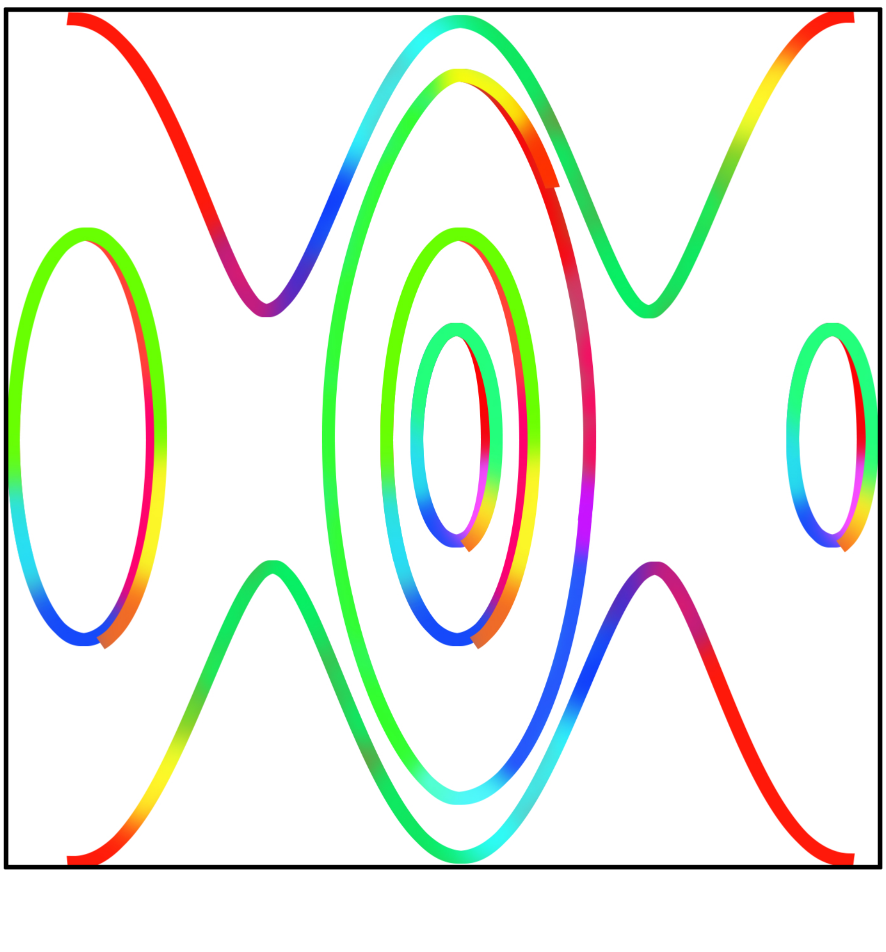
ORDINARY DIFFERENTIAL EQUATIONS AND APPLICATIONS II:

WITH MAPLE ILLUSTRATIONS



Benjamin Oyediran Oyelami

Bentham Books

Ordinary Differential Equations and Applications II: With Maple Illustrations

Authored by

Benjamin Oyediran Oyelami
Department of Mathematics,
Plateau State University,
Bokkos, Nigeria

National Mathematical Centre,
Abuja, Nigeria

Baze University,
Abuja, Nigeria

&

University of Abuja,
Abuja, Nigeria

Ordinary Differential Equations and Applications II: with Maple Illustrations

Author: Benjamin Oyediran Oyelami

ISBN (Online): 978-981-5313-86-4

ISBN (Print): 978-981-5313-87-1

ISBN (Paperback): 978-981-5313-88-8

First published in 2024.

Bentham Science Publishers Pte. Ltd.
80 Robinson Road #02-00
Singapore 068898
Singapore
Email: subscriptions@benthamscience.net

BENTHAM SCIENCE

CONTENTS

FOREWORD

I strongly endorse this exceptional book on the subject of differential equations. It covers all aspects of the field. It has a solid theoretical foundation and an applied focus, with many practical examples. It demonstrates how to program them using Maple, which is a leading mathematical software; and finally, it demonstrates how to generate graphics that clearly represent the nature of solutions and provide deep insights into them. All of these aspects are essential in the use of differential equations in modern mathematics, science, and technology. Thus, the book is equally useful for mathematicians, scientists, and engineers. As engineers should have some understanding of the theory of differential equations, also mathematicians should be able to program and generate graphical results.

