

INSTABILITY AND ITS CONTROL IN PARALLEL FLOW

**M. Sc. Thesis by
Oğuzhan SELÇUK, B. Sc.**

Department : Aeronautical Engineering

Programme: Aeronautical Engineering

JANUARY 2003

INSTABILITY AND ITS CONTROL IN PARALLEL FLOW

**M. Sc. Thesis by
Oğuzhan SELÇUK, B. Sc.**

(511981003)

Date of submission : 15 February 2003

Date of defence examination: 15 January 2003

Supervisor (Chairman) : Doç. Dr. İbrahim ÖZKOL

Members of the Examining Committee: Doç. Dr. M. Orhan KAYA

Doç. Dr. Erol UZAL

OCAK 2003

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

PARALEL AKIŞTA İNSTABİLİTİ VE KONTROLÜ

MASTIR TEZİ

Müh. Oğuzhan SELÇUK

Tezin Enstitüye Verildiği Tarih : 16 Şubat 2003

Tez Danışmanı : Doç. Dr. İbrahim ÖZKOL

Diğer Jüri Üyeleri: Doç. Dr. M. Orhan KAYA

Doç. Dr. Erol UZAL

OCAK 2003

ACKNOWLEDGEMENTS

Instability is an important problem because of increasing the drag force that cause more energy consume. For this reason the control of the instability is a major problem and control the flow are seen to be the most suitable option to solve this problem.

I would like to thank to my supervisor Assoc. Dr. İbrahim Özkol for allowing me to work together in this exciting subject. I gratefully acknowledge his support and assistance during preparation.

Lastly, I must thank to Prof. Dr. Elburus CAFEROV and Assoc. Dr. Muammer KALYON. They have brought me up and helped me in every subject and at every time they can.

January, 2003

Oğuzhan SELÇUK

CONTENTS

LIST OF ABBREVIATIONS	v
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF SYMBOLS	viii
SUMMARY	x
ÖZET	xi
1. INTRODUCTION	1
1.1. Description of the Problem	2
1.2. Linear Stability	4
1.3. Non-Linear Stability	7
1.4. Linear Stability: One-Dimensional Case	7
1.5. Linear Stability: Two-Dimensional Case	8
2. SOLUTION OFF THE ORR-SOMMERFEL EQUATION	9
2.1. Numerical Approaches	9
2.2. Method	10
2.3. Orr-Sommerfeld Problem	15
2.4 Small Disturbance Stability	17
2.5 Stability Curve Analyses	20
3. CONTROL OF TRANSITION	23
3.1. The Theory of Bunday Layer	24
3.2. Development of Transition	25
3.3. Control Algorithm for Laminar Plane Channel Flows	26
3.5. Continuous Form of Laminar Flow Equations	26
3.5. Discrete Form of Laminar Flow Equations	28
3.6. Wall Measurements Approximation	30
4. DISCUSSION	33
REFERENCES	35
APPENDIX-A	37
APPENDIX-B	42
APPENDIX-C	47
BIOGRAPHY	55

LIST OF ABBREVIATIONS

EHD	: Elektrohydrodynamic
N-S	: Navier-Stokes
O-S	: Orr-Sommerfeld
LTI	: Linear Time Invariant
T-S	: Tollmien-Schlichting

LIST OF TABLES

	<u>Page No</u>
Table 2.1 : Orr-Sommerfeld results for $\alpha = 1$, $Re = 2500$	16
Table 2.2 : Orr-Sommerfeld results for $\alpha = 1$, $Re = 10000$	16
Table 2.3 : Orr-Sommerfeld results for $\alpha = 1$, $Re = 100000$	17

LIST OF FIGURES

	<u>Page No</u>
Figure 1.1 :Geometry of the problem.....	2
Figure 2.1 :Relative stability of a ball at rest.....	17
Figure 2.2 :Stability curve for plane Poiseuille flow for $ci = 0$	21
Figure 2.3 :Stability curve for plane Poiseuille flow for $ci = 0.003$	21
Figure 2.4 :Stability curve for plane Poiseuille flow for $ci = 0.004$	21
Figure 2.5 :Stability curve for plane Poiseuille flow for $ci = 0.006$	21
Figure 2.6 :Stability curve for plane Poiseuille flow for $ci = 0.007$	21
Figure 2.7 :Stability curve for plane Poiseuille flow for $ci = 0.0076$	22
Figure 2.8 :Stability diagram for plane Poiseuille flow.....	22
Figure 2.9 :Instability regions for different nondimension pressure gradients.	22
Figure A.1 :The starting approximations p_0 , p_1 , and p_2 for Muller's method and the differences h_0 and h_1	36
Figure A.2 :The geometric construction of p_2 for the secont method.....	39
Figure C.1 :Estimated initial interval area for $ci = 0$	48
Figure C.2 :Stability Diagram for plane Poiseuille flow (After Shen(1954))	51
Figure C.3 :Stability Diagram for plane Poiseuille flow.....	53

LIST OF SYMBOLS

Re	: Reynold numbers
p	: Pressure
ρ	: Density
ν	: Kinematic vizkozity
t	: Time
x, y, z	: Pozition vector
u	: Streamwise velocity
w	: Spanwise velocity
U	: Undisturbed stream velocity in the x direction
$\tilde{u}_0, \tilde{p}_0$	: Small perturbations
$\tilde{u}$	: Velocity perturbation in streamwise
$\tilde{w}$	: Velocity perturbation in spanwise
$\tilde{p}$	: Pressure perturbation
α	: Real wave number in the x direction
c	: Complex phase speed.
s	: Stream function.
ϕ	: Amplitude function
$\tilde{s}$	: Small disturbance on stream function
γ	: Scalar parameter
∂	: Estiamtion parameter
λ	: Control parameter
$\mathbf{w}$	: Disturbance
$\mathbf{x}$	: State vector
$\mathbf{u}$	: Control
$\mathbf{L}$	:Esrimator feedback matrix
$\mathbf{K}$	: Controller feedback matrix
δ	: Channel half width
U_0	: Maximum velocity
v	: Perturbation in y direction
∇	: Laplacian
$\mathbf{w}$	: Vorticity
k_x, k_z	: Wave number pair
v_i	: Normal velocity fluctuation
$\mathbf{w}_i$	: Normal vorticity fluctuation
y_m	: Measurement

Üniversitesi	: İstanbul Teknik Üniversitesi
Enstitüsü	: Fen Bilimleri
Anabilim Dalı	: Uçak Mühendisliği
Programı	: Uçak Mühendisliği
Tez Danışmanı	: Doç. Dr. İbrahim ÖZKOL
Tez Türü ve Tarihi	: Mastır – Ocak 2003

ÖZET

PARALEL AKIŞTA İNSTABİLİTİ VE KONTROLÜ

Oğuzhan SELÇUK

Bu çalışmada, Poiseuille akışı incelendi, lineerstabiliti analizi yardımıyla Orr-Sommerfeld denklemi çıkartıldı. Bu denklem Runge-Kutta yöntemi ile çözülebilecek konuma getirilerek Matlab.6.'da hazırlanan programla Runge-Kutta yöntemini kullanarak çözüldü. Kritik Reynold sayısı değeri ve stabiliti eğrileri $ci = 0$, $ci = 0.0003$, $ci = 0.0004$, $ci = 0.0007$, $ci = 0.00076$ için elde edildi. Akışın kontrolünün nasıl yapılacağıyla ilintili yaklaşımlar verildi, optimal ve robust kontrol için gereken kontrol matrisi ve diğer matrisler teorik olarak elde edildi.

Anahtar Kelimeler: Orr-Sommerfeld denklemi, instabiliti, paralel akış, paralel akış kontrolü.

Bilim Dalı Sayısal Kodu: 618

University	: İstanbul Technical University
Institute	: Institute of Science and Technology
Science Programme	: Aeronautical Engineering
Programme	: Aeronautical Engineering
Supervisor	: Assoc. Dr. İbrahim ÖZKOL
Degree Awarded and Date	: M.Sc– January 2003

ABSTRACT

INSTABILITY AND ITS CONTROL IN PARALLEL FLOW

Oğuzhan SELÇUK

In this study, the Poiseuille flow analysed, the Orr-Sommerfeld equation obtained with the aid of linear stability analyses. This equation converted the suitable form for using Runge-Kutta method and solved with Matlab.6. program using the Runge-Kutta method. Critical Reynold number and stability curve for $ci = 0$, $ci = 0.0003$, $ci = 0.0004$, $ci = 0.0007$, $ci = 0.00076$ obtained. The approximation of the flow control is given and the theoretical matrices obtained for the optimal and robust control.

Keywords: Orr-Sommerfeld equation, instability, parallel flow , parallel flow control.

Science Code: 618

1. INTRODUCTION

Simple flows are extremely useful from a numerical standpoint. They play an important role, both as test cases during the development of new algorithms and as debugging tools in the construction of new codes or the updating of old ones. Among these simple flows, the most important might very well be the two-dimensional stationary laminar flow between parallel walls known as the Poiseuille flow. The geometry and boundary conditions are extremely simple. A parabolic velocity profile and constant pressure gradient solution of the analytic problem exist, for all values of the Reynolds number Re . However simple it may be, that flow has many important properties that can be useful for the numericist, and which raise a number of intriguing theoretical questions, in particular from the stability point of view. Here again, the simplicity of the geometry allows for an almost complete analysis without the use of powerful computers.

In fluid dynamics, the role of hydrodynamic stability, which focuses on the evolution with time of small disturbances of permanent flow, is of paramount importance due to the wide range of problems arising from tremendous engineering applications in many fields. One such field is climate modeling with questions like determining an explanation for the origin of the mid-latitude cyclone which in turn is responsible for producing the high and low pressure regions from which variable weather patterns arise. Another application is to shear flows in electrohydrodynamic (EHD) systems which have industrial relevance in the invention of devices employing the electroviscous effect or those utilizing charge entrainment, such as EHD clutch development, or EHD high voltage generators. Yet other important mundane applications include the prediction of landslides, and flow over an aeroplane wing covered in de-icer. The term stable can be defined precisely in terms of those disturbances: if they ultimately decay to zero, the flow is said stable, whereas if any of them remains permanently different from zero, it is unstable. The study of their evolution can follow at least two roads, depending on whether the governing

Navier-Stokes (N-S) equations have been linearized or not: we can conduct a linear or non linear stability analysis. It is well known from a mathematical standpoint that steady-state solutions to the N-S equations exist for large values of the Reynolds number Re . It is also known that for small Re the stationary solution is unique and it is of great interest, not only in the Poiseuille case for that matter, to determine for which value of Re that basic flow loses its stability. We will denote that critical value by Re_{cr} . For Reynolds numbers higher than Re_{cr} , a number of scenarios are possible leading to the steady of transition. Surprisingly enough, the Poiseuille flow exhibits rather complex transition at high Reynolds numbers.

1.1 Description of the Problem

We restrict our attention to the case of an arbitrary two-dimensional, steady, incompressible, Newtonian, viscous fluid, shear flows between two fixed, parallel plates (Figure 1.1). The dimensionless equations that govern the motion of the fluid are thus given by [1]

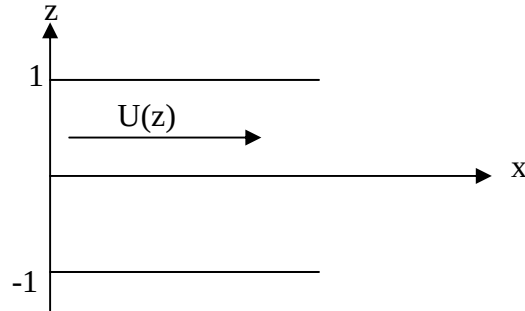


Figure 1.1 Geometry of the problem

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0, \quad (1.1)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left\{ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right\}, \quad (1.2)$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\frac{\partial p}{\partial z} + \frac{1}{Re} \left\{ \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial z^2} \right\}, \quad (1.3)$$

where (x, z) is the position vector of an arbitrary point in the flow domain, t is the time, u and w are the velocity and p the pressure. Assuming the channel to be sufficiently ‘long’ (so that variations of the velocity field in the streamwise direction

may be neglected), an arbitrary two-dimensional, steady, incompressible flow field is given by $U = (U(z), 0, 0)$, $P = P(x)$, where U can be at most quadratic in z for viscous flows. It is easy to see that a steady state solution exists for all values of Re which is given by

$$U(z) = 1 - z^2, \quad P(x) = -\frac{1}{Re}x \quad (1.4)$$

We are interested in the stability of this solution as the Reynolds number increases. Physically this can be interpreted as whether or not the steady state solution can be observed. To investigate the instability, we first construct a laminar steady solution (U, P) to equations (1.1)-(1.3). Then the turbulent flow is imposed on the basic flow U, P as a small disturbance. The basic idea is that a solution can be observed only if it is not sensitive to small perturbations. We thus suppose that at some initial moment, a small perturbation $(\tilde{u}_0, \tilde{p}_0)$ is superposed to the laminar solution (1.4). This induces a perturbation, which is a function of both time and space, $\tilde{u} = (\tilde{u}(x, z, t), 0, \tilde{w}(x, z, t))$, $\tilde{p} = \tilde{p}(x, z, t)$; for all time $t > 0$. Stability deals with the evolution of $(\tilde{u}, \tilde{p})$ with time. It is thus necessary to obtain a set of equations describing this evolution. Replacing (u, p) by

$$u = U + \varepsilon \tilde{u}, \quad p = P + \varepsilon \tilde{p}, \quad \text{with } 0 < \varepsilon \ll 1 \quad (1.5)$$

the motion is now by the above components. What is assumed now, is that while U and P are solutions of the Navier Stokes equations, described in the N-S system (1.1)-(1.3), then the above resultant flow also satisfies the Navier Stokes equations. Also it is supposed that these disturbances are small, such that their quadratic and higher order terms may be neglected. Upon substitution, of the above into the Navier Stokes equations and noting the fact that the mean flow is also satisfies the Navier Stokes equations, (U, P) is a steady state solution, we may arrive at the following set of equations which is the initial value problem:

$$\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{u}}{\partial z} = 0, \quad (1.6)$$

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{u} \frac{\partial U}{\partial z} + (U + \tilde{u}) \left(\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{u}}{\partial z} \right) = -\frac{\partial \tilde{p}}{\partial x} + \frac{1}{\text{Re}} \left\{ \frac{\partial^2 \tilde{u}}{\partial x^2} + \frac{\partial^2 \tilde{u}}{\partial z^2} \right\}, \quad (1.7)$$

$$\tilde{u}(x, z, 0) = u_0(x, z, 0) - U = \tilde{u}_0(x, z, 0) \quad (1.8)$$

We have three equations for our three unknowns $\tilde{u}$, $\tilde{w}$ and $\tilde{p}$, and we also make use of the no slip boundary condition. Problem (1.6)-(1.8) represents the mathematical problem of hydrodynamic stability.

1.2 Linear Stability

In the bid to understand the transition process, a theory based on the effects of small disturbance to laminar flow became established. The idea being that small disturbances originating at the inlet to the channel may somehow be a factor in the transition process. What was supposed was the superimpose such small disturbances upon a main flow, and to see if these small perturbations should amplify or diminish with time. Should they diminish the flow may be considered 'stable', and unstable should they grow. Reynolds had hypothesized similarly, and much mathematical work ensued, by the likes of Rayleigh in the following decades. In 1930 Prandtl had formed a method for prediction the critical Reynolds number, and experimental evidence for stability came some ten years later with improved low disturbance wind tunnels.

