Department of Aeronautics
Imperial College of Science, Technology, and Medicine

הטכניון - מכון טכנולוגי לישראל
Technion - Israel Institute of Technology

CALCULATION OF TURBULENT COUETTE FLOW USING VARIOUS TURBULENT MODELS

ABSTRACT

This report describes some of the different turbulence models and their application to a one-dimensional steady incompressible turbulent Couette flow. In particular it looks at the mixing length turbulence model and at the standard k-ε turbulence model. A mixing length turbulence model code was created. Results from the program were compared with experimental and DNS data from several Reynolds numbers from 1300 to 20000. The results show a good agreement with the DNS and experimental data.

June 2000

Javier Cuadriello Rodríguez

Aeronautics 4th Year.

CONTENTS

APPENDICES

1. Introduction

The objective of this project is to produce programs and results for the velocity profile of a one-dimensional turbulent Couette flow using different turbulence models. This is involves both a channel flow and a boundary layer problem. Results are to be compared with DNS and Experimental data. Turbulence models provide an accurate prediction of a turbulent flow with-out the need to perform expensive experimental tests and do not require the computing power needed to perform direct numerical solutions (DNS) . There is a large number of these models, which can be applied to multiple cases. In this project the models used are the mixing length turbulence model and several variations of the k-epsilon model.

2. Description of the problem, the boundary layer and the Couette flow

This project investigates the modeling of a Couette flow , this kind of flow is both a boundary layer problem and a channel flow problem. This section gives fist an introduction to a general boundary layer and then concentrates in the description of channel flows, and in particular the Couette flow.

2.1 Description of the non-dimensional "+ " system.

When dealing with a boundary layer it is common to non-dimensionalize units using the '+ " system. This allows us to compare results from different Reynolds numbers more easily.
In the "+"system the velocity (u) and the vertical position (y) become :

$$u^+ = \frac{u}{u_\tau} = \frac{u}{u_e}\sqrt{\frac{2}{Cf}} \qquad \text{(Eq. 1)}$$

$$y^+ = \frac{y \cdot u_\tau}{\nu} = \frac{y}{H}\frac{u_t}{u_e}\mathrm{Re} = \mathrm{Re}\frac{y}{H}\sqrt{\frac{Cf}{2}} = \mathrm{Re}\frac{y}{H}\gamma \qquad \text{(Eq. 2)}$$

where :

$$\gamma = \sqrt{\frac{Cf}{2}} \qquad \text{(Eq. 3)}$$

Where H = wall to wall distance , U = velocity of the wall

2.2 STRUCTURE OF A GENERIC BOUNDARY LAYER

This section describes the different regions in a boundary layer. Not all there regions are necessary present in the problem studied in this project, in particular the defect layer doesn't appear in a channel flows..

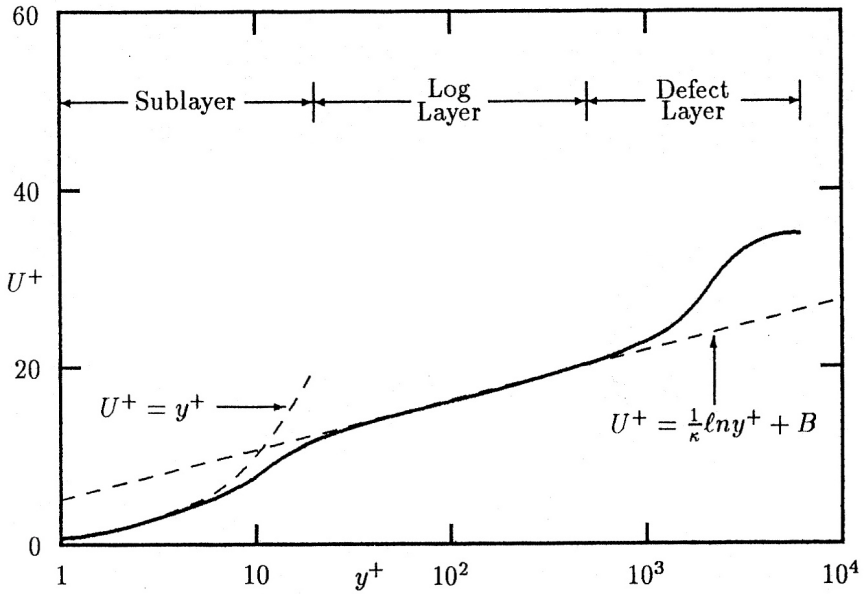


Figure 1 from Ref (9)

A boundary layer can be divided into four different zones. Turbulent models have terms that help model particular parts of the boundary layer and that are damped in the regions where they do not apply. The most important parts are the viscous sublayer, the log layer and the defect layer (see Figure 1). Section 3.3 describes the approximations to the velocity profiles for the viscous sub-layer and the Logarithmic layer. The approximate ranges of the different zones can be seen below.

a) Viscous sublayer y+< 5
b) Buffer layer 5<y+<30
c) Logarithmic layer y+>30 , y<0.1δ
d) Defect layer 0.2 < y/δ < 1

a) The viscous sublayer is the region between the wall and the log layer. Close to the surface, the velocity varies approximately linearly with y+ . A further explanation and derivation of an approximate expression for this model can be found in section 3.3.1

b) The buffer layer is a region where the viscous and turbulent stresses are both important . The velocity profile there may be fitted by the von Karman's empirical formula :

$$U^{+}=4.74 \log y^{+} -2.63 \qquad \text{(Eq. 4)}$$

c) The log layer is often referred to as the fully turbulent wall layer, it is the portion of the boundary layer sufficiently close to the surface that inertial terms can be neglected yet sufficiently far such that the viscous stress is negligible compared to the Reynolds Stress. This region lies between y+=30 and y=0.1δ, where the upper boundary depends on Re. A further explanation and derivation of an approximate expression for this model can be found in section 3.3.2

c) The defect layer lies between the log layer and the edge of the boundary layer. The velocity asymptotes to the law of the wall as $y/\delta \to 0$. However this layer is modeled in this project, as it is not present in channel flows.

2.3 Types of Channel flow

There are two main types of planar channel flows. The Couette Flow and the Poiseuille flow.

Poiseuille flow:
The two walls are stationary and the flow is kept in motion by a pressure gradient along the channel.

The Couette Flow:
There is no pressure gradient but one of the walls is in movement

And of course a combination of the two.

This project treats the Couette flow case.

2.4 The couette flow

In a Couette flow, we have a viscous fluid contained between two flat plates separated by a distance H . It is common in some papers define the wall to wall distance as 2H , but in this project it is defined as H. The velocity of the moving wall is U_e. Due to the no-slip condition the velocity of the flow at the moving wall is also U_e. As there is no pressure gradient the driving force for the flow is the motion of the upper wall, pulling the flow along with it due to friction. The upper plate exerts a shear stress action on the fluid causing a movement towards the right. By opposite reaction the flow exerts a shear force on the wall as well. We assume that the driving force of

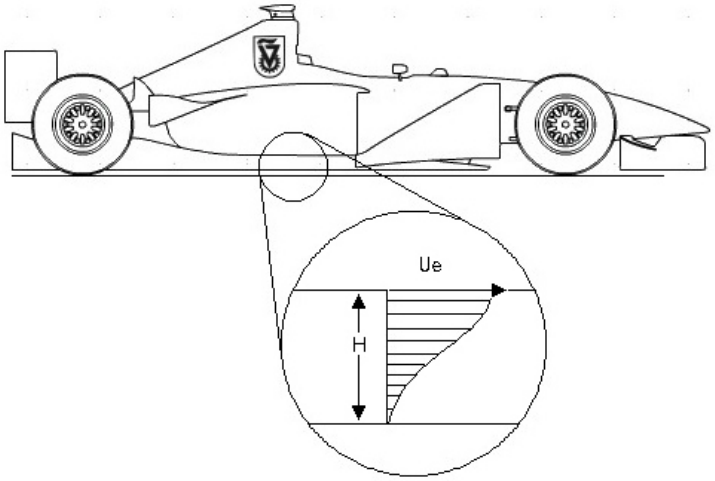


Figure 2

the wall is strong enough to overcome this force. The flow under a racing car (see Figure 1) somehow approaches a Couette flow, as the bottom of the car is flat and very close to the ground. Furthermore, some kinds of racing cars have skirts on the sides that drop almost to the ground to maximize the suction effect from the ground. Despite this the flow under a car is a more complex problem than the one analyzed in this project. A more similar problem could be found in bearings for example.

In a fully developed planer Couette flow it is easy to show that the normal velocity component v is zero and the streamwise component u depends on the normal distance y only. Thus the problem in one dimensional although the velocity and the spatial coordinate do not point at the same direction.. This will be further discussed in later sections when comparing with experimental data.
There are some further assumptions noted below.

Assumptions:

- It is a 1-D problem
- Incompressible flow
- Steady flow
- Plates are at the same temperature
- Temperature changes are neglected

3. The models and representations for a turbulent boundary layer

3.1 the mixing length model

3.1.1 The model

In order to describe the turbulent boundary layer using a mixing length model a mixing length distribution is required. This must be:

(A) Proportional to y^3 in the viscous sub-layer

(B) Proportional to y in the logarithmic layer

(C) Proportional to δ in the outer, wake-like region

The mixing length model implemented in the program combines an exponentially damped mixing length for the viscous layers (Van Driest (1956a)) with a 'ramp' distribution.

i.e. $l = \kappa \cdot \hat{y} \cdot \left[1 - e^{\frac{y+}{A}}\right]$ where "l " is the length scale. (Eq. 5)

and $\hat{y} = \min(y, h - y)$ i.e. the distance to the closest wall (Eq. 6)

The exponential term in equation (Eq. 5) ensures that the mixing length is zero on the wall and that it approaches the correct linear distribution in the logarithmic region.

This model has the merit of being simple and fast to compute. Still it is used very often

FORMULATING THE PROBLEM

From the 2-D Navier Stokes in x-direction

$$\rho \cdot u \cdot \frac{\partial u}{\partial x} + \rho \cdot v \cdot \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left[\mu \cdot \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right)\right] \quad \text{(Eq. 7)}$$

For a 1-D Couette Flow

$$\frac{\partial u}{\partial x} = 0, \ \frac{\partial p}{\partial x} = \mathrm{P}, \ v = 0 \quad \text{(Eq. 8)}$$

For a turbulent case the viscosity will be have a turbulent component, i.e. the turbulent viscosity, referred here as μ_t or ν_t for the dynamic viscosity.

