AN ANALYSIS OF BLASIUS BOUNDARY LAYER SOLUTION WITH DIFFERENT NUMERICAL METHODS

MUSTAFA SAIFUDEEN BIN ABDUL WALID

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: AN ANALYSIS OF BLASIUS BOUNDARY LAYER SOLUTION WITH DIFFERENT NUMERICAL

SESI PENGAJIAN: 2011/2012

Saya, **MUSTAFA SAIFUDEEN BIN ABDUL WALID (890901-11-5209)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda / ~~Sarjana / Doktor Falsafah~~)* ini disimpan di perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis ini adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

[ ]	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
[ ]	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan)
[√]	**TIDAK TERHAD**	

Disahkan oleh:

____________________	____________________
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat Tetap:	
2899, Desa Sri Bayas Permai, **21100 Kuala Terengganu** **Terengganu**	**IDRIS BIN MAT SAHAT** (Nama Penyelia)
Tarikh: **22 JUNE 2012**	Tarikh: **22 JUNE 2012**

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

◆ Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

AN ANALYSIS OF BLASIUS BOUNDARY LAYER SOLUTION WITH DIFFERENT NUMERICAL METHODS

MUSTAFA SAIFUDEEN BIN ABDUL WALID

Report submitted in partial fulfillment of the requirement
for the award of Bachelor of Mechanical Engineering

Faculty of Mechanical Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2012

UNIVERSITI MALAYSIA PAHANG
FACULTY OF MECHANICAL ENGINEERING

I certified that the report entitled "An Analysis of Blasius Boundary Layer Solution with Different Numerical Methods" is written by Mustafa Saifudeen bin Abdul Walid. I have examined the final copy of this report and in my opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. I herewith recommend that it be accepted in partial fulfillment of the requirements for the degree of Bachelor of Mechanical Engineering.

PROF DR. ABDUL GHAFFAR BIN ABDUL RAHMAN
Examiner Signature

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project report and in our opinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering.

Signature :
Name of Supervisor : IDRIS BIN MAT SAHAT
Position : LECTURER
Date : 22 JUNE 2012

STUDENT'S DECLARATION

I hereby declare that the work in this report is my own except for quotations and summaries which have been duly acknowledged. The report has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature :
Name : MUSTAFA SAIFUDEEN BIN ABDUL WALID
ID Number : MA08126
Date : 25 JUNE 2012

To my beloved parents, siblings and friends

ACKNOWLEDGEMENTS

Bismillahirrahmanirahim, In the name of Allah, the Most Gracious, the Most Merciful and peace be upon Prophet Muhammad Sallallahu 'alaihi wassalam.

Alhamdulillah, I would like to extend my gratitude to many people for the completion of this thesis in due course of time. Firstly I would like to thank and express my sincere gratitude to my supervisor; Mr. Idris Bin Mat Sahat for his valuable guidance, endless support and encouragement in completing this thesis. Besides that, guidance from Dr. Mohd Zuki Bin Salleh in the usage of Maple software was much appreciated.

Thank you to all my friends for their support. Last but not least, I would like to thank my family and those who are involved either directly or indirectly along the completion of this project and thesis.

ABSTRACT

The nonlinear equation from Prandtl has been solved by Blasius using Fourth order Runge-Kutta methods. The thesis aims to study the effect of solving the nonlinear equation using different numerical methods. Upon the study of the different numerical methods be use to solve the nonlinear equation, the Predictor-Corrector methods, the Shooting method and the Modified Predictor-Corrector method were used. The differences of the methods with the existing Blasius solution method were analyzed. The Modified Predictor-Corrector method was developed from the Predictor-Corrector method by adjusting the pattern of the equation. It shows the graphs of the f, f' and f'' against the *eta*. All the methods have the same shape of graph. The Shooting method is closely to the Blasius method but not stable at certain value. The Variational Iteration method that has been used cannot be proceeding because the method only valid for the earlier flows and lost the pattern at the higher value of eta. It can be comprehend that the Predictor-Corrector methods, the Shooting method and the Modified Predictor-Corrector method achieve the conditions and can be applied to solve the nonlinear equation with minimal differences. The methods are highly recommended to solve the Sakiadis problem instead of the stationary flat plate problem.

ABSTRAK

Persamaan tidak linear daripada Prandtl telah diselesaikan oleh Blasius dengan menggunakan kaedah penyelesaian Runge-Kutta keempat. Kaedah penyelesaian persamaan tidak linear tersebut dikaji melalui penggunaan kaedah penyelesaian berangka yang berbeza di dalam tesis ini. Kaedah Peramal-Pembetul, Kaedah Tembakan dan juga Kaedah Ubahan Peramal-Pembetul telah digunakan. Perbezaan kaedah-kaedah ini dengan kaedah yang sedia ada Blasius di analisis. Kaedah Ubahan Peramal-Pembetul telah dikeluarkan daripada kaedah asal Peramal-Pembetul dengan melaraskan corak persamaannya. Ia menunjukkan graf f, f' dan f'' terhadap *eta*. Semua kaedah mempunyai bentuk graf yang sama dengan kaedah penyelesaian Blasius. Kaedah Tembakan adalah paling hampir dengan kaedah penyelesaian Blasius namun terdapat sedikit ketidakstabilan pada titik-titik tertentu. Kaedah Lelaran Perubahan pula telah digunakan namun tidak dapat diteruskan kerana kaedah ini hanya sah pada aliran permulaan sahaja dan hilang corak pada nilai eta yang lebih tinggi. Kaedah Peramal-Pembetul, Kaedah Tembakan dan juga Kaedah Ubahan Peramal-Pembetul mencapai syarat-syarat dan boleh digunakan untuk menyelesaikan persamaan tidak linear dengan perbezaan yang kecil. Kaedah-kaedah ini amat disyorkan untuk menyelesaikan masalah Sakiadis iaitu plat rata yang tidak statik.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT BACKGROUND

The research work involved the analysis of Blasius boundary layer solution. Blasius come out with the solution of the Prandtl theory of boundary layer. Prandtl deriving the momentum equation into the final boundary layer equation on the flat plate. The equation is in the form of nonlinear third order ordinary differential equation. Blasius then solve the equation using numerical methods.

1.2 PROBLEM STATEMENT

There are no exact values when solving the numerical methods. Using different types of numerical methods will give the different results and error. The objective includes seeing the pattern of difference between the methods. Furthermore, the numerical solution is too much hard to be solving manually by hands. There should be a proper way to solve it.

1.3 OBJECTIVE

The objective of this project is to study the result of different type of numerical methods towards Blasius solution.

1.4 SCOPE

The project scope is firstly to know briefly about the boundary layer theory and how the boundary layer happened. The boundary layer that formed allowed us to determine the values that related; as example temperature, pressure and etc. Hence, the theory that comes out from Prandtl later being solved by the Blasius to be derived and proved with numerical methods. Once the methods proven, there should be a proper way proposed to solve the equation usually using software; as example *Fortran*, *C++* or *MATLAB.*

1.5 FLOW CHART

Figure 1.1 shows the project flow chart for this Final Year Project (FYP) 1. A first meeting has been arranged with the supervisor to discuss about the project title. I have required finding any related article, journal or references related to the project title. Then the proposal can be start to write containing introduction, literature review and methodology. The FYP1 will be ending with the presentation to the panel on week of 14^{th} on this semester.

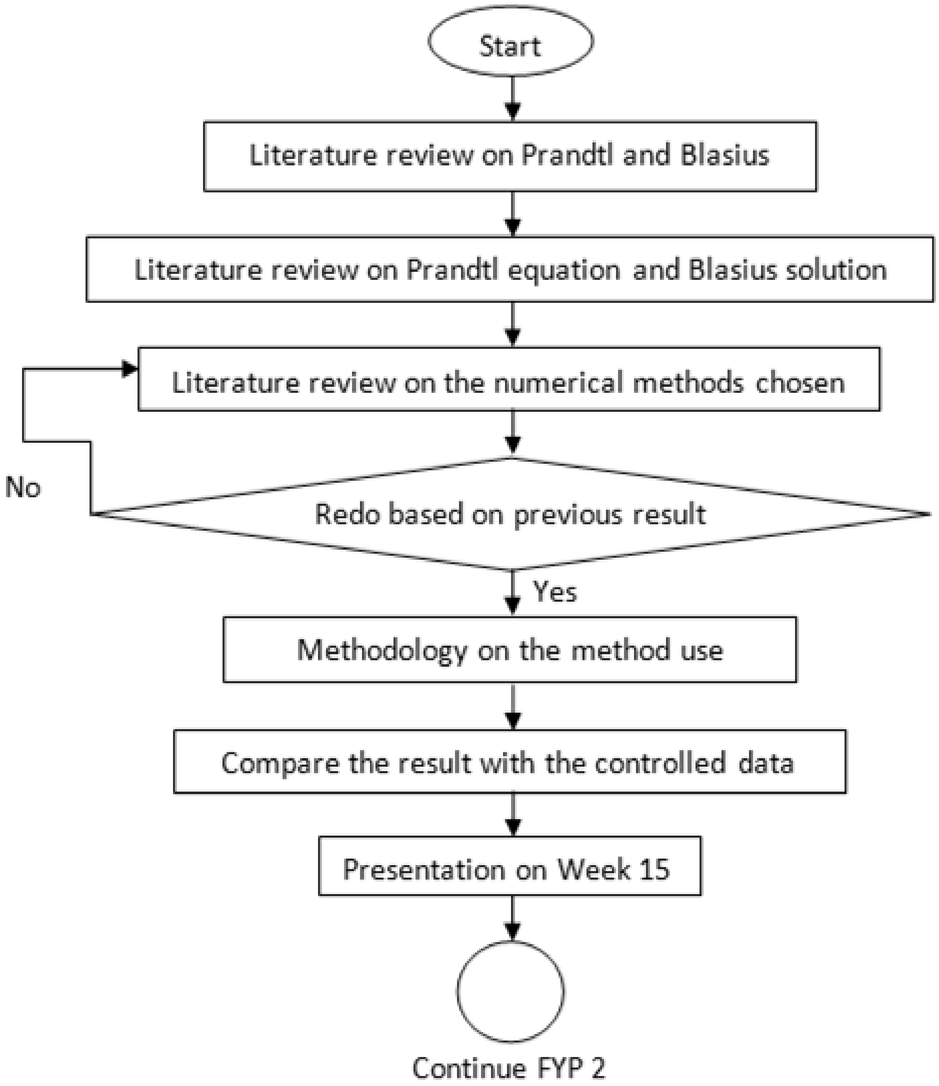


Figure 1.1: Project Flow Chart

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The literature review consists of the brief explanations of elements that related to this project. The analysis of Blasius boundary layer solution is related to the boundary layer theory and also boundary layer equation. The research of the boundary layer was done by the German scientist, Ludwig Prandtl with his presented benchmark paper on boundary layer in 1904 (Prandtl 1904). Later, solution of the boundary layer theory was done by his student, Blasius. In solving the boundary layer theory, several approximation that eliminate terms reducing the Navier-Stokes equation to a simplified form that is more easily solvable.

The mathematical parts of this project include the numerical methods. There are quite a number of numerical methods to solve the differential equation. As the boundary layer equation is in third order ordinary differential equation, the numerical method such as Runge-Kutta, Euler and also Predictor-Corrector methods are the available method that solve the problem. These elements will be briefly discussed in the further part of this chapter.

2.2 HISTORY

2.2.1 Sir Ludwig Prandtl

Sir Ludwig Prandtl was born in Freising, Bavaria (Beyond the Boundary Layer Concept). His father, Alexander Prandtl, was a professor of surveying engineering at

The Agricultural College at Weihenstephen, near Freising. The Prandtls had three children, but two died at birth and Ludwig grew up as an only child. His mother suffered from a protracted illness, Ludwig became very close to his father. He became interested in his father's books on physics, machinery and instruments at an early age.

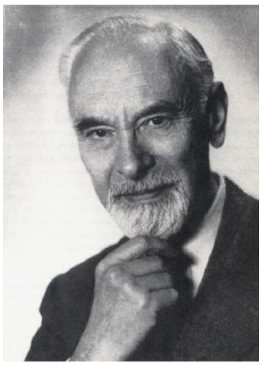


Figure 2.1: Ludwig Prandtl

(Source:Anderson, 2005)

Table 2.1: Prandtl's Chronology

Year	Events
1875	February 4 – born in Freising, Bavaria
1894	Begin scientific studies at the Technische Hochshule in Munich with well-known mechanics Professor August Foppl
1900	Graduated with Ph.D from University of Munich Continue research in solid mechanics Joined the Nurnberg Works of Machinenfabrik Augsburg as an engineer
1901	Became professor of mechanics in The Mechanical Engineering Department at The Technische Hochschule in Hanover Develop boundary layer theory and began work on supersonic flow through nozzle
1904	Third International Mathematics Congress (Heidelberg) famous presentation fluid flow in very little friction that highlight his name on the research
1918-1919	Result on the problem of a useful mathematical tool for examining lift from real world wing were published known as Lanchester-Prandtl wing theory
1920s	Developed the mathematical basis for the fundamental principles of subsonic aerodynamics in particular; and in general up to and including transonic velocities
1953	August 15 – died in Gottingen

He spent the remainder of his life to become director of the institute for technical physics in the prestigious University of Gottingen and built his laboratory into the greatest aerodynamics research center in early 20th century.