Linear stability deals only with disturbances of a particular type, the form of which is suggested by experiments. In this section, we follow the approach of Georgescu[2] to get to the Orr-Sommerfeld equation. Neglecting the nonlinear term in (1.7) the mathematical problem of linear hydrodynamic stability becomes:

$$\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{u}}{\partial z} = 0, \quad (1.9)$$

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{u} \frac{\partial U}{\partial z} + U \left(\frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{u}}{\partial z} \right) = -\frac{\partial \tilde{p}}{\partial x} + \frac{1}{\text{Re}} \left\{ \frac{\partial^2 \tilde{u}}{\partial x^2} + \frac{\partial^2 \tilde{u}}{\partial z^2} \right\}, \quad (1.10)$$

$$\tilde{u}(x, z, 0) = \tilde{u}_0(x, z, 0) \quad (1.11)$$

It is assumed [3] that any perturbation $\tilde{u}$ can be obtained by the superposition of some perturbations of the form

$$\tilde{u}(x, z) = \tilde{u}_0(x, z, 0) e^{\sigma t}, \quad (1.12)$$

called normal modes. In the case of unbounded domains in one direction the perturbations are assumed to be periodic along this direction and we assume a two-dimensional disturbance for which the z component of the perturbation velocity is proportional to the real part of the expression with α real. They are then of the form

$$\tilde{u}(x, z) = \hat{u}_0(z) e^{(i\alpha x + \sigma t)} = \hat{u}_0(z) e^{i\alpha(x + \frac{\sigma}{i\alpha}t)} \quad (1.13)$$

and called transversal Tollmien-Schlichting (T-S) waves. Here α is a wave number in the x -direction ($\alpha = \frac{2\pi}{L}$), L is the wave length (length of the domain) and σ , $\tilde{u}_0$ and $\tilde{p}$ are solutions of the following eigenvalue problem:

$$\frac{\partial \tilde{u}_0}{\partial x} + \frac{\partial \tilde{u}_0}{\partial z} = 0, \quad (1.14)$$

$$\sigma \tilde{u}_0 + \tilde{u}_0 \frac{\partial U}{\partial z} + U \left(\frac{\partial \tilde{u}_0}{\partial x} + \frac{\partial \tilde{u}_0}{\partial z} \right) = -\frac{\partial \tilde{p}}{\partial x} + \frac{1}{\text{Re}} \left\{ \frac{\partial^2 \tilde{u}_0}{\partial x^2} + \frac{\partial^2 \tilde{u}_0}{\partial z^2} \right\}, \quad (1.15)$$

which is obtained from (1.9)-(1.11) through the change of the unknown (1.13). Let us set

$$\frac{-\sigma}{i\alpha} = c = c_r + i c_i \quad (1.16)$$

so that

$$\tilde{u}(x, z) = \hat{u}_0(z) e^{i\alpha(x - ct)} \quad (1.17)$$

for any eigenvalue and eigenfunction $(\sigma, \tilde{u}_0)$, the corresponding wave (1.13) will decay to zero if and only if

$$c_i < 0. \quad (1.18)$$

Thus, we have linear stability if it is violated by any of them. The case where the first of the c_i becomes zero is called neutral. Since the perturbation is two dimensional and $\text{rot } u = 0$ [4], we can define the stream function $s = s(x, z, t)$ which represents the disturbance [25] and satisfies $u = \partial s / \partial z$, $w = -\partial s / \partial x$. For a particular oscillation we introduce a small disturbance $\tilde{s}(x, z, t) = \phi(z) e^{i\alpha(x-ct)}$ to the stream function s , where ϕ is the amplitude function of $\tilde{s}$, α is a real wavenumber (if α were complex, spatial instabilities could occur in addition to the time dependent ones) and c is the complex phase speed. Here the theory divides into two lines of thought, that of spatial, and temporal theory [25]. α and c may be considered to be imaginary or real. In the temporal case for example, α may be considered real and we have complex $r = \alpha c = r_r + i r_i$. The imaginary component r_i is the amplification factor, and should it be < 0 , the disturbance will be damped, and the flow will remain laminar, whereas if it > 0 then the flow will become unstable. Namely c_i determines the stability of the disturbance because of the real part of the temporal growth rate of $\tilde{s}$ is $e^{\alpha c_i t}$: if $c_i < 0$, the amplitude of the disturbance decreases with time, hence the perturbation is stable.

Comparing with (1.17), we get

$$\hat{u}_0(z) = \begin{pmatrix} \phi'(z) \\ i\alpha\phi(z) \end{pmatrix}, \quad (1.19)$$

where the $'$ stands for the derivative with respect to z . Setting $\tilde{p}(x, z, t) = p(z) e^{i\alpha(x-ct)}$, the eigenvalue problem (1.14)-(1.15) becomes

$$-i\alpha\phi U' - i\alpha c\phi' + iU\alpha\phi' = -i\alpha\tilde{p} + \frac{1}{\text{Re}}(-\alpha^2\phi' + \phi''') \quad (1.20)$$

$$-\alpha^2 c\phi + \alpha^2 U\phi = -\tilde{p}' + \frac{1}{\text{Re}}(i\alpha^3\phi - i\alpha\phi'') \quad (1.21)$$

Upon elimination of the pressure, neglecting terms of order higher than ε and forming to vorticity equation leads to

$$(U - c)(\phi'' - \alpha^2\phi) - \phi U'' = \frac{1}{i\alpha \text{Re}}(\phi''' - 2\alpha^2\phi'' + \alpha^4\phi), \quad (1.22)$$

We obtain the well known Orr-Sommerfeld (O-S) fourth order ordinary differential equation. In the Poiseuille case the undisturbed stream velocity in the x direction is

$U = (1 - z^2)$ the side walls are at $z = \pm 1$ and equation (1.22) is subjected to the boundary conditions

$$\phi(-1) = \phi'(-1) = \phi(1) = \phi'(1) = 0, \quad (1.23)$$

Therefore, instability of two-dimensional laminar flow can now be discussed in terms of the eigenvalue problem (1.22) and (1.23). The linear stability analysis of the sensitivity of the Poiseuille flow to perturbations of the form of transversal Tollmien-Schlichting waves is equivalent to the eigenvalue problem (1.22),(1.23) where c is the eigenvalue and ϕ the eigenfunction.

1.3 Non-linear Stability

The non-linear stability analysis is based on the direct solution of (1.1)-(1.3) with an initial condition of the form

$$u(x, z, 0) = U + \tilde{u}_0(x, z), \quad (1.24)$$

where U is again the basic flow and $\tilde{u}_0(x, z)$ is a perturbation. The reader will easily convince himself that the resulting problem can also be obtained from (1.6)-(1.8) by setting $u = U + \tilde{u}$. this leads to a system that differs from (1.9)-(1.11) only by the inertial term $u \cdot \nabla u$.

1.4 Linear Stability: One-Dimensional Case

In the preceding section, we have shown that the linear stability analysis of the Poiseuille flow with respect to an infinitesimal two dimensional disturbance, is equivalent to the O-S eigenvalue problem

$$(U - c)(\phi'' - \alpha^2 \phi) - \phi U'' = \frac{1}{i\alpha \text{Re}} (\phi'''' - 2\alpha^2 \phi'' + \alpha^4 \phi), \quad (1.25)$$

$$\phi(-1) = \phi'(-1) = \phi(1) = \phi'(1) = 0, \quad (1.26)$$

The first attempt to obtain a numerical solution of (1.25), (1.26) is due to Thomas[5]. He successfully used a finite differences scheme to tackle the numerical difficulty arising from the sharp boundary layer near the channel walls. Orszag [6] solved this problem by using an expansion in Chebyshev polynomials, and obtained

$Re_{cr} = 5772.22$ for $\alpha = 1.020545$. This is the smallest Reynolds number for which linear instability occurs. By varying the value of α . One can obtain the classical linear stability curve (see Figure 1.2)

We have also solved the O-S equation in order to determine critical Reynolds number and the corresponding eigenfunctions ϕ which will be useful for the construction of initial solutions for the two-dimensional case of the following section. But we choose the different method that explained following section.

1.5 Linear stability: Two-Dimensional Case

In this section we carry out two-dimensional numerical calculations for the eigenvalue problem (1.14)-(1.15) in which U is given by (1.4) and the geometry is that of Figure 1.1.

2. SOLUTION OF THE ORR-SOMMERFELD EQUATION

The Orr-Sommerfeld equation occurs in hydrodynamic stability theory [7]; it governs the stability of subsonic shear flows of viscous, Newtonian incompressible fluids, whose velocity field u satisfies $\text{rot } u = 0$ (thus we have a potential flow). These flow may exist under various conditions, for instance, flows in a pipe or channel, flows of superposed immiscible fluids, wakes, jets, plumes, and free-streams in general. These flows may be laminar or turbulent and the transition from the former to the latter is essentially the above mentioned instability.

The general approach is to construct a laminar flow solution to the governing differential equations and boundary conditions and then superimpose the turbulent flow as small variations of the laminar flow. In the context of linearized modelling equations, the Orr-Sommerfeld equation is the from which a mathematical analysis of flow instabilities starts. The task is to determine the complex eigenvalues of the Orr-Sommerfeld equation, because, as explained section 2.1, linear instability occurs when one of the the real part of the temporal growth rate of the disturbances c_i 's becomes positive. Since $\sigma = -i\alpha c$, instability will correspond to an eigenvalue σ crossing the imaginary axis from right to left. When all eigenvalues have negative real part, the least stable mode corresponds to the σ nearest the imaginary axis.

Orr-Sommerfeld problem is in fact quite a difficult problem to solve, taking about 20 years, after it was first derived by Orr (1907) and Sommerfeld (1908) to yield solutions, for the Blasius boundary layer by Tollmien (1929) and Schlichting (1933) [26-30]. With a specified mean flow U there exist the four parameters α , c_i , c_r and Re .

2.1 Numerical Approaches

Early on, one is forced to employ numerical techniques in solving the Orr-Sommerfeld eigenvalue. In earlier work by Gersting [8] and Gersting and Jankowski [9] various techniques for computational solution of stiff differential eigensystems, in particular the Orr-Sommerfeld equation, were evaluated. Work of Ng and Reid [10] calls for a significant revision of the conclusions in the earlier work [8,9]. What follows is a presentation of the method proposed by Ng and Reid and comparisons of the computational efficiency of this method with other methods using the example

stiff eigensystem proposed in [8]. Results for solution of Orr-Sommerfeld equation are also presented.

2.2 Method

Consider the fourth order linear ordinary differential equation

$$L(u) = u^{(4)} + a_3 u^{(3)} + a_2 u'' + a_1 u' + a_0 u = 0 \quad (2.1)$$

with the homogeneous boundary conditions

$$u(0) = u'(0) = 0 \text{ and } u(1) = u'(1) = 0. \quad (2.2)$$

In the usual way define

$$u = (u, u', u'', u''')^T, \quad (2.3)$$

then (2.1) may be written in matrix form as

$$u' = Au \quad (2.4)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 \end{bmatrix}$$

Consider a linear combination of two linearly independent solutions of (2.4) which satisfy the initial conditions $u(0) = u'(0) = 0$, and construct the solution

$$u = u_1 \beta_1 + u_2 \beta_2 = U\beta \quad (2.5)$$

where $\beta = (\beta_1, \beta_2)^T$ and

$$U = \begin{bmatrix} u_1 & u_2 \\ u_1' & u_2' \\ u_1'' & u_2'' \\ u_1''' & u_2''' \end{bmatrix} \quad (2.6)$$

Then (2.1) becomes

$$U' = AU \quad (2.7)$$

With initial conditions

$$U(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.8)$$

As suggested in [8] and [9] the next step is to integrate the two initial value problems (2.6) while maintaining the linear independence of the solution vectors u_1 and u_2 by using an orthonormalization process at selected points in the interval. Iteration is performed to adjust the eigenvalue until the determinant, the requirement that

$u(1) = u'(1) = 0$, becomes zero, that is,

$$\begin{vmatrix} u_1(1) & u_1'(1) \\ u_2(1) & u_2'(1) \end{vmatrix} = 0 \quad (2.9)$$

A simplified explanation of the above process is that two vectors u_1 and u_2 are constructed and they establish the plane in which the solution u must lie. Orthonormalizations are performing during the integration to maintain the linear independence of u_1 and u_2 . At the end of the interval the constants β_1 and β_2 in (2.5) are determined and the solution u may be reconstructed.

Ng and Reid suggest a method that avoids the orthonormalization and the ‘arithmetic’ evaluation, from (2.9), of the determinant. Continuing the simplified explanation, one way to look at the new method is to note that in the iteration process for the eigenvalue only u_1 and u_2 are used which means only the knowledge of the plane of the solution is necessary. If u_1 and u_2 were three dimensional vectors the plane could easily established by forming the vector cross product $u_1 \times u_2 = Y$. if $Y(x)$ were known across the interval, the plane of u would also be known. This should be enough information to determine the eigenvalue.

Since the vectors u_1 and u_2 are 4-tuples the idea of the cross product must be generalized to the exterior (or wedge) product [11]. The components of the exterior product of the two vectors in (2.6) are:

$$\begin{vmatrix} u_1 & u_2 \\ u_1' & u_2' \\ u_1'' & u_2'' \\ u_1''' & u_2''' \end{vmatrix} \quad (2.10)$$

or

$$t^{ik} = u_1^i u_2^k - u_1^k u_2^i \quad (2.11)$$

where i and k range over the set $\{1-4\}$. In this case t^{ik} produces the six distinct components

$$\begin{aligned} y_1 &= u_1' u_2 - u_1 u_2' \\ y_2 &= u_1'' u_2 - u_1 u_2'' \\ y_3 &= u_1'' u_2' - u_1' u_2'' \\ y_4 &= u_1''' u_2 - u_1 u_2''' \\ y_5 &= u_1''' u_2' - u_1' u_2''' \\ y_6 &= u_1''' u_2'' - u_1'' u_2''' . \end{aligned} \quad (2.12)$$

using (2.12) ,(2.7) may be replaced by the system

$$Y' = BY \quad (2.13)$$

With

$$Y(0) = (0, 0, 0, 0, 1)^T \quad (2.14)$$

where

$$Y(x) = (y_1, y_2, y_3, y_4, y_5, y_6)^T$$

and

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -a_1 & -a_2 & 0 & -a_3 & 1 & 0 \\ a_0 & 0 & -a_2 & 0 & -a_3 & 1 \\ 0 & a_0 & a_1 & 0 & 0 & -a_3 \end{bmatrix} \quad (2.15)$$

The determinant (2.9) is just

$$y_1(1) = u_1'(1)u_2(1) - u_1(1)u_2'(1) \quad (2.16)$$

Thus (2.13) and (2.16) replace (2.7) and (2.9). The basic superposition technique, (2.7), involves integration of several solutions and, if the system is stiff, parasitic error allows the dominant solution unstable. To allow meaningful evaluation of (2.9). Techniques must be found for coping with parasitic error. Integration of (2.13) is the dominant solution, and yields values directly for (2.16), the eigentest condition, which can be used in an iteration to locate eigenvalues. This solution is stable may involve large growth rates if the system is stiff. Machine arithmetic overflows may be counteracted by choosing appropriately small values of $y_6(0)$ or by periodically scaling the Y vector.