This volume is especially valuable because it presents all of these aspects in an integrated fashion. It is written by a true expert in the field, an experienced teacher who has also carried out significant applied research. As a master teacher, Dr. Oyelami presents the material in a simple, straightforward, easy-to-follow manner. As an expert researcher, he knows first-hand the power of differential equations as a modeling tool, and his love for the field is clearly visible. The volume is also comprehensive in its coverage, especially in the areas of differential equations of the greatest practical interest. The students who study this material will be thoroughly prepared for employment in technical fields that use differential equations for modeling purposes. Such a student will also find the book to be a valuable continuing reference, both for its clear theoretical presentations and its useful and generalizable computer codes.

Christopher Thron
Associate Professor of Mathematics,
Texas A&M University, Central Texas, USA

ENDORSEMENT

I have thoroughly gone through this book, which can be considered to be a compendium of knowledge on Differential Equations at the Undergraduate levels in all ramifications. The book presents poignantly insightful views on quantitative and qualitative modeling, as it unleashes the tremendous power of differential equations techniques, with applications on current multifarious trends, including population dynamics, spread of viruses and diseases and neural networks.

This book places the generally neglected implementation aspects of mathematical results on the front burner, with special implementations on the platform of Maple. In the above regard, the contributions of this book are exceptional and unprecedented. In terms of scope and diversity, the reader will be surprised by the unfathomable depth of knowledge and broad horizon of the author on the mathematical modelling of continuous processes by the deft deployment of differential equations.

The book must be highly acclaimed for its balanced coverage of the theory, applications, and computational issues of differential equations and their solutions. It gives an effective exposition of differential equations and concepts with functional analytic support, as needed, with meticulously chosen examples, exercises and extensive use of Maple, currently regarded as the best mathematical software. This is the main thrust of the book, as it encompasses and emphasizes current trends of modern computational tools in enhancing the effectiveness of differential equations as an indispensable and core tool for modelling of processes that exist in the continuum.

On the other hand, the book reinforces the reader’s understanding of ordinary differential equations, which, on the other hand, simulates and enhances the readers’ interest and curiosity about the immense modelling possibilities on ordinary differential equations platforms.

This book vividly brings to the fore, the inconvertible fact that, for the most part, ordinary differential equations cannot be precluded in the modelling of real-life phenomena. This being an exceptionally well-crafted book with an abundance of realistic, well-researched examples, illustrations and exercises, will enliven discussions of ordinary differential equations, techniques and key modelling objectives that the reader will likely encounter in undergraduate courses and much more.

In view of the aforementioned attribute coupled with its lucid presentation, novelty of the abstract of each chapter and emphasis on digital implementations, this book deserves the highest recommendation. The book is a 'must-read 'and 'unputdownable'.

Professor Ukwu Chukwunenye
Functional Differential Equations,
Control theory & Industrial Engineering Specialist,
Department of Mathematics,
University of Jos, Jos, Nigeria

COMMENTS FROM RENOWNED SCIENTISTS

‘‘I have gone through the whole book. It is simple, clear and easy to read and understand .I have no doubt in my mind that the book is a must for all students of Mathematics in Tertiary Institutions’’

Professor M.O. Ibrahim
University of Ilorin
Ilorin and former President of Mathematics Association of Nigeria (MAN)

‘‘The book will be a very good choice for both professionals across all fields of endeavours. The fact that the book does not assume familiarity with some basic mathematical concepts is an incentive in its appeal to those who have been out of school for some time. These qualities will increase its sellable quality in the market place as well as a recommendation to students on mathematical courses’’.

Professor Emeritus A.A. Asere
Department of Mechanical Engineering,
Obafemi Awolowo University, Osun,
Nigeria

PREFACE

Ordinary differential equations are powerful tools for modeling and analyzing complex phenomena in various fields. Understanding ODEs is essential for making accurate predictions, optimizing systems, and solving real-world problems. ODEs are vital tools in study involving climate change, population dynamics, economic growth, chemical reaction, resource management, epidemiological growth of diseases and pandemic and drug administration.