2.2.2 Blasius

Paul Richard Heinrich Blasius was born in Berlin, Germany (Hager, 2003). He was studied at the University of Marburg and Gottingen from 1902 to 1906. Accordingly, Blasius spent only six years in science and moved to teaching which he loves it more than doing research. After World War II, Blasius was specially acknowledged for having rebuilt the lecture rooms and laboratories. Officially, he stayed at the mechanical engineering department from 1912 to 1950, and heads the department from 1945 to 1950

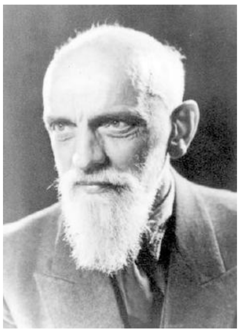


Figure 2.2: Blasius

(Source: Hager, 2003)

Table 2.2: Blasius's Chronology

Year	Events
1883	August 9 - born in Berlin, Germany
1902-1906	Studies and scientific collaborator with Ludwig Prandtl at the university of Marburg and Gottingen
1908	Research assistance at the hydraulics laboratory of Berlin Technical University.
	Paper on flow separation behind circular cylinder, development of boundary layer flow due to sudden initiation of flow and separation from a cylinder for unsteady flow
1909	Started working on Pitot tube

Table 2.2: Continue

Year	Events
1910	Second key paper- classical potential theory applied; (i)force the exerted body immersed in a fluid flow; (ii)potential flow over weirs
1911	Investigated the curve airfoil using the Kutta method.
	Blasius reconsider mathematical methods applied to potential flow and derived an expression for the force of an obstacle positioned in a stream.
1912	Publish relating friction coefficient of turbulent smooth pipe flow.
	First to derive a law relating to so-called turbulent smooth pipe flows.
	Teacher at The Technical College Of Hamburg
1931	Undergraduate books on heat transfer
1934	Undergraduate book on mechanics
1912-1950	Continued lecturing in Hamburg
1970	April 24 – passed away in Hamburg

2.2.3 Navier-Stokes Equation

The traditional model of fluids used in physics is based on a set of partial differential equations known as the Navier-Stokes equations. These equations were originally derived in the 1840s on the basis of conservation laws and first-order approximations. For very low Reynolds numbers and simple geometries, it is often possible to find explicit formulas for solutions to the Navier-Stokes equations. But even in the regime of flow where regular arrays of eddies are produced, analytical methods have never yielded complete explicit solutions. In this regime, however, numerical approximations are fairly easy to find.

The ability of computers has been capable enough to allow computations at least nominally to be extended to acceptably higher Reynolds numbers since about the 1960s. And indeed it has become increasingly common to see numerical results given far into the turbulent region that leading sometimes to the assumption that turbulence has somehow been derived from the Navier-Stokes equations. But just what such numerical results actually have to do with detailed solutions to the Navier-Stokes equations is not clear. For in particular it ends up being almost impossible to distinguish whatever genuine instability and apparent randomness may be implied by the Navier-Stokes equations from artifacts that get introduced through the discretization procedure used in solving the equations on a computer. At a mathematical level analysis of the Navier-

Stokes has never established the formal uniqueness and existence of solutions. Indeed, there is even some evidence that singularities might almost inevitably form, which would imply a breakdown of the equations.

The Navier-Stokes equation of incompressible flow of Newtonian fluid with constant properties

$$\frac{\rho D\vec{V}}{Dt} = \rho\left[\frac{\partial \vec{V}}{\partial t} + \left(\vec{V}.\vec{\nabla}\right)\vec{V}\right] = -\vec{\nabla}P + \rho\vec{g} + \mu\nabla^2\vec{V} \qquad (2.1)$$

2.3 BOUNDARY LAYER

Boundary layer is a fluid character that forms in the flow of fluid through a body of surface. For this scope of project, we are about to discuss about the boundary layer that form due to the fluid flow through a stationary and parallel flat plate. Boundary layer on a flat plate happened due to the friction of the wall and the fluid particle along it surface. The boundary layer has the characteristic of increasing the value along the static plate. In this project, the boundary layer of the flat plate incompressible flow is taken into consideration. The study was done by Prandtl. It is about when a fluid flow in a horizontal direction passing through a flat plate that in x-direction, by assuming that it is a incompressible flow, the velocity of the fluid at the surface of the plate is equal to zero. There will be a layer of boundary layer will be formed along the flat plate.

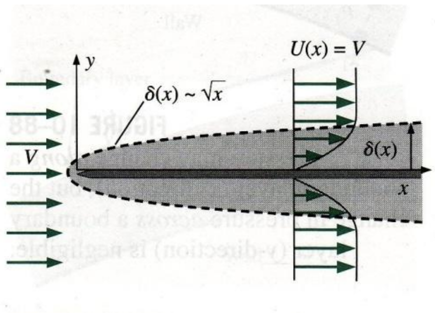


Figure 2.3: Laminar Boundary Layer along a Flat Plate

(Source: Cengel and Cimbala, 2010)

Blasius solution is about boundary layer theory of fluid flow. Blasius solution originally solves simplified momentum equation and continuity equation which were simplified by Prandtl. These simplified equations are in partial differential forms. By introducing a similarity variable, Blasius used numerical methods to solve the partial differential equations to obtain the results. Blasius solve the Prandtl boundary layer problem using 4th order Runge-Kutta numerical methods (Cengel et al., 2010).

2.4 CONTINUITY EQUATION

Continuity equation focus on conservation of mass on a motion of fluid flow with the assumption made that the flow is in steady condition which is not varying with the time. The application of continuity equation one of it is to determine the change in fluid velocity due to an expansion or contraction in the diameter of a pipe.

Incompressible continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.2}$$

x-component of the incompressible Navier-Stokes equation

$$\begin{aligned}\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}\right) \\ = -\frac{\partial P}{\partial x} + \rho g_x + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)\end{aligned} \tag{2.3}$$

y-component of the incompressible Navier-Stokes equation

$$\begin{aligned}\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}\right) \\ = -\frac{\partial P}{\partial x} + \rho g_y + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right)\end{aligned} \tag{2.4}$$

z-component of the incompressible Navier-Stokes equation

$$\rho\left(\frac{\partial w}{\partial t}+u\frac{\partial w}{\partial x}+v\frac{\partial w}{\partial y}+w\frac{\partial w}{\partial z}\right)=-\frac{\partial P}{\partial x}+\rho g_z+\mu(\frac{\partial^2 w}{\partial x^2}+\frac{\partial^2 w}{\partial y^2}+\frac{\partial^2 w}{\partial z^2}) \tag{2.5}$$

2.5 MOMENTUM EQUATION

Momentum equation is a nonlinear set of differential equation that describes the flow of a fluid whose stress depends linearly on velocity gradient and pressure. The Navier-Stokes equation is one of the momentum equations

$$\frac{\rho D\vec{V}}{Dt}=\rho\left[\frac{\partial\vec{V}}{\partial t}+(\vec{V}.\vec{\nabla})\vec{V}\right]=-\vec{\nabla}P+\rho\vec{g}+\mu\nabla^2\vec{V} \tag{2.6}$$

$$u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=\frac{\partial^2 u}{\partial y^2} \tag{2.7}$$

2.6 NUMERICAL METHODS

Many problems in science and engineering required the mathematical parts to solve the problems. For this project, the Blasius solution is a nonlinear ordinary differential equation which arises in the boundary layer flow. The method reduces solving the equation to solving a system of nonlinear algebraic equation. The equation can be solved using these numerical methods:

i. Taylor's method
ii. Fourth order Runge-Kutta Method
iii. Heun's Method
iv. Euler Method
v. Predictor-Corrector Method
vi. Shooting Method

2.7 DERIVATION OF BOUNDARY LAYER EQUATION

Given that boundary layer equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \qquad (2.8)$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = U\frac{dU}{dx} + v\frac{\partial^2 u}{\partial y^2} \qquad (2.9)$$

With boundary conditions

$$y = 0, \qquad u = 0, \qquad v = 0 \qquad (2.10)$$

$$y = \infty, \qquad u = U \qquad \frac{\partial u}{\partial y} = 0 \qquad (2.11)$$

Eq. (2.8) and Eq. (2.9), with the boundary conditions of Eq. (2.10) are in nonlinear, partial differential equations for unknown velocity field u and v. Blasius reasoned that to solve them, the velocity profile, $\frac{u}{U}$ should be similar for all values of x when plotted versus a nondimensional distance from the wall. The boundary layer thickness, δ, was a natural choice for nondimensionalizing the distance from the wall. Thus the solution is of form

$$\frac{u}{U} = g(\eta) \qquad \eta \propto \frac{y}{\delta} \qquad (2.12)$$

Based on the solution of Stokes (Fox et al., 2009), Blasius reasoned that $\delta \propto \sqrt{vx/U}$ and set

$$\eta = y\sqrt{\frac{U}{vx}} \qquad (2.13)$$

The stream function, ψ were introduced, where

$$u = \frac{\partial \psi}{\partial y} \qquad\qquad v = -\frac{\partial \psi}{\partial x} \tag{2.14}$$

satisfies the continuity equation Eq. (2.8) identically. Replacing for u and v into Eq. (2.9) reduces the equation to which ψ is the single dependent variable. The dimensionless stream function is defined as $f(\eta) = \frac{\psi}{\sqrt{vxU}}$ makes $f(\eta)$ the dependent variable and η the independent variable in Eq. (2.9) with the ψ defined by Eq. (2.13) and η defined by Eq. (2.12), we can evaluate each of the terms in Eq. (2.8).

The velocity components are given by

$$u = \frac{\partial \psi}{\partial y} = \frac{\partial \psi}{\partial \eta}\frac{\partial \eta}{\partial y} = \sqrt{vxU}\frac{df}{d\eta}\sqrt{\frac{U}{vx}} = U\frac{df}{d\eta} \tag{2.15}$$

and

$$\begin{aligned} v &= -\frac{\partial \psi}{\partial x} = -\left[\sqrt{vxU}\frac{\partial f}{\partial x} + \frac{1}{2}\sqrt{\frac{vU}{x}}f\right] \\ &= -\left[\sqrt{vxU}\frac{\partial f}{\partial x}\left(-\frac{1}{2}\eta\frac{1}{x}\right) + \frac{1}{2}\sqrt{\frac{vU}{x}}f\right] \end{aligned} \tag{2.16}$$

or

$$v = \frac{1}{2}\sqrt{\frac{vU}{x}}\left[\eta\frac{df}{d\eta} - f\right] \tag{2.17}$$

By differentiating the velocity components, it also can be shown that

$$\frac{\partial u}{\partial x} = -\frac{U}{2x}\eta\frac{d^2 f}{d\eta^2} \tag{2.18}$$

$$\frac{\partial u}{\partial y} = U\sqrt{U/vx}\frac{d^2 f}{d\eta^2} \tag{2.19}$$

and

$$\frac{\partial^2 u}{\partial y^2} = \frac{U^2}{vx}\frac{d^3 f}{d\eta^3} \tag{2.20}$$

Substituting these expressions into Eq. (2.9), yield

$$2\frac{\partial^3 f}{\partial\eta^3} + f\frac{d^2 f}{d\eta^2} = 0 \tag{2.21}$$

With boundary conditions:

$\eta = 0$

$$f = \frac{df}{d\eta} = 0 \tag{2.22}$$

$\eta \to \infty$

$$\frac{df}{d\eta} = 1 \tag{2.23}$$

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter is focusing on explaining clearly the steps taken to complete the project in order to obtain the result and discussion. The procedure must be done systematically to make sure there is no mistake and conflict on the result obtain. A good methodology can describe the project flow smoothly and the project framework that contains the process element hence it becomes the guideline to find the objective required.

3.2 METHODOLOGY FLOW CHART

The planning is very important to give an illustration about the project flow process to make sure the progress project is satisfied with the time required. The flow chart can describes the project flow and process briefly. Hence the project will run smoothly as scheduled. This methodology will show the sequence of the project flow including in method choosing, literature review on related methods, and data analysis and discussion on the result obtained.