Next consider the determination of the eigenfunction, $u(x)$, for (2.7). Returning to the simplified explanation in 3-dimensional space, at the end of the interval Y specifies the direction normal to the plane of u , so that

$$Y \cdot u = 0 \text{ or}$$

$$(u_1 \times u_2) \cdot u = 0 \quad (2.17)$$

which is the scalar triple product. In fact, since Y is known in the interval from the integration of (2.13), (2.17) can be thought of as a system of equations for the determination of u . Integration can be initiated from the end point of the original interval with initial conditions constructed from the boundary conditions at that point and then proceed back to the original starting point.

In terms of the exterior product for 4-tuple vectors a condition equivalent to (2.17) would be

$$\begin{vmatrix} u_1 & u_2 & u_3 \\ u_1' & u_2' & u_3' \\ u_1'' & u_2'' & u_3'' \\ u_1''' & u_2''' & u_3''' \end{vmatrix} \quad (2.18)$$

$$\begin{aligned} t^{ikl} &= u_1^i u_2^k u_3^l + u_1^l u_2^i u_3^k + u_1^k u_2^l u_3^i - u_1^k u_2^i u_3^l - u_1^i u_2^l u_3^k - u_1^l u_2^k u_3^i \\ &= (u_1^i u_2^k - u_1^k u_2^i) u_3^l + (u_1^k u_2^l - u_1^l u_2^k) u_3^i + (u_1^l u_2^i - u_1^i u_2^l) u_3^k \end{aligned} \quad (2.19)$$

The first form in (2.22) shows the even and odd permutations of i, k, l . The second form is reminiscent of the scalar triple product and also contains factors similar to (2.11). As i, k and l range over the set $\{1-4\}$ four distinct component equations result. Equation (2.17) may now be implemented by choosing u_3 as u in (2.19) and setting the resulting components to zero. Ng and Reid show that all four of the resulting equations are equivalent but some are more tractable computationally than others. After making use of (2.12) two of the equations are

$$y_1 u'' + y_2 u' + y_3 u = 0 \quad (2.20)$$

and

$$y_2 u''' + y_4 u'' + y_6 u = 0 \quad (2.21)$$

The procedure to reconstruct the eigenfunction for this type of problem is to save the values of Y during the forward integration ($x = 0$ to $x = 1$) of (2.13) and to determine $u(x)$ from (2.20) by reverse integration ($x = 1$ to $x = 0$) to avoid growth problems. In this case $y_1(1) = 0$ it is necessary to use (2.21) for one step away from $x = 1$ and then continue the integration using (2.20). Details of the mathematical behavior of (2.19) and (2.20) and a proof that the resulting $u(x)$ is the solution to (2.7) are fairly lengthy may be found in [10].

2.3 Orr-Sommerfeld Problem

One reason for pursuing methods of obtaining eigenvalues for stiff systems is to be able to obtain results for equations like the Orr-Sommerfeld problem. For plane poiseuille flow the Orr-Sommerfeld eigensystem given with (1.22) converted in the form of (2.1). That is

$$\phi'''' + (i\alpha \text{Re}(c - U) - 2\alpha^2)\phi'' + (\alpha^4 - i\alpha \text{Re}((c - U)\alpha^2 - U''))\phi = 0 \quad (2.22)$$

with the boundary conditions

$$\phi(-1) = \phi'(-1) = 0 \text{ and } \phi(1) = \phi'(1) = 0 \quad (2.23)$$

where U is the primary flow, α is the wave number, Re is the Reynolds number and $c = c_r + ic_i$ is the complex eigenvalue.

To implement solution of (2.22) using (2.13) and (2.16) more suitable than using (2.7) and (2.12) because of abnormally terminates due to arithmetic error occurring regardless of the number of steps at the later ones [12]. So, to obtain solution of (2.22) cast in the form of (2.13) and the matrix B must be formed. Since ϕ and c are complex (2.13) and (2.16) will also be complex so that the integration must be complex also.

Method given above for solving (2.23) is from the code given in the Appendix-A and uses (2.16) and Matlab.5. Table1-2-3 below compares various methods for solving (2.23), S (SUPPORT [13], near-orthonormalization, Runge-Kutta-Fehlberg,

$\text{RE} = 10^{-4}$, $\text{AE} = 10^{-4}$), O (ORTNRM [14], near-orthonormalization, Runge-Kutta, 100 steps, $\text{ANG} = 60$ degrees) are from [8] are implementations based on (1.5), ORSZAG and THOMAS are from [2], DONGARRA is from [15], R (400 steps) is from the code given in the Appendix-A and uses (2.16) and Matlab.5. The code given in [16] is transcribed to Matlab.5. The entry marked T is the work of Thomas [5] using finite difference and is included as a standard value for the comparison. In all cases $\alpha = 1$, $\text{Re} = 2500$, initial estimate $c = 0.3231 - 0.0262i$ except R.

Table 2.1 Orr-Sommerfeld results for $\alpha = 1$, $Re = 2500$

Method	$c = c_r + ic_i$	iteration
O	0.301148 - 0.014179I	10
S	0.301150 - 0.014199	15
R	0.301135 - 0.01418	3
THOMAS	0.3011 - 0.0142	

Table 2.2 Orr-Sommerfeld results for $\alpha = 1$, $Re = 10000$

	$c = c_r + ic_i$	Method
ORSZAG		Cheyshev-Tau
	0.23752649 + 0.00373967i	K = 28
R		Runge-Kutta
	0.23752529 + 0.00373996i	M = 1200
THOMAS		FDM
	0.2375006 + 0.0035925i	50 Grid points
	0.2375243 + 0.0037312i	100 Grid points
DONGARRA		D4 method
	0.23752708 - 0.00373980i	K = 50

Table 2.3 Orr-Sommerfeld results for $\alpha = 1$, $Re = 100000$

	$c = c_r + ic_i$
DONGARRA	0.145247829 - 0.0150203085i
R	0.145938489 - 0.0150119924i

Again the code in the Appendix-A was used with the modification that for large Re the solution was scaled by setting $y_6(0)$ to a small value to avoid arithmetic overflow. Initial estimates were taken from Thomas [5] for the first two cases. The iteration tolerance in the form of the relative change in the eigenvalue has set at 0.0001. M is the number of integration steps. Detailed results are also given in Appendix-B.

2.4 Small Disturbance Stability.

The stability and transition are important words in this chapter . The general concept of stability has been discussed many times. The discussion always boils down to one question: can a given physical state withstand a disturbance and still return to its original state? If so, its stable. If not, that particular state is unstable. It is the study of the stability analyst to test the effect of particular disturbance. The simple suitable example is shown in Figure 2.1 [21], where

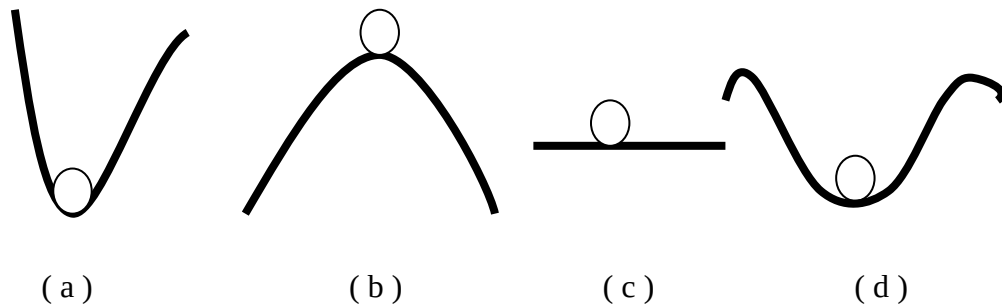


Figure 2.1 Relative stability of a ball at rest: (a) stable; (b) unstable; (c) neutral stability; (d) stable for small disturbances but unstable for large disturbances.

a ball lies at rest under various conditions. In Fig. 2.1-a, its position is unconditionally stable, because it would return to its original position even if disturbed by a large displacement. Conversely, Fig 2.1-b shows an unstable situation, since any slight disturbance would topple the ball, never to return original state. A flat tabletop, as in figure 2.1-c, is an example of neutral stability, since any disturbance applied on the ball, the ball will rest anywhere it is displaced. Finally, 2.1-d illustrate a more complicated case, where, small disturbance applied on the ball, the ball is stable but will diverge or unstable if disturbed far enough to drop over the edge. The boundary layer-flow is an example of this type of case, the otherwise stable laminar flow to become turbulent with a large trip wire. Note that stability requires simply a yes or no answer. One can show that a physical state is unstable without being able to determine to what true stable state the disturbance will lead. In viscous flow, we can show that laminar flow is unstable above certain Reynolds numbers, but that is all: the analysis does not predict turbulent flow. The proper stable state for turbulent flow is an experimentally yielded at high Reynolds numbers. Thus we can discuss only qualitatively our second important word, transition of a laminar flow into a turbulent flow defined as the change, over space and time and a certain Reynolds number range. It is not yet stability theory has been widely accepted, there is still no theory of transition although there is, a modestly successful suitable empirical prediction of transition of the flow based on the spatial amplification rates of the linearized stability theory.

The mechanism of instability in fluid mechanics is one of the topics, which has not been fully understood yet. Since, there are numerous external and internal agents for a flow field to lose its laminar behavior and to find itself in transition to turbulence. Some of which of these agents are free-stream turbulence, sound, pressure gradient, oscillation of the external flow, roughness, suction or blowing, wall curvature. One of these agents itself is required a detailed experimental, analytical or numerical work.

When instability takes place, whatever the origin is, in a fluid flow system, the effective disturbances are responsible for the instability to extract a sufficient amount of kinetic energy from the basic flow that upsets the equilibrium of the forces, which are operative on the basic flow. On the other hand, Laminar-turbulent transition is

extraordinary complicated process, consisting of great number of competing events. The initial is the transformation of external disturbances into internal instability oscillations of the boundary layer, taking the well-known form of Tollmien-Schlichting (T-S) waves. In relatively quiet flows, the initial amplitude of waves is insufficient to provoke immediate transition. (T-S) waves must first amplify in the boundary layer to trigger non-linear effects, which are characteristic of the transition process. The extent of the amplification region and hence the location of the “transition point” on the body surface, is strongly dependent not only on the amplitude and/or the spectrum of external disturbance but also on their physical nature. All the external disturbances are not always sufficient to provoke these (T-S) waves. Once these (T-S) waves are generated by any mechanism, either external or internal, they are first amplified in the boundary layer to trigger non-linear effects, which are the characteristics of the transition process.

Some of the disturbances early penetrate into the boundary layer and turn into (T-S) waves but others do not. For many applications, it might be very interesting to delay this transition towards turbulence and to preserve the flow field laminar. Linear disturbance waves can grow by viscous mechanism if the velocity profile is non-inflectional and both viscous and inviscid instabilities are present for inflectional profiles. When disturbances become non-linear, the resulting dynamical interactions depend upon the type of instability and particularly on whether the critical layer lies within the viscous wall layer or separates from it. It is quite clear that the classification of a wave as viscous or inviscid is important to the theory of transition.

Laminar flows have a fatal weakness: poor resistance to high Reynolds numbers. For any given laminar flow, there is a finite value of its Reynolds number which threatens its very existence. Since this critical Reynolds number, as we shall call it, has only a modest value, being of the order of 1000 when referred to a transverse thickness, it follows that laminar flows are the exception rather than the rule in most engineering situations. At higher Reynolds numbers, the flow is always turbulent, i.e., disorderly,, randomly unsteady, apparently impossible to analyze exactly, but fortunately amenable to study of its average values.

Thus laminar flow is found to be unstable, and its critical Reynolds numbers are of such a magnitude that flows of low-viscosity fluids (water, mercury, ammonia,

gases,...) are normally turbulent, not laminar. Coffee stirred in a cup mixes turbulently. Smoke rises turbulently from chimney. Water in a bathroom shower pipe flows turbulently. The boundary layer on a commercial jet airplane wing is turbulent. Any river worthy of its name flows turbulently. Meanwhile, laminar flows should not be disregarded, because many practical situations arise which are indeed laminar, such as low speed flows, small scale bodies, very viscous fluids, or leading edge problems.

2.5 Stability Curve Analyses

The Orr-Sommerfeld equation is to reduce the problem of stability to that of an eigenvalue problem. With the Reynolds number specified and a particular wave length λ of interest, then the differential equation together with its boundary condition lead to the finding of the eigenvalue c and the eigenfunction ϕ .

Where $c_i = 0$ we have a point of neutral stability, and for all values of Re and $\alpha(A)$ we have the corresponding so called neutral curve in $Re - \alpha$ space. Outside the curve the flow is laminar, and within it, the flow will be unstable. As can be seen the from the following stability diagram, in Figure 2.2, the flow in this case is stable to all disturbances, of any wave number below a certain Re number, see Appendix-C for more information. In the case of a Blasius boundary layer, if $\frac{dp}{dx} = 0$ then the stability curve will close on itself as Re tends to infinity. The wave numbers for points within the neutral stability curve, are a set of unstable waves called Tolmien-Schlichting waves. It should be noted that in the case where $\frac{dp}{dx} < 0$ and thus the velocity profile must have a point of inflection, then the neutral curve will not close entirely back on itself as $Re \rightarrow \infty$. This corresponds with Reyleigh's point of inflection theorem. Stability curves, in Figure 2.2-2.6, yielded the case of the linear Couette profile generated between a fixed and a moving plate with solving of the Orr-Sommerfeld problem $c_i = 0$, $c_i = 0.003$, $c_i = 0.004$, $c_i = 0.006$, $c_i = 0.007$ and $c_i = 0.0076$, respectively, and also stability diagram for plane Poiseuille flow was given by the super position of these stability curves in Figure 2.8. This diagram approximately the same Figure in 2.9, given [22].

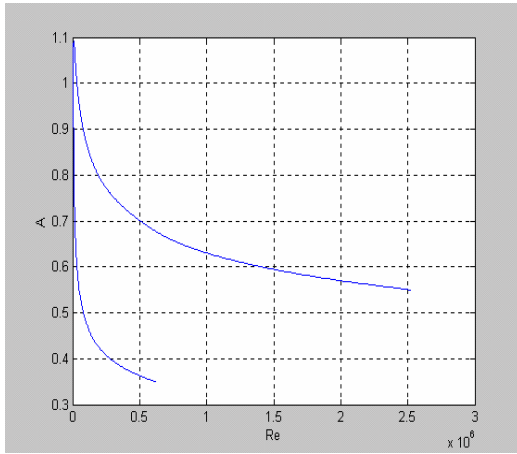


Figure 2.2 Stability curve for plane Poiseuille flow for $ci = 0$.