This textbook is an encyclopedia of techniques for finding solutions to ordinary differential equations. It was developed when lecturing students and researching at the Abubakar Tafawa Balewa University, Bauchi; Kaduna State University, Kaduna, Nigerian; Nile University, Abuja ; Plateau State University Bokkos, University of Abuja and Baze University Abuja all in Nigeria.

This book comprises nine chapters and it is on 'Vector valued ordinary differential equations and applications'. The Chapters are written bearing in mind beginners in the field of study who have little or no background on the course. This requirement is met by deployment of lucid and self-instructional language and utilization of scintillating examples throughout the book as well as illustration using Maple modeling and simulation software.

The first chapter contains preliminaries like set theory, topological concepts and the formulation of vector differential equations. The second chapter and the third chapter are on Linear differential equations in the linear space , basic concepts related to topological structures are discussed such structures are Normed and Banach spaces as applicable to solutions of ordinary differential equations. The proof of existence and uniqueness of solution for initial value problems (IVP), the 'power house' of course finds its shape from fixed points. Peano's existence theorem and Picard Lindelof theorem are exploited in no small measure. The fourth chapter is on solutions to matrix initial value problems.

The fifth chapter is about canonical transformation, a kind of transformation from scalar equations to vector equations. This chapter ends with the treatment of exponential matrices and estimation theory. The sixth is on Stability theory, Stability is a kind of graduation from continuous dependence on initial data localized to some finite interval of $E = (-\infty, +\infty)$ to more global generalized concepts. The seventh chapters examine the linear periodic systems with the

Floquent rule extensively utilized. Also treated in this chapter are stability of linear perturbed systems and applications to neural firing models, avian influenza, population models. The ninth chapter is on numerical solutions to ODEs and applications to some models respectively

Every part of the chapters in this textbook contains preambles without assuming students' familiarity with some basic mathematical concepts. Hence it will prove to be a valuable and supplementary textbook for other courses in Mathematics and Engineering.

Benjamin Oyediran Oyelami
Department of Mathematics, Plateau State University,
Bokkos, Nigeria, National Mathematical Centre, Abuja, Baze University,
Abuja, Nigeria

ACKNOWLEDGEMENTS

My profound gratitude goes to Professor P. Smoczynski of the Department of Mathematics and Statistics, Simon Fraser University, Canada who first introduced me to Differential Equations and sustained my interest in the field. I am indebted to Professor Styr University of Botswana and late Professor Olaofe, University of Ibadan both of them taught me Numerical Analysis at undergraduate and postgraduate levels respectively. In ship of thanks are Prof. Ukwu Chukwunenye my Lecturer in the University of Jos Nigeria; Mr. Salam Mukaila a friend, late Professor M. Ibiejugba Koji State University Ayingba, Professor G. Abimbola and Professor M. O Ibrahim University of Ilorin, Nigeria; Professor Christopher Thron and Gwenda Lynn Anders, Texas A&M University-Central Texas, USA; late Professor P.C. Ram, and Professor M S.Sesay , Abubakar Tafawa Balewa University Bauchi, Nigeria.

Furthermore, I am grateful to: Professor S O Ale, National Mathematical Centre, Abuja, Nigeria; Professor A.A. Asere, Obafemi Awolowo University Ile-Ife Nigeria; Late Professor D.D. Bainov, Medical University Sofia Bulgaria; Professor Olusola Akinyele, Bowie State University, Maryland USA for their exposure to Impulsive differential equations and my mentor, Professor Emeritus Trench William Trinity College USA , and Professor R.A.T. Solarin and Professor Stephen Onah, the former Directors and Chief Executive of National Mathematical Centre(NMC), Abuja, Nigeria respectively. Professor Promise Mebine, the Present Director and Chief Executive of NMC. I am grateful to Professor Femi Taiwo Obafemi Awolowo University Ile-Ife. We are Co-Trainer for Maple Software across some Nigerian Universities. I am greatly indebted to the former Vice Chancellor, Plateau State University Bokkos, Professor Danjuma Sheni who through TETfund Research grant for preparation of this book. I am also grateful to Colleagues at the National Mathematical Centre, University of Abuja, Abuja and the Baze University all in Abuja, Nigeria.