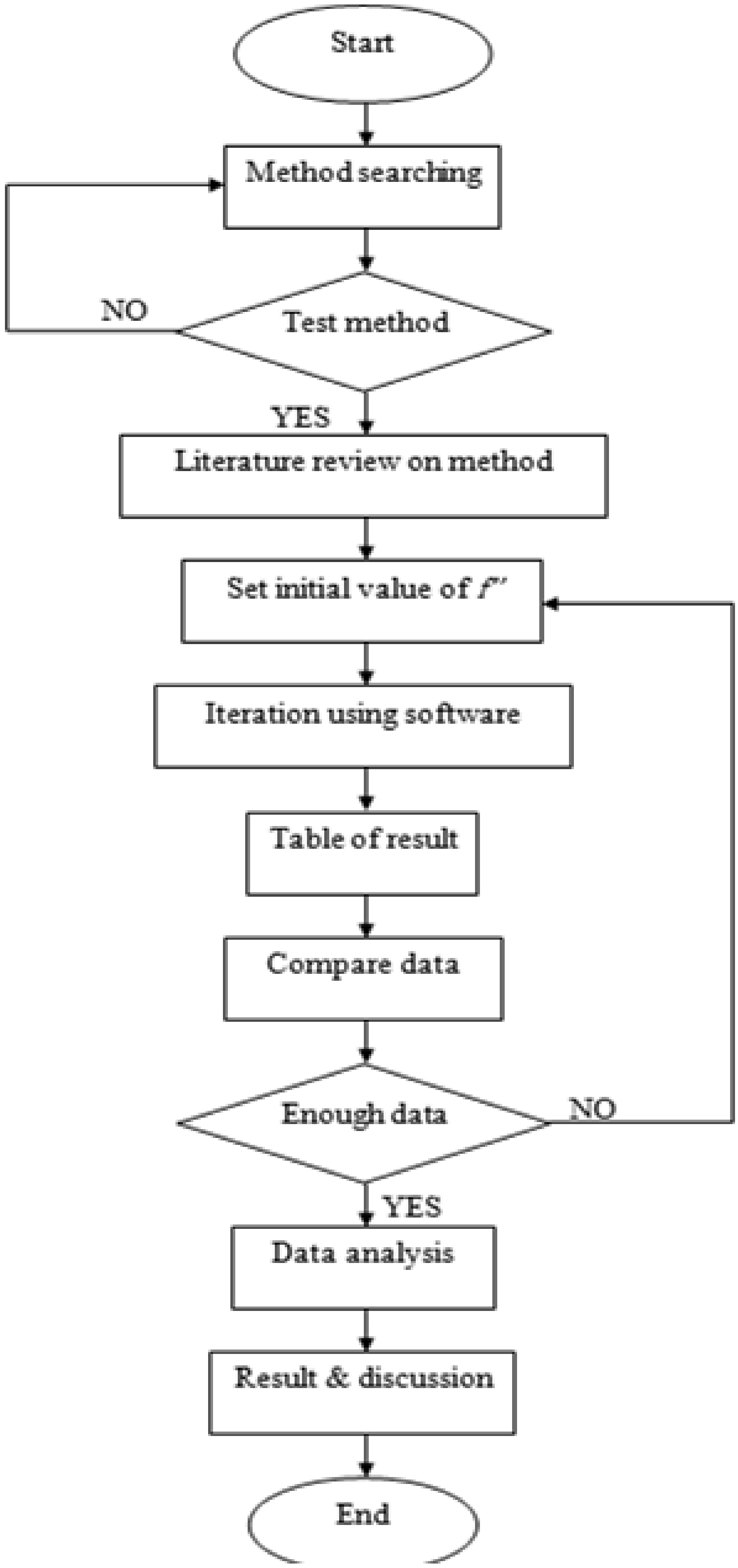


Figure 3.1: Methodology Flow Chart

3.3 LITERATURE STUDY

In order to give a more understanding based on project title a research journal, conference article, reference book and others are used as reference. The main term like boundary layer and Blasius solution are the key to the related article found. Based on the article found, it is important to know the process of method of solving equation used until the data table was gathered.

3.4 BLASIUS SOLUTION'S TABLE (CONTROLLED DATA)

Table 3.1: Result of Blasius Solution Using 4th Order Runge-Kutta Methods

TABLE 10–3

Solution of the Blasius laminar flat plate boundary layer in similarity variables*

η	f''	f'	f	η	f''	f'	f
0.0	0.33206	0.00000	0.00000	2.4	0.22809	0.72898	0.92229
0.1	0.33205	0.03321	0.00166	2.6	0.20645	0.77245	1.07250
0.2	0.33198	0.06641	0.00664	2.8	0.18401	0.81151	1.23098
0.3	0.33181	0.09960	0.01494	3.0	0.16136	0.84604	1.39681
0.4	0.33147	0.13276	0.02656	3.5	0.10777	0.91304	1.83770
0.5	0.33091	0.16589	0.04149	4.0	0.06423	0.95552	2.30574
0.6	0.33008	0.19894	0.05973	4.5	0.03398	0.97951	2.79013
0.8	0.32739	0.26471	0.10611	5.0	0.01591	0.99154	3.28327
1.0	0.32301	0.32978	0.16557	5.5	0.00658	0.99688	3.78057
1.2	0.31659	0.39378	0.23795	6.0	0.00240	0.99897	4.27962
1.4	0.30787	0.45626	0.32298	6.5	0.00077	0.99970	4.77932
1.6	0.29666	0.51676	0.42032	7.0	0.00022	0.99992	5.27923
1.8	0.28293	0.57476	0.52952	8.0	0.00001	1.00000	6.27921
2.0	0.26675	0.62977	0.65002	9.0	0.00000	1.00000	7.27921
2.2	0.24835	0.68131	0.78119	10.0	0.00000	1.00000	8.27921

* η is the similarity variable defined in Eq. 4 above, and function $f(\eta)$ is solved using the Runge–Kutta numerical technique. Note that f'' is proportional to the shear stress τ, f' is proportional to the *x*-component of velocity in the boundary layer ($f' = u/U$), and f itself is proportional to the stream function. f' is plotted as a function of η in Fig. 10–99.

(Source: Cengel, Cimbala. 2010)

3.5 PREDICTOR-CORRECTOR METHOD

The first method that has been used is the Predictor-Corrector method (Numeric Solution to Blasius Equation). Predictor-Corrector method uses the previous known value to compute for the next value. The Predictor-Corrector method needs a starting value to get the process done because it is self-generated. Predictor-Corrector methods consist of two formulas which are; a predictor formula and a corrector formula. The predictor formula extrapolates the existing data to be used to estimate the next value while the corrector formula improves the estimations (Griffiths et al., 2006). Given a first order differential equation in the standard form, $y' = f(x, y)$ with $y(x_0) = y_0$,

The predictor using the Rectangle rule

$$\bar{y}_{i+1} = y_i + hf(x_i, y_i) \tag{3.1}$$

And the corrector uses the Trapezoid rule

$$y_{i+1} = y_i + \frac{1}{2}h[f(x_i, y_i) + f(x_{i+1}, \bar{y}_{i+1})] \tag{3.2}$$

The corrector needs a prior estimation of $y_{i+1}^{'}$

For the case of Blasius flat plate solution equation that is

$$2\frac{d^3f}{d\eta^3} + f\frac{d^2f}{d\eta^2} = 0 \tag{3.3}$$

Also can be written as

$$2f''' + ff'' = 0 \tag{3.4}$$

The generate equation will be

Predictor

$$\bar{f}_{i+1} = f_i + \Delta\eta f_i' \tag{3.5}$$

$$\bar{f}_{i+1}' = f_i' + \Delta\eta f_i'' \tag{3.6}$$

$$\bar{f}''_{i+1} = f''_i + \Delta\eta f'''_i \tag{3.7}$$

$$\bar{f}'''_{i+1} = -\frac{f_{i+1} + f''_{i+1}}{2} \tag{3.8}$$

Corrector

$$f_{i+1} = f_i + \Delta\eta \frac{(f'_i + \bar{f}'_{i+1})}{2} \tag{3.9}$$

$$f'_{i+1} = f'_i + \Delta\eta \frac{(f''_i + \bar{f}''_{i+1})}{2} \tag{3.10}$$

$$f''_{i+1} = f''_i + \Delta\eta \frac{(f'''_i + \bar{f}'''_{i+1})}{2} \tag{3.11}$$

$$f'''_{i+1} = -\frac{f_{i+1} f''_{i+1}}{2} \tag{3.12}$$

The Microsoft Office Excel is used to complete the iteration for the generated equations with the initial estimation of $f''(0) = 0.332057$.

3.6 SHOOTING METHOD WITH MAPLE

Instead of using the manual iteration, the numerical methods also being solve by using software nowadays. The second methods that being used was the Shooting Method with *Maple13* software. The shooting method is to convert a boundary value problem to an initial value problem and solve the problem iteratively (Rao, 2002). The shooting method has the similarity of its procedure to the Newton-Raphson method. Hence, it is often called as Newton-Raphson method. The shooting methods get its name because of the trial and error approach is used to solve the problem as initial value problem instead of the original boundary value problem. The shooting method can be applied to linear boundary value problems.

In this project, instead of using the conventional method of solving the shooting method the computational method was used to solve the shooting method. *Maple13* software has the build in shooting method that should be called out during the programming process. The command known as *'shoot'* process the existing file as the calculation of shooting method. The result of iterations from the programming will be shown in the graph and were tabulated. The shooting method was used to determine the initial value problem to start further process. In the Blasius flat plate solution equation problem, the initial value that need to be determine in the first is the value of $f''(0)$. The following value is determined by the shooting method. Once the value of $f''(0)$ is determined, the further process of iteration will be continued to attain the complete table of differential result.

3.6.1 Iteration of $f''(0)$ Using Shooting Method

In this project includes the Shooting Method with Maple software. One of the specialty of this method compared with the other two methods is, the value of $f''(0)$ is determined by using iteration, while the other two methods using the existing value. By using the Maple software, the initial value of $f''(0)$ first is guessed by any value of its range and the software will iterate and stop when the final value of iteration is get. Hence the last value of iteration is the value of $f''(0)$.

To solve the iteration, the value of maximum of eta value also needs to be determined. From the boundary condition of Blasius equation, $f'(+\infty) = 1$, the value of infinity is unknown. In solving the ordinary differential equation of nonlinear, there is a property that reflects to this problem. In order to know what is the minimum value of infinite that can be guess, the iterated graph that being formed must be stable at a longer time. Hence the value of infinite can be guess by that value. The graph that stable at longer time is known as asymptotic by its shape (Salleh M.Z., 2012). The asymptotic can be denoted by the Figure 3.2 below:

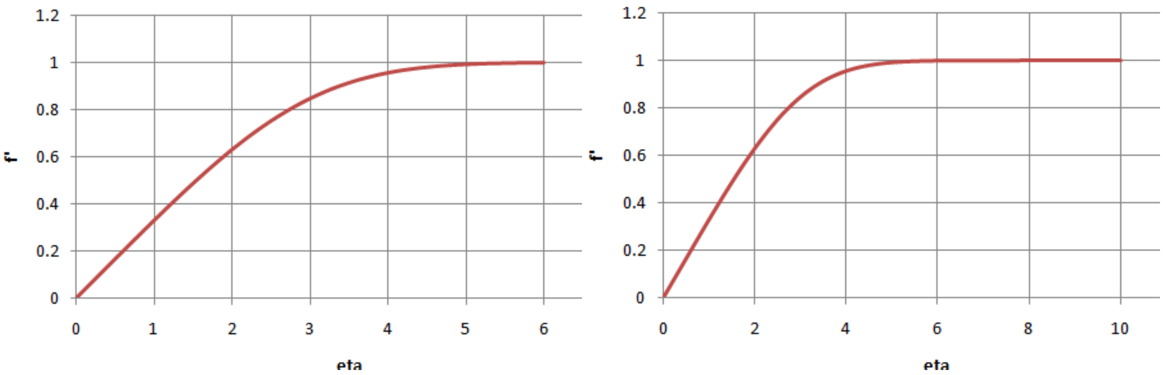


Figure 3.2: Asymptotic Graph Comparison

For the left figure, the value of $\eta = 6$ is taken as the infinite value while the right figure $\eta = 6$ is taken as the infinite value. The left on the left is not yet experience the asymptotic and hence, the iteration value of $f''(0)$ will not be accurate while the right figure can be considered as asymptotic and hence, the result to be obtained is more accurate.

3.7 PREDICTOR-CORRECTOR METHOD WITH CENTRAL DIFFERENCE

Upon solving the Blasius solution of numerical method using the existing method, this project also include a method that has been derived by the existing method that has been used before. The Predictor-Corrector method that has been used earlier shows the pattern that the equations using the forward difference method of boundary value problem. The method can be modified by using the central difference method of boundary value problem. The equation is in the shape of forward difference approximation. In this project of research, the method of forward difference was replaced by the central difference as to determine the changes of the methods to the value of iterations.

Predictor

$$\bar{f}_{i+1} = f_{i-1} + 2\Delta\eta f_i' \tag{3.13}$$

$$\bar{f}'_{i+1} = f'_{i-1} + 2\Delta\eta f''_i \tag{3.14}$$

$$\bar{f}''_{i+1} = f''_{i-1} + 2\Delta\eta f'''_i \tag{3.15}$$

$$\bar{f}'''_{i+1} = -\frac{f_{i+1} + f''_{i+1}}{2} \tag{3.16}$$

Corrector

$$f_{i+1} = f_{i-1} + \Delta\eta(f'_{i-1} + \bar{f}'_{i+1}) \tag{3.17}$$

$$f'_{i+1} = f'_{i-1} + \Delta\eta(f''_{i-1} + \bar{f}''_{i+1}) \tag{3.18}$$

$$f''_{i+1} = f''_{i-1} + \Delta\eta(f'''_{i-1} + \bar{f}'''_{i+1}) \tag{3.19}$$

$$f'''_{i+1} = -\frac{f_{i+1} + f''_{i+1}}{2} \tag{3.20}$$

The Microsoft Office Excel is used to complete the iteration for the generated equations with the initial estimation of $f''(0) = 0.332057$.

3.8 VARIATIONAL ITERATION METHOD

The Variational Iteration Method is known by the usage of the Langrangian multiplier in its derived equation (Moghimi et al., 2006). The Variational Iteration Method solves equation easily and accurately a huge class of nonlinear equation with rapidly converged approximation to the exact value. The basic concept of the Variational Iteration Method shown in the following:

$$Lu + Nu = g(x) \tag{3.21}$$

Where, L is a linear operator, N is a nonlinear operator and *g(x)* is a known analytic function. According to He’s Variational Iteration Method (He, 1999), a correctional function can be constructed as follows;

$$u_{n+1}(x) = u_n(x) + \int_0^x \lambda[Lu_n(\tau) + N\hat{u}_n(\tau) - g(\tau)]d\tau \qquad (3.22)$$

Where, λ is a general Lagrangian multiplier by the Variational theory. $\hat{u}_n$ is a restriction variation. The equation above is called a correctional function.