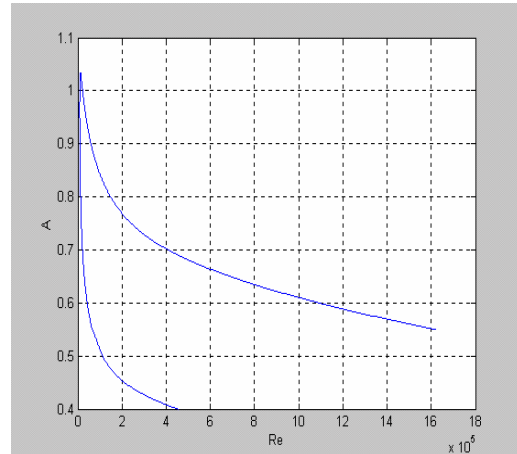


Figure 2.3 Stability curve for plane Poiseuille flow for $ci = 0.003$

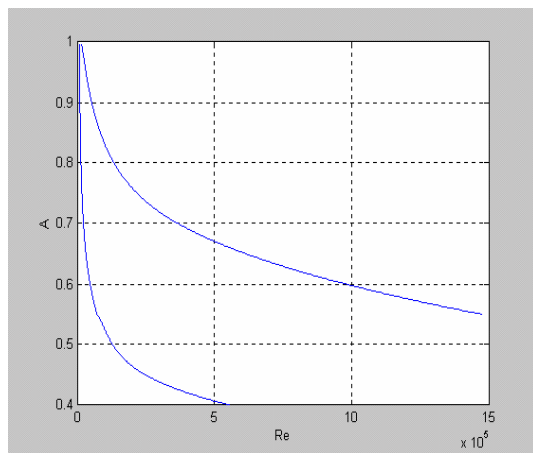


Figure 2.4 Stability curve for plane Poiseuille flow for $ci = 0.004$

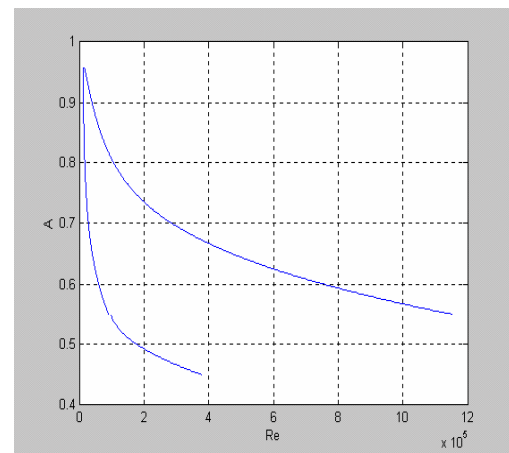


Figure 2.5 Stability diagram for plane Poiseuille flow for $ci = 0.006$

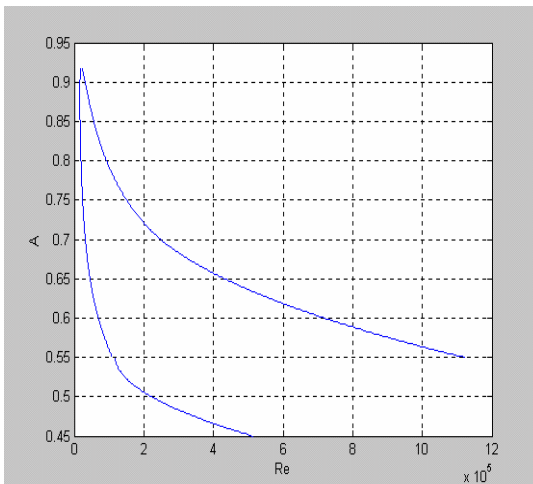


Figure 2.6 Stability diagram for plane Poiseuille flow for $ci = 0.007$

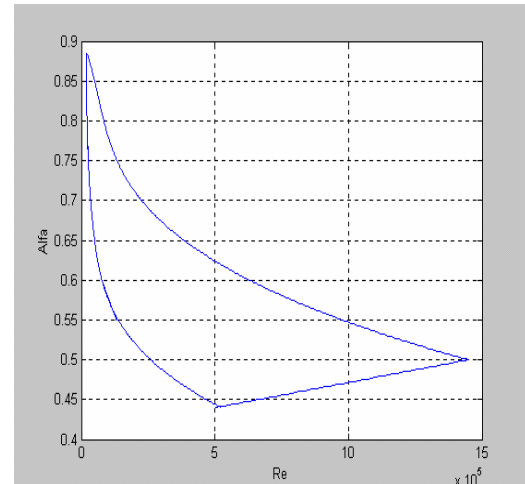


Figure 2.7 Stability diagram for plane Poiseuille flow for $ci = 0.0076$

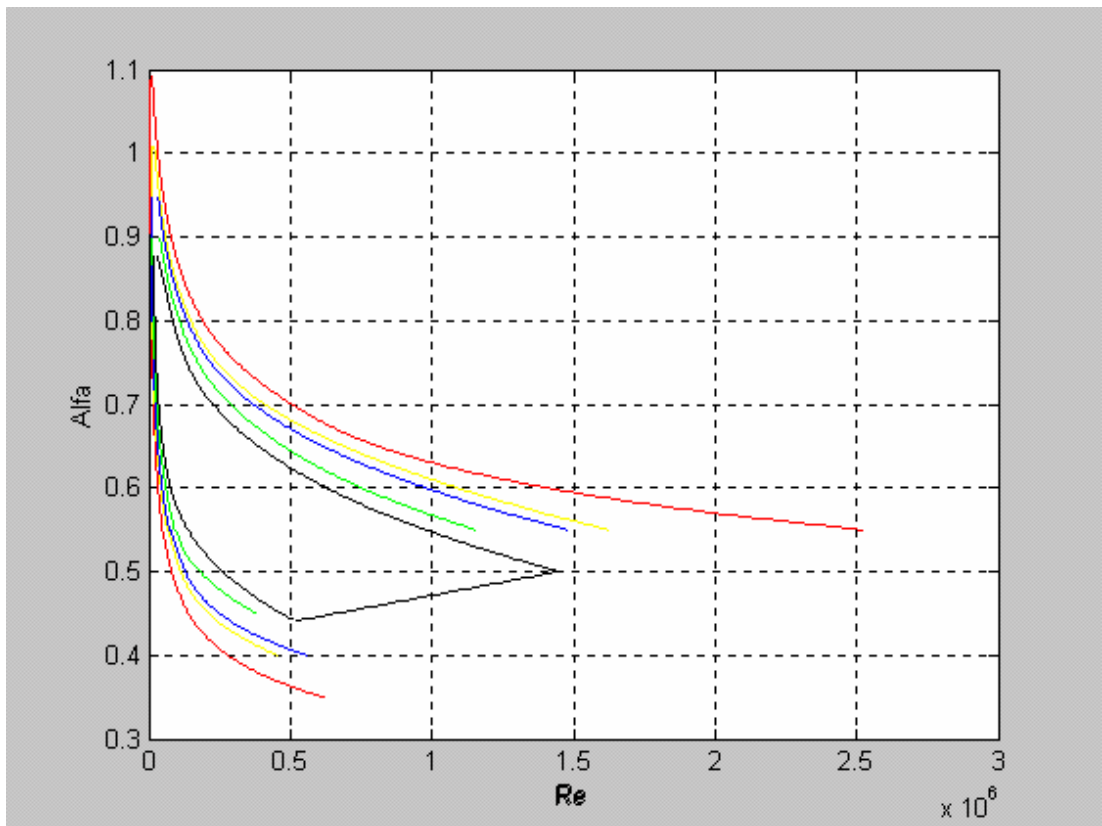


Figure 2.8 Stability diagram for plane Poiseuille flow.

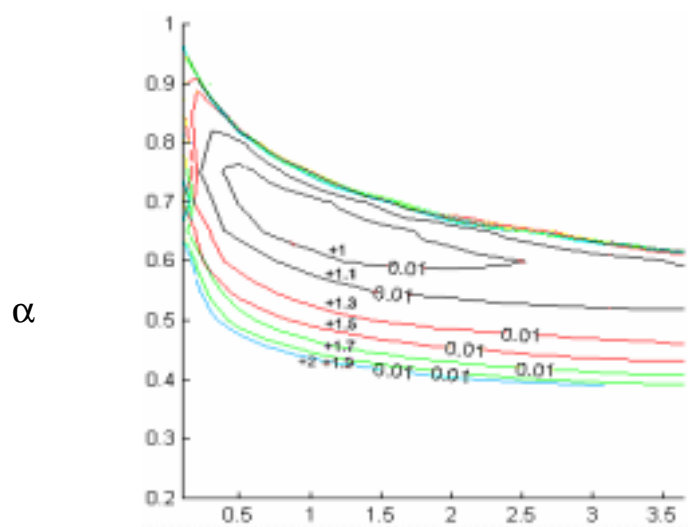


Figure 2.9. Instability regions for different nondimension pressure gradients

3. CONTROL OF TRANSITION

One of the fundamental goals in the design of a body passing through a fluid, is a reduction of the drag induced by the flow. In particular, the reduction of ‘skin Friction’ drag with regards to aerodynamic shapes is of high importance, indeed typically half the total drag on a supersonic aircraft is accounted for by Skin Friction Drag [23].

A way in which this may be achieved is by delaying the laminar to turbulent transition of a boundary layer. The skin friction drag in laminar flow can differ by an order of magnitude less than that of turbulent flow. The benefits are clear in aircraft design, provided longer range and reduced fuel costs. It should be noted however that a turbulent boundary layer is more ‘resistant’ to separation and where separation is prevented lift is enhanced and ‘Form’ drag is reduced. Another area where maintaining laminar flow is of importance applies to the reduction of flow induced noise for operation of underwater solar [24], however turbulent is an efficient mixer, and higher rates of heat transfer and the like are achieved with turbulent motion.

As a means of implementing these delays, methods fall into either of the Passive or Active Techniques. In the passive case the aim is in typically of a pressure-gradient / wall-shaping strategy. Active techniques may take the form of transpiration, wall heating / cooling, wall motion. In general these methods are in some way trying to alter the growth of unstable waves. The Theory leading to these unstable waves is that of linear stability. The linear stability of a flow being governed by the famous Orr-Sommerfeld equation.

The initial stages of the project have been concerned with the establishment of numerical techniques for investigating the control of Plane Poiseuille Flow. This flow is of interest as it is both very well understood flow, yet still a difficult control problem. In particular the control under consideration is that of transpiration.

3.1 The Theory of Boundary Layer

A huge amount of work got underway in the following centuries for understanding fluid flow, leading to many great advances, but also to the development of two, very distinct. The first of which being “ theoretical hydrodynamics”, which was based upon a development of Euler’s equations of motion for a frictionless, non viscous fluid. The second of which was “hydraulics”, this being based on the many results from experiments, which differed greatly from what was expected from the hydrodynamics point of view. This discrepancy is of high relevance, as without its resolution, current flow control most probably wouldn’t have advanced.

D’Alembert’s paradox is perhaps the most famous case of these discrepancies. The theory being based upon a perfect fluid, that is frictionless and incompressible, leads to the result that a body immersed in a fluid flow should experience no drag. What was missing, was simply the fact that the fluid does indeed not only permit normal forces but tangential forces also. The equations which had taken account for the frictional properties of the fluid have been developed, the Navier-Stokes equations, but unfortunately solutions to these have proved to be extremely difficult, (and generally unsurmountable) in all but very particular examples. Another important aspect [25], not to be overlooked, was the simple fact that important fluids such as water and air, are of very low viscosity, and as such the influence of friction forces should be small (which is true in general) in relation to the pressure forces. All of this did not help in resolving the discrepancies between the two lines of thought.

Then in 1904, came possibly the most famous paper, “ Fluid motion with very small friction”, in fluid dynamics by L. Prandtl. With it came the notion of the Boundary layer and a huge step towards rectifying the gap between theoretical work, with known practical results. The idea was that flow about a body may be divided in two regions. The first being a thin layer very close to the body, and a second layer outside of this. Within the thin boundary Layer the frictional forces become important, and outside of it the frictional forces are negligible. What is of most importance here, is that not only had he formed an intriguing insight to the physical process, but had also greatly simplified the inherent mathematical problem.

With the development of boundary layer theory, comes an understanding of drag in its various forms, from skin friction to form. In the case of form drag it is seen that

within the boundary layer, flow may become reversed very close to the surface. With separation, comes the formation of eddies and a marked change in the pressure distribution from that of a frictionless stream [25]. This pressure difference is what leads to the form drag.

With the development of the boundary layer came, the concepts, of the Laminar boundary layer, and that of the turbulent. Of particular interest is the transition process from the laminar to the turbulent state.

3.2 Development of Transition

One of the important elements that was missing from the ideal fluid theory was that of turbulence. The transition process was first studied in detail in the 1980's. In particular the transition process in pipe flow was studied. At low Reynolds numbers, the flow is orderly, with the effect of viscosity slowing the motion towards the wall. For higher Reynolds numbers however, the order breaks down, and mixing occurs between the regions of fluid, flowing at different distances from the surface walls. Reynolds [27] made this visible, by using a dye, which would form a distinct line as the flow moved in its laminar state, but would diffuse, as result of the turbulent motion in a uniform mixture further down stream. As well as the main velocity in the pipe direction, which in turn results in an exchange of momentum from the faster layers to the slower layers. Another contrast of the two flows is that of the cross-sectional velocity profiles. In the Laminar case it is parabolic, but due to the 'mixing' of momentum in the turbulent case, the center region of the profile is more or less uniform. Reynolds, with that aid of this dye experiment, set about understanding, this change in flow property, leading to the famous Reynolds number, where the Reynolds number itself being the relation, as above mentioned, $Re = (u d / \nu)$, where u is mean velocity and d is diameter of the pipe. Provided that Re was kept below the critical Reynolds number Re_c , the flow would remain laminar, and above this value the flow would become turbulent. The actual value of the flow is dependant upon conditions that prevail with in the initial section of the pipe.

3.3 Control Algorithm for Laminar Plane Channel Flows

First of all continuous form of laminar flow equation given, then discrete form of laminar flow equation given for writing control algorithm, then measurement system explained and theoretical approximation given. Information yielded from [21] for this section, so see [21] for more information.

3.4 Continuous Form of Laminar Flow Equations

Imagine a steady plane channel flow with maximum velocity U_0 and channel half-width δ . Non-dimensionalizing all velocities by U_0 and lengths by δ , the mean velocity profile in the streamwise direction (x) may be written $U(y) = 1 - y^2$ on the domain $y \in [-1, 1]$. The equations governing small, incompressible, three-dimensional perturbations $\{u, v, w, p\}$ to the mean flow U are given by the linearized Navier-Stokes and continuity equations

$$\dot{u} + U \frac{\partial}{\partial x} u + U' v = -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}} \Delta u, \quad (3.1a)$$

$$\dot{v} + U \frac{\partial}{\partial x} v = -\frac{\partial p}{\partial y} + \frac{1}{\text{Re}} \Delta v, \quad (3.1b)$$

$$\dot{w} + U \frac{\partial}{\partial x} w = -\frac{\partial p}{\partial z} + \frac{1}{\text{Re}} \Delta w, \quad (3.1c)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \quad (3.2)$$

where $\Delta \equiv \partial^2 / \partial x^2 + \partial^2 / \partial y^2 + \partial^2 / \partial z^2$ is the Laplacian, $\text{Re} \equiv U_0 \delta / \nu$ is the Reynolds number, ν is the kinematic viscosity, dot ($\dot{\cdot}$) denotes $\partial / \partial t$, and prime ($'$) denotes d / dy . A single equation for the normal component of velocity v , found by taking the Laplacian of (3.1b), substituting for Δp from the divergence of (3.1), and applying (3.2), is

$$\Delta \dot{v} = \{-U \frac{\partial}{\partial x} \Delta + U' \frac{\partial}{\partial x} \Delta(\Delta / \text{Re})\}v. \quad (3.3a)$$

The equation for the normal component of vorticity $w \equiv \partial u / \partial z - \partial w / \partial x$, found by subtracting $\partial / \partial x$ of (3.1c) from $\partial / \partial z$ of (3.1a), is

$$\dot{w} = \{-U' \frac{\partial}{\partial z}\}v + \{-U \frac{\partial}{\partial x} + \Delta / \text{Re}\}w. \quad (3.3a)$$

The flow perturbation problem in $\{u, v, w, p\}$ with second-order partial derivatives in (3.1)-(3.2) has been reduced to a problem in $\{v, w\}$ with fourth-order partial derivatives in (3.3) with no loss of generality; essentially, the three-component velocity field has been projected onto a two-component divergence-free manifold by eliminating the pressure from the equations and applying continuity.