I am grateful to the above academicians for their technical advice, thought provoking suggestions and eagle-eye proofreading of the whole manuscript. I am grateful to Mr. Anthony Oluloye of Tangier Company, who provided me with Maple 17, 18, 2015,2018,2019,2020,2021,2022,2023, MapleSim 7 and MapleSim 2023 gratis.

Finally, I must not forget my family especially my wife Keith E. Oyelami and my children; Moses, Ruth, Victoria, Miracle, David and Hannah for their contribution in making the publication of this book successful. I am grateful, God bless you all.

DEDICATION

Dedicated to God Almighty, most merciful, the author of life, the giver of knowledge, wisdom and understanding. The procreator, sustainer, and annihilator of all life processes. God is the greatest problem solver who can solve a problem in an infinitely many ways.

CHAPTER 1

Vector-Valued Differential Equations and Related Analysis Concepts

Abstract: This chapter starts with the revision of basic concepts in real, complex, and functional analyses. Vector-valued differential equations are formulated and conditions for generating solution bases for the differential equations are stated.

Keywords: Basic concepts functional analyses, Solution bases, Vector-valued differential equations.

INTRODUCTION

What are Ordinary Differential Equations (ODEs)?

The branch of mathematics that studies equations involving derivatives of unknown functions is called differential equations. There are two classes of such equations that are classified according to the number of unknown variables involved. A differential equation is a relationship between an independent variable, x and dependent variable y, and one or more derivatives of y with respect to x. Differential equations with a single unknown variable are called ordinary differential equations (ODEs). ODEs find applications in mathematical physics, electrical engineering, and mechanical engineering, for example in the vibration of strings [5-8]

Ordinary Differential Equations (ODEs) are mathematical equations that describe how a quantity changes over time or space. They involve an unknown function and its derivatives, and are used to model a wide range of phenomena in science, engineering, economics, and other fields [1, 8].

ODE describes change over time or space .Typically involves rates of change (*e.g.*, velocity, acceleration) and can be linear or nonlinear. ODEs can be solved using various methods, including: Analytical methods, numerical methods, approximation methods (perturbation theory)

Benjamin Oyediran Oyelami

In this textbook, efforts will be devoted to vector valued differential equations [5-8]. The system will be formulated in matrix form and solutions obtained in matrix form.

In most studies on scalar differential equations, the fundamental assumption made is that the solutions of the scalar equations exist and are uniquely determined. This may not be true in the general setting, therefore the need to establish the framework for existence and uniqueness of solutions of ODEs. Theorems that guarantee the existence and uniqueness of solutions of ordinary differential equations [5-8] will be given in chapter three. Here we are considering the building blocks of tools for theorems on existence and uniqueness of solutions of ODES and associated analysis.

PRELIMINARIES

Let us briefly review some of the familiar notions in the set theory relevant to our discussion in subsequent chapters.

Open Sets

A set X is open, if there exists a neighborhood or ball that lies entirely in the set. Geometrically, a set X is open if we consider a sphere (ball) centered at x_0 with arbitrary radius r_0 which lies entirely inside the set. In set notation, we write $S(x_0) \subset X$ (See [3,4]

Example 1.1

An arbitrary sphere in n-tuples Euclidean space:

1. $$S(x_0, r_0) = \left\{ \begin{array}{l} (x_1, x_2, ..., x_n) \in E^n : (x_1 - x_{01})^2 + (x_2 - x_{02})^2 + (x_3 - x_{03})^2 + ... + (x_n - x_{0n})^2 < r_0 \\ , (x_{01}, x_{02}, ..., x_{0n}) \in E^n \end{array} \right\}$$

2. $$S(x_0, r_0) = \left\{ (x, y) \in E^2 : |x| < a, |y| < a, a \in E^1, a > 0 \right\}$$

Closed Set

A set X is closed if every open set in X lies in it, its boundary is inclusive.

Example 1.2

A square $N(x_0, r_0)$ such that $N(x_0, r_0) = \{(x_1, x_2) \in E^2 : |x_1| \le a, |x_2| \le a, a > 0\}$.