Therefore from (Eq. 8) and (Eq. 7) ,

$$\frac{\partial}{\partial y}\left[\left(\nu+\nu_t\right)\cdot\left(\frac{\partial u}{\partial y}\right)\right]=0 \qquad \text{(Eq. 9)}$$

Adding mixing length turbulence model:

$$\nu_t=\ell^2\frac{\partial u}{\partial y} \qquad \text{(Eq. 10)}$$

Where, as said before $\ell=\kappa\cdot\hat{y}\cdot\left[1-e^{\frac{y+}{A}}\right]$

for κ=0.4 , A=26 and $\hat{y}=\min(y,h-y)$

The required boundary conditions are.

Y = 0 at u = 0
Y = H at u = u_e

Dimensions are normalized as :

η= y / H
z = u / u_e
n = ν_t / u_e H

Where u_e is the velocity of the moving wall and H is the distance between the two walls.
The resulting normalized equation is:

$$\frac{\partial}{\partial\eta}\left[\left(\frac{1}{\mathrm{Re}}+n\right)\frac{\partial z}{\partial\eta}\right]=0 \qquad \text{(Eq. 11)}$$

With boundary conditions z (0) = 0 and z (n) = 1

and with $n=\kappa^2\hat{\eta}^2\left[1-e^{\frac{-\mathrm{Re}\cdot\hat{\eta}\cdot\gamma}{A}}\right]^2\left(\partial u/\partial y\right)$ (Eq. 12)

Where $\hat{\eta} = \min(\eta, 1-\eta)$ (Eq. 13)

where $\gamma = \sqrt{\frac{Cf}{2}}$ (Eq. 14) and $Cf = \frac{2\left(\frac{\partial u}{\partial y}\right)_{wall} \rho \cdot D^2}{\mathrm{Re}^2 \cdot \mu}$ (Eq. 15)

3.1.2 The program

3.1.2.1 General information about the program

The program was written in a mixture of Fortran 77 and Fortran 90. It was written and compiled using Microsoft Fortran Power Station 4.0. Minor changes are required to make it run under UNIX/LINUX environments but it must be compiled with the Fortran 90 compiler due to the way memory is dynamically allocated.
The program asks for an input of Reynolds number, von Karman constant and number of grid points. The output of the program is stored in a file named "*solution.dat*" which contains the y- and U- coordinates for the velocity profile.

3.1.2.2 The numerical discretisation

In a general case the discretisation of this type of equation is:

General equation:

$$\frac{\partial}{\partial y}\left[a \cdot \frac{\partial}{\partial y}(b \cdot \phi)\right] + c\frac{\partial}{\partial y}(e \cdot \phi) + f \cdot \phi = h \quad \text{(Eq. 16)}$$

Which is discretised as

$$\frac{a_{i+1/2}\left(\frac{b_{i+1}\phi_{i+1} - b_i\phi_i}{y_{i+1} - y_i}\right) - a_{i-1/2}\left(\frac{b_i\phi_i - b_{i-1}\phi_{i-1}}{y_i - y_{i-1}}\right)}{\frac{1}{2}y_{i+1} - y_{i-1}} + c_i\frac{e_{i+1}\phi_{i+1} - e_{i-1}\phi_{i-1}}{y_{i+1} - y_{i-1}} + f_i\phi_i = h_i \quad \text{(Eq. 17)}$$

In this case: c = 0 , f = 0 and h =0 $\forall$ i

And as the step is constant y_{i+1}-y_{i-1} = Δy = constant

Therefore

$$\frac{a_{i+1/2}\left(\frac{b_{i+1}\phi_{i+1}-b_i\phi_i}{\Delta y}\right)-a_{i-1/2}\left(\frac{b_i\phi_i-b_{i-1}\phi_{i-1}}{\Delta y}\right)}{\Delta y}=0 \qquad \text{(Eq. 18)}$$

Where :

$$a=\frac{1}{\mathrm{Re}}+\kappa^2\hat{\eta}^2\left[1-e^{\frac{-\mathrm{Re}\cdot\hat{\eta}\cdot\gamma}{A}}\right]^2\left(\partial u/\partial y\right) \qquad \text{(Eq. 19)}$$

With $\hat{\eta}=\min(\eta,1-\eta)$, as in (Eq. 13)

re- arranging:

$$\underbrace{\phi_{i-1}\left(a_{i-1/2}\cdot b_{i-1}\right)}_{A}+\underbrace{\phi_i\left(-a_{i+1/2}\cdot b_i-a_{i-1/2}\cdot b_i\right)}_{B}+\underbrace{\phi_{i+1}\left(a_{i+1/2}\cdot b_{i+1}\right)}_{C}=0 \qquad \text{(Eq. 20)}$$

Generate tri-diagonal matrix, where A , B and C are the 3 non-zero diagonals. This is solved using subroutine adapted from Numerical Recipes for Fortran 77[6] , See Appendix D .

3.1.2.3. The Solving Algorithm.

The actual Fortran code for this algorithm can be found in Appendix D in the subroutine called ‘solve’.

3.1.2.3.1. description

The program has to solve a tri-diagonal system. Naturally the whole matrix is not stored in memory, only the non-zero components. The method used is the Thomas algorithm. This algorithm is a form of the LU decomposition which is valid only for tri-diagonal systems. It is a very fast algorithm with a computational cost of only O(10N) operations.

The algorithm is adapted from Numerical Recipes for Fortran 77[6]

The tri-diagonal system is $a_i u_{i-1} + b_i u_i + c_i u_{i+1} = r_i$

The algorithm has two steps:

Forward Step

$\beta_0 = b_0$

$$u_0 = \frac{r_0}{\beta}$$

$$\gamma_i = \frac{c_{i-1}}{\beta}$$

$\beta = b_i - a_i \cdot \gamma_i$ For i=1 to n

$$u_i = \frac{r_i - a_i \cdot u_{i-1}}{\beta}$$ For i=1 to n

Backward Step

$u_i = u_i - \gamma_{i+1} \cdot u_{j+1}$ For i= n-1 to 0

3.1.2.3.2 Testing the algorithm

In order to check the accuracy of the Thomas algorithm subroutine a simpler case, with known analytical solution was tested.

The following equation ,

$$\frac{\partial^2 z}{\partial \eta^2} = (p+1) \cdot (p+2) \cdot \eta^p$$ Where p is a constant ≥ 0

Can be analytically integrated to a solution of: $z = \eta^{p+2}$

Solving numerically using he Thomas algorithm.

First the second order derivative is approximated using a second order discertisation:

$$\frac{\partial^2 z}{\partial \eta^2} = \frac{z_{i-1} - 2 \cdot z_i + z_{i+1}}{\Delta \eta^2} \quad \text{(for constant intervals)} \qquad \text{(Eq. 21)}$$

Therefore:

$$\frac{z_{i-1} - 2 \cdot z_i + z_{i+1}}{\Delta \eta^2} = (p+1) \cdot (p+2) \cdot \eta^p \qquad \text{(Eq. 22)}$$

$$\begin{bmatrix} 1 & 0 & 0 & & & \\ 1 & -2 & 1 & 0 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ & & & 1 & -2 & 1 \\ & & & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix} = \Delta \eta^2 \cdot \begin{bmatrix} 0 \\ (p+1)(p+2) \cdot \eta_1 \\ \vdots \\ \vdots \\ (p+1)(p+2) \cdot \eta_{n-1} \\ 1 \end{bmatrix}$$

Solve using an adapted Thomas algorithm with the same subroutine used in the programs calculating the Couette flow.

The differences between the numerical and exact solutions are very small, Table 1 shows the maximum error for different numbers of mesh points, this is more clearly shown in . Figures 3 and 4. The error is defined here as the maximum difference between numerical and exact solution at any point.

Table 1

POINTS	P = 0	P = 1	P=2	P=3
10	0	0	2.49E-3	6.39E-3
100	0	0	2.49E-5	6.41E-5
1000	9.32E-14	9.55E-14	2.50E-7	6.4E-7
2000	2.58E-13	1.81E-13	6.25E-8	1.60E-7
4000	1.49E-12	1.00E-12	1.56E-8	4.00E-8
8000	4.26E-12	3.49E-12	3.90E-9	1.00E-8
10000	4.39E-12	2.66E-12	2.50E-9	6.41E-9

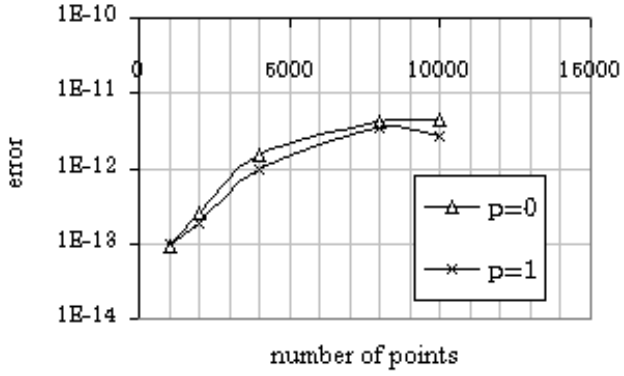


Figure 3

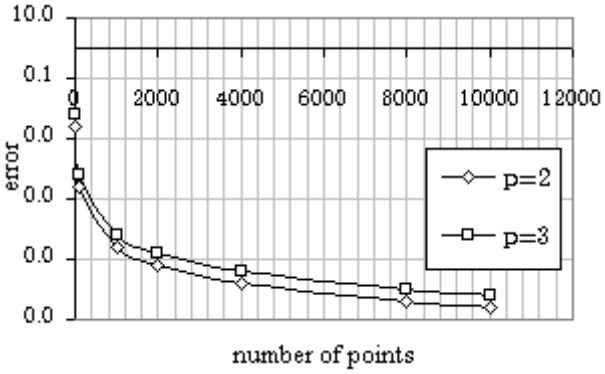


Figure 4

3.1.2.4. Mesh Size

All tests for the mixing length turbulence model case were done using 10000 mesh points, this was a good compromise between accuracy (see previous section). Using less mesh points gave less accurate results in the velocity profile. No improvement in the accuracy of the solution can be found by further increasing the number of points. However, as memory for the arrays is dynamically allocated, the number of points is easy to change. The user is asked for the number of points at the beginning of the program.

3.1.2.5. convergence of the solution

Tests were run on a Pentium 133 MHz with 64Mb of memory. Convergence times were reasonable, always less that a minute even for large Reynolds numbers. Convergence is required for the friction coefficient and the velocity.
Convergence criteria was based on the absolute value of the difference of the solution for the velocity at any point and the friction coefficient between consecutive iterations. This value was required to be smaller than 1.0E-8

3.1.2.6 Symmetry in the solution and Y^+ initial value

The velocity profile data created by the program is supposed to be symmetric. Several tests were performed in order to check the symmetry of the solution. Table 2 shows the maximum error or deviation from perfect symmetry.