As the domain is homogeneous in the x - and z -directions, we may Fourier transform the solution yielded

$$v(x, y, z, t) = \sum_{k_x, k_z} \hat{v}(k_x, y, k_z, t) \exp[(ik_x x + k_z z)],$$

$$w(x, y, z, t) = \sum_{k_x, k_z} \hat{w}(k_x, y, k_z, t) \exp[(ik_x x + k_z z)].$$

As the various Fourier modes are orthogonal and equations (3.3a) and (3.3b) are linear, the solution for each wavenumber pair (k_x, k_z) is decoupled and obeys the equations

$$\Delta \dot{v} = \{-i k_x U \Delta + i k_x U' + \Delta(\Delta / \text{Re})\}v \quad (3.4a)$$

$$\dot{w} = \{-i k_z U'\}v + \{-i k_x U + \Delta / \text{Re}\}w \quad (3.4b)$$

where the hat accents ($\hat{}$) have been dropped for notational relevance and the Laplacian now takes the form $\Delta \equiv \partial^2 / \partial y^2 - k_x^2 - k_z^2$. Equation (3.4a) is the well-known (fourth-order) Orr-Sommerfeld equation for the wall-normal velocity

where the hat accents ($\wedge$) have been dropped for notational relevance and the Laplacian now takes the form $\Delta \equiv \partial^2 / \partial y^2 - k_x^2 - k_z^2$. Equation (3.4a) is the well-known (fourth-order) Orr-Sommerfeld equation for the wall-normal velocity modes, and (3.4b) is the (second-order) equation for the wall-normal vorticity modes. Note the one-way coupling between these two equations. Also note that, from any solution $\{v, w\}$, the values of u and w may be evolved by manipulation of the above equations into the forms

$$u = \frac{i}{k_x^2 + k_z^2} \left(k_x \frac{\partial v}{\partial y} - k_z w \right) \text{ and } w = \frac{-i}{k_x^2 + k_z^2} \left(k_z \frac{\partial v}{\partial y} - k_x w \right), \quad (3.5)$$

and p may be found by solution of the equation $\Delta p = -2ik_x U'v$. Control is carried out at the wall as a boundary condition on the wall-normal component of velocity v . The boundary conditions on u and w are no-slip ($u = w = 0$), which implies that, at the wall, $w = 0$ and (by continuity) $\partial v / \partial y = 0$

3.5 Discrete Form of Laminar Flow Equations

The continuous equations for the $\{v, w\}$ perturbations in (3.4) are now discretized on a grid of $N + 1$ Chebyshev-Gauss-Lobatto points in the wall-normal direction such that

$$y_\kappa = \cos(\pi\kappa / N) \text{ for } 0 \leq \kappa \leq N.$$

An $(N+1) \times (N+1)$ matrix D may be expressed [29] such that the derivative of w with respect to y on the discrete set of $N + 1$ points is given by

$$w' = D w, \text{ and } w'' = D w',$$

$$\tilde{D}_{iK} = \begin{cases} 0, & i = 0, N \\ D_{iK}, & 1 \leq i \leq N-1. \end{cases}$$

Differentiation of v with respect to y is then given by

$$v' = \tilde{D}v, \quad v'' = Dv', \quad v''' = Dv'' \quad \text{and} \quad v'''' = Dv'''$$

With these derivative matrices, it is straightforward to write (3.4) in matrix form. This is executed by first expressing the matrix form of (3.4) on all $N + 1$ collocation points such that

$$\mathbf{v}' = L\mathbf{v}, \quad (3.6a)$$

$$\mathbf{v}'' = B\mathbf{v} + G\mathbf{w}, \quad (3.6b)$$

where the $(N + 1) \times (N + 1)$ matrices L , B , and G represent the spatial discretization of the bracketed operations in (3.4). The Dirichlet boundary conditions are explicitly prescribed as separate 'forcing' terms. To execute this, decompose L , B , and G according to

$$L = \begin{pmatrix} * & * & * \\ b_{11} & L_c & b_{12} \\ * & * & * \end{pmatrix}, \quad B = \begin{pmatrix} * & * & * \\ b_{21} & B_c & b_{22} \\ * & * & * \end{pmatrix}, \quad G = \begin{pmatrix} * & * & * \\ * & G_c & * \\ * & * & * \end{pmatrix},$$

where L_c , B_c , G_c are $(N-1) \times (N-1)$ and b_{11} , b_{12} , b_{21} , and b_{22} are $(N-1) \times 1$. Noting that $w_0 = w_N = 0$ by the no-slip condition, and describing

$$\mathbf{x} \equiv \begin{pmatrix} v_1 \\ \mathbf{M} \\ v_{N-1} \\ w_1 \\ \mathbf{M} \\ w_{N-1} \end{pmatrix}, \quad \mathbf{A} \equiv \begin{pmatrix} L_c & 0 \\ B_c & G_c \end{pmatrix}, \quad \mathbf{B} \equiv \begin{pmatrix} -b_{11} & b_{12} \\ -b_{21} & b_{22} \end{pmatrix}, \quad \mathbf{u} \equiv \begin{pmatrix} -v_0 \\ v_N \end{pmatrix},$$

Where $\mathbf{x}$ is $2(N-1) \times 1$, $\mathbf{A}$ is $2(N-1) \times 2(N-1)$, $\mathbf{B}$ is $2(N-1) \times 2$, and u is 2×1 , we may define (3.6) in the form

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}u. \quad (3.7)$$

The vector $\mathbf{x}$, which includes the normal velocity fluctuations v_i and normal vorticity fluctuations w_i at the grid points on the interior of the channel, is referred to as the 'state'. The vector u , which contains the blowing/suction velocity at the top and bottom walls, is referred to as the 'control'.

3.6 Wall Measurements Approximation

We will imagine control algorithms using both full flow field data and wall data only. For the latter case, we will suppose that measurements made at the wall ensure information about the streamwise and spanwise skin friction, from which (subtracting out the known influence of $\partial v / \partial x$ and $\partial v / \partial z$ from the stress tensor at the wall) we may establish the following four quantities:

$$y_{m1} = -\frac{1}{\text{Re}} \frac{\partial u}{\partial y} \Big|_{\text{upper wall}}, \quad y_{m2} = \frac{1}{\text{Re}} \frac{\partial u}{\partial y} \Big|_{\text{lower wall}}, \quad (3.8)$$

$$y_{m3} = -\frac{1}{\text{Re}} \frac{\partial w}{\partial y} \Big|_{\text{upper wall}}, \quad y_{m4} = \frac{1}{\text{Re}} \frac{\partial w}{\partial y} \Big|_{\text{lower wall}},$$

Note that, for $k_x^2 + k_z^2 \neq 0$, the matrix form of the left-hand side of (3.4a) is invertible, so the form (3.6a) is easily determined.

With (3.5), we may explain these measurements as linear combinations of v and w . Describing $a \equiv ik_x / (k_x^2 + k_z^2) / \text{Re}$ and $b \equiv -ik_z / (k_x^2 + k_z^2) / \text{Re}$, the measurements are shown in terms of the discrete vectors v and w as

$$y_{m1} = (-aD\tilde{D}v - bDw) \Big|_{\text{upper wall}}, \quad y_{m2} = (aD\tilde{D}v + bDw) \Big|_{\text{lower wall}},$$

$$y_{m3} = (-bD\tilde{D}v - aDw) \big|_{upper \text{ wall}}, \quad y_{m4} = (bD\tilde{D}v + aDw) \big|_{lower \text{ wall}},$$

Decompose D , $\tilde{D}$, and $(D\tilde{D})$ according to

$$D = \begin{pmatrix} * & c_3 & * \\ * & * & * \\ * & c_4 & * \end{pmatrix}, \quad \tilde{D} = \begin{pmatrix} * & * & * \\ * & \tilde{D}_c & * \\ * & * & * \end{pmatrix}, \quad (D\tilde{D}) = \begin{pmatrix} d_1 & c_1 & d_3 \\ * & * & * \\ d_2 & c_2 & d_4 \end{pmatrix},$$

where $\tilde{D}_c$ (to be used in the following section) is $(N-1) \times (N-1)$, c_1, c_2, c_3 , and c_4 are $1 \times (N-1)$, and d_1, d_2, d_3 , and d_4 are 1×1 . Finally, defining

$$y_m \equiv \begin{pmatrix} y_{m1} \\ y_{m2} \\ y_{m3} \\ y_{m4} \end{pmatrix}, \quad \mathbf{C} \equiv \begin{pmatrix} -ac_1 & -bc_3 \\ ac_2 & bc_4 \\ -bc_1 & -ac_3 \\ bc_2 & ac_4 \end{pmatrix}, \quad \mathbf{D} \equiv \begin{pmatrix} ad_1 & -ad_3 \\ -ad_2 & ad_4 \\ bd_1 & -bd_3 \\ -bd_2 & bd_4 \end{pmatrix},$$

where y_m is 4×1 , $\mathbf{C}$ is $4 \times 2(N-1)$, and $\mathbf{D}$ is 4×2 , allows us to express y_m in the form of a linear combination of the state $\mathbf{x}$ and the control $\mathbf{u}$

$$y_m = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}. \tag{3.9}$$

The vector y_m is referred to as the 'measurement'. Define the inner product for two discrete vectors $\mathbf{u}$ and $\mathbf{v}$ discretized on the collocation points $y_\kappa = \cos(\pi\kappa/N)$ by

$$(\mathbf{u}, \mathbf{v}) \equiv \sum_{\kappa=0}^N u_\kappa^* v_\kappa \zeta_\kappa, \quad \text{where } \zeta_\kappa \equiv \begin{cases} \frac{\pi}{2N}, & \kappa = 0, N \\ \frac{\pi}{N}, & 1 \leq \kappa \leq N-1. \end{cases}$$

4. DISCUSSION

The method suggested by Ng and Reid for solving the Orr-Sommerfeld equation (in section 2.1) accomplishes the following tasks: It reduces the eigenvalue problem to iteration involving single initial value problem. It avoids ‘arithmetic’ computation of the determinant required in the iteration process for the eigenvalue. It allows computation to be carried out using ‘standard’ integration methods. The net result is greatly increased speed of computation with large reduction in the effort required to implement a problem on the computer. Orr-Sommerfeld equation solved with MATLAB6 using Ng and Reid method. First of all the program given [16] transcribed Matlab with the aid of [17-18-19]. Original program want to three initial values for calculate the Orr-Sommerfel equation, this problem solved with sub Matlab6 program that give approximation values [p1, p2, p3] for a given Re numbers. This sub program calculate the approximation values like; first initial approximation values written for only one Re number (these values get in [16]) in program then program calculate the approximation values using linear and cubic-spline interpolations for a given Re space (for this study Re space is $[3000-10^6]$). Cubic-spline interpolation used to calculate the values for $Re = Re + \text{value}$. Original program want to M value (number of sub interval, steps, in Runge-Kutta method) for a given Re number. This program only want to enter Re number and calculate the Orr-Sommerfeld equation approximately between 1 to 3 minutes according to Re number. If Re number is large, the M value, using in Runge-Kutta method, becomes large and the calculation time is long. Muller’s method used to find the c values because of more convenient for finding complex root. Orr-Sommerfeld equation’s initial data values multiplied with 10^{-100} or 10^{-200} or 10^{-300} according to given large Re number, to prevent to arithmetic overflows. Yielded results using this program is suitable for engineering applications and the comparison with the other results given above shows that this program is more convenient for small Re numbers. If you want to more accurate solution for a given Re number, you should use greater M value.

Stability diagram obtained for different c_i values using this program as subprogram. First, initial interval area yielded for given c_i and curves plotted Re versus c_i for

visuality then other sub program compare the Re value on initial interval and then chose the required Re value for approximation to ci value. Secant method was used the approximate given ci value in program. Stability diagram yielded superposition of stability curves of the different ci value.

Finally, measurement approximation written for control and the control matrix of the parallel flow written for the control algorithm.

REFERENCES

- [1] **KIRCHNER, N. P.**, 2000. Computational Aspects of the Spectral Galerkin FEM for the Orr-Sommerfeld Equation, *International Journal for Numeical Methods in Fluids*, **32**, 119-137
- [2] **GEORGEUSCU, A.**, 1985. Hydrodynamic Stability Theory, *Martinus Nijhoff publishers, Doldrecht*.
- [3] **ORAZW, P. G., REIO, W. H.**, 1981. Hydrodynamic Stability, *Cambridge University Press, Cambridge*.
- [4] **FORTIN, A., JARDAK, M., GERVAIS, J. J., PIERRA, R.**, 1994. Old and New Results on the Two-Dimensional Poiseuille Flow, *Department de Mathematique Appliquees, Ecole Polytechnique*
- [5] **THOMAS, L. H.**, The Stability of the Plane Poiseuille Flow, *Physics. Rev* (2), **91**, pp.780-783.
- [6] **ORSZAG, S. A.**, 1971. Accurate Solution of the Orr-Sommerfeld Stability Equation, *J. Fluid Mech*, **50**, pp.689-703
- [7] **ORAZIN, P. G., REID, W. H.**, 1981. Hydrodynamic Stability, *Cambridge Unniversity Press, Cambridge*.
- [8] **GERSTING, J. M.**, 1977. Numerical Methods for Eigensystem: the Orr – Sommerfeld Problem. *Comp. Maths. Applics*. **3**, 47-52.