A set could be open and closed simultaneously, examples are:

1. $X = \{x \in E^1 : 0 \le x < 1\}$

2. $Y = \{x \in E^2 : |x_1| \le a, |x_2| < b, x = (x_1, x_2)\}$

3. $X_3 = \{x \in E^2 : (x_1 - x_{01})^2 + (x_2 - x_{02})^2 < 1\} \bigcap \{x \in E^2 : |(x_1, x_2)| \le 1\}$.

Bounded Set

X is a bounded set if there exists a positive constant M such that $|x| \le M$ for every $x \in X$.

Compact Set

X is compact if any closed open subset whose union contains X has a finite subclass whose union also contains X. Heine-Borel theorem asserts that for a finite-dimensional Euclidean space, compactness is equivalent to closedness and boundedness, for example:

$R(x, y) = \{(x, y) \in E^2 : |x| \le a, |y| \le b\}$ is compact in E^2.

Reformulation of compactness in terms of open cover is given by William [4]. The equivalent definition to compactness is the Weistrass-Bolzano theorem, which is stated as follows: Any infinite sequence $\{x_n\}$ of X has a subsequence $\{x_{nk}\}$ which converges to $x \in A$. This is often called subsequent compactness from a topological point [2,3].

Connected Set

Set X is connected if there exist two sets or points in X joined by an arbitrary line segment that lies entirely within X.

CHAPTER 2

Differential Equations in the Linear Spaces

Abstract: Fundamental concepts in normed spaces are elucidated and linear systems are considered and applied to some problems.

Keywords: Gronwall's inequality, Linear systems, Norms, Normed spaces.

INTRODUCTION

In this chapter, we introduce fundamental concepts useful for studying differential equations in linear space. The topological structure on spaces such as the Norm and Normed Space will be defined with some examples. We will also present Gronwall's inequality, the cornerstone of estimation theory in normed spaces [1, 6, 7, 8]. Many problems in differential equations are from Banach spaces, that is, complete normed spaces.

PRELIMINARIES

A scalar function defined on the linear space $X(F)$ $i.e., \|.\|: X \to E^+, E^+ = [0, +\infty)$ is called a norm on X (see [1-8])

If the following conditions are satisfied:

1. $\|x\| \geq 0, \|x\| = 0$ if and only if $x = 0$, for $x \in V$
2. $\|x + y\| \leq \|x\| + \|y\|$
3. $\|\alpha x\| = |\alpha| \|x\|$

Property (ii) is called triangular inequality. This can be generalized to a finite number of arbitrary vectors in V $i.e., \|x_1 + x_2 + ... + x_n\| \leq \|x_1\| + \|x_2\| + ... + \|x_n\|$. At times, the triangular inequality is often referred to as Minskwoski 's inequality, if $\|.\|$ is defined on a metric space and satisfies all the properties of a norm, then the metric space is a normed space.

The couple $[\|.\|, X]$ forms a normed space. A normed space with the additional structure of being linear (Chapters 5 and 6) is referred to as a normed linear space.

Example 2.1

1. The space of continuous functions $C(E^n)$ defined in E^n is equipped with a norm given by the pair $[C(E^n), \|.\|]$, forms a normed space [3, 4].
2. The vector space of bounded continuous functions from the interval J to the set of positive real numbers. *i.e.*, with the norm, $\| f \| = \sup_{x \in E^n} | f | < \infty$,

$$X = \left\{ f \in C(J, E^+) : \| f \| = \sup_{t \in J} | f(t) |, J = [0, +\infty), \forall t \in J \right\}.$$

3. The Euclidean space E^n endowed with any of the following norms forms a normed space:

$$\|x\| = \sum_{i=1}^{n} |x_i| \text{ for } x = (x_1, x_2, ..., x_n) \in E^n$$

$$\|x\| = \max_{1 \le i \le n} |x_i|, \|x\| = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2} \tag{2.1}$$

4. $L(J, Y)$ is the Banach space+ of all Lipchitzian functions in J strongly differentiable everywhere except for some finite number of points with range in the Banach space Y. $L(J, Y)$ is endowed with the sup norm $\| f \| = \sup_{f \in L(J,Y)} | f |$.