The application of the Variational Iteration Method to the Blasius equation yields

$$f_{n+1}(\eta) = f_n(\eta) + \int_0^{+\infty} \lambda(f'''(\tau) + \frac{1}{2}\hat{f}''(\tau)\hat{f}(\tau))d\tau \qquad (3.23)$$

The correctional functional equation is stationary by

$$\delta\hat{f}_n = 0 \qquad (3.24)$$

$$\delta\hat{f}_n'' = 0 \qquad (3.25)$$

$$\delta f_{n+1}(\eta) = \delta f_n(\eta) + \delta \int_0^{+\infty} \lambda(f'''(\tau) + \frac{1}{2}\hat{f}''(\tau)\hat{f}(\tau))d\tau \qquad (3.26)$$

$$\begin{aligned} \delta f_{n+1}(\eta) = \delta f_n(\eta) &+ \left(\frac{\partial^2\lambda}{\partial\tau^2}\delta f_n(\tau)\right)\Big|_{\tau=\eta} \\ &- \left(\frac{\partial\lambda}{\partial\tau}\delta f'_n(\tau)\right)\Big|_{\tau=\eta} + \left(\lambda\delta f''_n(\tau)\right)\Big|_{\tau=\eta} \\ &- \int_0^{+\infty}\left(\frac{\partial^3\lambda}{\partial\tau^3}\delta f_n(\tau)\right)d\tau \end{aligned} \qquad (3.27)$$

The stationary condition

$$\delta f_n: 1 + \frac{\partial^2\lambda}{\partial\tau^2}\Big|_{\tau=\eta} = 0 \qquad (3.28)$$

$$\delta f'_n: 1 + \left.\frac{\partial\lambda}{\partial\tau}\right|_{\tau=\eta} = 0 \tag{3.29}$$

$$\delta f''_n: \lambda|_{\tau=\eta} = 0 \tag{3.30}$$

$$\delta f_n: \left.\frac{\partial^3\lambda}{\partial\tau^3}\right|_{\tau=\eta} = 0 \tag{3.31}$$

The Langrangian multiplier then can be identified as

$$\lambda(\tau) = -\frac{1}{2}(\tau-\eta)^2 \tag{3.32}$$

The formula is then obtained

$$f_{n+1}(\eta) = f_n(\eta) - \frac{1}{2}\int_0^{+\infty}(\tau-\eta)^2\left(f'''(\tau) + \frac{1}{2}f''(\tau)f(\tau)\right)d\tau \tag{3.33}$$

The initial condition of Blasius is substituted in the above equation

$$f_0(\eta) = A + B + \frac{1}{2}\eta^2 C \tag{3.34}$$

Where *A*, *B* and *C* are constant that be determined. The value of $f''(0)$ is represent by

$$f_0(\eta) = \frac{1}{2}\sigma\eta^2 \tag{3.35}$$

$$f_1(\eta) = \frac{1}{2}\sigma\eta^2 - \frac{1}{240}\sigma^2\eta^5 \tag{3.36}$$

$$f_0(\eta) = \frac{1}{2}\sigma\eta^2 - \frac{1}{240}\sigma^2\eta^5 + \frac{11}{161280}\sigma^3\eta^8 - \frac{1}{5702400}\sigma^4\eta^{11} \tag{3.37}$$

$$f_0(\eta) = \frac{1}{2}\sigma\eta^2 - \frac{1}{240}\sigma^2\eta^5 + \frac{11}{161280}\sigma^3\eta^8 - \frac{5}{4257792}\sigma^4\eta^{11} + \frac{10033}{1394852659200}\sigma^5\eta^{14} - \frac{5449}{125076897792000}\sigma^6\eta^{17} + \frac{83}{571875655680000}\sigma^7\eta^{20} - \frac{1}{6282355064832000}\sigma^8\eta^{23} \tag{3.38}$$

The Microsoft Office Excel is used to complete the iteration for the generated equations with the initial estimation of $f''(0) = 0.332057$.

3.9 USAGE OF MICROSOFT EXCEL SOFTWARE

The methods used in this project mostly use the Microsoft Excel software. The usage of the Microsoft Excel more than the other software is due to the pre-programmed software that enabled the user use the design function in the software. Unlike the other software such as *Matlab*, the usage of the function need to be programmed manually using specified command. The Microsoft Excel eases the calculation of linear equation. The Microsoft Excel is user friendly software that guides the user to the function. The usage of programming software for the new learning person sometimes will affect the working schedule. For example of to make a multiple value in a graph, the Microsoft Excel is easier than the *Matlab* because of the existing program prepared which the *Matlab* requires more time than the Microsoft Excel. The Microsoft Excel and the *Matlab* software also can be linked together. The array of axis from Excel can be transferred to the *Matlab* for the plotting function and also vice-versa. It is important to know how to link these two softwares in order to maximize the usage of these two.

CHAPTER 4

RESULT AND DISCUSSION

4.1 SHOOTING METHODS WITH MAPLE

Table 4.1: Iteration of $f''(0)$ Using Shooting Methods with Maple

Shoot Step	$f''(0)$ **From Shooting Method**
1	1.5
2	0.733737059782952894e-1
3	0.263463957771544799
4	0.329380680840193618
5	0.332053745164111624
6	0.332057357121743779

Table 4.1 above shows the final value of Shooting Method using Maple. The value of initial is set for 1.5 hence generate the value of $f''(0) = 0.332057357121743779$. The value that generated is depending on the maximum value of eta that has been set earlier. Any value that has been set must meet the asymptotic element of the profile to make sure that the solution is close to accuracy.

4.2 COMPARISON OF THE GRAPHS BETWEEN PREDICTOR-CORRECTOR METHOD, SHOOTING METHOD AND MODIFIED PREDICTOR-CORRECTOR METHOD

Figure 4.1: Graph of $f/6$, f' and f'' VS eta (Predictor-Corrector)

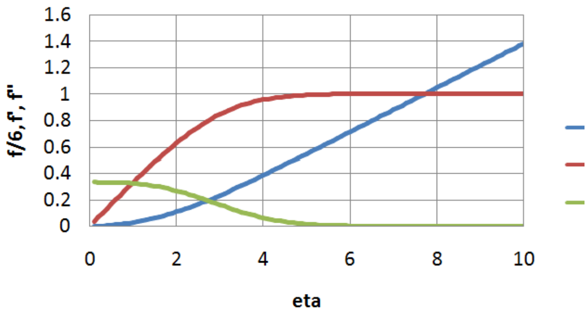


Figure 4.2: Graph of $f/6$, f' and f'' VS eta (Shooting)

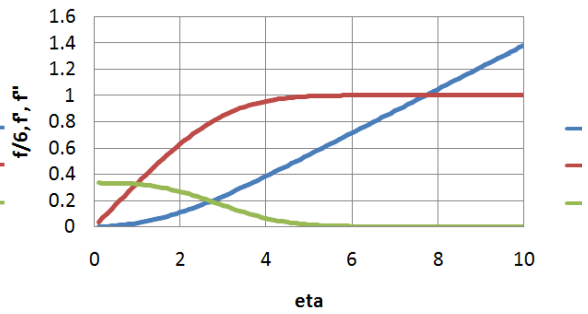
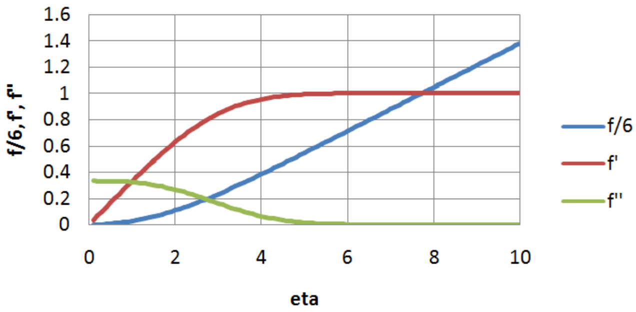


Figure 4.3: Graph of $f/6$, f' and f'' VS eta (Modified Predictor-Corrector)



From the Figure 4.1, 4.2 and 4.3 above, the shape of the graphs is closely identical between all the three methods. The low percentage difference between all the three methods causes the values obtained have only a small difference to each other. The shape of the graph is also due to the nature of Blasius flow over flat plate behavior that being reflected by these equations of solution. Hence, the difference is hardly determined by comparing the graphs.

4.2.1 Calculating the Error Percentage

The difference between any values that to be compare can be clearly seen by calculating the error percentage of the difference. The small value of difference can be seen in the percentage value. It has been used in any data comparison analysis. The formula for calculating the error percentage is

$$error\ percentage(\%) = \left|\frac{other\ method - Blasius\ method}{Blasius\ \ method}\right| X100 \quad (4.1)$$

The error percentage calculation value of eta, η when $f'(\eta) = 0.99$ was calculated for all the three methods to the Blasius solution and the results were tabulated.

Error percentage of Predictor-Corrector Method to the Blasius solution:

$$error\ percentage = \left|\frac{4.87697709102085 - 4.93599334995844}{4.93599334995844}\right| \text{X100} = 1.196\%$$

Error percentage of Shooting Method to the Blasius solution:

$$error\ percentage = \left|\frac{4.91073201558469 - 4.93599334995844}{4.93599334995844}\right| \text{X100} = 0.512\%$$

Error percentage of Modified Predictor-Corrector Method to the Blasius solution:

$$error\ percentage = \left|\frac{4.90120080088182 - 4.93599334995844}{4.93599334995844}\right| \text{X100} = 0.705\%$$

Table 4.2: Comparison Result of $f'(\eta) = 0.99$

Methods	$f'(\eta) = 0.99$	**Error (%)**
Blasius (Forth Order Runge-Kutta)	4.93599334995844	-
Predictor-Corrector	4.87697709102085	1.196
Shooting with Maple	4.91073201558469	0.512
Current (Modified Predictor-Corrector)	4.90120080088182	0.705

Table 4.2 shows that the Predictor-Corrector method has the largest error when compare to the Blasius solution that is 1.196 percent. While the second largest is the current method that is 0.705 percent and the least error is the Shooting Method that is 0.512 percent. The Shooting method using Maple is the most closer to the Blasius solution is due to the usage of the software. The software usually iterates and solves the equation more accurate than the other method. The specialty of the shooting method is the value initial value of $f''(\eta) = 0.99$ is develop from the iteration. While the other method we let the value of $f''(\eta) = 0.332057$ obtain from the existing.

4.2.2 Error Percentage of $f'(\eta)$with the Other Methods to the Blasius Solution Method

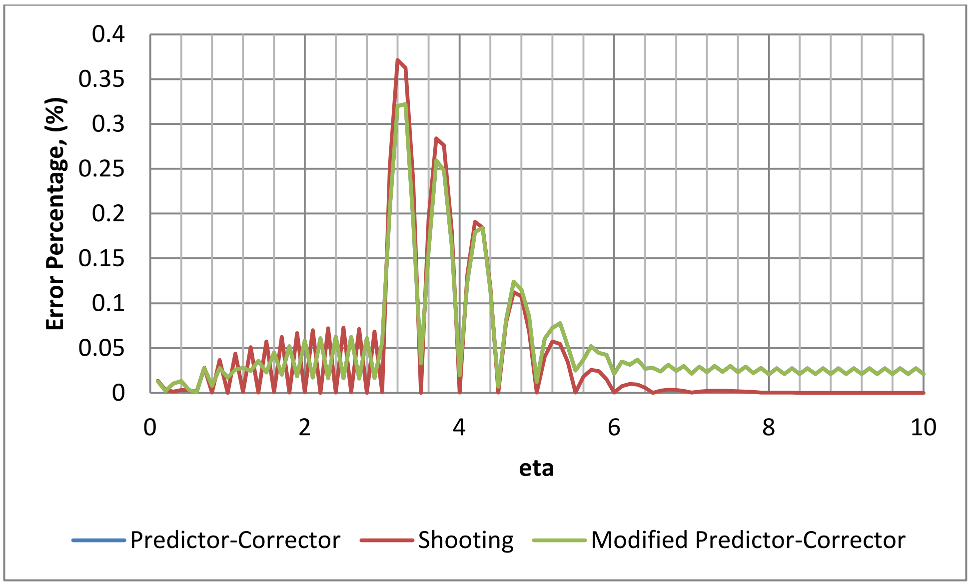


Figure 4.4: Error Percentage of *f'* of Other Methods towards Blasius Solution

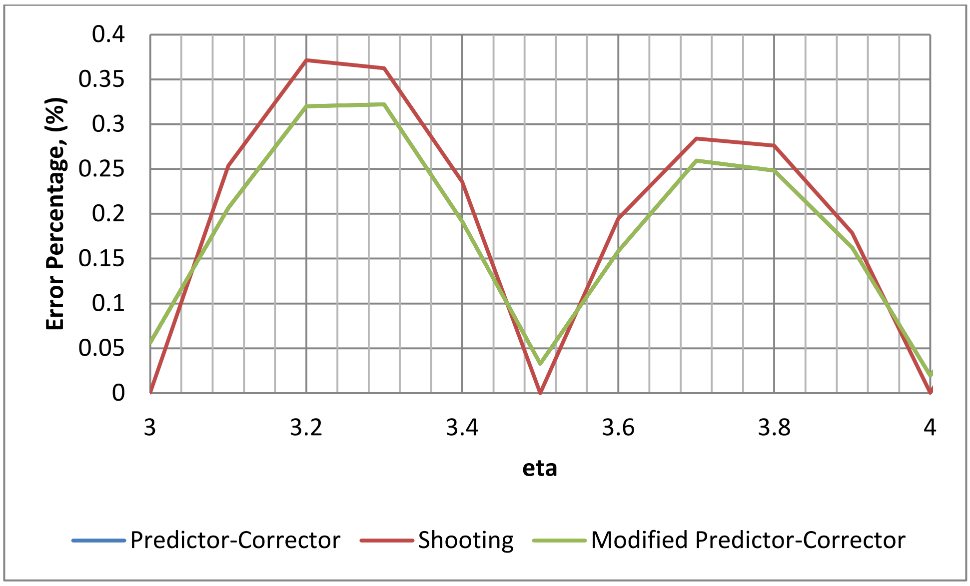


Figure 4.5: Error Percentage of *f'* of Other Methods towards Blasius Solution (Maximum Value)

The Figure 4.5 is the error percentage of *f'* with other methods to the Blasius solution method. The latter graph is to capture the highest error among others. It shows that the Shooting Method has the highest error compare to the Blasius solution up to 0.4 percent. The error percentage of Predictor-Corrector Method with the Modified Predictor-Corrector to the Blasius solution shows the similar pattern as the methods is derived using the similar method. But when the eta reaches the value of 6, the Shooting Method is marked as the lowest error to the Blasius equation among the other two methods.