Error is defined here as the maximum absolute difference between the obtained solution and the inverted solution at any point i.e. the point with the worst agreement from all the 10000 points.

Table 2

Re	3000	5000	7000	10000	15000	20000
Error	2.03E-01	3.13E-01	4.18E-01	5.62E-01	7.59E-01	8.88E-01

Errors are small although they seem to increase with Reynolds number . The values shown in Table 2, represent the worst case, the average difference is about 10 times smaller, Figure 3 shows a plot where both the normal solution and the inverted solution are plotted together for Re=20000 , which shows the worst agreement in Table 2. It can be seen that the agreement is still very good.

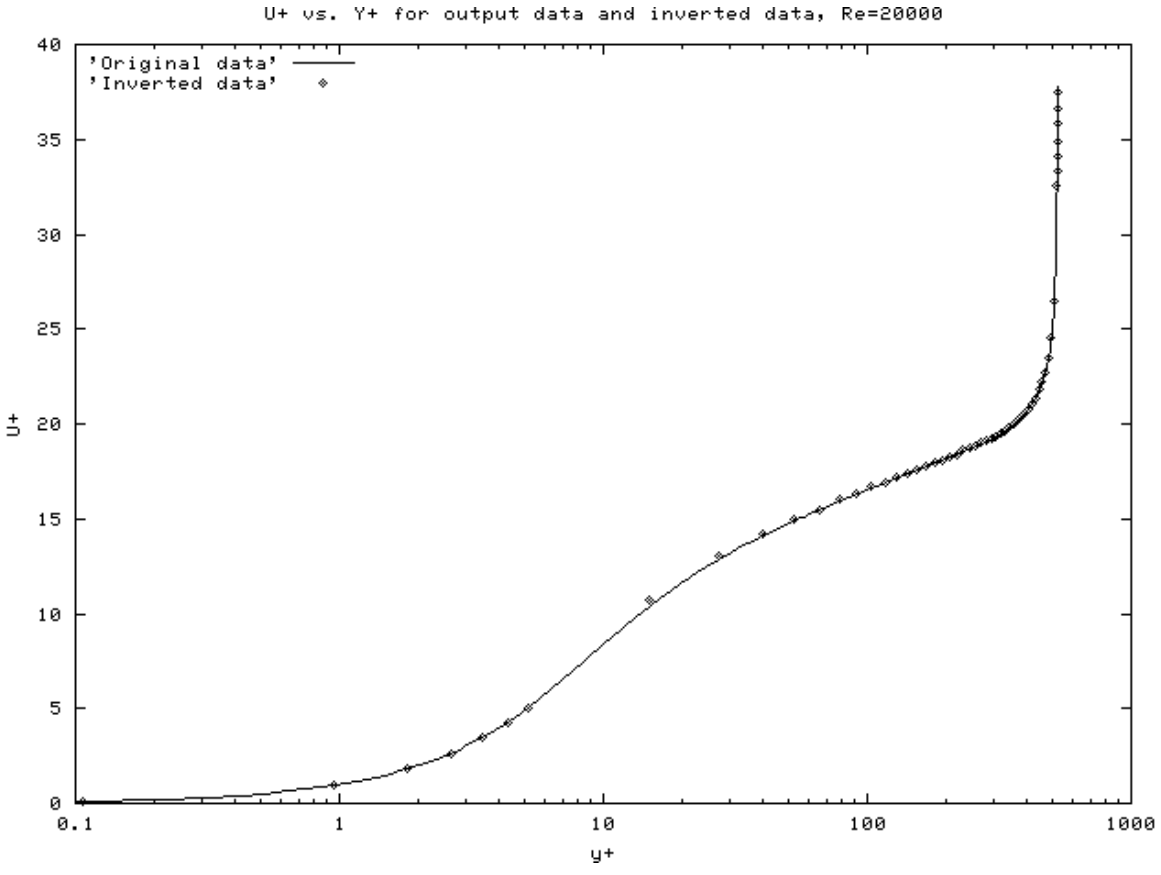


Figure 3

Another important check is that the value of $y^+(1)$ must be smaller than one to obtain a good solution. It can be seen in Table 3 that for all tested Reynolds numbers, $y^+(1)$ is always significantly smaller than one.

Table 3

Re	3000	5000	7000	10000	15000	20000
$y^+(1)$	1E-02	3E-02	2E-02	2.8E-02	4E-02	2.6E-02

3.1.2.7. POSSIBLE IMPROVEMENTS TO THE PROGRAM

In order to improve even further the speed of convergence of the program and to produce smaller output files a variable step size should be implemented. However as convergence times were already reasonable even when run in this modest hardware configuration it didn't seem necessary to add that extra complexity to the program.

3.1.2.8 DIAGRAM OF THE PROGRAM

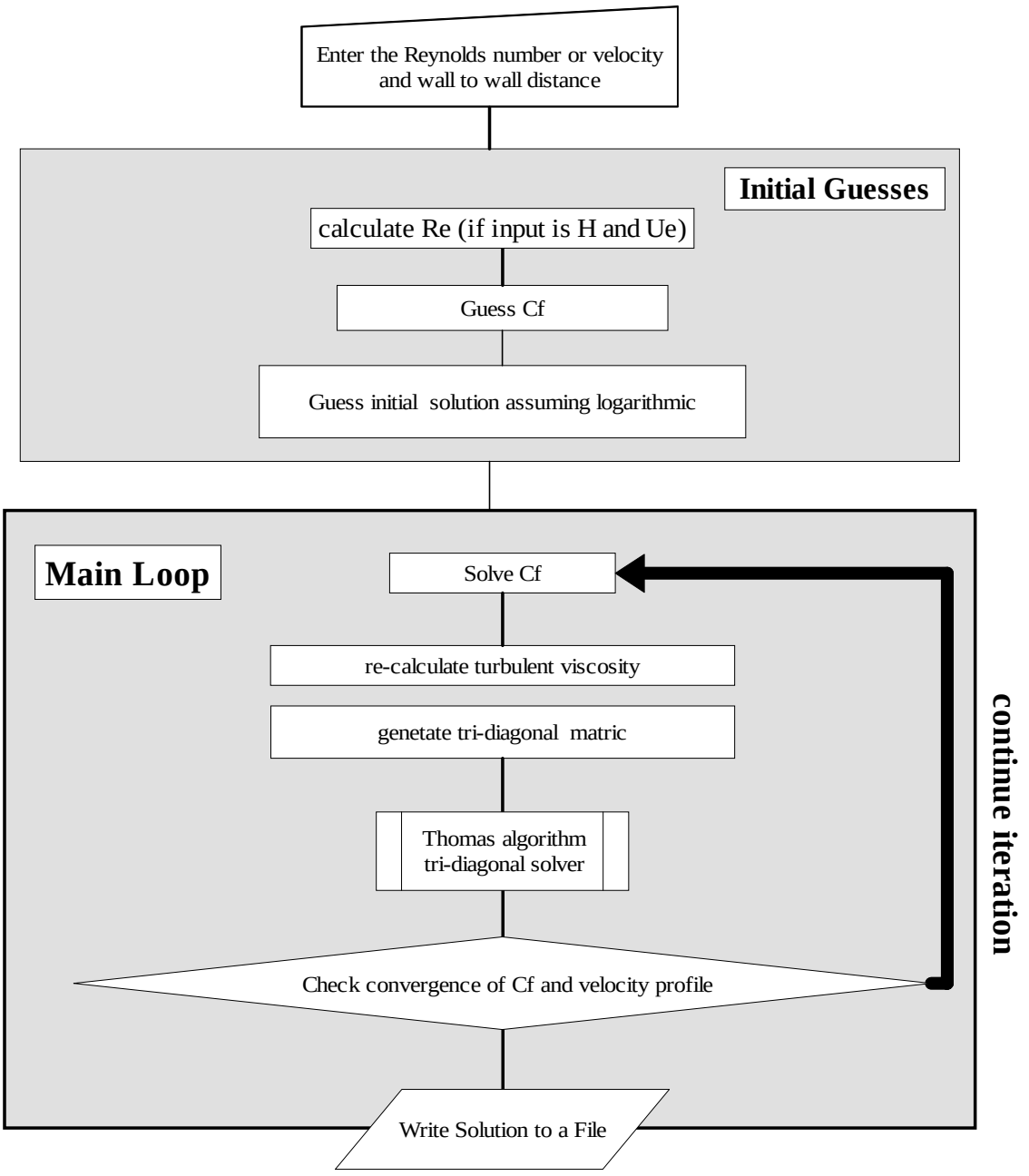


Figure 4

3.2 THE κ- ε TURBULENCE MODEL

The κ–ε turbulence model initially proposed by Harlow and Nakayama in 1968 and later modify by Jones and Launder (*Ref(2)*) is by far the most widely-used two-equation eddy-viscosity turbulence model, mainly because the was long believed to require no extra terms near walls. Despite the several modifications of the models the Jones and Launder variant is still the most commonly used and it is usually called "the standard "κ–ε model. Other variant of the model are the Launder Sharma, Chien , Nagano Tawara , Myon and Kasagi and So.

The popularity of the model, and its wide use and testing, has thrown light on both its capabilities and its defects.

