***value***

Specify the initial (freestream) value of the turbulent kinetic energy to *value*. The initial (freestream) value of ω is calculated from the values of `TKEINF` and `RMUTNF`.

`'RMUTNF'` ***value***

Specify the initial (freestream) value of the turbulent eddy viscosity to *value*. The initial (freestream) value of ω is calculated from the values of `TKEINF` and `RMUTNF`.

`'MINIMUM TKE'` ***value***

Specify the minimum value of the turbulent kinetic energy allowed by the `FIXQ` option to *value*.

`'MINIMUM OMEGA'` ***value***

Specify the minimum value of the specific dissipation rate allowed by the `FIXQ` option to *value*.

4.10.4 Reynolds Stress Model

The default Reynolds stress model is the Zhang, So, Gatski, Speziale model. Additional `REYNOLDS STRESS` models are:

`'SSG'`	Sarkar, Speziale, Gatski
`'FLT'`	Fu, Launder, Tselepidakis
`'ZSGS'`	Zhang, So, Gatski, Speziale
`'GIBSON LAUNDER'`	Gibson, Launder
`'SSG-ADRM'`	SSG pressure-strain with anisotropic dissipation rate model

The default Zhang-So-Gatski-Speziale (`'ZSGS'`) model is valid for integration to the wall. All of the additional Reynolds stress models are high Reynolds number models and must be used with wall functions for wall bounded flows.

Other inputs that are specified under the `K OMEGA` command can be used to modify model constants or initial conditions. These are:

```
'TURBULENT DIFFUSION' value
     'DH'
```

Specify Daly-Harlow as the turbulent diffusion model. Additional turbulence diffusion models are `'HL'` for the Hanjalic-Launder model and `'MH'` for the Mellor-Herring model.

```
'PRESSURE STRAIN' value
     'SSG'
```

Specify the pressure-strain model to be the Speziale-Sarkar-Gatski model. Additional pressure-strain models are `'LRR1'` for the Launder-Reece-Rodi model, `'FLT'` for the Fu-Launder-Tselepidakis model, `'LAI SO'` for the Lai-So model, and `'GIBSON LAUNDER'` for the Gibson-Launder model.

```
’PSC1’ value
’PSC2’ value
’PSCW’ value
’PSALFA’ value
’PSBETA’ value
’PSGAMA’ value
```

Specify coefficients for the pressure strain model to override the default values.

```
’CMU’ value
```

Specify C_μ to be *value.*

```
’CEPSLN’ value
```

Specify C_ε to be *value.*

```
’CSUBS’ value
```

Specify C_s to be *value.*

```
’CEPS1’ value
```

Specify C_{ε_1} to be *value.*

```
’CEPS2’ value
```

Specify C_{ε_2} to be *value.*

```
’CEPS3’ value
```

Specify C_{ε_3} to be *value.*

```
’CEPS4’ value
```

Specify C_{ε_4} to be *value.*

```
’CEPS5’ value
```

Specify C_{ε_5} to be *value.*

```
’SIGMAK’ value
```

Specify σ_k to be *value.*

```
’SIGMAE’ value
```

Specify σ_ε to be *value.*

`'SIGMARHO'` ***value***

Set the coefficient for the pressure velocity correlation term to *value*.

`'ALF1'` ***value***

Set the coefficient in Sarkar's compressibility correction to *value*. The default value is 0.

`'ALF2'` ***value***

Set the coefficient in Sarkar's pressure dilatation model for the production term to *value*. The default value is 0.

`'ALF3'` ***value***

Set the coefficient in Sarkar's pressure dilatation model for the dissipation term to *value*. The default value is 0.

`'PRT'` ***value***

Specify turbulent Prandtl number, Pr_t, to be *value*.

`'PRDLIM'` ***value***

Set the value of the production limiter to *value*. The production limiter limits the maximum value that the production can be to *value* $\times \rho\varepsilon$. The code outputs a counter for the number of times that this limit is applied in the flowfield. The default value is 100. The minimum value is ≈ 10.

`'FULL EPSILON PROD'` ***value***

Use the full production term in the ε transport equation instead of the production limited by the value of `PRDLIM`. This can sometimes improve the robustness of a run at the expense of slightly slower convergence.