4.2.3 Comparison between the Three Methods

From the error percentage result, the Shooting Method is the most closely identical to the Blasius solution followed by the Modified Predictor-Corrector Method and the Predictor-Corrector Method. There are many factors that contribute to the result shown. Such as:

1. The Shooting Method using the value of $f''(0)$ by iteration. While the other two methods, the $f''(0)$ value cannot be determine. Hence the $f''(0)$ value is taken from the existing value that is $f''(0) = 0.332057$.
2. The specialty of shooting method also, the equation is solved fully by using the Maple software. While the other two methods, the Blasius equation has been derived to get the simplified equation before being solved using software. The difference of the software usage had caused to the difference of the result obtained.
3. Difference methods have their own stability. The higher the iteration, the results become more unstable. The shooting method is more stable than the other two methods hence, the results yielded are more accurate.

4.2.4 Error Percentage of $f(\eta)$ With the Other Methods to the Blasius Solution Method

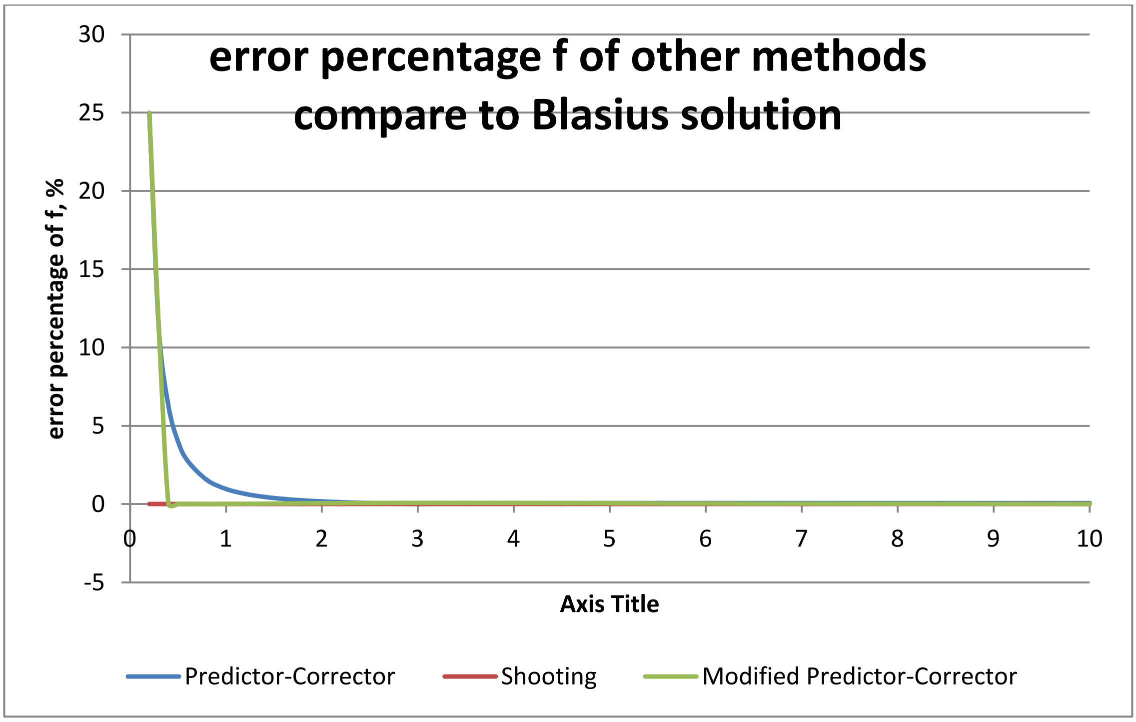


Figure 4.6: Error Percentage of f with other methods to the Blasius Solution

The Figure 4.6 shows the error percentage of the f from the other methods compare to the Blasius solution. At the starting, the Predictor-Corrector Method and the

Modified Predictor-Corrector method differ from the Blasius of 25 percent error. While the shooting method experience an opposite manner that the error is only 0.01 percent.

4.2.5 Error Percentage of $f''(\eta)$ With the Other Methods to the Blasius Solution Method

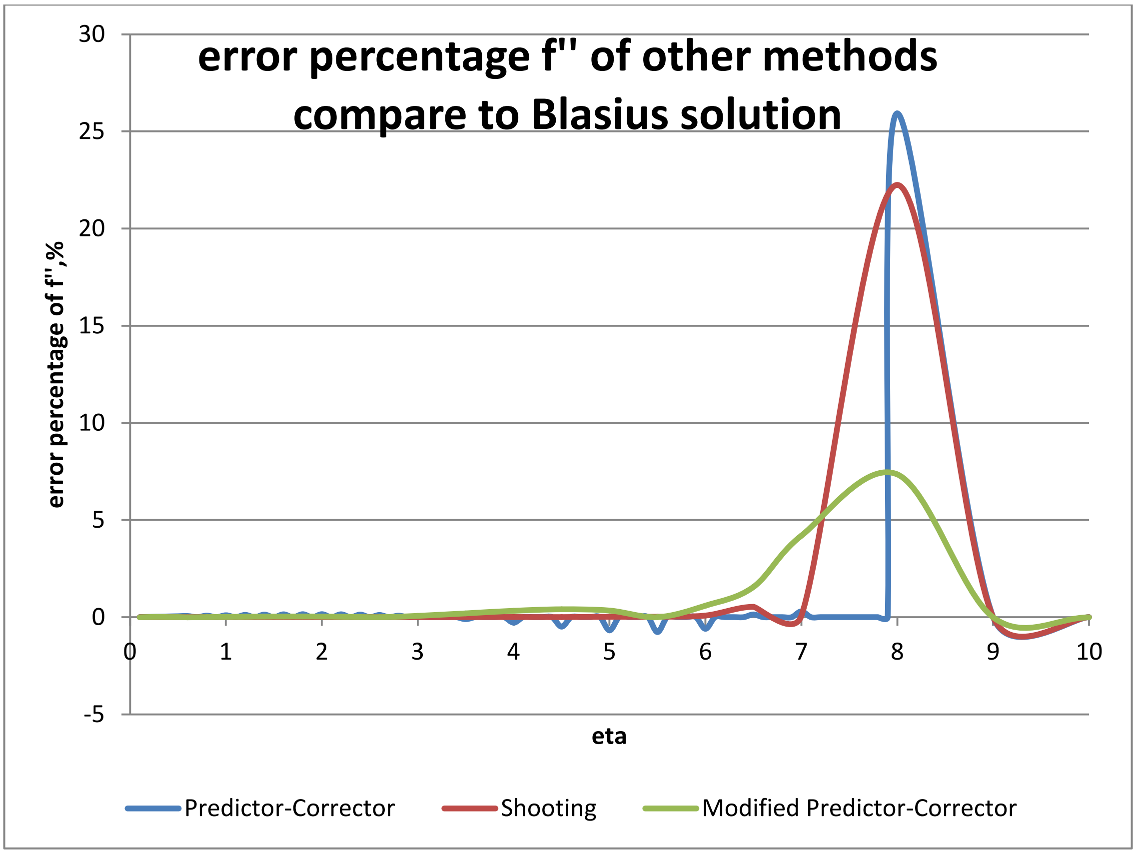


Figure 4.7: Error Percentage of f'' with other methods to the Blasius Solution

The Figure 4.7 shows the error percentage of the f'' from the other methods compare to the Blasius solution. At the starting, the all the methods shows the low percentage of error. Upon approaching the eta equal to 8, the error tend to increase to infinity because of the zeroes value of error. Because of the Blasius solution table does not give the full value of the f'' there are points where the lines are connected computationally. And the line is not accurately to be taken as consideration.

4.3 COMPARISON OF THE GRAPH OBTAIN IN MAPLE

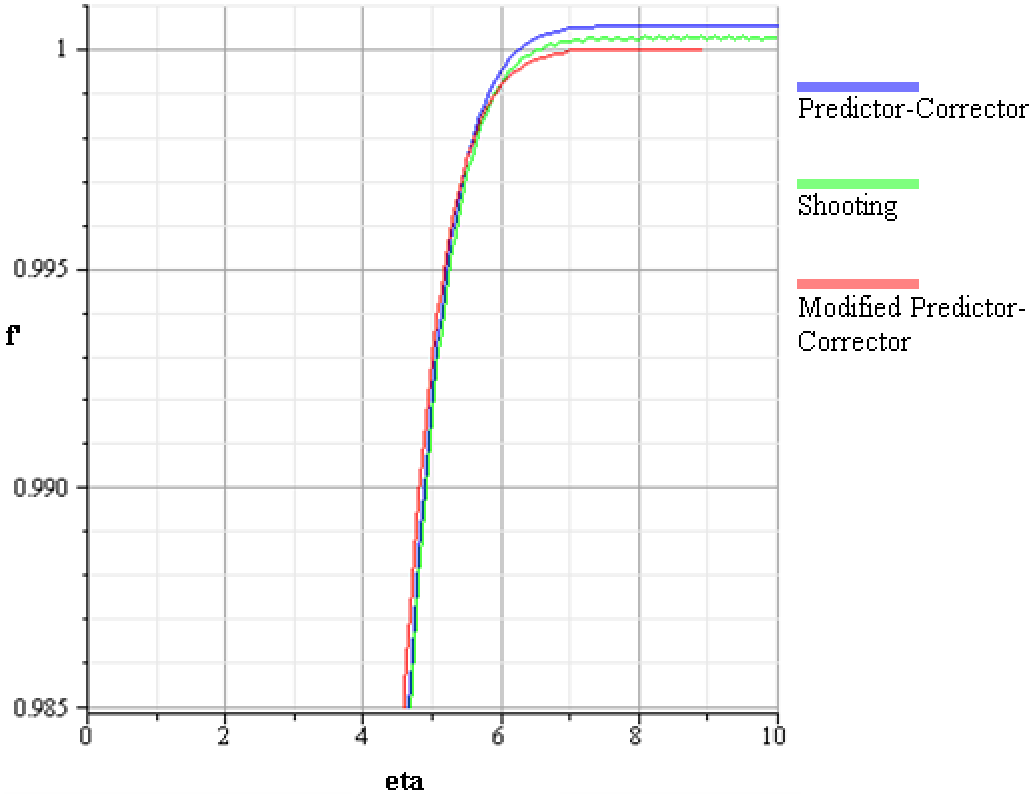


Figure 4.8: Comparison Graph of f' against η

The Figure 4.8 shows that the value of f' against eta for different method. The value is not much difference due to the same profile to the Blasius solution. The obvious part from the figure is the Predictor-Corrector Method is at the top of the stable line followed by The Shooting Method and the Modified Predictor-Corrector Method. It shows that the Predictor-Corrector Method and the Shooting Method has exceeded the value of $f' = 1$ but The Modified Predictor-Corrector Method is on the line. Although the boundary condition of Blasius solution is to obtain the value of f' to infinite is 1, it still acceptable that the difference is too small.