3.2.1 THE MODEL

FORMULATING THE PROBLEM

FOR HIGH REYNOLDS NUMBERS

TURBULENT ENERGY

$$\rho \cdot u \frac{\partial k}{\partial x} + \rho \cdot v \frac{\partial k}{\partial y} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial y}\right] + \mu_t \left(\frac{\partial u}{\partial y}\right)^2 - \rho \cdot \varepsilon \qquad \text{(Eq. 23)}$$

Which for the1-D case becomes:

$$\frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial y}\right] + \mu_t \left(\frac{\partial u}{\partial y}\right)^2 - \rho \cdot \varepsilon = 0 \qquad \text{(Eq. 24)}$$

ENERGY DISSIPATION

$$\rho \cdot u \frac{\partial \varepsilon}{\partial x} + \rho \cdot v \frac{\partial \varepsilon}{\partial y} = \frac{\partial}{\partial x_j}\left[\left(\frac{\mu_t}{\sigma_k}\right)\frac{\partial \varepsilon}{\partial y}\right] + c_1 \frac{\varepsilon}{k}\mu_t \left(\frac{\partial u}{\partial y}\right)^2 - c_2 \frac{\rho \cdot \varepsilon^2}{k} \qquad \text{(Eq. 25)}$$

Which for the1-D case becomes:

$$\frac{\partial}{\partial x_j}\left[\left(\frac{\mu_t}{\sigma_k}\right)\frac{\partial \varepsilon}{\partial y}\right] + c_1 \frac{\varepsilon}{k}\mu_t\left(\frac{\partial u}{\partial y}\right)^2 - c_2 \frac{\rho \cdot \varepsilon^2}{k} = 0 \qquad \text{(Eq. 26)}$$

and

$$\mu_t = C_\mu \cdot \rho \frac{k^2}{\varepsilon} \qquad \text{(Eq. 27)}$$

FOR LOW REYNOLDS NUMBERS

TURBULENT ENERGY

$$\rho \cdot u \frac{\partial k}{\partial x} + \rho \cdot v \frac{\partial k}{\partial y} = \frac{\partial}{\partial y}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial y}\right] + P_k - D_k + S_k \qquad \text{(Eq. 28)}$$

which for the1-D case becomes:

$$\frac{\partial}{\partial y}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial y}\right] + P_k - D_k + S_k = 0 \qquad \text{(Eq. 29)}$$

ENERGY DISSIPATION

$$\rho \cdot u \frac{\partial \varepsilon}{\partial x} + \rho \cdot v \frac{\partial \varepsilon}{\partial y} = \frac{\partial}{\partial y}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial \varepsilon}{\partial y}\right] + P_\varepsilon - D_\varepsilon + S_\varepsilon \qquad \text{(Eq. 30)}$$

which for the1-D case becomes:

$$0 = \frac{\partial}{\partial y_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial y}\right] + P_\varepsilon - D_\varepsilon + S_\varepsilon \qquad \text{(Eq. 31)}$$

where:

P, D and S are the Production rate of kinetic energy by shear forces , Destruction and Source terms respectively and:

$$P_k = \mu_\tau \cdot \left(\frac{\partial u}{\partial y} \right)^2 \quad \text{(Eq. 32)}$$

$$D_k = \rho \cdot \varepsilon \quad \text{(Eq. 33)}$$

$$P_\varepsilon = C_{\varepsilon 1} \cdot \frac{\varepsilon}{k} \cdot P_k \quad \text{(Eq. 34)}$$

$$D_\varepsilon = C_{e2} \cdot f2 \cdot \rho \cdot \frac{\varepsilon^2}{k} \quad \text{(Eq. 35)}$$

In the implementation of this model the turbulent viscosity is modified by a damping function to account for wall proximity:

$$\mu_t = C_\mu \cdot f_\mu \cdot \rho \frac{k^2}{\varepsilon} \quad \text{(Eq. 36)}$$

3.2.1.1 BOUNDARY CONDITIONS

For K

The turbulent kinetic energy is obviously always zero at the walls.

For ε

For the Jones Launder, Launder Sharma and Chien models the k-epsilon equations are modified so that the wall dissipation is zero.

For the NT , MK and S models the wall boundary of the dissipation is:

$$\varepsilon_W = \left[2\mu \frac{\partial^2 k}{\partial y^2} \right]_w \quad \text{(Eq. 37)}$$

However , as the first derivative of the energy vanishes on the wall we can also write:

$$\varepsilon_1 = \frac{2 \cdot k_1}{y_1^2} \quad \text{(Eq. 38)}$$

3.2.1.2 Initial guesses

Initial guesses for k and ε (in the + system)

$$\left.\begin{aligned} k^{+} &= \min(0.16 \cdot y^{+2}, 10/3) \\ \varepsilon^{+} &= \min(0.32, 1/\kappa \cdot y^{+}) \end{aligned}\right\} 0 < y^{+} < \frac{h}{2} \qquad \text{(Eq. 39)}$$

where

$$\varepsilon^{+} = \frac{\varepsilon \cdot \nu}{(u_{\tau})^{4}} = \frac{\varepsilon \cdot \nu}{\left(\mu \cdot \left(\frac{\partial u}{\partial y}\right)_{wall}\right)^{4}} \qquad \text{(Eq. 40)}$$

$$k^{+} = \frac{k}{(u_{\tau})^{2}} = \frac{k}{\left(\mu \left(\frac{du}{dy}\right)_{wall}\right)^{2}} \qquad \text{(Eq. 41)}$$

3.2.1.3 Different coefficients for different variants of the k-epsilon model

JL= Jones Launder , LS = Launder Sharma, C = Chien , NT = Nagano Tawara , MK = Myon and Kasagi , S= So

	f_μ	f_2	S_k	ε_w	S_ε
JL[2]	$e^{\left(\frac{-2.5}{1+Rt/50}\right)}$	$1-0.3\cdot e^{-Rt^2}$	$-2\mu\left(\frac{\partial\sqrt{k}}{\partial y}\right)^2$	0	$2\mu\cdot\mu_t\left(\frac{\partial^2\overline{u}}{\partial y^2}\right)$ (1971 paper) $2\mu\cdot\mu_t\left(\frac{\partial^2\overline{u}}{\partial y^2}\right)^2$ (1972 paper)
LS[4]	$e^{\left(\frac{-3.4}{(1+Rt/50)^2}\right)}$	$1-0.3\cdot e^{-Rt^2}$	$-2\mu\left(\frac{\partial\sqrt{k}}{\partial y}\right)^2$	0	$2\frac{\mu\cdot\mu_t}{\rho}\left(\frac{\partial^2\overline{u}}{\partial y^2}\right)^2$
C[3]	$1-e^{-0.0115y+}$	$1-0.3^{Rt^2/36}$	$-2\mu\left(\frac{k}{y}\right)$	0	$-2\mu\frac{\varepsilon}{y^2}e^{-0.5y+}$
NT[10]	$\left(1-e^{\frac{-y+}{26}}\right)^2 *\left(1+\frac{4.1}{Rt^{\frac{3}{4}}}\right)$	$\left(1-e^{-\frac{y+}{6}}\right)^2 *\left(1-0.3e^{-\left(\frac{Rt}{6.5}\right)^2}\right)$	0	$\left[2\mu\frac{\partial^2 k}{\partial y^2}\right]_w$	0
MK[1]	$\left(1-e^{\frac{y+}{70}}\right)^2 *\left(1+\frac{3.45}{Rt^{\frac{1}{2}}}\right)$	$\left(1-e^{-\frac{y+}{5}}\right)^2 *\left(1-0.222e^{\left(\frac{Rt}{6}\right)^2}\right)$	0	$\left[2\mu\frac{\partial^2 k}{\partial y^2}\right]_w$	0
S[7]	$\tanh\left(\frac{y+}{115}\right)*\left(1+\frac{3.45}{Rt^{\frac{1}{2}}}\right)$	1	0	$\left[2\mu\left(\frac{\partial\sqrt{k}}{\partial y}\right)^2\right]_w$	$C_{\varepsilon 2}\cdot\rho\cdot\frac{\varepsilon_w}{k}-\frac{4}{3}\rho\cdot\varepsilon\frac{\overline{\partial}u}{\partial y}$

CONSTANTS

	C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_k	σ_ε	κ
JL[2]	0.090	1.57-155	2.00	1.00	1.30	0.4095
LS[4]	0.090	1.44	1.92	1.00	1.30	0.4330
C[3]	0.090	1.35	1.80	1.00	1.30	0.4190
NT[10]	0.090	1.45	1.90	1.40	1.30	0.4190
MK[8]	0.090	1.40	1.80	1.40	1.30	0.3950
S[7]	0.096	1.50	1.83	0.75	1.45	0.3850

3.2.2 THE PROGRAM

There are three equations to be solved in every iteration. The first two calculate the turbulence model quantities: k and epsilon, and the third equation calculates the velocity . K and epsilon are used to calculate the turbulent viscosity to solve the third equation. All the equations are discretised in a similar manner and solved by the same subroutine using the Thomas algorithm.

The discretisation from a generic equation like (EQ. 16)

$$\frac{\partial}{\partial y}\left[a \cdot \frac{\partial}{\partial y}(b \cdot \phi)\right] + c\frac{\partial}{\partial y}(e \cdot \phi) + f \cdot \phi = h$$

$$\frac{a_{i+1/2}\left(\frac{b_{i+1}\phi_{i+1} - b_i\phi_i}{\frac{1}{2}(y_{i+1} - y_i)}\right) - a_{i-1/2}\left(\frac{b_i\phi_i - b_{i-1}\phi_{i-1}}{\frac{1}{2}(y_i - y_{i-1})}\right)}{\frac{1}{2}y_{i+1} - y_{i-1}} + P - D + S = 0 \qquad \text{(EQ. 42)}$$

Grouping terms:

$$\phi_{i-1}\left(a_{i-1/2} \cdot b_{i-1}\right) + \phi_i\left(-a_{i+1/2} \cdot b_i - a_{i-1/2} \cdot b_i\right) + \phi_{i+1}\left(a_{i+1/2} \cdot b_{i+1}\right) + (P - D + S) \cdot \frac{1}{4}dy^2 = 0$$

(Eq. 43)

For the k equation

$$A = \left[\frac{\phi_{i-1} \left(a_{i-1/2} \cdot b_{i-1} \right)}{\frac{1}{4} dy^2} \right] \quad \text{(Eq. 44)}$$

$$B = \left[\frac{\phi_i \left(-a_{i+1/2} \cdot b_i - a_{i-1/2} \cdot b_i \right)}{\frac{1}{4} dy^2} \right] \quad \text{(Eq. 45)}$$

$$C = \left[\frac{\phi_{i+1} \left(a_{i+1/2} \cdot b_{i+1} \right)}{\frac{1}{4} dy^2} \right] \quad \text{(Eq. 46)}$$

$$\text{R} = R = -(P_k - D_k + S_k) \quad \text{(Eq. 47)}$$

For the Epsilon Equation,

$$A = \left[\frac{\phi_{i-1} \left(a_{i-1/2} \cdot b_{i-1} \right)}{\frac{1}{4} dy^2} \right] \quad \text{(Eq. 48)}$$

$$B = \left[\frac{\phi_i \left(-a_{i+1/2} \cdot b_i - a_{i-1/2} \cdot b_i \right)}{\frac{1}{4} dy^2} \right] \quad \text{(Eq. 49)}$$

$$C = \left[\frac{\phi_{i+1} \left(a_{i+1/2} \cdot b_{i+1} \right)}{\frac{1}{4} dy^2} - D_\varepsilon \right] \quad \text{(Eq. 50)}$$

$$R = -(P_\varepsilon + S_\varepsilon) \quad \text{(Eq. 51)}$$

Where A , B , C are the three non zero diagonals and R is the right hand side vector of the equation as seen below an “b” is equal to one for all “i”

$$\begin{bmatrix} 1 & 0 & & & \\ A & B & C & & \\ & \ddots & \ddots & \ddots & \\ & & A & B & C \\ & & & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} boundary \\ r_1 \\ \vdots \\ r_{n-1} \\ boundary \end{bmatrix} \qquad \text{(Eq. 52)}$$

(note: "x" could be y,k or epsilon depending on the case)

3.2.2.1 The Solving Algorithm.

As in the mixing length program the algorithm used is the Thomas Algorithm , (See Section 3.1.2.3)

3.2.2.2 Mesh size

As in the mixing length program 10000 points are used between the two walls , (See Section 3.1.2.4)

3.2.2.3 Convergence

The convergence criteria is the same as in the mixing length program, (See Section 3.1.2.5)

3.2.2.4 stability

The program produced unstable results, oscillating from iteration to iteration between two different kind of solutions, one of which seemed to be in the right path , and another one that gave very high values.
As seen in (Eq. 50) the negative term in the epsilon equation is added to the middle diagonal , as in this way its absolute value will be higher and the middle diagonal would be dominant , which increases stability. However this didn't solve the problem in the program.