`'TKEINF'` ***value***

Specify the initial (freestream) value of the turbulent kinetic energy to *value*. The initial (freestream) value of ε is calculated from the values of `TKEINF` and `RMUTNF`.

`'RMUTNF'` ***value***

Specify the initial (freestream) value of the turbulent eddy viscosity to *value*. The initial (freestream) value of ε is calculated from the values of `TKEINF` and `RMUTNF`.

`'MINIMUM TKE'` ***value***

Specify the minimum value of the turbulent kinetic energy allowed by the `FIXQ` option to *value*. The minimum normal stresses are calculated assuming an isotropic distribution.

```
'MINIMUM EPSILON' value
```

Specify the minimum value of the dissipation rate allowed by the `FIXQ` option to *value*.

```
'COMPRESSIBLE DISS' value
     'SARKAR'
```

Specify Sarkar's compressible dissipation model. The default is no compressible dissipation model. The compressible dissipation model can be turned off by specifying `'NONE'`.

```
'PRESSURE DILATATION' value
     'SARKAR'
```

Specify Sarkar's pressure dilatation model. The default is no pressure dilatation model. The pressure dilatation model can be turned off by specifying `'NONE'`.

4.10.5 General Turbulence Inputs

```
'FIRST ORDER TURBULENCE' value
```

Specifies the use of first order upwind differencing for the advection terms on the turbulence transport equations. This option is used to improve the robustness of the solution procedure, especially in early iterations.

```
'SOURCE' value
```

Evaluate source terms in equations or not. Setting *value* to 1 evaluates the source terms; setting *value* to 0 turns off the evaluation of the source terms. This option can be useful in the initial iterations of a case.

```
'NITALG' value
```

Run *value* iterations evaluating the turbulent eddy viscosity with the Baldwin-Lomax algebraic model. This can only be used with the source terms turned off to initialize a flowfield, but `NITALG` should always be set to zero when the source terms are evaluated.

```
'FIXQ' value
     'MINIMUM'
```

Control of the positivity preservation scheme for the flow variables. Valid Options:

`'NONE'`	do not run positivity scheme
`'MINIMUM'`	set $Q = \max(Q, small)$
`'AVERAGE'`	if $Q < small$ then set Q to be the average of neighboring values of Q

The default is `MINIMUM`. The value of *small* can be modified with the `MINIMUM TKE`, `MINIMUM EPSILON`, or `MINIMUM OMEGA` inputs.

```
'ENFORCE POSITIVE PROD.' value
```

Specifying this option results in the production term for the K, ε, and/or the ω equations being reset to the absolute value of the production. This option is off by default.

4.10.6 Transition Specification

```
'ENFORCE LAMINAR FLOW' value
      block       istrt    jstrt    kstrt       iend     jend     kend
```

Transition is specified by setting the regions of the grid zones that are to be laminar. This input reads *value* additional inputs as above that specify rectangular sub-domains with laminar flow. Multiple sub-domains can be specified for each grid zone. Transition is simulated with this command by setting the production term in the turbulence transport equations to zero in the specified sub-domain. This option is currently not available for the Baldwin-Lomax model.

4.10.7 Turbulence Model Conversion

```
'CONVERT' value
     'K EPSILON' 0
```

Convert from specified model to input model. The input line following this one specified the model converting FROM. In this case, convert from k-epsilon to Reynolds stress as Reynolds stress is specified as the turbulence model. DEFAULT is to not perform the conversion.

Bibliography

[1] Jones, W. P. and Launder, B. E., "The Prediction of Laminarisation with a Two-Equation Model of Turbulence," *Int. J. of Heat and Mass Transfer*, 1972, Vol. 15.

[2] Yakhot, V., Orszag, S. A., Thangam, S., Gatski, T. B., and Speziale, C. G., "Development of Turbulence Models for Shear Flows by a Double Expansion Technique," *Phys. Fluids A*, Vol. 4, No. 7, 1992, pp. 1510-1520.

[3] Speziale, C. G. and Gatski, T. B., "Analysis and Modeling of Anisotropies in the Dissipation Rate of Turbulence," *J. Fluid Mech.*, Vol. 344, 1997, pp. 155-180.