4.4 VARIATIONAL ITERATION METHOD RESULT

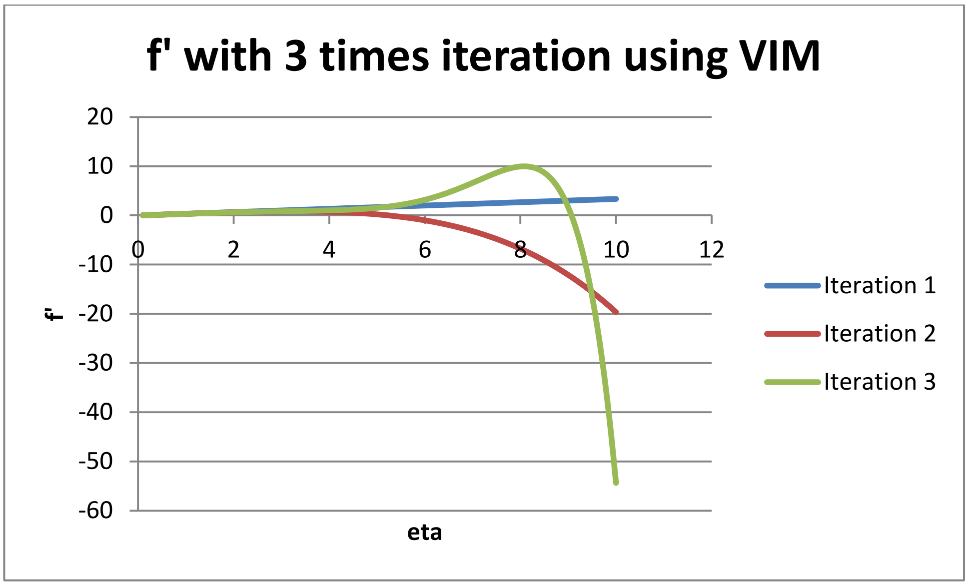


Figure 4.9: Graph of f' against eta using Variational Iteration Method

The Figure 4.9 shows the result of iterations of f' using Variational Iteration Method. For the first iteration (Iteration 1), the graph shows a shape to increase but it quite different with the Blasius profile. For the second iteration (Iteration 2), the graph shows an improvement as the shape is more likely to follow the Blasius profile, but the value drops due to instability. And for the third iteration (Iteration 3), the shape of graph is identical to the Blasius profile, but the value to infinity is not equal to one. Instead, it drops to negative due to the instability.

According to the result of the VIM above, the more iteration, the more accurate it is to the Blasius but, it only valid with eta less than 4. With the increasing the value of eta, the method become unstable and loss the profile. This method is not used for further parts of this project as the boundary value of Blasius is not satisfied. The phenomenon is common in project of research when the method is halfway to complete, but failed to satisfy the needs of the research.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Numerical that has been used earlier is no longer the basic method of solving the nonlinear equation. Most of the method has been derived and modified to satisfy the based on the assumption to the real application.

In this project of research, the various methods have been applied to solve the boundary layer equation. There are method that can be applied and satisfied to the initial and boundary condition of the equation. There also methods that closely accurate by not stable at high value. We have used the Variational Iteration Method by He and also Homotopy Perturbation Method to solve the nonlinear equation. The result of the two methods only valid for the earlier flow and lost the pattern at the higher value of eta. Hence, the further comparison and analysis between the existing data cannot be proceeding.

The objective of comparing the different numerical method to solve the nonlinear ordinary differential equation has been completed using the three methods that are Predictor-Corrector Method, Shooting Method and also Modified Predictor-Corrector Method. The results are successfully compared and analyses in this project. The comparison includes the shape of the graph, the starting value, the error percentage and the value of infinite point.

5.2 RECOMMENDATION

The work presented suggests several future areas of study especially according to the boundary layer solution. In this study, the different methods have been used for solving the stationary flat plate of boundary layer phenomena which correspond to how the boundary layer is form by equation. However, another way of comparing the results is might be by enhancing the method from stationary flat plate into a moving flat plate that is also known as Sakiadis. The research later also can be compared with this project of research.

REFERENCES

This project is prepared on the following references

Beyond The Boundary Layer Concept. (December 2005). Retrieved June 14, 2011, from http://www.physicstoday.org.

Boyce, M. A., W.E, Richard C. DiPrima. *Elementary Differential Equations, Ninth edition.* John Wiley & Sons, Inc. United State of America. 2009.

Cengel, Y. A. and *Cimbala, J. M. Fluid Mechanics: Fundamentals and Applications,* 2nd ed.: McGraw Hill, 2010, p. 542.

Cortell, R. *Numerical Comparison of Blasius and Sakiadis Flows*. Universidad Politcnica de Valencia, 2010.

Cortell, R. *Numerical Solution of the Classical Blasius flat-plate Problem*. 2005.

Donald F. Young, Bruce Munson,Theodore H. Okiishi, Wade W. Huebsch. *A Brief Introduction to Fluid Mechanics*, 4th Edition. Wiley. 2008

Gahan, B., J. J. H. Miller, G. I. Shishkin. *Accurate Numerical Method for Blasius' Problem for Flow past a Flat Plate with Mass Transfer*. University of Dublin, 2000.

Ganopal, B. D. *Highly Accurate Solutions of the Blasius and Falkner-Skan Boundary Layer Equations via Convergence Acceleration*. University of Arizona.

Griffiths, D.V., I.M. Smith. *Numerical Methods for Engineers Second Edition.* Chapman & Hall/Crc. United States of America. 2006.

Hager, W.H. Blasius: A Life in Research and Education. *Experiments in Fluids 34*. 2003)

He, J. H. *Variation Iteration Method – a Kind of Nonlinear Analytical Technique: Some Examples*, Int. J. Nonlinear Mechanics, Vol. 34, No. 4, pp.699-708, 1999.

Howard, L. *On the Solution of the Laminar Boundary Layer Equations*. King's College, Cambridge, 1937.

Hutton, D.V. *Fundamentals of Finite Element Analysis*. McGraw Hill. Singapore. 2004.

Latif M. Jiji. *Heat Convection Second edition*. Springer, India. 2009.

Moghimi, M., Khoramishad, H., Massah, H. R., Mortezaei. *Approximation Analytical Solution to Flow over a Flat Plate by Variational Iteration Method*, Int. J. Mech. & Aerospace Eng., Vol. 2, No.2, pp.63-69, 2006.

Nagle, Saff, Snider. *Fundamentals of Differential Equations, Seventh edition.* Pearson Education. United State of America. 2008.

Numeric Solution to Blasius' Equation. (n.d.). Retrieved June 3, 2011, from http://moon.pr.erau.edu/~gallyt/ae302/NumericBlasius.pdf.

Raisinghania, M.D. *Fluid Dynamics with Complete Hydrodynamics and Boundary Layer Theory*. S.Chand & Co. Ltd. New Delhi. 2005.

Rao, S. S.. *Applied Numerical Methods for Engineers and Scientists*. Prentice Hall, Inc. United States of America. 2002.

Robert W. Fox, Alan T. McDonald, Philip J. Pritchard. *Introduction to Fluid Mechanics* 7th Edition. John Wiley & Sons. 2009.

Salleh, M. Z.(2012, April 23). How The Infinite Value of Boundary Condition Can Be Determine. (M. S. Walid, Interviewer)

Schlichting, H., Gersten K. *Boundary-Layer Theory*, McGraw Hill, New York, 1968.

APPENDIX

Table A-1: Iteration Result Using Predictor-Corrector Method

η	Predictor			Corrector		
	$f(\eta)$	$f'(\eta)$	$f''(\eta)$	$f(\eta)$	$f'(\eta)$	$f''(\eta)$
0.0	0	0	0	0.00000	0.00000	0.33206
0.1	0.00000	0.03321	0.33206	0.00166	0.03321	0.33206
0.2	0.00498	0.06641	0.33203	0.00664	0.06641	0.33200
0.3	0.01328	0.09961	0.33189	0.01494	0.09960	0.33184
0.4	0.02490	0.13279	0.33159	0.02656	0.13278	0.33151
0.5	0.03984	0.16593	0.33107	0.04150	0.16590	0.33096
0.6	0.05809	0.19900	0.33027	0.05974	0.19897	0.33013
0.7	0.07964	0.23198	0.32915	0.08129	0.23193	0.32898
0.8	0.10448	0.26483	0.32765	0.10613	0.26476	0.32746
1.0	0.16398	0.32997	0.32333	0.16561	0.32986	0.32309
1.2	0.23642	0.39405	0.31696	0.23802	0.39389	0.31669
1.4	0.32153	0.45664	0.30827	0.32309	0.45642	0.30798
1.6	0.41897	0.51724	0.29709	0.42048	0.51696	0.29678
1.8	0.52830	0.57535	0.28336	0.52975	0.57500	0.28305
2.0	0.64896	0.63046	0.26716	0.65033	0.63006	0.26687
2.2	0.78030	0.68210	0.24871	0.78159	0.68164	0.24847
2.4	0.92160	0.72986	0.22839	0.92279	0.72935	0.22821
2.6	1.07203	0.77340	0.20668	1.07312	0.77285	0.20656
2.8	1.23073	0.81250	0.18415	1.23171	0.81193	0.18411
3.0	1.39680	0.84705	0.16142	1.39767	0.84649	0.16146
3.5	1.83829	0.91402	0.10766	1.83888	0.91350	0.10787
4.0	2.30687	0.95638	0.06405	2.30723	0.95599	0.06433
4.5	2.79171	0.98025	0.03381	2.79191	0.97998	0.03409
4.8	3.08713	0.98846	0.02174	3.08726	0.98826	0.02197
4.9	3.18608	0.99046	0.01858	3.18619	0.99029	0.01880
5.0	3.28522	0.99217	0.01580	3.28532	0.99202	0.01600
5.5	3.78284	0.99745	0.00653	3.78288	0.99737	0.00665
6.0	4.28217	0.99951	0.00239	4.28219	0.99948	0.00245
6.5	4.78214	1.00023	0.00077	4.78215	1.00022	0.00080
7.0	5.28232	1.00045	0.00022	5.28233	1.00045	0.00023
8.0	6.28283	1.00052	0.00001	6.28283	1.00052	0.00001
9.0	7.28335	1.00053	0.00000	7.28335	1.00053	0.00000
10.0	8.28388	1.00053	0.00000	8.28388	1.00053	0.00000

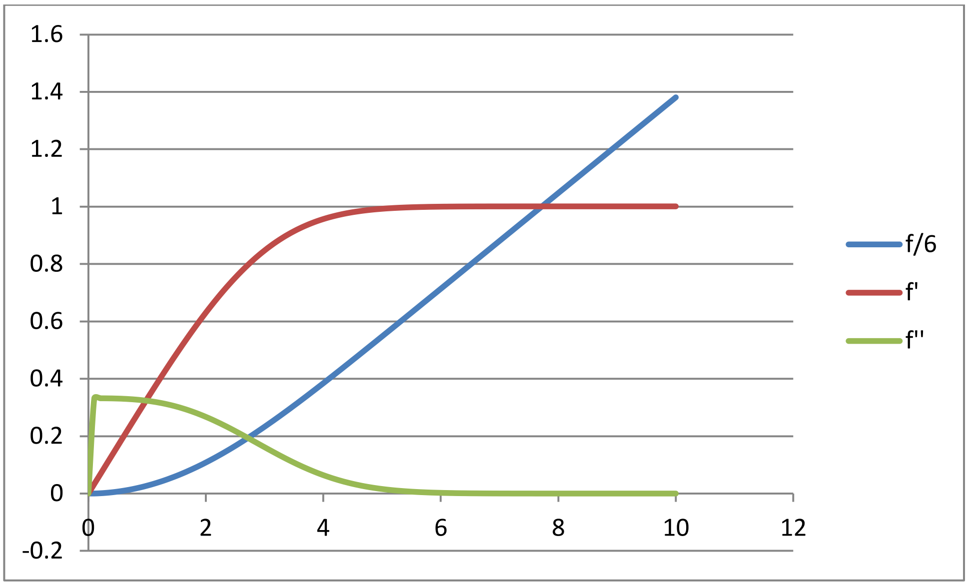


Figure A-1: Graph of $f\,/6$, $f\,'$ and $f\,''$ against η Using Predictor-Corrector Method (Excel)

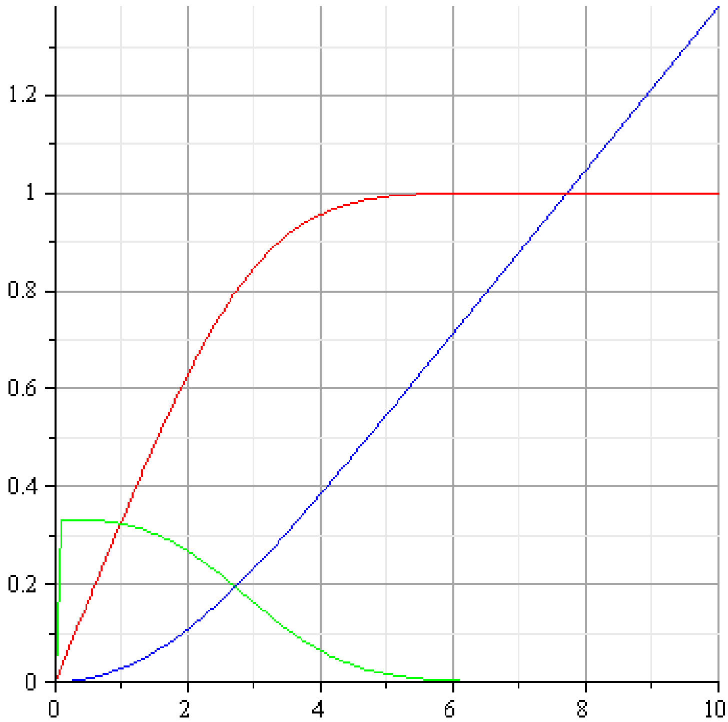


Figure A-2: Graph of $f\,/6$, $f\,'$ and $f\,''$ against η Using Predictor-Corrector Method (Maple)