3.2.2.5 DIAGRAM OF THE PROGRAM

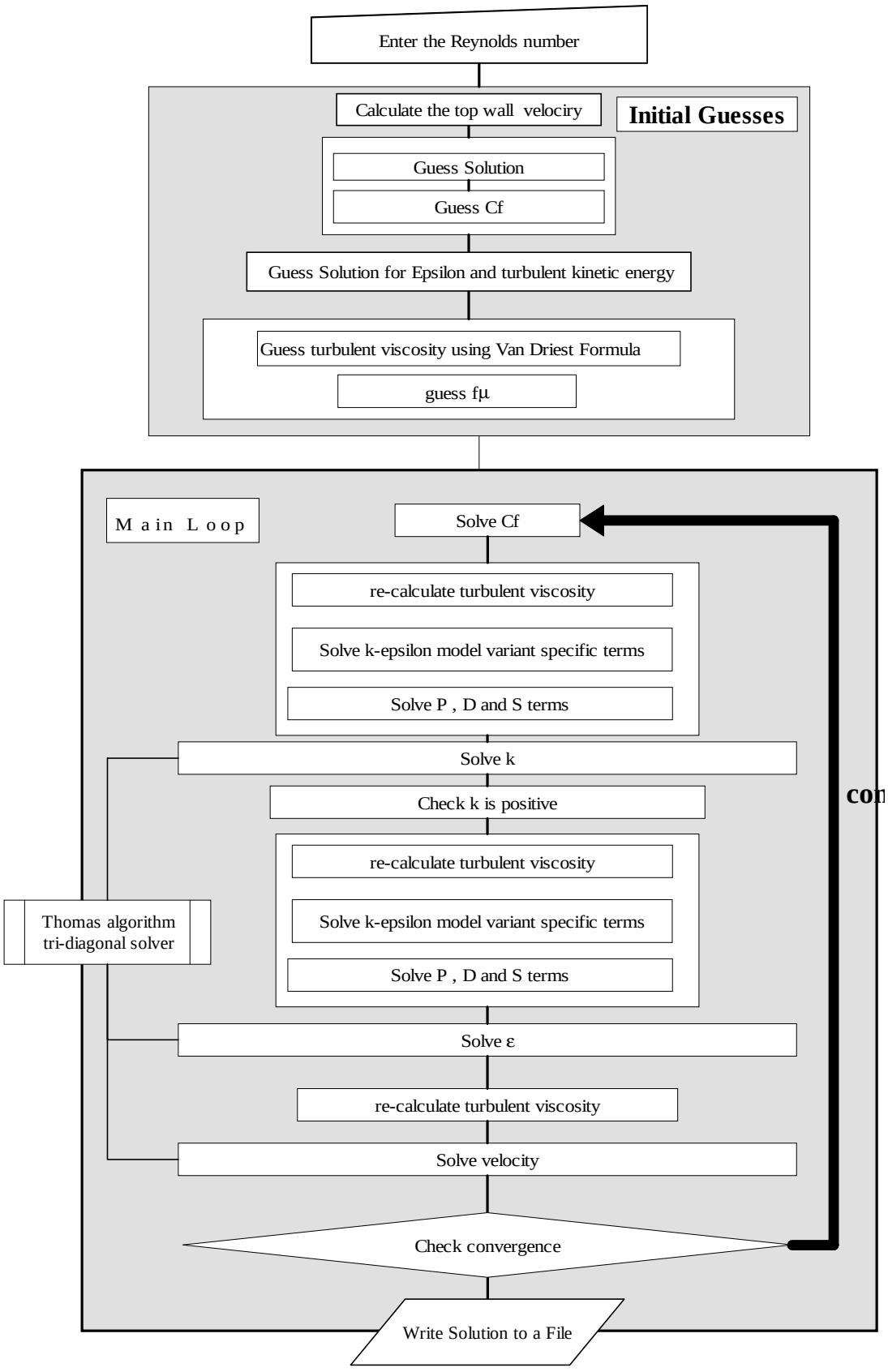


Figure 5

3.3 LAW OF THE WALL

The simpler approximation to the velocity profile of a boundary layer flow is given by the law of the wall. It has two parts, the linear region , which represents the viscous sub-layer and the logarithmic law which represents the logarithmic region .The expressions for the simple formulas for the rest of the regions are not described here as these regions are very small. The law of the wall provides a general approximation to the boundary layer velocity profile which is very useful to initially assess the results from more complicated models. This representation based on some assumptions provides a surprisingly good estimate of the boundary layer velocity profile.

3.3.1. LINEAR DISTRIBUTION

Near a smooth solid surface, the velocity fluctuations are strongly damped. Thus there is likely to be some small layer dominated by the viscous rather than the turbulent stresses. This layer is called the viscous sub-layer. The viscous sublayer is surely very thin, i.e. it spans a small *y*. One can therefore suppose that as $\tau = \tau_w + \frac{\partial P}{\partial x} y$, then $\tau = \tau_W$ in this region.

Therefore,

$$\frac{\partial U_+}{\partial y_+} = 1 \qquad \text{(Eq. 53)}$$

Which can be integrated with boundary condition U(0)=0 to give :

$$u_+ = y_+ \qquad \text{(Eq. 54)}$$

3.3.2 THE LOGARITHMIC LAW

The so called logarithmic law represents an approximation to the logarithmic region of the boundary layer. The expression of the log-law can be obtained as a exact solution to the standard boundary layer equations assuming that the length scale is $\ell = \kappa \cdot y$, with κ being the Von Karman constant.

Derivation

Starting from the general boundary layer equation:

$$\rho \cdot U \frac{\partial U}{\partial x} + \rho \cdot V \frac{\partial U}{\partial y} = \frac{\partial}{\partial y}\left[(\mu + \mu_\tau)\frac{\partial U}{\partial y}\right] \quad \text{(Eq. 55)}$$

for the 1-D case

$$\frac{\partial}{\partial y}\left[(\mu + \mu_\tau)\frac{\partial U}{\partial y}\right] = 0 \quad \text{(Eq. 56)}$$

or $$\frac{\partial}{\partial y_+}\left[(1 + \mu_+)\frac{\partial U_+}{\partial y_+}\right] = 0 \text{ in the "+" system} \quad \text{(Eq. 57)}$$

Integrating equation (Eq. 56)we obtain:

$$(\mu + \mu_\tau)\frac{\partial u}{\partial y} = C' \text{ or } (\nu + \nu_\tau)\frac{\partial u}{\partial y} = C \quad \text{(Eq. 58)}$$

or in the "+" system from (Eq. 57):

$$(1 + \nu_+)\frac{\partial u_+}{\partial y_+} = 1 \quad \text{(Eq. 59)}$$

Furthermore we know that in this region $\nu_+ >> \nu$

Using the Prandtl mixing length hypothesis we get

$$\nu_t = \ell^2\left(\frac{\partial u}{\partial y}\right) \text{ or } \nu_+ = \ell_+^{\,2}\left(\frac{\partial u_+}{\partial y_+}\right) \text{ in the "+ "system} \quad \text{(Eq. 60)}$$

Where ℓ is the length scale and for this simplified case:

$$\ell = \kappa \cdot y \quad \text{(Eq. 61)}$$

Therefore

$$\kappa \cdot y^{+}\left(\frac{\partial u^{+}}{\partial y^{+}}\right)=1 \qquad \text{(Eq. 62)}$$

Integrating equation (Eq. 62)we obtain the expression for the low-law.

$$u^{+}=\frac{1}{\kappa}\log_{e} y^{+}+C_{2} \qquad \text{(Eq. 63)}$$

Where C_2 is the integration constant and has to be found empirically. This limits the use of the logarithmic law as this value can be different for different boundary layer problems, therefore, apart from the limited accuracy of its results, it lacks the necessary generality. For this case C_2 is usually a value close to 5.2 , and this is the value used in the graphs for this project.

4. Comparison of results with experimental and DNS data and discussion

The data from the mixing length program is compared with three different sets of data. The first set of data was produced using DNS (*Ref. (11)*) and the other two come from experiments (*Ref.(13)* & (14)) .
The DNS data from Bech (*Ref. (11)*) and one of the Aydin & Leutheusser Ref (14) sets are for a low Reynolds, in this case Re=1300.
The other set of experimental data from Nakabayashi, Kitoh & Nishimura (Ref*(13))* has several measurements from Reynolds numbers ranging from 3000 to 20000

Low Reynolds numbers

Graph 2 in Appendix A shows the velocity profile obtained by the mixing length program, the Bech DNS data and the Aydin & Leutheusser experimental data.
The agreement between the three data sets is very good. The Aydin & Leutheusser seems to provide the worst agreement, but this is due probably due to the fact that the data as read by hand from the paper (this data appear also Ref (12) and it was copied from there, not the original paper). Apart from this the experimental data seem to be slightly above the DNS calculation and the mixing length results at about y+ = 30. This can be seen in Graph 2 in Appendix A but can be more clearly seen in Ref (12)

The slopes of U+ also agree well between the mixing length values and the DNS values from Bech , this can be clearly appreciated in Graph 3 with compares $\frac{\partial u^+}{\partial y^+}$ versus y+ . The experimental data from Aydin & Leutheusser could not be included in this plot due to the lack of points and reliable values.

High Reynolds Numbers

For higher Reynolds numbers the results are compared with the data from the Nakabayashi, O. Kitoh and F. Nishimura (*Ref* 32) . This experimental data goes up to Re=20000. I do not believe many experiments for higher Reynolds numbers can be found due to the difficulty of reproducing high Reynolds numbers in wind tunnel testing.
Results can be observed in Graph 1 in Appendix A. Agreement between data and experiment is good for both all Reynolds numbers. Agreement is also good with the logarithmic law and the U+ =y+ approximation for the viscous sublayer which are also plotted in Graph 1. Once again the main cause for the small disagreements observed is due to the fact that the data was read by hand from the paper. However, in general experimental data seems to give a slightly higher value for u+ in the logarithmic region.