[4] Launder, B. E., Reece, G. J., and Rodi, W., "Progress in the Development of a Reynolds Stress Turbulence Closure," *J. Fluid Mech.*, Vol. 68, 1975, pp. 537-566.

[5] Gibson, M. M. and Launder, B. E., "On the Calculation of Horizontal, Turbulent, Free Shear Flows Under Gravitational Influence," *Journal of Heat Transfer*, Vol. 98, 1976, pp. 81-87.

[6] Speziale, C. G., Sarkar, S. and Gatski, T. B., "Modeling the Pressure-Strain Correlation of Turbulence: An Invariant Dynamical Systems Approach," *J. Fluid Mech.*, Vol. 227, 1991, pp. 245-272.

[7] Daly, B. J. and Harlow, F. H., "Transport Equations of Turbulence," *Phys. Fluids B*, 1970, pp. 2634-2649.

[8] Hanjalic, K. and Launder, B. E., "Contribution Towards a Reynolds-Stress Closure for Low-Reynolds-Number Turbulence," *J. Fluid Mech.*, Vol. 74, 1976, pp. 593-610.

[9] Mellor, G. L. and Herring, H. J., "A Survey of the Mean Turbulent Field Closure Models," *AIAA Journal*, 1973, Vol. 11, pp. 590-599.

[10] Speziale, C. G., Abid, R., and Anderson, C., *AIAA Journal*, Vol. 33, No. 1, January 1995.

[11] Wilcox, D. C., "Reassessment of the Scale Determining Equation for Advanced Turbulence Models," *AIAA Journal*, Vol. 26, No. 11, November 1988.

[12] Zhang, H. S., So, R. M. C., Gatski, T. B., and Speziale, C. G., "A Near-Wall Second-Order Closure for Compressible Turbulent Flows," *Near-Wall Turbulent Flows*, Eds. R. M. C. So, C. G. Speziale, and B. E. Launder, Elsevier Science Publishers B.V., 1993.

[13] Morrison, J. H., "Flux Difference Split Scheme for Turbulent Transport Equations," AIAA Paper 90-5251, October 1990.

[14] Morrison, J. H., "A Compressible Navier-Stokes Solver with Two-Equation and Reynolds Stress Turbulence Closure Models," NASA CR-4440, May 1992.

[15] Morrison, J. H. and Korte, J. J., "Implementation of Vigneron's Streamwise Pressure Gradient Approximation in Parabolized Navier-Stokes Equations," *AIAA Journal*, Vol. 30, No. 12, November 1992.

[16] Morrison, J. H., Gatski, T. B., Sommer, T. P., Zhang, H. S., and So, R. M. C., "Evaluation of a Near-Wall Turbulent Closure Model in Predicting Compressible Ramp Flows," *Near-Wall Turbulent Flows*, Eds. R. M. C. So, C. G. Speziale, and B. E. Launder, Elsevier Science Publishers B.V., 1993.

[17] Abid, R., Gatski, T. B., and Morrison, J. H., "Assessment of Pressure-Strain Models in Predicting Compressible, Turbulent Ramp Flows," *AIAA Journal*, Vol. 33, No. 1, January 1995.

[18] Abid, R., Morrison, J. H., Gatski, T. B., and Speziale, C. G., "Prediction of Aerodynamic Flows with a New Explicit Algebraic Stress Model," *AIAA Journal*, Vol. 34, No. 12, December 1996.

[19] Klebanoff, P. S., "Characteristics of Turbulence in a Boundary Layer with Zero Pressure Gradient," NACA Report 1247, *Forty-First Annual Report of the National Advisory Committee for Aeronautics*, 1955, pp. 1135-1153.

[20] Spina, E. F. and Smits, A. J., "Organized Structures in a Compressible, Turbulent Boundary Layer," *J. Fluid Mech.*, Vol. 182, 1987, pp. 85-109.

[21] Cooke, P. H., McDonald, M. A., and Firmin, M. C. P., "Airfoil RAE2822 - Pressure Distributions, and Boundary Layer and Wake Measurements," AGARD Advisory Report No. 138, *Experimental Data Base for Computer Program Assessment*, May 1979.

[22] Moir, I. R. M., "Measurements on a Two-Dimensional Aerofoil with High-Lift Device," Paper A2, AGARD Advisory Report No. 303 Vol. II, *A Selection of Experimental Test Cases for the Validation of CFD Codes*, August 1994.