+Readers familiar with metric spaces will quickly recognize that in the above equation, the three norms are equivalent in E^n and as a matter of fact, this forms what is called topological isomorphism in the normed space. A linear space is a vector space, which, in addition, is linear (See William [4] and Chen [1]). Here it is not our interest to study topological structures in detail [2-4].

Linear Systems

The general first-order n-dimensional linear system is a system of the form:

$$\dot{x}(t) = A(t)x(t) + f(t) \quad \textbf{(2.2)}$$

where $A(t)$ is the $n \times n$ matrix function of t whose elements $[a_{ij}]_{i,\, j=1,\ 2\ ..\ n}$ are functions of $t \in E^1 = (-\infty, +\infty)$; I is an open and connected in the sub-interval of R. If $f(t) = 0$, equation (2.2) is said to be a homogeneous linear system, otherwise, it is termed nonhomogeneous.

We remark that; if $(t_0, x_0) \in I \times \Omega$ *with* $|t_0| < \infty.\ \|x_0\| < \infty$, $y(t_0) = x_0$. Then there exists a unique solution (t, x) passing through $(t_0, x(t_0))$ [3, 5]. This, would be justified in due course when the existence and uniqueness theorems are treated.

Lemma 2.1 (Gronwall's inequality) (See [5-8])

Let α be a nonnegative real constant and let ϕ and β be nonnegative and integrable on some interval $[a,b]$ such that:

$$\phi(t) \le \alpha + \int_{t_0}^{t} \beta(s)\phi(s)ds, a \le t \le b \quad \textbf{(2.3)}$$

For $a \le t_0 \le t \le b$.

Then:

$$\begin{aligned} &\phi(t) \le \phi(a)\exp(\int_{t_0}^{t} \beta(s)ds) \\ &\text{or } \phi(t) \le \alpha \exp(\int_{t_0}^{t} \beta(s)ds) \end{aligned} \quad \textbf{(2.4)}$$

CHAPTER 3

Fixed Point Theorems, Existence and Uniqueness of Solutions of Differential Equations

Abstract: In differential equations, one of the cornerstones of most theorems and their framework (hypotheses) is the existence and uniqueness theorem. We consider theorems that guarantee the existence and uniqueness of solutions of differential equations. We consider the Carathedory theorem, Peano existence theorem, Picard-Linderlof existence and uniqueness theorem, Brower and Schauder fixed point theorems. Picard successive approximation method is applied to establish the existence and uniqueness of solution to some initial value problems. Moreover, the conditions for the continuation of solutions from a given interval to an extended interval are also derived for ordinary differential equations.

Keywords: Existence, uniqueness, Differential equations, Carathedory theorem, Fixed point theorems, Peano existence theorem, Picard-Linderloft existence, uniqueness theorem, Picard successive approximation method.

FIXED – POINT THEORY

Modern theorems on the existence and uniqueness of solutions to differential equations are from the so-called fixed theorems for which there are many versions [3,6-10]. Fixed point theory significantly utilizes some elements in functional analysis. This approach makes it a sophisticated and effective tool for solving differential equations.

Fixed point theory has a variety of applications widely used in both integral equations and operator theory [9-11]. Our goal or prime concern in this chapter is its application to initial value problems (IVP). Erwin [4], pp. 316 – 326) contains catalogues of applications of fixed-point theorems to integral equations and a systems of equations [14-15]. In particular, Garret and Gian [5] developed many iterative algorithms on fixed point theorems.

Furthermore, the following three types of fixed-point theorems are given: Schauder, Banach – Caccioppolis and Brower's fixed-point theorems. A concise statement of the theorems as was observed by Smart [14] is that every continuous mapping of a compact convex set to itself must have a fixed point. Besides, fixed point is a topological concept. It is fair for us to conclude from Smart's assertion that any closed interval $[a,b]$ in E and a unit disc in E^2 must have a fixed-point property.

Benjamin Oyediran Oyelami

Finally, the celebrated Picard-Linderlof existence and uniqueness theorem will be proved by a fixed-point theorem, to be specific, by Banach-Caccioppolis theorem. An alternative proof is found in Jack [7] where the proof was made by Schauder's fixed point theorem.