Both the results from the low and the high Reynolds numbers agree well with experimental and DNS data, therefore the 1-D assumption for the fully developed Couette flow proves to hold well .

5. CONCLUSION

It has been shown that the mixing length turbulence model can provide a reasonable estimate of the mean velocity for a one dimensional, incompressible turbulent Couette flow for a range of Reynolds numbers up to 20000. The assumption of a one-dimensional flow for this problem is valid since the results are very close to those produced by experiments. It has also been shown how the law of the wall provides a good initial estimation for the velocity profile for this problem. Unfortunately, the k-e turbulence model code did not function properly before the end of the project and therefore its results could not be analysed.

6. References

(1) H.K. Myong and N. Kasagi , JSME Intern. J., Vol 33, (1990) , pp 63-72

(2) JL, W.P Jones and B.E. Launder , International J. Heat and Mass Transfer , Vol 15 (1972) , pp 301-314

(3) K.Y. Chien , AIIA J. Vol 20 , (1982) pp 33-38

(4) LS , B.E. Launder and B.I. Sharma , Lett Heat and ass Trasnfer , Vol 1 (1974) pp 131-138

(5) Millikan (1938) Velocity Defect Law.

(6) Numerical Recipes for FORTRAN 77 The Art of Scientific Computing , William H. Press, Saul A. Teukolsky (Contributor), William T. Vetterling, Brian P. Flannery (Contributor) (online version at www.nr.com) Cambridge University press.

(7) RCM So. NASA CR 189608 (1991)

(8) Van Driest (1946) Mixing length ramp distribution

(9) Wilcox, David C. , Turbulence Modeling for CFD, DCW Industries , (1993)

(10) Y. Nagano and M. Tagawa, ASME Trans, J. Fluid Eng., Vol 112 (1990) pp 33-39

(11) BECH, K.H, Turbulent plane Couette flow – DNS, Appl. Sci. Res 51,237

(12) Knut h Bech , Nils Tillmark , P. Henrik Alfredsson , Helge I. Anderson , An Investigation of turbulent plane Couette flow at low Reynolds numbers. Journal of Fluid Mechanics (1995) , 00 291-325.

(13) Nakabayashi, Kitoh, Nishimura, Experimental Study of a turbulent Couette flow at low Reynolds number.

(14) Aydin , E. M . & Leutheusser, H.J . 1987 Experimental investigation of turbulent plae-Couette flow. ASME Forum on turbulent Flows, FED vol 51, p.51

7. Bibliography

Anderson J.D., Fundamentals of Aerodynamics,McGraw-Hill 1991.

B. Debusschere * and C. J. Rutland , Turbulent Scalar Transport Mechanisms in Plane Channel and Couette Flows, University of Wisconsin–Madison

Knut H Bech , Nihls Tillmark, P. Henrix Alfreddon , Helge Angersson, An Investigation of Turbulent Plane Couette Flow at Low Reynolds Numbers
J. Fluid Mechanics (1995) vol 286, pp. 291-325

B. Debusschere * and C. J. Rutland ,Turbulent Scalar Transport Mechanisms in Plane Channel and Couette Flows, Submitted to J. Fluid Mech., June 1998

Tuncer Cebeci / A.M.O. Smith , Analysis of turbulent Boundary Layers, Academic Press, New York 1974

Schetz , Joseph A. Boundary Layer Analysis, Prentice-Hall , 1993

Carol Braester. Introduction to numerical methods, Civil Engineering , Israel Institute of Technology, (1998)

Eif N. Persen, Boundary layer Theory, Tapir, (1972)

M. Wolfshtein , D. Naot , A. Lin , Topics in Transport Phenomena , Chapter 1, John Wiley and Sons 1975

Frost and Moulden , Handbook of turbulence, volume 1, Plenum Press, (1977)

Schlichting , Hermann , Boundary Layer theory, McGraw Hill , series in Mechanical Engineering, , 3rd edition (1968)

Peiró J. , Sherwin , CFD Lecture Notes, Imperial College of Science , Technology and Medicine , Aeronautics Department (1999)

M. Wolfshtein , Toulouse Lectures , March 1996

Morrison J.F. Turbulence Modelling Lecture notes, Imperial College of Science , Technology and Medicine , Aeronautics Department (1995)

M. Wolfshtein, The velocity and temperature distribution in one dimensional flow with turbulence augmentation and pressure gradient, Imperial College, Dept. Mechanical Eng. (1968)

Gibson M.M . Turbulence Modelling Lecture notes, Imperial College of Science , Technology and Medicine , Mechanical Eng. Dept. (1997)

F-1 picture from jaguar F1 website.

Numerical Recipes in Fortran 90 : The Art of Parallel Scientific Computing (Fortran Numerical Recipes , Vol 2) , William H. Press (Editor), Saul A. Teukolsky (Contributor), Michael Metcalf, Cambridge University Press 1996

Young . A. D. Boundary Layers. AIAA education Series, !989

P.H. Oosthuizen and D. Naylor An Introduction to Convective Heat Transfer Analysis" McGraw-Hill.

B. Aupouix , Lectures on Turbulence.

8. Acknowledgements

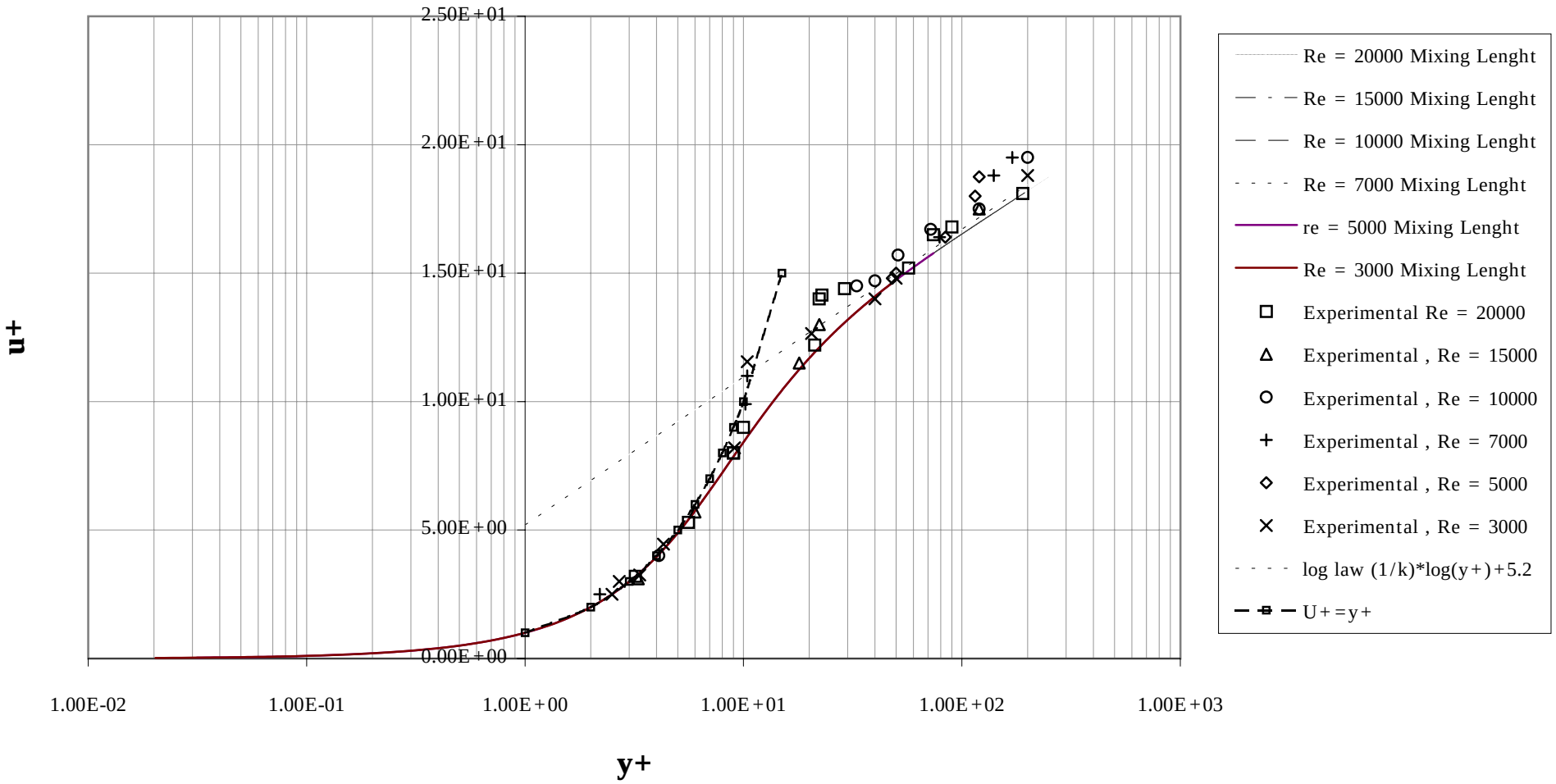
I would like to thank Prof. Micha Wolfshtein and Dr. B. Aupoix for their help and their patience in this project and Dr. R. E. Brown for his support.