[23] Fejtek, I., "Summary of Code Validation Results for a Multiple Element Airfoil Test Case," AIAA Paper 96-xxxx.

[24] Comte-Bellot, G. and Corrsin, S., "Simple Eulerian Time Correlation of Full- and Narrow-Band Velocity Signals in Grid-Generated 'Isotropic' Turbulence," *J. Fluid Mech.*, Vol. 48, Part 2, 1971, pp. 273-337.

Appendix A

Installation and Execution

A.1 Porting and Compilation

ISAAC has been written in FORTRAN-77 to facilitate porting to various operating systems and architectures including vector machines like Cray supercomputers. The standard distribution is set to run on machines running the Unix operating system but should be easily adaptable to other operating systems with minor modifications to the file I/O and compilation.

Programmer's note: file unit numbers are defined in a single `BLOCK DATA` defining mnemonic names to facilitate porting.

The standard distribution includes source files and makefiles in tar files `isaac.tar` and `wsmake.tar` (alternatively `make.tar` for Cray Unicos). Installing ISAAC requires creating a local directory structure `ISAAC/src` in the users home directory, untarring the distribution, and compiling.

```
% cd
% mkdir ISAAC
% cd ISAAC
% mkdir src
% cd src
% tar -xvf isaac.tar
% tar -xvf wsmake.tar
% make
```

A.2 Running ISAAC

ISAAC executes by reading an input file containing a description of the flow case to be run, reading a grid file, running the calculation and outputing a run history with messages and a separate restart file. The input file contains information about the flow conditions, boundary conditions, file I/O, and run control. The input file is read from the Unix standard in file and the output file is written to Unix standard out. Unix file redirection is generally used so that the execution is normally invoked as follows

```
~/ISAAC/src/isaac < input_file > output_file &
```

where `input_file` is the file containing the flow data and `output_file` is written with a run history and error messages.

A.3 Using the STOP File

Early termination of a running ISAAC job can be triggered by creating a file named 'STOP' in the directory where the ISAAC job is running. ISAAC checks every iteration/time step for the existence of this file. If a file named 'STOP' exists in the current directory, ISAAC terminates the current run and outputs restart files as specified in the input data file.

A.4 Memory Usage

Memory for global and work storage is allocated in the main routine (`ISAAC/src/main/main.f`) and then passed into other routines and modules as needed. The memory allocation is controlled through `PARAMETER` statements which can be used to tailor the memory to optimize memory usage for individual user's needs.

`MXPTS`: Number of Solution Points

Set the maximum number of solution points. The number of solution points is determined from:

$$\texttt{MXPTS} = \sum_{nlvls} \sum_{nblocks} (idim + 3)(jdim + 3)(kdim + 3)$$

where *nlvls* is the number of grid levels for the run, *nblocks* is the number of zones for the run, and *idim*, *jdim*, *kdim* are the dimensions of each zone.

`MXNODE`: Number of Grid Nodes

Set the maximum number of grid nodes to store the grid. The number of grid nodes is determined from:

$$\texttt{MXNODE} = \sum_{nlvls} \sum_{nblocks} (idim + 2)(jdim + 2)(kdim + 2)$$

where *nlvls* is the number of grid levels for the run, *nblocks* is the number of zones for the run, and *idim*, *jdim*, *kdim* are the dimensions of each zone.

`MXPPTS`: Number of Property Points

Set the maximum number of property points. The property array contains the molecular viscosity, turbulent eddy viscosity, y^+, normal distance to the wall, and other variables as required by specific models. For laminar or turbulent calculations this should be the same size as the number of solution points, i.e. `MXPPTS` = `MXPTS`. For inviscid calculations this can be set to `MXPPTS` = 1 to minimize storage.

MXQNPT: Number of Points for Unsteady Solution

The unsteady solution procedure requires the storage of additional time levels. If the unsteady solution procedure is to be used, then this should be set to the number of solution points, i.e. MXQNPT = MXPTS. For steady calculations this can be set to MXQNPT = 1 to minimize storage.