- [9] GERSTING, J. M., JANKOWSKI, D.F., 1972. Numerical Methods for Orr-Sommerfeld Problems. *Inter. J. Numer. Math. Engr*, **4**, 195-206.
- [10] NG, B. S., REID, W. H., 1976. An Initial Value Method for eigenvalue Problems Using Compound Matrices, *J. Comp. Phys*, **30**, 125-136.
- [11] BRILLOUIN, L., 1964. Tensors in Mechanics and Elasticity, Academic Press, New York.
- [12] SCOTT, M. R., 1953. Initial Value Method for the Eigenvalue Problem for Systems of Ordinary Differential Equations, *J Comput. Phys*, **12**, **91**, pp. 780-783.
- [13] SCOTT, M. R., WATTS, H. A., 1975. SUPPORT-A Computer Code for two-point Boundary Value Problems via Orthonormalization, *Sandia Laboratories, Report SAND75-0198*.
- [14] SLVERSTON, S., 1968. ORTNORM-A FORTRAN Subroutine Pakage for the Solution of Linear Two-point Baundry Value Problems, Purdue University, Computer Science Department, *Report CSD TR 18*.
- [15] DONGORRA, J. J., SRAUGHAN, B., WALKER, D. W., Chebyshev tau-QZ Algorithm Methods for Calcualting Spectra of Hyrodynamic Stability Problems, Department of Computer Science, University of Tennesse, Knoxville, Tennessee 379996-13001, U.S.A.
- [16] ERVIN, Y. RODIN. 1980. Numerical Methods for Eigen Systems: The Orr-Sommerfeld Problem as an Initial Value Problem, *Comp & Maths with Appls*. **6**, pp. 167-174.
- [14] SLVERSTON, S., 1968. ORTNORM-A FORTRAN Subroutine Pakage for the Solution of Linear Two-point Baundry Value Problems, *Purdue University, Computer Science Department, Report CSD TR 18*.
- [15] DONGORRA, J. J., SRAUGHAN, B., WALKER, D. W., “Chebyshev tau-QZ Algorithm Methods for Calcualting Spectra of Hyrodynamic Stability Problems”, *Department of Computer Science, University of Tennesse, Knoxville, Tennessee 379996-13001, U.S.A.*
- [16] ERVIN, Y. RODIN. 1980. “Numerical Methods for Eigen Systems: The Orr-Sommerfeld Problem as an Initial Value Problem”, *Comp & Maths with Appls*. **6**, pp. 167-174.
- [17] YÜKSEL, İ. 2000. MATLAB ile Mühendislik Sistemlerinin Analizi ve Çözümü, Baskı 2, *VİPAŞ A.Ş Basım*.
- [18] MOLER, C., 1997. Engineering Problem Solving with MATLAB, *Second Edition, The MathWorks, Inc.*

- [19] **MTHEWS, J. H., FINK, K. U.**, Numerical Methods Using Matlab, *Third Edition*.
- [20] **WHITE, F. M.**, Viscous Fluid Flow, *Second Edition*.
- [21] **BEWLEY, T. R., LIU, S.**, 1998. "Optimal and Robust Control and Estimation of Linear Paths to Transition", *J. Fluid Mech*, vol. 365, pp. 305-349
- [22] **SÖYLEMEZ, H. T., ÖZKOL, İ.**, "The Stability Dependence on the Different Pressure Gradient in a Channel Flow"
- [23] **KURT, KLEINER.**, 19.10.1996, New Scientist .
- [24] **GAD-EL-HAK, MOHAMED.**, 1989, Flow Control, *Depth. Of Aerospace and Mechanical Engineering* .
- [25] **SCHLICHTING, HERMAN.**, 1960, *Boundary Layer Theory*.
- [26] **L, PANTON, RONALD.**, 1984, Incompressible Flow.
- [27] **REYNOLDS, O.**, 1883,"On the Experimental Investigation of the Circumstances Which determine Whether the Motion of Water Shall be Direct or Sinuous, and the Law of Resistance in Parallel Channels" *Phil. Trans. Royal Society*.
- [28] **SHEN, S.F.**, 1954, Calculate Amplified Oscillations in Plane Poiseuille and Blasius flows, *J. Aeronaut. Sci.*, vol. 21, pp. 62-64
- [29] **CANUTO, C., HUSSAINI, M. Y., QUARTERONI, A. & ZANG, T. A.**, 1988 Spectral Methods in Fluid Dynamics. *Springer*.
- [30] **O' DEA, ENDAÇ., TUTTY, OWEN.**, June.30.200. " Robust Flow Control", *Nine Month Report, School of Engineering Sciences, University of Southampton*.

Appendix-A

METHODS USED CALCUALATE THE c VALUE AND ci VALUE

1 MULLER'S METHOD FOR APPROXIAMATING THE c VALUE

It is an iterative method that requires three starting points $(p_0, f(p_0))$, $(p_1, f(p_1))$, and $(p_2, f(p_2))$. A parabola is constructed that passes through the three points; then the quadratic formula is used to find a root of the quadratic for the next approximation. It has been proved that near a simple root Muller's method converges faster than the secant method and almost as fast as Newton's method. The method can be used to find real or complex zeros of a function and can be programmed to use complex arithmetic.

Without loss of generality, we assume that p_2 is the best approximation to the root and consider the parabola through the three starting values shown in Figure A.1.

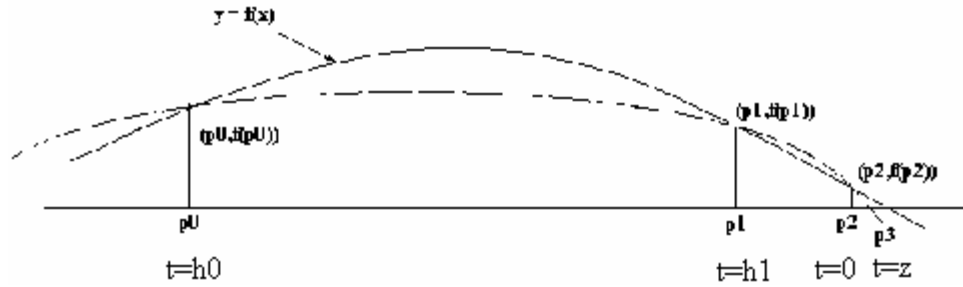


Figure A.1 The starting approximations p_0 , p_1 , and p_2 for Muller's method and the differences h_0 and h_1 [19].

Make the change of variable

$$t = x - p_2 \quad (1.1)$$

and use the difference

$$h_0 = p_0 - p_2 \quad \text{and} \quad h_1 = p_1 - p_2 \quad (1.2)$$

Consider the quadratic polynomial involving the variable t :

$$y = a t^2 + b t + c. \quad (1.3)$$

Each point is used to obtain an equation involving a , b , and c :

$$\begin{aligned} \text{At } t = h_0: & \quad a h_0^2 + b h_0 + c = f_0. \\ \text{At } t = h_1: & \quad a h_1^2 + b h_1 + c = f_1. \\ \text{At } t = 0 : & \quad a 0^2 + b 0 + c = f_2. \end{aligned} \quad (1.4)$$

From the third equation in (1.4), we see that

$$c = f_2 \quad (1.5)$$

Substituting (1.5) into the first two equations in (1.4) and using the definition $e_0 = f_0 - c$ and $e_1 = f_1 - c$ results in the linear system

$$\begin{aligned} a h_0^2 + b h_0 &= f_0 - c = e_0. \\ a h_1^2 + b h_1 &= f_1 - c = e_1. \end{aligned} \quad (1.6)$$

Solving the linear system for a and b results in

$$\begin{aligned} a &= \frac{e_0 h_1 - e_1 h_0}{h_1 h_0^2 - h_0 h_1^2} \\ b &= \frac{e_1 h_1 - e_0 h_0}{h_1 h_0^2 - h_0 h_1^2}. \end{aligned} \quad (1.7)$$

The quadratic formula is used to find the roots $t = z_1, z_2$ of (1.3):

$$z = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}. \quad (1.8)$$

Formula (1.8) is equivalent to the standard formula for the roots of a quadratic and is better in this case because we know that $c = f_2$.

To ensure stability of the method, we choose the root in (1.8) that has the smallest absolute value. If $b > 0$, use the positive sign with the square root, and if $b < 0$ use the negative sign. Then p_3 is shown in Figure A.1 and is given by

$$p_3 = p_2 + z. \quad (1.9)$$

To update the iterates, choose p_0 and p_1 to be the two values selected from among $\{p_0, p_1, p_3\}$ that lie closest to p_3 (i.e., throw out the one that is farthest away). Then replace p_2 with p_3 . Although a lot of auxiliary calculations are done in Muller's method, it only requires one function evaluation per iteration.

If Muller's method is used to find the real roots of $f(x) = 0$, it is possible that one may encounter complex approximations, because the roots of the quadratic in (1.8) might be complex (nonzero imaginary components). In these cases the imaginary components will have a small magnitude and can be set equal to zero so that the calculations proceed with real numbers.

2 FOURTH-ORDER RUNG-KUTTA (RK4) METHOD FOR CALCULATE THE c VALUE WITH THE AID OF NG-REID METHOD.

The Taylor methods have the desirable feature that the final global error is of order $O(h^N)$, and N can be chosen large so that this error is small. However, the shortcomings of the Taylor methods are the priori determination of N and the computation of the higher derivatives, which can be very complicated. Each Runge-Kutta method is derived from an appropriate Taylor method in such a way that the final global error is of order $O(h^N)$. A trade-off is made to perform several function evaluations at each step and eliminate the necessity to compute the higher derivatives. This method can be constructed for any order N . The Runge-Kutta method of order $N = 4$ is most popular. It is a good choice for common purposes because it is quite accurate, stable, and easy to program. Most authorities proclaim that it is not necessary to go to higher order method because the increased accuracy is offset by the additional computational effort. If more accuracy is required, than either a smaller step size or an adaptive method should be used.

The fourth-order runge-Kutta method (RK4) simulates the accuracy of the Taylor series method of order $N = 4$. The method is based on computing y_{k+1} as follows[19]:

$$y_{k+1} = y_k + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4, \quad (2.1)$$

Where k_1, k_2, k_3 , and k_4 have the form

$$\begin{aligned} k_1 &= h f(t_k, y_k), \\ k_2 &= h f(t_k + a_1 h, y_k + b_1 k_1), \\ k_3 &= h f(t_k + a_2 h, y_k + b_2 k_1 + b_3 k_2), \\ k_4 &= h f(t_k + a_3 h, y_k + b_4 k_1 + b_5 k_2 + b_6 k_3), \end{aligned} \quad (2.2)$$

By matching coefficients with those of the Taylor series method of order $N = 4$ so that the local truncation error is order $O(h^5)$, Runge and Kutta were able to obtain the following system of equations:

$$\begin{aligned} b_1 &= a_1, \\ b_2 + b_3 &= a_2, \\ b_4 + b_5 + b_6 &= a_3, \\ w_1 + w_2 + w_3 + w_4 &= 1, \\ w_2 a_1 + w_3 a_2 + w_4 a_3 &= 1/2, \\ w_2 a_1^2 + w_3 a_2^2 + w_4 a_3^2 &= 1/3, \\ w_2 a_1^3 + w_3 a_2^3 + w_4 a_3^3 &= 1/4, \\ w_3 a_1 b_3 + w_4 (a_1 b_5 + a_2 b_6) &= 1/6, \\ w_3 a_1 a_2 b_3 + w_4 a_3 (a_1 b_5 + a_2 b_6) &= 1/8, \\ w_3 a_1^2 b_3 + w_4 (a_1^2 b_5 + a_2^2 b_6) &= 1/12, \\ w_4 a_1 b_3 b_6 &= 1/24. \end{aligned} \quad (2.3)$$

The system involves 11 equations in 13 unknowns. Two additional conditions must be supplied to solve the system. The most useful choice is

$$a_1 = 1/2 \text{ and } b_2 = 0. \quad (2.4)$$

Then the solution for the remaining variables is

$$a_2 = 1/2, a_3 = 1, b_1 = 1/2, b_3 = 1/2, b_4 = 0, b_5 = 0, b_6 = 1, \quad (2.5)$$

$$w_1 = 1/6, w_2 = 1/3, w_3 = 1/3, w_4 = 1/6.$$

The values in (2.4) and (2.5) are substituted into (2.2) and (2.1) to obtain the formula for the standard Runge-Kutta of order $N = 4$, which is stated as follows. Start with the initial point (t_0, y_0) and generate the sequence of the approximations using

$$y_{k+1} = y_k + h*(f_1 + 2f_2 + 2f_3 + f_4)/6 \quad (2.6)$$

where

$$\begin{aligned} f_1 &= f(t_k, y_k), \\ f_2 &= f(t_k + h/2, y_k + h f_1 / 2), \\ f_3 &= f(t_k + h/2, y_k + h f_2 / 2), \\ f_4 &= f(t_k + h, y_k + h f_3), \end{aligned} \quad (2.7)$$

3- SECANT METHOD FOR APPROXIMATING TO THE $c_i = 0$ VALUE

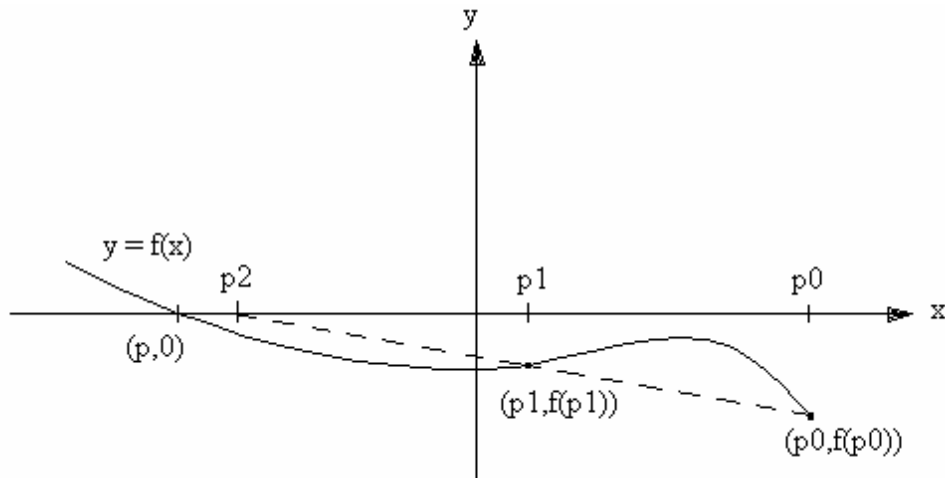


Figure A.2 The geometric construction of p_2 for the secant method [19].

The Newton-Raphson algorithm requires the evaluation of two functions per iteration, $f(p_{k-1})$ and $f'(p_{k-1})$. Traditionally, the calculation of derivatives of elementary functions could involve considerable effort. But, with modern computer algebra software packages, this has become less of an issue. Still many functions have nonelementary forms (integrals, sums, etc.), and it is desirable to have a method that converges almost as fast as Newton's method yet involves only evaluations of $f(x)$ and not $f'(x)$. the secant method will require only one evaluation of $f(x)$ per step and at a simple root has an order of convergence $R \approx 1.618033989$. It is almost as fast as Newton's method, which has order 2.

The formula involved in the secant method is the same one was used in the regula falsi method, except that the logical decisions regarding how to define each succeeding term are different. Two initial points $(p_0, f(p_0))$ and $(p_1, f(p_1))$ near the point $(p, 0)$ are needed, as shown in Figure A-2. Define p_2 to be the abscissa of the point of intersection of the line through these two points and the x-axis; then Figure A.2. shows that p_2 will be closer to p than to either p_0 or p_1 . The equation relating p_2 , p_1 , and p_0 is found by considering the slope

$$m = \frac{f(p_1) - f(p_0)}{p_1 - p_0} \quad \text{and} \quad m = \frac{0 - f(p_1)}{p_2 - p_1}. \quad (3.1)$$

The values of m in (3.1) are the slope of the secant line through the first two approximations and the slope of the line through $(p_1, f(p_1))$ and $(p_1, 0)$, respectively. Set the right-hand sides equal in (3.1) and solve for $p_2 = g(p_1, p_0)$ and get

$$p_2 = g(p_1, p_0) = p_1 - \frac{f(p_1)(p_1 - p_0)}{f(p_1) - f(p_0)}. \quad (3.2)$$

the general term is given by the two-point iteration formula

$$p_{k+1} = g(p_k, p_{k-1}) = p_k - \frac{f(p_k)(p_k - p_{k-1})}{f(p_k) - f(p_{k-1})}. \quad (3.3)$$

Appendix-B

1- Estimate initial c versus Re for calculating initial approximated values (p_1, p_2, p_3) that required more accurate solution .