APPENDICES

Appendix A – graphs

Graph 1

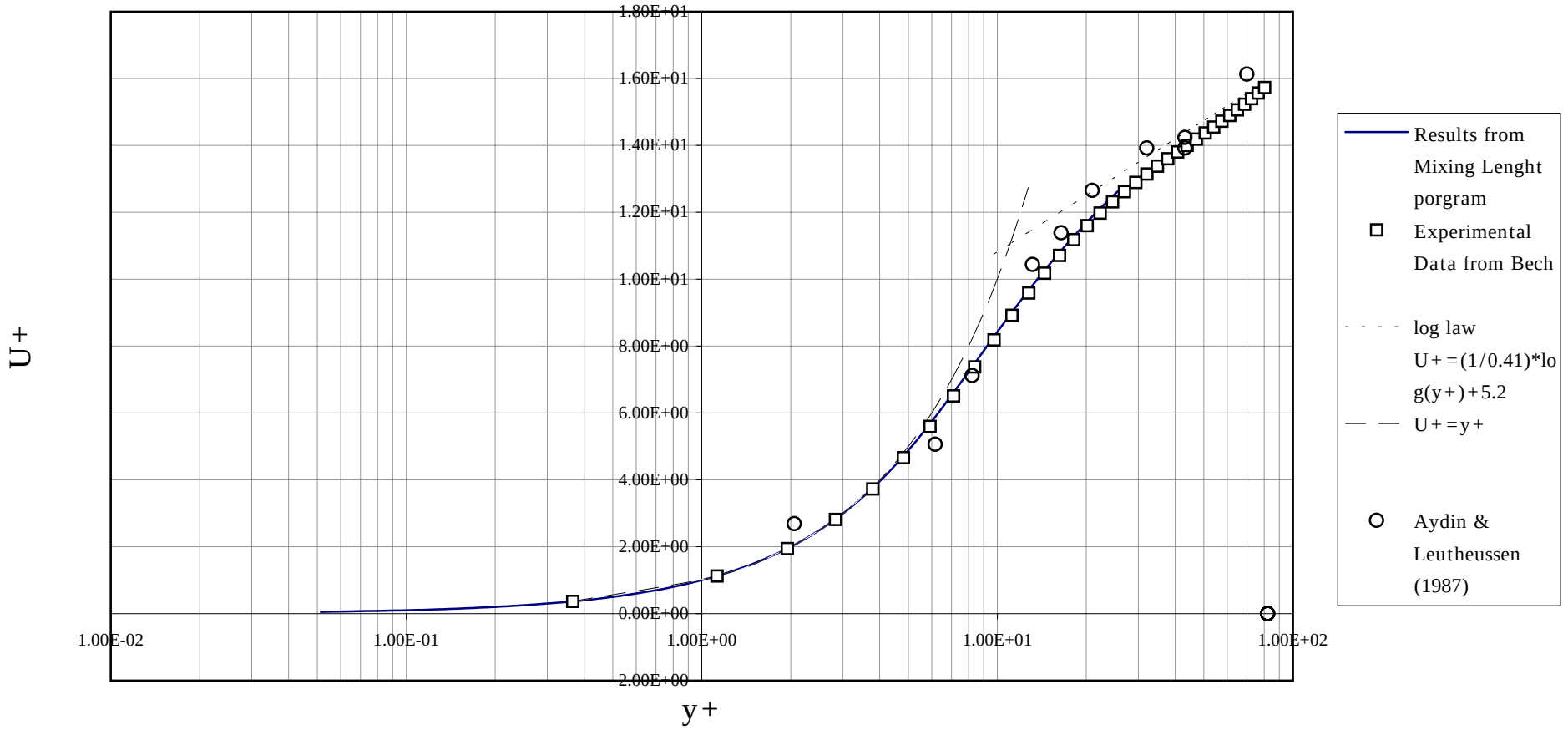
Mixing Lenght program results compared with experimental data from K.Nakabashi, O.Kitoh , F. Nishimura paper



Graph 2

Bech and Aydin & Leutheussen data and Mixing Lenght comparison.

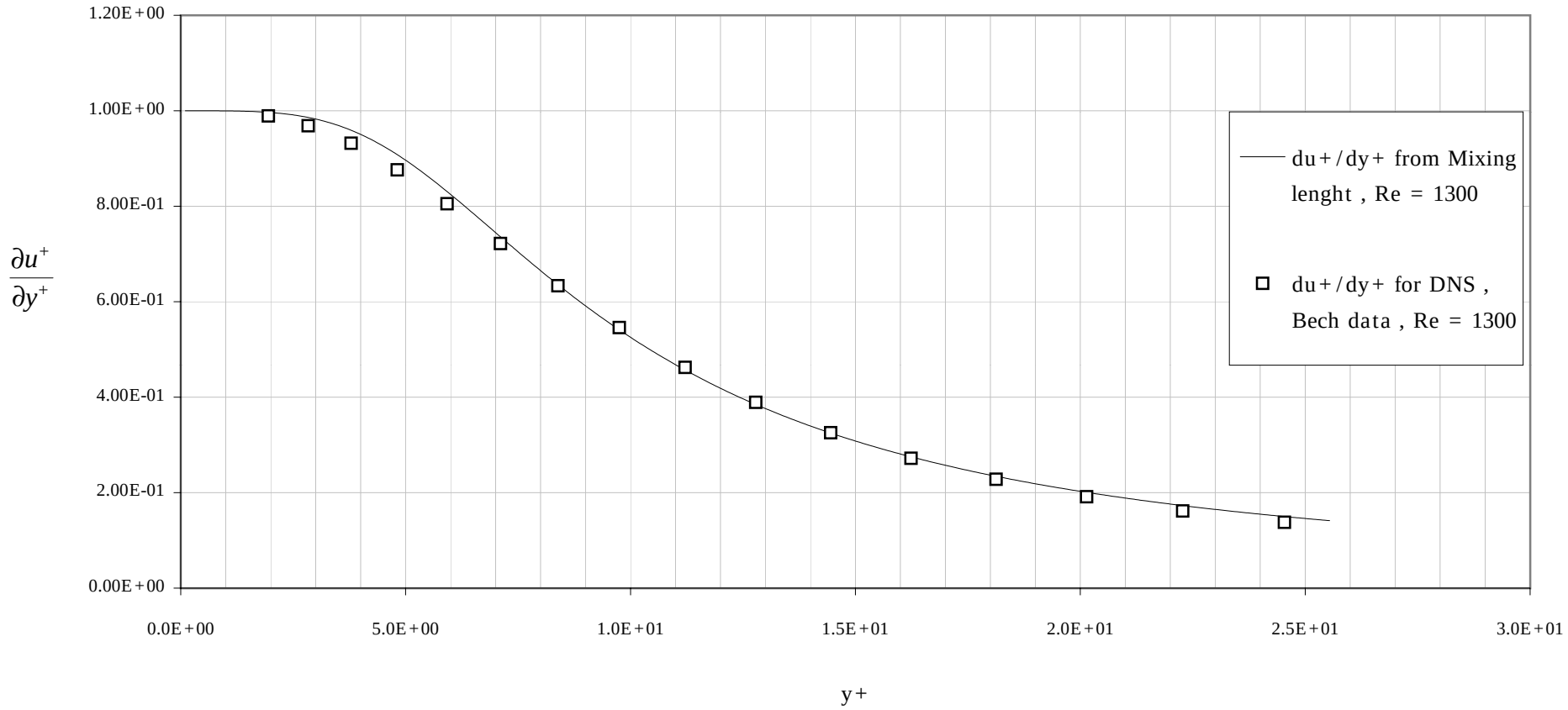
Low Reynods Numbers, Re = 1300



du+/dy+ vs. y+ for mixing lenght results and data for Bech data ,

Low Reynolds Numbers , Re = 1300

Graph 3



Appendix B - Definition of the Symbols

A = van Driest Constant

U = mean velocity in the horizontal direction

V = velocity in y direction, always zero in this project

Cf = Friction Coefficient

ℓ = length scale

U+ = mean velocity in the plus system

Y+ = position in y (vertical) axis in the plus system

$u_\tau = u^* =$ friction velocity, or wall shear velocity

ε = energy dissipation

$\varepsilon+$ = energy dissipation in the "+ " system

k = turbulent kinetic energy

k+ = turbulent kinetic energy in plus system

γ = gamma , the square root of half the friction coefficient

$\eta = \frac{y}{H}$ dimensionless vertical position in the channel

δ = boundary layer thickness

$z = \frac{u}{u_e}$, dimensionless velocity

μ = viscosity

$\nu = \frac{\mu}{\rho}$ = dynamic viscosity

ρ = density

κ = Von Karman constant , always taken as 0.41 or 0.4 in this project

τ = shear stress

τ_w = wall shear stress

Re = Reynolds numbers

$R\tau$ = Turbulent Reynolds Number

U_e = velocity of the moving wall , velocity of the flow at the moving wall

H = Wall to wall distance

APPENDIX C - SOME USEFUL RELATIONS

$$\nu_t = \frac{\mu_t}{\rho}$$

Rt= turbulent Re $= \dfrac{k^2}{\varepsilon \cdot \nu}$

$$u^+ = \frac{u}{u_\tau} = \frac{u}{u_e}\sqrt{\frac{2}{Cf}}$$

$$y^+ = \frac{y \cdot u_\tau}{\nu} = \frac{y}{H}\frac{u_t}{u_e}\mathrm{Re} = \mathrm{Re}\frac{y}{H}\sqrt{\frac{Cf}{2}} = \mathrm{Re}\frac{y}{H}\gamma \text{ where : } \gamma = \sqrt{\frac{Cf}{2}}$$