MXRKPT: Number of Points for Runge-Kutta Scheme

The Runge-Kutta scheme requires an additional intermediate storage level. If the Runge-Kutta integration procedure is to be used, then this should be set to the number of solution points, i.e. MXRKPT = MXPTS. For implicit calculations this can be set to MXRKPT = 1 to minimize storage. This storage is also used when the CONVERT option is used to convert between turbulence models.

MAXQ: Number of Solution Variables

Specify the maximum number of solution variables per solution point. The number of solution variables per grid point required is shown in Table A.1.

MAXF: Number of Flux Variables

Specify the maximum number of fluxes per solution point. The number of flux variables per grid point required is shown in Table A.1.

MXRANK: Solution Matrix Rank

Specify the rank of the implicit matrices. The number of flux variables per grid point required is shown in Table A.1 for the implicit schemes.

	MAXQ	MAXF	MXRANK
Inviscid	5	5	5
Laminar	5	5	5
Two-Equation Turbulence Model	7	7	7
Reynolds Stress Turbulence Model	12	12	12

Table A.1: Storage required for each model type.

MXSIZE: Size of Largest Zone

Specify the maximum size of the largest zone for solution work space.

$$\texttt{MXNODE} = \max\left[(idim+3)\times(jdim+3)\times(kdim+3)\right]_{nblocks}$$

MXSIZ4: Size of Largest Zone for Fourth Order Scheme

MXSIZ4 = 1.

Appendix B

ISAAC Public Source License

ISAAC Public Source License

Version 1.0
October 2001

Copyright (c) 2001 Joseph H. Morrison

Everyone is hereby permitted to copy and distribute verbatim copies of this license.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTING, AND MODIFYING THE PROGRAM

0. This License applies to the ISAAC (Integrated Solution Algorithm for Arbitrary Configurations) computational fluid dynamics code, its related documentation, and related sample cases and helper applications. The ”Author” below refers to Joseph H. Morrison, the original author of the ISAAC source code, documentation, sample cases, and helper applications. The ”Program” below refers to the ISAAC source code and associated helper applications that are distributed with ISAAC by the Author. Each licensee is referred to as ”You”.

1. You may use, reproduce, and distribute verbatim copies of the Program source code provided that You retain and reproduce intact all of the copyright notices, references to this license, notices of absence of any warranty, and disclaimers; give any other recipients of the Program a copy of this License along with the Program.

2. You may modify Your copy or copies of the Program or any portion therof and copy and distribute Your modifications under the terms of Section 1, provided that:

   a) You must include a prominent notice in the modified files stating that you changed the files and the date of the changes

   b) You must make Source Code of changes that you distribute publicly available under the terms of this license.

3. You may copy and distribute the Program or a modified version of the Program in object code or executable form under the terms of Sections 1 and 2 provided that:

   a) You accompany it with the complete corresponding machine readable source code distributed under the terms of Section 1 and 2 above; OR

   b) You include a prominent notice in the code itself, the executable form of the code, and the related documentation, stating that the source code is available under the terms of this License with information on how and where to obtain the source code of the Program.

4. You may only copy, modify, sublicense, or distribute the Program as expressly provided by this License. Any attempt to copy, modify, sublicense, or distribute the Program in a manner not expressly provided by this License is void and will automatically terminate Your rights under this License. However, parties who have received copies of the Program from You but remain in compliance with this License will not have their licenses terminated.

5. You are not required to accept the terms of this License. However, by downloading, using, modifying, or distributing the Program, You acknowledge that you have read this License Agreement, that you understand it, and that you agree to be bound by its terms.

6. Any recipient of the Program from You automatically receives a license from the original licensor to copy, modify, or distribute the Program subject to these terms and conditions. You may not impose additional restrictions on the recipient's rights granted herein.

7. If any part of this License is found invalid or unenforceable by a court of competent jurisdiction, the remainder of this License shall be interpreted so as to reasonably effect the intentions of the parties. If you cannot distribute the Program so as to satisfy the terms of this License, then You may not distribute the Program at all.

8. The Author of this License may publish revised and/or new versions of the License from time to time. Each new version will be given a distinguishing version number. Once a version of the Program has been published under a particular version of the License, You can continue to use the Program under that version of the License or You may choose to use the Program under the terms of any subsequent version of the License published by the Author.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions differ from this License, contact the Author to ask for permission. ALL distributions of the Program require copmpliance with Sections 11, 12, 13, and 14 below.