1.1 Programs

```
global A Re C sw
iter=1;sr=0;km=1;kr=1;tr=1;kk(1)=0; Re=3000; M=400;
EC=rci(M);H(1,:)=EC; %Obtain estimated c value ,used below, for
Re=3000
%Estimated Re intervals are given as Re*1e-3
rr=[6,10,15,20,30,40,50,60,70,80,90,100,...
    110,120,130,140,150,160,175,200,225,250,...
    275,300,325,350,375,400,425,450,475,500,...
    525,550,575,600,625,650,675,700,725,750,...
    775,800,825,850,875,900,925,950,975,1000,1010];
r=rr(iter);
while r <= 1010
    iter=iter+1; Re=1000*r; %Obtain M number for a given Re
    if r <= 5 M=400;
    elseif (r >5 & r <10) M=1200;
    elseif (r >=10 & r <13) M=1400;
    elseif (r >=13 & r <60) M=1500;
    elseif (r >=60 & r <=150)M=1600;
    elseif (r >150 & r <500)M=1600;
    elseif (r > 500& r <650) M=2000;
    else M=3000;
    end
    load rey_cest %Yielded with rci function at Re = 3000m, supply C12=c
    and re2 = Re
    % Estimate C2 = c value using linear interpolation for later Re
    C2=C12(end-1,:)+(r-re2(end-2))/(re2(end-1)-re2(end-2))*(C12(end,:)-
    C12(end-1,:));
    C12=[C12;C2]; % Assigning this c in C12 that required, below,
    interpolation
    If sw==10%sw enrolled in rci func that control the c is founded or
    not
        r2=r+0.108; r3=r-0.1869; %Required linear interpolation
    end
    c1=interp1(re2,C12,r,'linear');
    c2=interp1(re2,C12,r2,'spline'); % No c value Re=Re+.. so I use
    the spline
                                %interpolation
    c3=interp1(re2,C12,r3,'linear');
    P=[(c1(1)+c1(2)*i);(c2(1)+c2(2)*i);(c3(1)+c3(2)*i)]; % Estimated
    initial c
                                % values
    P=[p1,p2,p3]
    [estc,iteration]=cest(P,M); %Calculate the c values
    if sw==1 %Enrolled in cest func and this represent the c value
    found for a
        %given P=[p1,p2,p3]
        sw=10;km=km+1; c=[real(estc),imag(estc)];
```

1.2 Results

This program yield estimated c values versus wanted Re numbers and use rci and cest function to calculate the c values. These are mentioned following section.

2- Calculating the c values for any given Re number.

Muller's method used to approximate the c values and Fourth-order Runge-Kutta (RK4) method used to calculate the c value with the aid of Ng-Reid method.

2.1 Program

2.1.1 Main program

```
global A Re C
tol=1e-4;
'***IF FIRST RUN THIS PROGRAM, PLEASE RUN INIT.M TO OBTAIN
RCINI.MAT ,THEN RUN THIS ONE****'
Re =input(' Please Enter Reynolds Number ');
load rcini %Enrolled in rcini func that supply required Re for rf
control
rf=find(re==Re); %rf control: if c value yielded for a given Re
, this
                %supply not do process
if size(rf)~=0
    '****This result had already been obtained for a given Re****'
    result=[re(rf) H(rf,:)]
    %break
end
[P,M]=ini(Re/1000); %Supply M and initial value for approximate
to c
[estc,iteration]=cest2(P,M,tol) %Calculate the c values using
mentioned above
                                %techniques
c=[real(estc),imag(estc)]; % this values saved for the purpose of
viewing
H1=[tol M c iteration];
H=[H;H1];re=[re;Re];rc=[re H];
save Res.dat rc -ascii -double
    . . . . .
```

3. Used functions and program for calculate the c values in main program.

3.1 Program of init.m

Supply the initial data required main program and viewing

```
global Re
Re=2000;re=Re;r=Re/1000;
[P,M]=ini(r);
[estc,iteration]=cest2(P,M,tol)
c=[real(estc),imag(estc)];
H=[tol M c iteration];
rc=[re H];
save Res.dat rc -ascii
save rcini re H
```

3.2 Function of ini.m

Give the initial $P=[p_1,p_2,p_3]$, used Muller's method to approximate to c , and M value. Use linear interpolation to find the P values.

```
function [P,M]=ini(r)
global Re
load rey_cest
re=re2;C=C12;
r2=r+0.52;r3=r-0.1;
if r>=2 & r<=2.1 r3=2.001;end
if r <= 5 M=400; end
if (r >5 & r <10) M=1200; end
if (r >=10 & r <13) M=1400; end
if (r >=13 & r <60) M=1500; end
if (r >=60 & r <=150)M=1600;end
if (r >150 & r <500)M=1600;end
if (r > 500) M=2000; end
if (r > 750) M=5000; end
c1=interp1(re,C,r,'linear');
c2=interp1(re,C,r2,'linear');
c3=interp1(re,C,r3,'linear');
P=[(c1(1)+c1(2)*i);(c2(1)+c2(2)*i);(c3(1)+c3(2)*i)];
```

3.3 Function of orr.m

Give the function value for a given Re , α and c that required to RK4

```
function F=orr(y,Z)
global A Re C
U=1-y^2;U2=-2.0;
A0=A^4-1.0i*A*Re*((C-U)*A^2-U2);
A2=1.0i*A*Re*(C-U)-2.0*A^2;
y1=Z(2);
y2=Z(3)+Z(4);
y3=Z(5);
y4=-A2*Z(2)+Z(5);
y5=A0*Z(1)-A2*Z(3)+Z(6);
y6=A0*Z(2);
F=[y1,y2,y3,y4,y5,y6];
```

3.4 Function of rks4.m

Give the ϕ value with using RK4 method.

```
function Z1=rks4(F,a,b,Za,M)
%Input - F is the system input as a string 'F'
%       - a and b are the end points of the interval
%       - Za=[x(a) y(a)] are the initial condition
%       - M is the number of subinterval (steps)
%Output- Z=[x1(t)...xn(t)];where xk(t) is the approximation
%         to the kth dependent variable
h=(b-a)/ M; y=a; Z=Za;
for j=1:M
    k1=h*feval(F,y,Z);
    k2=h*feval(F,y+h/2,Z+k1/2);
    k3=h*feval(F,y+h/2,Z+k2/2);
    k4=h*feval(F,y+h,Z+k3);
    Z=Z+(k1+2*k2+2*k3+k4)/6;
    y=y+h;
end
Z1=Z(1);
```

3.5 Function of cest2.m

Give the c value against to Re with using fourth-order Runge-Kutta and Muller's method

```
function [estimatedc, iteration]=cest2(P,M,tol)
global A Re C
A=1.0;y1=-1;y2=1;F='orr';cr=1e5;ci=1e5;max1=8;it=0;
fi=[(0+0i) (0+0i) (0+0i) (0+0i) (0+0i) (1+0i)];
if Re >=150000 & Re<=350000 fi=1e-100*fi;
elseif Re > 350000 & Re<650000 fi=1e-200*fi;
elseif Re >= 650000 fi=1e-300*fi;
end
Y=zeros(3,1);
for k=1:3
    C=P(k); Yy=rks4(F,y1,y2,fi,M);Y(k,1)=Yy;
end
for k=1:max1
    it=it+1; hi=P(3)-P(2); hi1=P(2)-P(1);
    ei=hi/hi1;di=1.0+ei; gi=Y(1)*ei^2-Y(2)*di^2+Y(3)*(ei+di);
    if abs(gi)>1e153
        '*****Suitable c not found,because of arithmetic overflows.*****'
        '**** (1e54^2 can not be calculalate)****'
        break; end
    temp=sqrt(gi^2-4.0* Y(3)*di*ei*(Y(1)*ei-Y(2)*di+Y(3)));
    denom=gi+temp; gim=gi-temp;
    if abs(denom)<abs(gim) denom=gim; end
    if abs(denom)==0.0 h=hi; C=P(3)+h;
    else e=-2.0*Y(3)*di/denom; h=e*hi; C=P(3)+h;
end
P(1)=P(2);P(2)=P(3); P(3)=C;Y(1)=Y(2); Y(2)=Y(3);
xy=rks4(F,y1,y2,fi,M);
Y(3)=xy; croid=cr; ciold=ci; cr=abs(real(C));
ci=abs(imag(C)); resid1=abs(cr-croid)/croid;
resid2=abs(ci-ciold)/ciold; Result(it,:)= [it C];
if resid1<tol & resid2<tol
    Result; iteration = it;
    estimatedc = C;
    break; end
end
if it>max1 '---Suitable c not found---' end
```


1Appendix-C

Stability Curve Analyses

1- Initial estimate program and curves for $ci = 0$ values.

Programs

Initial_estimate.m

```
global A Re sw tol
tol = 1e-4;
A = input(' Please Enter A Number ');
while A > 0.5
    A = A - 0.5 ;
    iter = 1; sr = 0; km = 2; kr = 1; tr = 1; kk(1) = 0; sw = 10; tp = 0; fg = 0; ty = 0; ty2 = 0;
    fg2 = 0;
    H = esinit; %Obtain estimated c value, used below, for Re = 3000
    rr = 1000*[3,6,10,15,20,30,40,50,60,70,80,90,100,...
        110,120,130,140,150,160,175,200,225,250,...
        275,300,325,350,375,400,425,450,475,500,...
        525,550,575,600,625,650,675,700,725,750,...
        775,800,825,850,875,900,925,950,975,1000,...
        1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,...
        2200,2400,2600,2800,3000,3200,3400,3600,3800,4000,4200,...
        4400,4600,4800,5000,5200,5400,5600,5800,6000,6200,6400,6600,...
        6800,7000,7200,7400,7600,7800,8000,8200,8400,8600,8800,9000,...
        9200,9400,9600,9800,10000,10500,11000,11500,12000];
    load reynoldi_cest %Yielded with esinit function at Re = 3000, supply c and re =
    Re.Estimate
        % C2 = c value using linear interpolation for later Re
    r = rr(iter)
    Re = r; re = [re;Re];
    C2 = c(end-1,:) + (Re-re(end-2))/(re(end-1)-re(end-2))*(c(end,:)-c(end-1,:));
    cc = [c; C2]; % Assigning this c in C12 that required, below, interpolation
    while r <= 1e6
        iter = iter + 1; %Obtain M number for a given Re
        if r <= 5e3 M = 400;
        elseif (r > 5e3 & r < 10e3) M = 1200;
        elseif (r >= 10e3 & r < 13e3) M = 1400;
        elseif (r >= 13e3 & r < 60e3) M = 1500;
        elseif (r >= 60e3 & r <= 150e3) M = 1600;
        elseif (r > 150e3 & r < 500e3) M = 1600;
        elseif (r > 500e3 & r < 650e3) M = 2000;
        else M = 3000;
    end
    if sw == 10 %sw enrolled in rci func that control the c is founded or not
        r2 = r + 100.8; r3 = r - 180.69; %Required linear interpolation
```

```

end
c1 = interp1(re,cc,r,'linear');
c2 = interp1(re,cc,r2,'spline'); % No c value Re=Re+.. so I use the spline interpolation
c3 = interp1(re,cc,r3,'linear');
P = [(c1(1)+c1(2)*i);(c2(1)+c2(2)*i);(c3(1)+c3(2)*i)] % Estimated initial c
                                     % value P = [p1, p2, p3]

[estc,iteration] = cest(P,M); %Calculate the c values
if sw == 1 %Enrolled in cest function and this represent the c value found for a taken
    % P = [p1, p2, p3]
    sw = 10; km = km + 1; c11 = [real(estc),imag(estc)]; c = [c;c11]; tp = 1;
    H(km,:) = [Re M c11 iteration kr];
    itr = [itr;kr]; kr = 1; result2 = [re c itr];
    sa = essve(H,re,c,itr) % Obtained c is saved for using later iteration
end
if Re == 875000 break;
end
if tp == 1
    tp = 2; Re = rr(iter-sr); re = [re;Re];
    C2 = c(end-1,:)+(Re-re(end-2))/(re(end-1)-re(end-2))*(c(end,:)-c(end-1,:));
    cc = [c;C2]; % Assigning this c in C12 that required, below, interpolation
end
if sw == 0 %Enrolled in cest function and this represent the c value not found for a
taken
    %P=[p1,p2,p3], so change the Re values and aproximate the c value for interpolating.
    kr = kr + 1; %Use obtainig the how many iteration doing for finding c value
    sr = sr + 1;
    r2 = r2 + 108 + kr*45
    r3 = r3 - 209 - kr*50
end
    r = rr(iter-sr) %sr control the received r value, supply if sw=0, r = present
    %else r = next r for supplying iteration

end
end

```

1.2 Graphic supply the plotting ci value versus Re number in initial interval for wiewing.

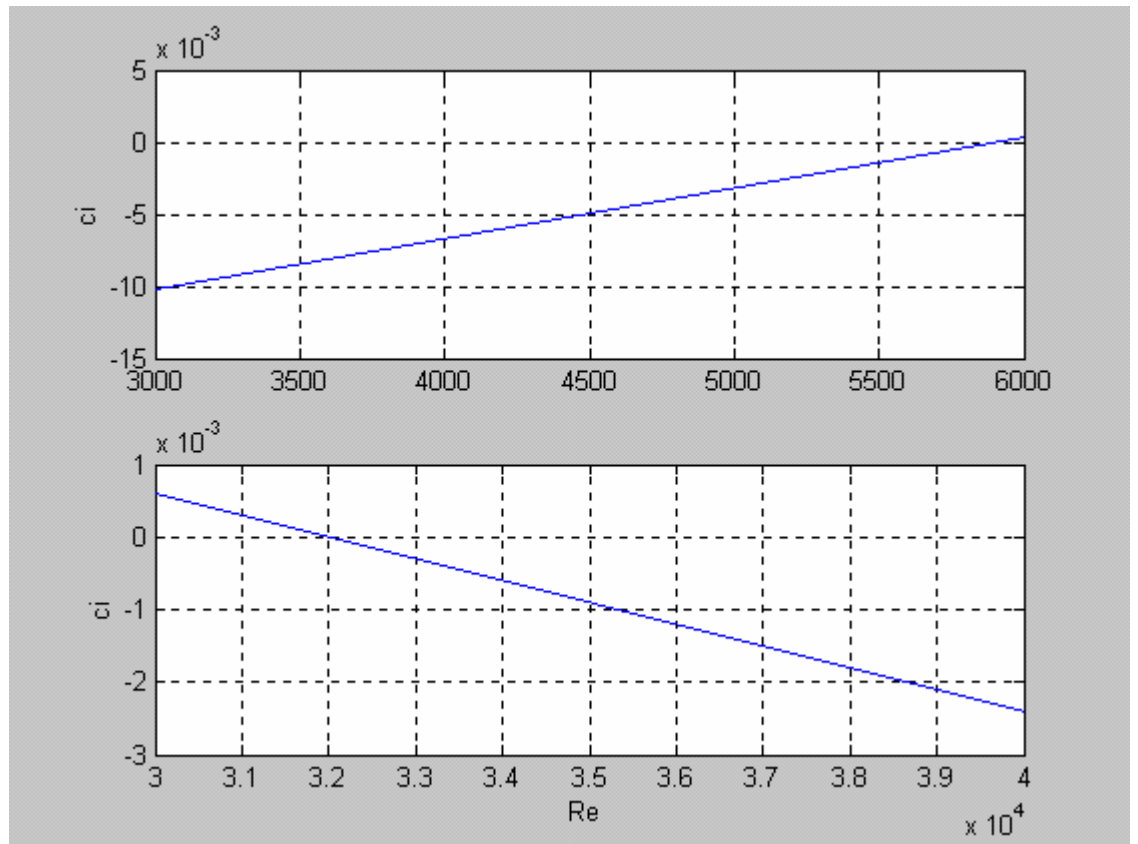


Figure C.1. Estimated initial interval area for $ci = 0$

2- Program that stretch initial estimated Re values for more approximate to $ci = 0$ values.