$$T_w = \mu \cdot \left(\frac{\partial u}{\partial y}\right)_{wall}$$

$$Cf = \frac{T_w}{\frac{1}{2}\rho \cdot u_e{}^2}$$

$$u_\tau = \sqrt{\frac{T_w}{\rho}}$$

APPENDIX D- MIXING LENGTH PROGRAM SOURCE CODE

```
C    COMPILE WITH FORTRAN 90 (WRITTEN IN MS FORTRAN 4.0)!!!
C    JAVIER CUADRIELLO MARCH 2000
C=============================================================
C=============================================================

     IMPLICIT NONE

C    DECLARING DYNAMICALLY ALLOCATABLE ARRAYS
C=============================================================
     REAL*8  ,ALLOCATABLE:: ZPAST(:), A(:),B(:),C(:)
     REAL*8  ,ALLOCATABLE:: AA(:),BB(:),CC(:)
     REAL*8  ,ALLOCATABLE:: AM(:),BM(:),CM(:)
     REAL*8  ,ALLOCATABLE:: AP(:),BP(:),CP(:)
     REAL*8  ,ALLOCATABLE:: DIFF(:), Z(:),DZDY(:), DZDYPAST(:)
     REAL*8  ,ALLOCATABLE:: ALFA(:) , BETA(:), GAMA(:), DELTA(:)
     REAL*8           ,ALLOCATABLE:: X(:),R(:),U(:),GAM(:),Y(:)
     REAL*8  , ALLOCATABLE:: TV(:) ! TV = TURBULENT VISCOSITY

C    DECLARING REST OF VARIABLES
     REAL*8    DY,EPSILON,VEL,H,RE ,DIV,DUDYWALL     ,CFPAST, DUMMYR
     REAL*8    K,CF,KK, G, DENSITY, VISCOSITY
     INTEGER   I,N,J , T ,JJ ,M  ,MM , OPT



     PARAMETER (EPSILON=1.0E-8)
                    !    PARAMETER (DY=0.001)
     PARAMETER (DENSITY=1.229)
     PARAMETER (VISCOSITY=1.789E-5)
C=============================================================



     WRITE(*,*) ' ENTER DISTANCE BETWEEN WALLS (1 IF DIMENSIONLESS)'
     READ(*,*) H
     WRITE(*,*) 'ENTER K'
     READ(*,*) K

     WRITE(*,*)
     WRITE(*,*) '(1) ENTER RE'
     WRITE(*,*) '(2) ENTER MOVING WALL VELOCITY'
     READ(*,*) OPT

     IF (OPT.EQ.2) THEN
       WRITE(*,*) 'ENTER VELCITY OF UPPER WALL'
       READ(*,*) VEL
```

```
      RE=DENSITY*VEL/(VISCOSITY)
    ELSE IF (OPT.EQ.1) THEN
      WRITE(*,*) 'ENTER RE BASED ON MOVING WALL SPEED'
      READ(*,*) RE
      WRITE(*,*) 'GENERATE SPEED'
      VEL=RE*VISCOSITY/(DENSITY)
    END IF

    WRITE(*,*) 'ENTER NUMBER OF POINTS'
    READ(*,*) N
    DY=N
    DY=1/DY


C    CALCULATE THE NUMBER OF NECESSARY MESH POINTS

C    DIV=(1.0/DY)
C    N=IDNINT(DIV)
    CF = 2/RE              ! ASSUMING LINEAR, INITIAL GUESS
    G=SQRT(CF/2.0)
    KK=K*K                 ! KK= K SQUARE

    WRITE(*,*) '-----------------PARAMETERS---------------'
    WRITE(*,*)
    WRITE(*,*) 'STEP: ',DY
    WRITE(*,*) 'CF: ', CF
    WRITE(*,*) 'K: ', K
    WRITE(*,*) 'GAMMA: ', G
    WRITE(*,*) 'N: ', N
    WRITE(*,*) 'REYNOLDS NUMBER: ', RE , 'WALL VELOCITY', VEL
    WRITE(*,*)
    WRITE(*,*) '-------------------------------------------'
    WRITE(*,*)      'PRESS ENTER TO START'
    READ(*,*)

C    ALLOCATING MEMORY

    ALLOCATE (ZPAST(0:N))
    ALLOCATE (A(0:N))
    ALLOCATE (B(0:N))
    ALLOCATE (C(0:N))
    ALLOCATE (AA(0:N))
    ALLOCATE (BB(0:N))
    ALLOCATE (CC(0:N))
    ALLOCATE (AM(0:N))
    ALLOCATE (BM(0:N))
    ALLOCATE (CM(0:N))
    ALLOCATE (AP(0:N))
    ALLOCATE (BP(0:N))
    ALLOCATE (CP(0:N))
```

```
      ALLOCATE (DIFF(0:N))
      ALLOCATE (DZDY(0:N))
      ALLOCATE (ALFA(0:N))
      ALLOCATE (BETA(0:N))
      ALLOCATE (GAMA(0:N))
      ALLOCATE (DELTA(0:N))
      ALLOCATE (X(0:N))
      ALLOCATE (U(0:N))
      ALLOCATE (R(0:N))
      ALLOCATE (Z(0:N))
      ALLOCATE (Y(0:N))
      ALLOCATE (GAM(0:N))
      ALLOCATE (DZDYPAST(0:N))
      ALLOCATE (TV(0:N))

      WRITE(*,*)'MATRICES ALLOCATED'
      WRITE(*,*)

C     CALCULATE INITIAL SOLUTION ASSUMING LINEAR

      DO M=0,N
        MM=M
        Y(M)=MM*DY
C     WRITE(*,*) 'Y=', Y(M), 'DY=', DY
        Z(M)=((VEL/H)*Y(M)*H)/VEL
C     WRITE(*,*) 'Z=', Z(M)
C     READ(*,*)

C     Z(M)=(1/K)*LOG(Y(M)*RE*0.004+0.01)+5.2

      END DO

      WRITE(*,*) Z(N),Z(0),Y(0),Y(N)


      ZPAST=2.0               ! MEANINGLESS VALUE TO PAST Z FOR 1ST ITERATION
            CFPAST=0.1

            WRITE(*,*)'INITIAL SOLUTION GUESSED'

            WRITE(*,*) 'STARTING MAIN ITERATION'



C=====================================================================
                  ! MAIN ITETERATION LOOP
C=====================================================================

            DO T=1,10000000
```

```
C     SHOW ITERATION PROGRESS AT CERTAIN INTERVALS

      IF ( (((REAL(T))/160)-(T/160)).EQ.0) THEN
        WRITE(*,*) T
        OPEN (88,FILE='MIDDLE.DAT')
             DO JJ=0,N
         WRITE (88,*)Y(JJ),Z(JJ)
       END DO
             CLOSE (88)
      END IF

      DZDY(0)=((Z(1))-(Z(0)))/(DY)


      DO I=1,N
        DZDY(I)=(Z(I)-Z(I-1))/(DY) ! DYDY IS ACTUALLY DZDY
      END DO



C     RE CALCULATE CF AND THEREFORE GAMMA (G)

      DUDYWALL=(((Z(N)*VEL)-Z(N-1)*VEL)/(DY*H))

                   !CFPAST=CF

      CF=(2.0*DUDYWALL*H*H*DENSITY)/(RE*RE*VISCOSITY)


      G=SQRT(ABS(CF)/2.0)

      CALL TURBULENTV(Y,KK,RE,G,TV,N,H,DZDY,VEL)
      DO I=0,N

        AP(I)= TV(I+1)+(1/RE)
        AM(I)= TV(I-1)+(1/RE)
        A(I)=  TV(I)+(1/RE)


      END DO


      I=0

      DO I=1,N-1

        AA(I)=((0.5*(A(I)+AM(I)))) !*DZDY(I) !*DZDYPAST(I)
        CC(I)= ((0.5*(AP(I)+A(I)))) !*DZDY(I) !*DZDYPAST(I)
        BB(I)=-AA(I)-CC(I)
```

```
      END DO

      AA(0)=0.0
      BB(0)=1.0
      CC(0)=0.0
      AA(N)=0.0
      BB(N)=1.0
      CC(N)=0.0


      R=0.0
      R(N)=1              !VEL
      R(0)=0.0

C     CALL SOLVING RUTINE

      CALL SOLVE(AA,BB,CC,R,Z,N)

C     CHECK CONVERGENCE


      IF (ABS(CF-CFPAST).LT.EPSILON) THEN
       CALL CONVERGE(N,Y,Z,ZPAST,DIFF, EPSILON,T,G,RE,H,VEL,CF)
      END IF

C     BACK UP MATRIX FROM PREVIUS ITERATION (OR INITIAL SOLUTON IN FIRST LOOP)

      ZPAST=Z
      CFPAST=CF

    END DO                              !END MAIN ITERATION LOOP
C=============================================================

    WRITE(*,*) 'SOLUTION DID NOT CONVERGE IN', T,'ITERATIONS'
    READ(*,*)
    STOP
    END


C=========================SUBROUTINES=======================

    SUBROUTINE CONVERGE(N,Y,Z,ZPAST,DIFF,EPSILON,T,G,RE,H,VEL,CF)
    IMPLICIT NONE


    INTEGER :: N,T
    INTEGER J,I      ,JJ
    REAL*8 :: EPSILON,G,RE,H,VEL,CF
    REAL*8 :: Z(0:N)
```

```
REAL*8 :: ZPAST(0:N)
REAL*8 :: DIFF(0:N)
REAL*8 :: Y(0:N)

DO J=1,N-1
  DIFF(J)=ABS(ZPAST(J)-Z(J))
END DO

IF (MAXVAL(DIFF).LT.EPSILON) THEN

  OPEN (10,FILE='SOLUTION.DAT')
              ! OPEN (33, FILE='RESUME.DAT')
  OPEN (22,FILE='SOLUTION_Z_ETA.DAT')
  OPEN (44, FILE = 'LOG2.TXT')
  WRITE(44,*)  'NUMBER OF ITERATIONS TO CONVERGE ', T
  WRITE(44,*)  'REYNOLDS NUMBER ', RE
  WRITE(44,*)  'NUMBER OF POINTS ', N
  WRITE(44,*)  'WALL SPEED', VEL
  WRITE(44,*)  'LAST CF ',CF, G
  CLOSE(44)



  WRITE(*,*)
  WRITE(*,*)'SOLUTION CONVERGED IN ',T,'ITERATIONS'
  WRITE(*,*)
  DO JJ=0,N
    WRITE (10,*)Y(JJ)*G*RE,Z(JJ)*(SQRT(2.0/ABS(CF))) !/VEL  ! WRITE Y+, U+
    WRITE (22,*)Y(JJ),Z(JJ)
  END DO

  CLOSE(10)
  CLOSE(22)
  WRITE(*,*)'SOLUTION WRITEN TO FILE'
  WRITE(*,*) 'FINAL GAMMA = ',G
  STOP
END IF

RETURN
END

SUBROUTINE TURBULENTV(Y,KK,RE,G,TV,N,H,DZDY,VEL)
IMPLICIT NONE
INTEGER ::N
INTEGER I
REAL*8 :: Y(0:N)
REAL*8 :: TV(0:N)
REAL*8 :: DZDY(0:N)

REAL*8 :: KK,RE,G,H,VEL
```

```
      DO I=1,N-1
        IF (1.0-Y(I).GE.Y(I))THEN

          TV(I)=(KK*(Y(I)*H)**2.0)*
          +         ((1.0-(EXP(-RE*G*Y(I)/26)) )**2.0) *DZDY(I) ! *VEL !/H !**2

        ELSE

          TV(I)=(KK*(1.0-Y(I)*H)**2.0)*
     +         ((1.0-(EXP(-RE*G*(1.0-Y(I))/26)))**2.0) *DZDY(I)
                ! 1-Y BECAUSE IT IS DIMENSIONLESS

        END IF
      END DO

      RETURN
      END

C     THE VECTORS A,B,C ARE THE VECTORS OF THE DIAGONAL AND TWO OFF-DIAGONALS.
C     THE VECTOR R IS THE RHS VECTOR, THE VECTOR U IS THE SOLUTION VECTOR, AND
C     N IS THE DIMENSION.
      SUBROUTINE SOLVE(A,B,C,R,U,N)
      IMPLICIT NONE
      INTEGER :: N
      INTEGER J
      REAL*8 :: A(0:N)
      REAL*8 :: B(0:N)
      REAL*8 :: C(0:N)
      REAL*8 :: R(0:N)
      REAL*8 :: U(0:N)
      REAL*8 BET
      REAL*8 :: GAM(0:N)

      IF(B(0).EQ.0.)PAUSE 'TRIDAG: REWRITE EQUATIONS'
      BET=B(0)
      U(0)=R(0)/BET
      DO 11 J=1,N

        GAM(J)=C(J-1)/BET
        BET=B(J)-A(J)*GAM(J)
        IF(BET.EQ.0.)PAUSE 'TRIDAG FAILED'
        U(J)=(R(J)-A(J)*U(J-1))/BET

 11   CONTINUE
      DO 12 J=N-1,0,-1

        U(J)=U(J)-GAM(J+1)*U(J+1)
 12   CONTINUE
      RETURN
      END
```