NO SALE TO U.S. GOVERNMENT

11. The PROGRAM, and/or any modified version thereof, shall not, in any manner, be offered for sale to the U.S. Government, without the written consent of the U.S. Government. The U.S. Government shall not pay a second time for the PROGRAM or any enhanced/modified version therof. The PROGRAM may be used in contract with the U.S. Government, but no charge may be made for its use. If the PROGRAM is modified using or enhanced using U.S. Government funds, the Government will be provided the complete source code of that modified/enhanced version and the intellectual property rights of the resulting modification/enhancement shall be controlled by such funding agreement.

NO WARRANTY

12. THE AUTHOR AND NASA MAKE NO REPRESENTATIONS REGARDING THE PROGRAM AND EXTENDS NO WARRANTIES OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE PROGRAM WILL CONFORM TO SPECIFICATIONS, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, ANY WARRANTY THAT THE DOCUMENTATION WILL CONFORM TO THE PROGRAM, AND ANY WARRANTY THAT THE PROGRAM WILL BE ERROR-FREE, WHETHER OR NOT THE ERROR IS DISCOVERABLE, AND ASSUMES NO RESPONSIBILITY WHATSOEVER WITH RESPECT TO USE, SALE, OR OTHER DISPOSITION BY UNDERSIGNED OR ITS VENDEES OR OTHER TRANSFEREES OF PRODUCTS INCORPORATING OR MADE BY THE USE OF PROGRAM OR INFORMATION, IF ANY, FURNISHED UNDER THIS AGREEMENT. NOTHING IN THIS AGREEMENT SHALL BE CONSTRUED AS A REPRESENTATION MADE OR WARRANTY GIVEN BY NASA OR THE AUTHOR THAT THE PRACTICE BY UNDERSIGNED OF THE USE GRANTED HEREUNDER SHALL NOT INFRINGE THE COPYRIGHT OR PATENT RIGHTS OF ANY THIRD PARTY. IN NO EVENT SHALL NASA OR THE AUTHOR BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO DIRECT, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF, RESULTING FROM, OR IN ANY WAY CONNECTED WITH THE USE OF THE PROGRAM, WHETHER OR NOT BASED UPON WARRANTY, CONTRACT, TORT, OR OTHERWISE, WHETHER OR NOT INJURY WAS ECONOMIC, LOST PROFITS OR SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, WHETHER OR NOT DAMAGES ARE BASED ON A THIRD PARTY CLAIM, WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF, THE RESULTS OR USE OF THE PROGRAM, AND REGARDLESS OF WHETHER NASA OR THE AUTHOR SHALL BE ADVISED, SHALL HAVE OTHER REASON TO KNOW, OR IN FACT SHALL KNOW OF THE POSSIBILITY OF THE FOREGOING.

13. UNDERSIGNED SHALL INDEMNIFY AND HOLD THE AUTHOR, THE U.S. GOVERNMENT AND ITS OFFICERS AND EMPLOYEES, HARMLESS AGAINST ALL CLAIMS, PROCEEDINGS, LIABILITIES, DEMANDS, DAMAGES, EXPENSES, OR LOSSES, INCLUDING LEGAL EXPENSES AND ATTORNEY'S FEES, ARISING (1) OUT OF THE USE BY UNDERSIGNED OR ITS TRANSFEREES OF THE PROGRAM OR ANY TECHNICAL DATA DERIVED FROM THE USE OF THE PROGRAM OR (2) OUT OF ANY USE, SALE, LEASE, OR OTHER DISPOSITION BY UNDERSIGNED OR ITS TRANFEREES OF PRODUCTS MADE BY USE OF SUCH PROGRAM OR TECHNICAL DATA DERIVED FROM USE OF THE PROGRAM.

14. The conditions imposed by this Agreement shall also apply to any software developed by UNDERSIGNED that incorporates any portion of the PROGRAM SOURCE CODE.

This Agreement constitutes the entire understanding and agreement between the parties hereto relating to release of the Program and may not be superseeded, modified, or ammended except by further written agreement duly executed by the parties.

END OF TERMS AND CONDITIONS