Table A-2: Iterations Result Using Shooting Method with Maple

$\boldsymbol{\eta}$	$\boldsymbol{f(\eta)}$	$\boldsymbol{f'(\eta)}$	$\boldsymbol{f''(\eta)}$
0.0	0.00000	0.00000	0.33206
0.1	0.00166	0.03321	0.33205
0.2	0.00664	0.06641	0.33198
0.3	0.01494	0.09960	0.33181
0.4	0.02656	0.13276	0.33147
0.5	0.04149	0.16589	0.33091
0.6	0.05973	0.19894	0.33008
0.7	0.08128	0.23189	0.32892
0.8	0.10611	0.26471	0.32739
0.9	0.13421	0.29735	0.32543
1.2	0.23795	0.39378	0.31659
1.4	0.32298	0.45626	0.30787
1.6	0.42032	0.51676	0.29666
1.8	0.52952	0.57476	0.28293
2.0	0.65002	0.62977	0.26675
2.2	0.78119	0.68131	0.24835
2.4	0.92229	0.72898	0.22809
2.6	1.07251	0.77246	0.20645
2.8	1.23098	0.81151	0.18401
3.0	1.39681	0.84604	0.16136
3.5	1.83770	0.91304	0.10777
4.0	2.30575	0.95552	0.06423
4.5	2.79013	0.97951	0.03398
4.9	3.18420	0.98981	0.01870
5.0	3.28327	0.99154	0.01591
5.5	3.78057	0.99688	0.00658
6.0	4.27962	0.99897	0.00240
6.5	4.77932	0.99970	0.00077
7.0	5.27924	0.99992	0.00022
8.0	6.27921	1.00000	0.00001
9.0	7.27921	1.00000	0.00000
10.0	8.27921	1.00000	0.00000

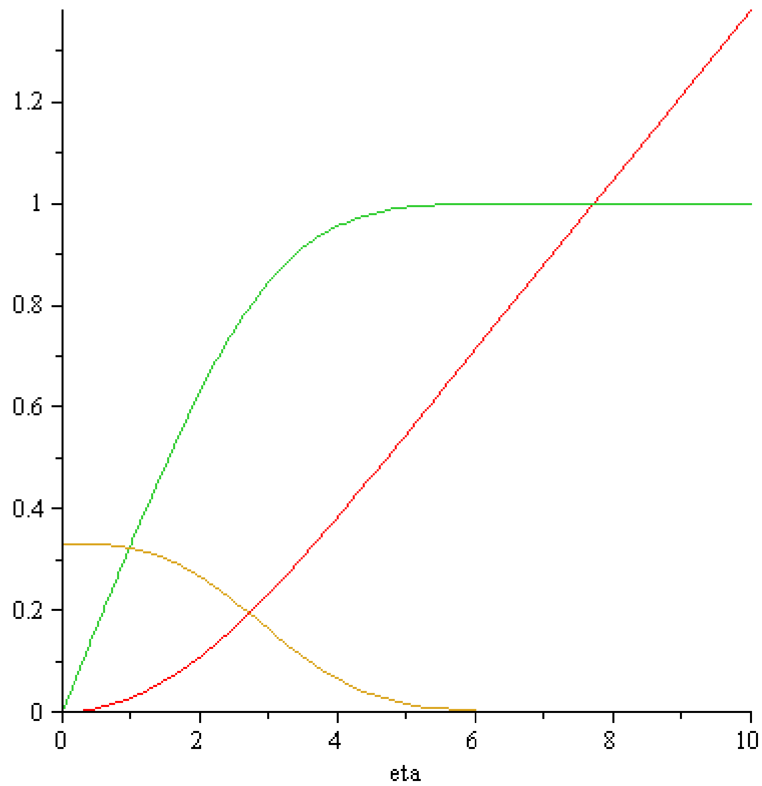


Figure A-3: Graph of *f* /*6*, *f '* and *f ''* against η Using Shooting Method with Maple

Table A-3: Iterations Result Using Modified Predictor-Corrector Method

η	Predictor			Corrector		
	$f(\eta)$	$f'(\eta)$	$f''(\eta)$	$f(\eta)$	$f'(\eta)$	$f''(\eta)$
0.0	0	0	0	0.00000	0.00000	0.33206
0.1	0.00000	0.03321	0.33206	0.00166	0.03321	0.33206
0.2	0.00498	0.06641	0.33203	0.00664	0.06641	0.33200
0.3	0.01328	0.09961	0.33189	0.01494	0.09960	0.33184
0.4	0.02656	0.13278	0.33151	0.02656	0.13276	0.33145
0.5	0.04149	0.16589	0.33096	0.04149	0.16588	0.33090
0.6	0.05974	0.19894	0.33008	0.05973	0.19891	0.33003
0.7	0.08127	0.23189	0.32893	0.08127	0.23187	0.32888
0.8	0.10610	0.26469	0.32735	0.10609	0.26465	0.32730
1.0	0.16555	0.32972	0.32294	0.16553	0.32968	0.32289
1.2	0.23789	0.39367	0.31649	0.23786	0.39361	0.31646
1.4	0.32288	0.45610	0.30775	0.32283	0.45603	0.30772
1.6	0.42015	0.51653	0.29653	0.42009	0.51646	0.29653
1.8	0.52925	0.57446	0.28280	0.52918	0.57439	0.28282
2.0	0.64965	0.62941	0.26664	0.64956	0.62934	0.26667
2.2	0.78068	0.68089	0.24827	0.78059	0.68083	0.24832
2.4	0.92163	0.72852	0.22805	0.92152	0.72847	0.22812

η	Predictor			Corrector		
	$f(\eta)$	$f'(\eta)$	$f''(\eta)$	$f(\eta)$	$f'(\eta)$	$f''(\eta)$
2.8	1.22998	0.81102	0.18407	1.22986	0.81099	0.18416
3.0	1.39564	0.84556	0.16148	1.39551	0.84556	0.16157
3.5	1.83623	0.91274	0.10799	1.83614	0.91277	0.10805
4.0	2.30388	0.95533	0.06445	2.30378	0.95538	0.06446
4.6	2.88621	0.98271	0.02961	2.88614	0.98276	0.02958
4.9	3.18233	0.98998	0.01877	3.18231	0.99002	0.01873
5.0	3.28121	0.99166	0.01597	3.28116	0.99170	0.01593
5.5	3.77878	0.99713	0.00658	3.77878	0.99715	0.00655
6.0	4.27767	0.99918	0.00239	4.27765	0.99920	0.00236
6.5	4.77778	0.99998	0.00076	4.77779	0.99998	0.00075
7.0	5.27750	1.00013	0.00021	5.27748	1.00014	0.00021
8.0	6.27768	1.00021	0.00001	6.27767	1.00021	0.00001
9.0	7.27789	1.00021	0.00000	7.27788	1.00021	0.00000
10.0	8.27810	1.00021	0.00000	8.27809	1.00021	0.00000

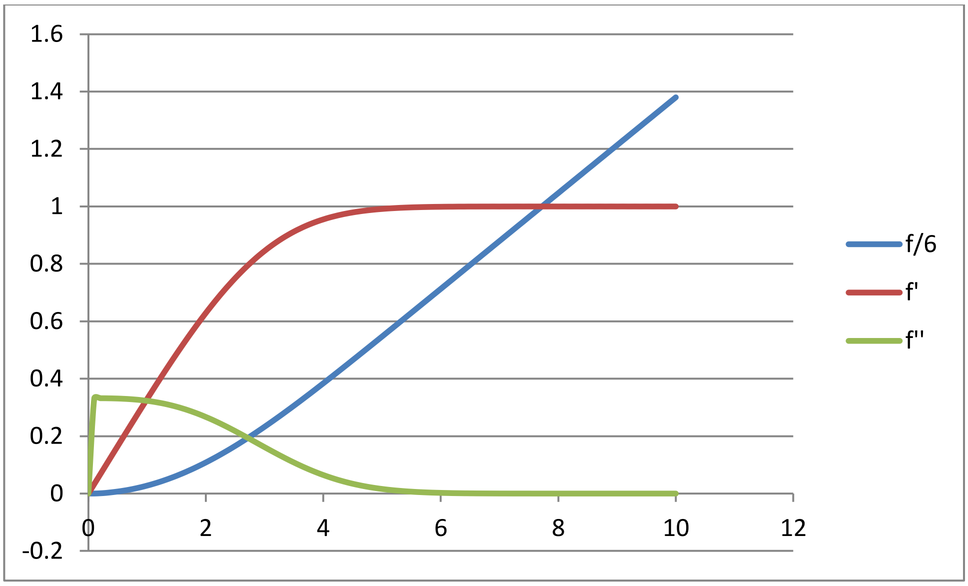


Figure A-4: Graph of $f/6$, f' and f'' against η Using Modified Predictor-Corrector Method (Excel)

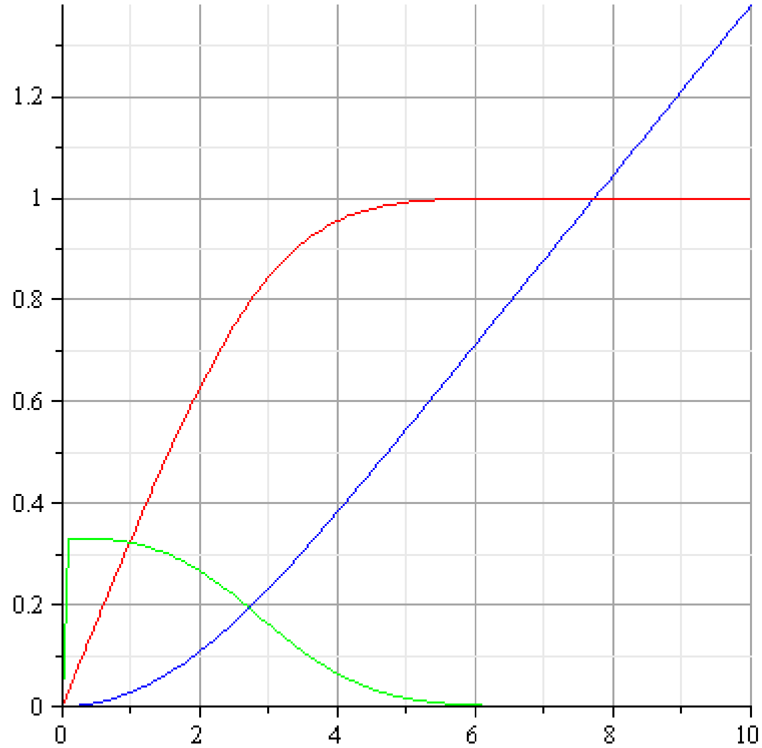


Figure A-5: Graph of *f* /6, *f '* and *f ''* against η Using Modified Predictor-Corrector Method (Maple)

Table A-4: Error Percentage of f'(n) with the Other Methods To The Blasius Solution Method

eta	Predictor-Corrector	Shooting	Modified Predictor-Corrector	Blasius
0.1	0.012948	0.013532	0.012948	0.03321
0.2	0.002108	0.003318	0.002108	0.06641
0.3	0.01025	0.001401	0.01025	0.0996
0.4	0.013054	0.00314	0.013054	0.13276
0.5	0.002995	0.002855	0.002995	0.16589
0.6	0.000608	0.001375	0.000608	0.19894
0.7	0.027621	0.028146	0.027621	0.231825
0.8	0.007727	0.00032	0.007727	0.26471
0.9	0.027872	0.036663	0.027872	0.297245
1	0.017176	1.43E-05	0.017176	0.32978
1.1	0.026125	0.043783	0.026125	0.36178
1.2	0.027847	0.000984	0.027847	0.39378
1.3	0.024571	0.051046	0.024571	0.42502
1.4	0.035654	0.000394	0.035654	0.45626
1.5	0.022629	0.057411	0.022629	0.48651

eta	Predictor-Corrector	Shooting	Modified Predictor-Corrector	Blasius
1.6	0.045007	0.000617	0.045007	0.51676
1.7	0.020269	0.062456	0.020269	0.54576
1.8	0.051879	0.000322	0.051879	0.57476
1.9	0.018357	0.066677	0.018357	0.602265
2	0.057916	0.000669	0.057916	0.62977
2.1	0.017029	0.069907	0.017029	0.65554
2.2	0.061114	5.46E-05	0.061114	0.68131
2.3	0.016406	0.072032	0.016406	0.705145
2.4	0.062879	0.000272	0.062879	0.72898
2.5	0.016173	0.072558	0.016173	0.750715
2.6	0.062413	0.000649	0.062413	0.77245
2.7	0.016075	0.071196	0.016075	0.79198
2.8	0.060978	4.12E-05	0.060978	0.81151
2.9	0.016704	0.068599	0.016704	0.828775
3	0.056357	0.000522	0.056357	0.84604
3.1	0.20656	0.253544	0.20656	0.85944
3.2	0.320017	0.371371	0.320017	0.87284
3.3	0.32198	0.362525	0.32198	0.88624
3.4	0.191562	0.235785	0.191562	0.89964
3.5	0.032817	4.22E-05	0.032817	0.91304
3.6	0.158391	0.194634	0.158391	0.921536
3.7	0.259225	0.283965	0.259225	0.930032
3.8	0.248101	0.275959	0.248101	0.938528
3.9	0.162295	0.178605	0.162295	0.947024
4	0.019576	0.000189	0.019576	0.95552
4.1	0.123336	0.131463	0.123336	0.960318
4.2	0.179352	0.190758	0.179352	0.965116
4.3	0.183874	0.1844	0.183874	0.969914
4.4	0.11479	0.118885	0.11479	0.974712
4.5	0.006661	0.000433	0.006661	0.97951
4.6	0.080437	0.078158	0.080437	0.981916
4.7	0.124225	0.112225	0.124225	0.984322
4.8	0.115186	0.107574	0.115186	0.986728
4.9	0.085512	0.068811	0.085512	0.989134
5	0.012055	0.000185	0.012055	0.99154
5.1	0.060637	0.040263	0.060637	0.992608
5.2	0.072456	0.057309	0.072456	0.993676
5.3	0.077566	0.054444	0.077566	0.994744
5.4	0.051998	0.034468	0.051998	0.995812
5.5	0.02493	0.000126	0.02493	0.99688
5.6	0.037199	0.018018	0.037199	0.997298
5.7	0.051928	0.025564	0.051928	0.997716
5.8	0.044444	0.024187	0.044444	0.998134
5.9	0.042515	0.015341	0.042515	0.998552