The function `src.m` in `srcrsi.m` function supply this situation

2.1 Function `srcrsi.m` supply initial estimate curve and initial estimate interval.

```
function [re1,re2] = srcrsi
[ re, c] = ld; cc = [re, c ]; cf1 = find (cc(:,3) > 0); cf2 = find (cc(:,3) < 0);
rcr1low = cc(cf1(1)-1,:); rcr1up = cc(cf1(1),:);
rcr2low = cc(cf1(end),:); rcr2up = cc(cf1(end)+1,:);
rc1 = [rcr1low(1,1); rcr1up(1,1)]; relow = [min(rc1); max(rc1)];
rc2 = [rcr2low(1,1); rcr2up(1,1)]; rcup = [min(rc2); max(rc2)];
f1 = find (reLOW(1) == cc); f2 = find (reLOW(2) == cc);
g1 = cc(f1,:); g2 = cc(f2,:); f3 = find (rcup(1) == cc);
f4 = find (rcup(2) == cc) ; g3 = cc(f3,:); g4 = cc(f4,:);
r1 = reLOW(:,1); ci1 = [g1(3) g2(3)]; r2 = rcup(:,1);
ci2 = [ g3(3) g4(3) ]; r11 = r1(1); r12 = r1(2);
r21 = r2(1); r22 = r2(2);
subplot(2,1,1);plot(r1,ci1);grid ;subplot(2,1,2);plot(r2,ci2);grid;
re1 = src (r11 r12); re2 = src (r21 r22); save resi re1 re2
```

2.2 Function src.m supply to stretch the initial estimated interval, shown in above graph, for more quickly approximation the Re number supplied $ci = 0$ value (Re_{cr}).

```
function rs = src(r1,r2)
[re,c] = ld; r12 = r2-r1; inc = 1;
if r12 > 5000 inc = r12/50; end
for k=1:inc:r12+1
    r = r1 + k; c1 = interp1(re,c(:,2),r,'linear');
    rr(k) = r; c11(k) = c1;
end
rf1 = find(c11<0); rf2 = find(c11>0);
[x1 y1] = max(c11(rf1)); [x2 y2] = min(c11(rf2));
re1 = rr(rf1(y1)); ci1 = c11(rf1(y1))
re2 = rr(rf2(y2)); ci2 = c11(rf2(y2))
ri = [re1,re2]; rs = [min(ri),max(ri)];
```

3 Programs that give approximated Re_{cr} values.

stability.m

```
global Re tol ch chw A
tol = 0.0001; it = 0; kr = 1; kr2 = 1; chw = 0; rt = 0; Rec = 0; ric = 0; rey2 = 0; cci = 0;
df = 1000; tp = 0; rd1 = 0; rd2 = 0;
A = input(' Please Enter A Number ');
[ re1, re2 ] = srcrsi
load resi
for z = 1:2
    rr = [ re1;re2 ];
    kk = rr(z,:);
    [ Re, ci, inc ] = sfapr(kk)
    load sfapri
    it1 = [1;2];
    if ch == 0 rd = [A Re ci]; Rec = rd; break;
end
for k = 1:100
    if chw == 566
        it = it + 1
        [P,M] = ini;
        [ estc, iteration ] = cest(P, M);
        c1 = [real(estc), imag(estc)]; c = [c; [c1,iteration]]; re = [re;Re]; yc = c1(2);
        it1 = [it1;2+it]; ric = [it1 re c ];
        w = [c(end-1,2); c(end,2)]; w1 = w(1); w2 = w(2);
        save sfaprcon.dat ric -ascii
        save sfaprcon re c
    end
    if chw == 566 & abs(yc) <= 1e-10 Rec(z,:) = [Re c1], chw = 50;
end
```

```

if chw == 566 & ch == 100 | ch == 300
    kr2 = kr2 + 1; inc = inc + 70*kr2; Re = Re - inc;
elseif chw == 566 & ch == 200 | ch == 400
    kr2=kr2+1;inc=inc + 70*kr2; Re=Re + inc;
end
if chw == 566 & w1*w2 < 0
    chw = 66
end
if chw ~= 566 & chw ~= 166 break;
end
if chw==166 w1=ci(end-1);w2=ci(end);chw=66; end
if chw==66
    it=0
    rt = re(end-1:end);
    r1=rt(1); r2=rt(2);
    ya=w1;yb=w2;
    max1=1+round(log(abs(r2-r1))/log(2))
    max1=100
end
for l=1:max1
    if rt == 0 break;
    end
    it = it + 1
    r3 = (r1 + r2)/2; Re = r3;
    [P,M] = ini; [estc, iteration] = cest(P, M);
    c1 = [real(estc), imag(estc)]; yc = c1(2);
    if abs(yc) <= 1e-10
        r1 = r3; r2 = r3;
    end
    if yb*yc > 0
        r2 = r3; yb = yc;
    else
        r1 = r3; ya = yc;
    end
    if r2 - r1 == 0
        kp = [kp;(1+it2(end)) Re yc];
        save secant.dat kp -ascii, break,
    end
    rey2(l,1) = Re;
    cci(l,1) = yc;
    it2(l,1) = it + it1(end);
    kp = [it2 rey2 cci];
    save secant.dat kp -ascii
    end
    it = 0; it2 = 0; rey2 = 0; cci = 0; tp = 1;
    rd1(z,1) = r3; rd2(z,1) = yc; Rec = [rd1 rd2];

```

```

    if tp == 1 & z == 1  sv = ssve(kp,ric);
    end
    end
    Rec = [[A;A] Rec]
    sv = ssve2(kp,ric,Rec)

```

3.1 Results

Obtained result for first $ci = 0$

Aproximated_Re = 5814.79

$ci = -9.938494243716746e-011$

Obtained result for second $ci = 0$

Aproximated_Re = 31975.03

$ci = 4.224670308485941e-011$

3.2 Compare

This value is convenient according to the below graph[20].

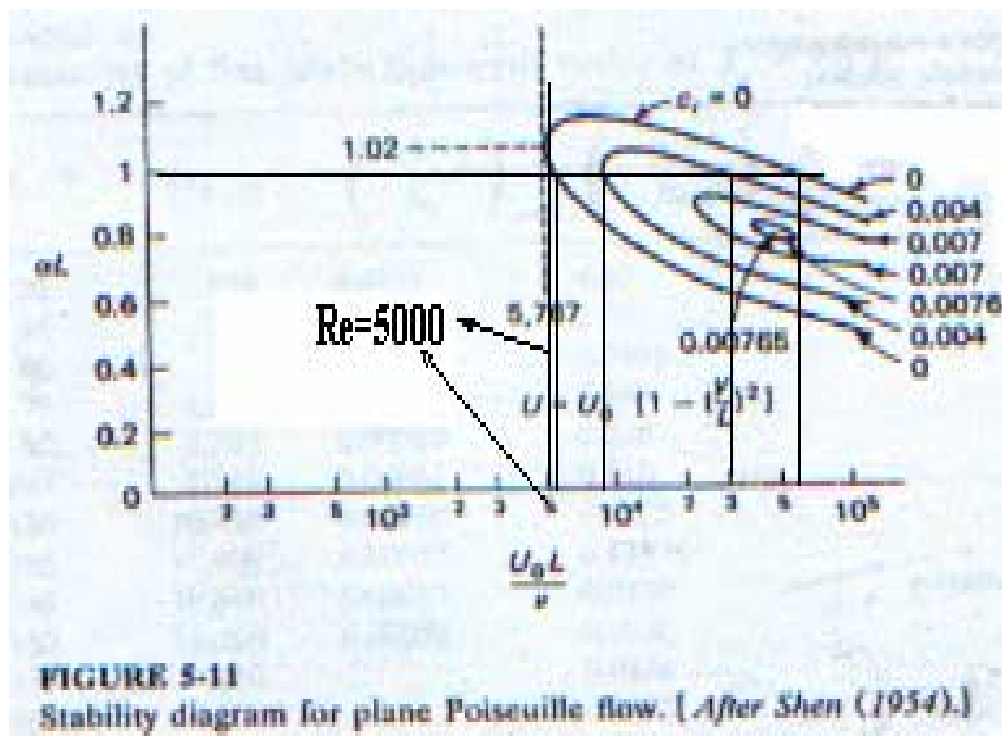


Figure C.2. Stability diagram for plane Poiseuille flow (After Shen(1954)[28]) .

4 Stability Curve.

This yielded using results of the stabilitiy.m program. Program of stabilitycurve.m yield stability curve. These results and curve are given following shape.

A number	Re	ci value
1.09100000e+000	1.0996891e+004	3.8987897e-012
1.09100000e+000	1.0996891e+004	8.5514623e-012
1.08900000e+000	6.9445902e+003	-4.4928996e-011
1.08900000e+000	1.1458954e+004	-7.7700488e-011
1.08700000e+000	6.8115875e+003	2.8484332e-011
1.08700000e+000	1.1896774e+004	-6.0925689e-011
1.08500000e+000	6.6989769e+003	-1.2423193e-011
1.08500000e+000	1.2318876e+004	-2.4969917e-011
1.08300000e+000	6.6017061e+003	-8.8395739e-011
1.08300000e+000	1.2730397e+004	-2.7697959e-011
1.08000000e+000	6.4776333e+003	6.8150924e-011
1.08000000e+000	1.3334225e+004	-9.9679525e-012
1.06000000e+000	6.0049642e+003	4.2374952e-011
1.06000000e+000	1.7308862e+004	7.1048371e-012
1.04000000e+000	5.8195575e+003	3.1410359e-011
1.04000000e+000	2.1582842e+004	5.4733883e-011
1.02000000e+000	5.7722010e+003	3.9417719e-011
1.02000000e+000	2.6412461e+004	-3.6011350e-012
1.00000000e+000	5.8147938e+003	8.7696499e-012
1.00000000e+000	3.1975030e+004	-6.5705883e-012
9.50000000e-001	6.2075371e+003	-5.4064608e-011
9.50000000e-001	5.0310148e+004	-5.5015228e-011
9.00000000e-001	6.9653006e+003	-6.5776717e-011
9.00000000e-001	7.8029880e+004	6.0856126e-011
8.50000000e-001	8.1409351e+003	-1.5734740e-012
8.50000000e-001	1.2122439e+005	-5.4008594e-011
8.00000000e-001	9.8826591e+003	-2.5370501e-011
8.00000000e-001	1.9082982e+005	-6.5277305e-012
7.50000000e-001	1.2461614e+004	-3.5594826e-011
7.50000000e-001	3.0915669e+005	-9.9280139e-011
7.00000000e-001	1.6355906e+004	7.4305201e-011
7.00000000e-001	5.0302460e+005	7.6720520e-011
6.50000000e-001	2.2427481e+004	-8.2805651e-011
6.50000000e-001	8.0868672e+005	-3.3718848e-011
6.00000000e-001	3.2301479e+004	1.2678143e-011
6.00000000e-001	1.4070146e+006	-2.2508037e-010

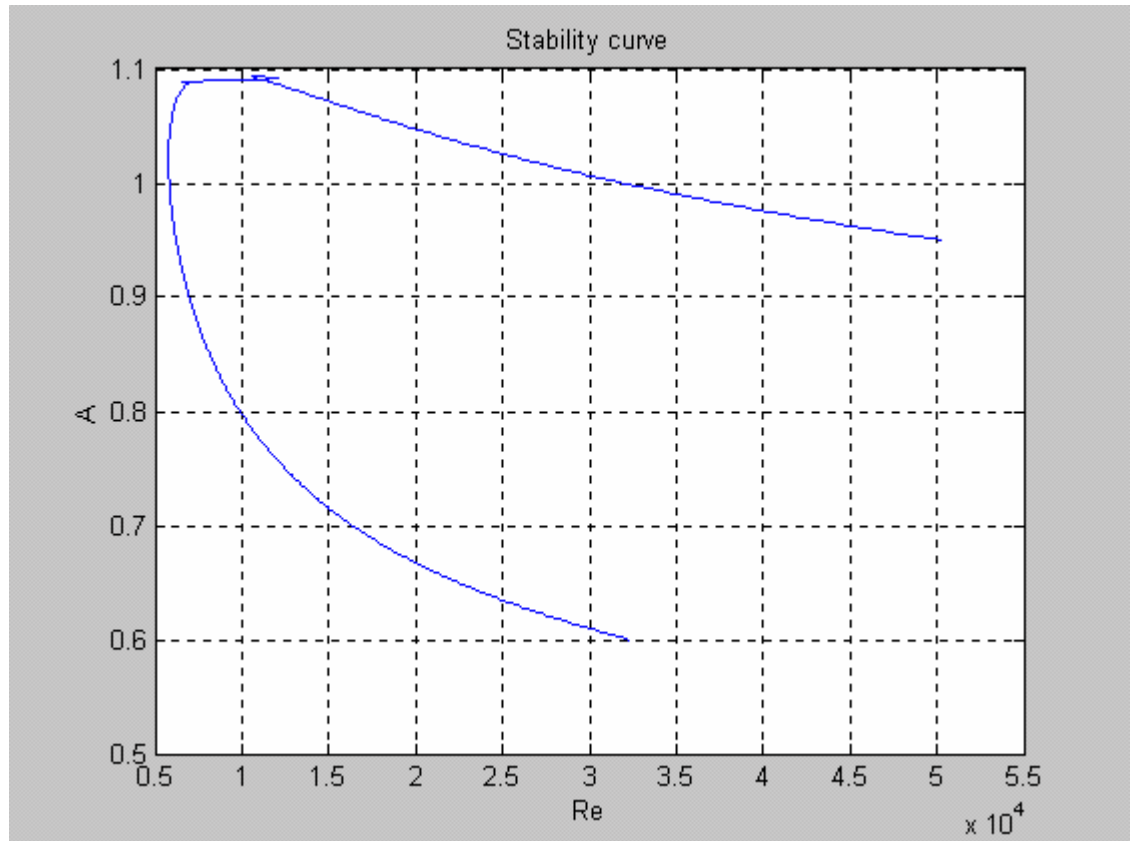


Figure C.3. Stability diagram for plane Poiseuille flow .

BIOGRAPHY

Oğuzhan Selçuk was born in Ankara, in 1975. He graduated from The High School of İnegöl Lisesi in 1992. He obtained his BSc degree from İ.T.U., Department of Aeronautical Engineering in 1998 and started his graduate education in İ.T.U., Institute of Science and Technology, Aeronautical Engineering Department. He has been working in T.E.I as a process engineer since 2002.