eta	Predictor-Corrector	Shooting	Modified Predictor-Corrector	Blasius
6	0.021179	0.000279	0.021179	0.99897
6.1	0.034924	0.007293	0.034924	0.999116
6.2	0.031302	0.010053	0.031302	0.999262
6.3	0.037183	0.009329	0.037183	0.999408
6.4	0.027163	0.005764	0.027163	0.999554
6.5	0.027824	0.000101	0.027824	0.9997
6.6	0.023812	0.002379	0.023812	0.999744
6.7	0.031287	0.003371	0.031287	0.999788
6.8	0.024579	0.003174	0.024579	0.999832
6.9	0.029902	0.002035	0.029902	0.999876
7	0.021501	0.000152	0.021501	0.99992
7.1	0.029092	0.001286	0.029092	0.999928
7.2	0.023254	0.001963	0.023254	0.999936
7.3	0.030034	0.002287	0.030034	0.999944
7.4	0.023574	0.002337	0.023574	0.999952
7.5	0.029875	0.002177	0.029875	0.99996
7.6	0.02305	0.001857	0.02305	0.999968
7.7	0.029073	0.001414	0.029073	0.999976
7.8	0.02204	0.000879	0.02204	0.999984
7.9	0.027905	0.000275	0.027905	0.999992
8	0.020755	0.000381	0.020755	1
8.1	0.027334	0.000277	0.027334	1
8.2	0.020921	0.0002	0.020921	1
8.3	0.027453	0.000145	0.027453	1
8.4	0.021005	0.000105	0.021005	1
8.5	0.027513	7.71E-05	0.027513	1
8.6	0.021048	5.68E-05	0.021048	1
8.7	0.027542	4.24E-05	0.027542	1
8.8	0.021068	3.23E-05	0.021068	1
8.9	0.027557	2.53E-05	0.027557	1
9	0.021078	2.03E-05	0.021078	1
9.1	0.027563	1.68E-05	0.027563	1
9.2	0.021083	1.44E-05	0.021083	1
9.3	0.027566	1.29E-05	0.027566	1
9.4	0.021085	1.17E-05	0.021085	1
9.5	0.027568	1.09E-05	0.027568	1
9.6	0.021085	1.04E-05	0.021085	1
9.7	0.027568	1.01E-05	0.027568	1
9.8	0.021086	9.94E-06	0.021086	1
9.9	0.027569	9.8E-06	0.027569	1
10	0.021086	9.68E-06	0.021086	1

The Programming Command of Solving Shooting Method Using Maple

CEC_Salleh et al (2008):

2f'''+ff''=0

f(0)=0   f'(0)=0

f'(infty)-->1

```
> restart ;
> Shootlib := "C:\\Program Files/Maple 13/Shoot9/:"
> libname := Shootlib, libname :
> with( Shoot ) :     #calling the shoot command
> with( plots ) :     #command to call for graph plotting
> blt := 10 :
      #the maximum value of eta to infinite that being fixed
> FNS := {F(eta), Fp(eta), Fpp(eta)};
```

$$FNS := \{F(\eta), Fp(\eta), Fpp(\eta)\}$$

```
> ODE := {diff(F(eta), eta) = Fp(eta), diff(Fp(eta), eta)
      = Fpp(eta), diff(Fpp(eta), eta) = -0.5* Fpp(eta)*F(eta)};
```

$$ODE := \left\{\frac{d}{d\eta} F(\eta) = Fp(\eta), \frac{d}{d\eta} Fp(\eta) = Fpp(\eta), \frac{d}{d\eta} Fpp(\eta) = -0.5\, Fpp(\eta)\, F(\eta)\right\}$$

```
> IC := {F(0) = 0,  Fp(0) = 0 , Fpp(0) = α};
```

$$IC := \{F(0) = 0, Fp(0) = 0, Fpp(0) = \alpha\}$$

```
> BC := {Fp(blt) = 1} :
> infolevel[shoot] := 1 :
> S := shoot(ODE, IC, BC, FNS, [α = 1.5]) :
      #value of alpha represent the value f''(0) - can be put any value,
      hence being solved by the shooting
```

shoot: Step # 1

shoot: Parameter values : alpha = 1.5

shoot: Step # 2

shoot: Parameter values : alpha = 0.733737059782952894e-1 shoot: Step # 3

shoot: Parameter values : alpha = .263463957771544799

shoot: Step # 4

shoot: Parameter values : alpha = .329380680840193618

shoot: Step # 5

shoot: Parameter values : alpha = .332053745164111624

shoot: Step # 6

shoot: Parameter values : alpha = .332057357121743779

```
> p := odeplot(S, [[eta, F(eta)], [eta, Fp(eta)], [eta, Fpp(eta)]], 0
      ..blt) :

> for eta to blt while true do S(eta) end do;
```

$[1 = 1., F(1) = 0.16557171950467414, Fp(1) = 0.329780047748175098, Fpp(1) = 0.323007140916909685]$

$[2 = 2., F(2) = 0.650024343889420653, Fp(2) = 0.629765785571107626, Fpp(2) = 0.266751511756883886]$

$[3 = 3., F(3) = 1.39680811898526434, Fp(3) = 0.846044417018699368, Fpp(3) = 0.161360259640695558]$

$[4 = 4., F(4) = 2.30574622683151764, Fp(4) = 0.955518199508671008, Fpp(4) = 0.0642340469626720562]$

$[5 = 5., F(5) = 3.28327340900751308, Fp(5) = 0.991541833365302439, Fpp(5) = 0.0159067518113343750]$

$[6 = 6., F(6) = 4.27962058198405516, Fp(6) = 0.998972792520609886, Fpp(6) = 0.00240200896237811792]$

$[7 = 7., F(7) = 5.27923837990029820, Fp(7) = 0.999921518318538148, Fpp(7) = 0.000220146738998743549]$

$[8 = 8., F(8) = 6.27921290733497540, Fp(8) = 0.999996185490805156, Fpp(8) = 0.0000122236972187083993]$

$[9 = 9., F(9) = 7.27921175294070012, Fp(9) = 0.99999979842824416, Fpp(9) = 4.02602048013621385\ 10^{-7}]$

$[10 = 10., F(10) = 8.27921163083987644, Fp(10) = 0.999999904189735006, Fpp(10) = 3.81473090990512305\ 10^{-9}]$

Table A-5: Iteration Results using Variational Iteration Method

eta	Iteration 1	Iteration 2	Iteration 3	Iteration 4
0.1	0.033206	0.033205	0.033205	0.033205
0.2	0.066411	0.066408	0.066408	0.066408
0.3	0.099617	0.099598	0.099598	0.099598
0.4	0.132823	0.132764	0.132764	0.132764
0.5	0.166029	0.165885	0.165885	0.165885
0.6	0.199234	0.198936	0.198937	0.198937
0.7	0.23244	0.231888	0.23189	0.23189
0.8	0.265646	0.264705	0.264709	0.264709
0.9	0.298851	0.297344	0.297354	0.297354
1	0.332057	0.32976	0.32978	0.32978
1.1	0.365263	0.361899	0.361938	0.361938
1.2	0.398468	0.393705	0.393777	0.393776
1.3	0.431674	0.425113	0.425238	0.425236
1.4	0.46488	0.456055	0.456265	0.456261
1.5	0.498086	0.486456	0.486796	0.486789
1.6	0.531291	0.516237	0.51677	0.516756
1.7	0.564497	0.545311	0.546126	0.5461
1.8	0.597703	0.573588	0.574803	0.574756
1.9	0.630908	0.600972	0.602743	0.602663
2	0.664114	0.62736	0.629893	0.629759
2.1	0.69732	0.652645	0.656204	0.655986
2.2	0.730525	0.676714	0.681635	0.681288
2.3	0.763731	0.699448	0.706153	0.705614
2.4	0.796937	0.720724	0.729738	0.728914
2.5	0.830143	0.740411	0.752381	0.751144
2.6	0.863348	0.758375	0.77409	0.772261
2.7	0.896554	0.774475	0.79489	0.792225
2.8	0.92976	0.788566	0.814826	0.810992
2.9	0.962965	0.800494	0.833968	0.828517
3	0.996171	0.810104	0.85241	0.844744
3.1	1.029377	0.817233	0.870274	0.859597
3.2	1.062582	0.821712	0.887713	0.872976
3.3	1.095788	0.823368	0.904916	0.884738
3.4	1.128994	0.822021	0.922108	0.894677
3.5	1.1622	0.817488	0.939552	0.902499
3.6	1.195405	0.809577	0.957554	0.907788
3.7	1.228611	0.798093	0.976466	0.909954
3.8	1.261817	0.782835	0.996685	0.908173
3.9	1.295022	0.763597	1.018658	0.901303
4	1.328228	0.740165	1.042884	0.887769
4.1	1.361434	0.712322	1.069912	0.865424

eta	Iteration 1	Iteration 2	Iteration 3	Iteration 4
4.2	1.394639	0.679845	1.100347	0.831354
4.3	1.427845	0.642505	1.134845	0.781636
4.4	1.461051	0.600067	1.174119	0.711015
4.5	1.494257	0.552293	1.218933	0.612498
4.6	1.527462	0.498936	1.270102	0.476824
4.7	1.560668	0.439746	1.328492	0.2918
4.8	1.593874	0.374466	1.395014	0.041444
4.9	1.627079	0.302834	1.47062	-0.29509
5	1.660285	0.224584	1.556296	-0.74486
5.1	1.693491	0.139442	1.653057	-1.34285
5.2	1.726696	0.047129	1.761939	-2.13405
5.3	1.759902	-0.05264	1.883984	-3.17614
5.4	1.793108	-0.16015	2.020233	-4.54269
5.5	1.826314	-0.2757	2.171705	-6.32722
5.6	1.859519	-0.39958	2.33939	-8.64811
5.7	1.892725	-0.53212	2.524218	-11.6546
5.8	1.925931	-0.67361	2.727048	-15.5343
5.9	1.959136	-0.82437	2.948635	-20.5219
6	1.992342	-0.98473	3.189608	-26.9102
6.1	2.025548	-1.15501	3.450435	-35.0629
6.2	2.058753	-1.33555	3.731387	-45.4307
6.3	2.091959	-1.52669	4.032503	-58.5701
6.4	2.125165	-1.72877	4.353543	-75.1656
6.5	2.158371	-1.94214	4.693942	-96.057
6.6	2.191576	-2.16715	5.052757	-122.271
6.7	2.224782	-2.40418	5.428606	-155.06
6.8	2.257988	-2.65358	5.819606	-195.946
6.9	2.291193	-2.91572	6.223306	-246.774
7	2.324399	-3.19099	6.636603	-309.774
7.1	2.357605	-3.47977	7.055667	-387.636
7.2	2.39081	-3.78244	7.475842	-483.594
7.3	2.424016	-4.09941	7.891556	-601.525
7.4	2.457222	-4.43106	8.296207	-746.071
7.5	2.490428	-4.77781	8.68205	-922.769
7.6	2.523633	-5.14007	9.040073	-1138.22
7.7	2.556839	-5.51824	9.35986	-1400.25
7.8	2.590045	-5.91276	9.629447	-1718.15
7.9	2.62325	-6.32406	9.835166	-2102.92
8	2.656456	-6.75256	9.961474	-2567.52
8.1	2.689662	-7.19869	9.990775	-3127.23
8.2	2.722867	-7.66292	9.903229	-3800.02
8.3	2.756073	-8.14568	9.676536	-4606.97
8.4	2.789279	-8.64743	9.285724	-5572.77
8.5	2.822485	-9.16864	8.702906	-6726.29

eta	Iteration 1	Iteration 2	Iteration 3	Iteration 4
8.6	2.85569	-9.70975	7.897031	-8101.23
8.7	2.888896	-10.2713	6.833611	-9736.85
8.8	2.922102	-10.8536	5.474432	-11678.8
8.9	2.955307	-11.4574	3.777257	-13980.2
9	2.988513	-12.0829	1.695488	-16702.4
9.1	3.021719	-12.7308	-0.82217	-19916.8
9.2	3.054924	-13.4015	-3.83208	-23705.6
9.3	3.08813	-14.0955	-7.39605	-28163.9
9.4	3.121336	-14.8134	-11.5818	-33401.4
9.5	3.154542	-15.5557	-16.4633	-39544.3
9.6	3.187747	-16.3228	-22.1213	-46737.7
9.7	3.220953	-17.1153	-28.6439	-55148.5
9.8	3.254159	-17.9338	-36.1268	-64968
9.9	3.287364	-18.7787	-44.6743	-76415.5
10	3.32057	-19.6506	-54.3992	-89742.1