Peano Existence Theorem [see [7], pp. 14 – 15]

If f is continuous in a domain D, then for any initial data (t_0, x_0), there exists at least one solution of the differential equation:

$$\dot{x}(t) = f(t, x(t)) \tag{3.1}$$

passing through $(t_0,\ x_0)$. Note if the assumption on f in the piano existence theorem is satisfied, then the existence of infinitely many solutions to the equation (3.1) is guaranteed. We will see later on that for initial value problems, whenever their solutions exist, it is always uniquely determined if f is bounded together with its first derivative or if it satisfies the Lipchitz condition.

Method of Picard Successive Approximation

Solutions of differential equations can be approximated by sequences of points starting from the initial data. The successive approximated sequence of solutions $\{x_k\}$ converges to the actual solution of initial value problem (IVP) as $t \to \infty$, *i.e.* $x_k \to x$ as $t \to \infty$.

Picard presents an iteratively appealing method now termed Picard successive approximation method. This method overcomes the problem of obtaining solutions to differential equations via approximations. The method is due to J. Liouville and others in the early 1800s, but, often credited to E. Picard and hence it is called the Picard iterative method. E. Picard further developed the method in 1893 See [1] and [7].

The method is as follows:

Let $\dot{x}(t) = f(t, x(t)), x(t_0) = x_0$, where t belongs to the half real line, $J = [0, +\infty)$ such that $f(t, x(t))$ is continuous and locally Lipchitzian in a rectangular domain

$R(\alpha,\beta)=\left\{(t,\alpha(t):\left|t-t_0\right|\le\alpha,\left|x(t)-x_0\right|\le\beta\right\}$.Then that there exists a unique solution to equation (3.1) given by the successive approximation scheme:

$$x_k(t)=x_0+\int_{t_0}^{t} f(s,x_{n-1}(s))ds, t\ge t_0 \text{ for } k\in\left\{0,1,2,...\right\}$$

The sequence $\left\{x_k\right\}$ converges uniformly to the solution $x(t)$ of equation (3.1) in the interval $J=[t_0,+\infty)$.

Lemma 3.1

Let $$x_n(t)=x_0+\int_{t_0}^{t} f(s,x_{n-1}(s))ds, t\ge t_0, m=\sup_{t\in[0,T]}\left|f(t,x(t))\right|$$

Then:

$$\left\|x_1-x_0\right\|\le m\left|t-t_0\right| \tag{3.2}$$

$$\| x_2 - x_1 \|\le\frac{L^2 m}{2!}\left|t-t_0\right| \tag{3.3}$$

L is Lipchitz constant. By induction on k,

$$\| x_{k+1} - x_k \|\le\frac{L^k m}{(k+1)!}\left|t-t_0\right|^{k+1} \tag{3.4}$$

Hence:

$$\begin{aligned}\sum_0^\infty\left\|x_{k+1}-x_k\right\| &\le\frac{m}{L}\sum_0^\infty\frac{[L\left|t-t_0\right|]^{k+1}}{(k+1)!}=\frac{m}{L}[\exp L\left|t-t_0\right|-1]\\ &\le\frac{M}{L}[\exp(\alpha L)-1] \qquad (3.5)\\ &\le\frac{M}{L}[\exp(L\left|t-t_0\right|)-1]\end{aligned}$$

CHAPTER 4

Matrix Solution to Initial Value Problems

Abstract: In this chapter, we will consider methods for estimating the norm of a matrix and matrix exponents. The conditions for the existence and uniqueness of solutions are considered for ordinary differential equations using the Lipchitz conditions. Adjoint systems are revisited together with the application of the Carathedory theorem to some selected problems.

Keywords: Adjoint systems, Carathedory theorem, Lipchitz conditions, Matrix exponents, Ordinary differential equations, Uniqueness.

INTRODUCTION

We extend the idea of the norm of a vector to a matrix. Matrix exponentials of a vector-valued differential equation (VDEs) are found in this chapter. Fundamental matrix solutions will be obtained for VDEs together with their corresponding adjoint systems using a vector version of variation of constant parameters.

Consider:

$$\dot{x}(t) = A(t)x(t) \tag{4.1}$$

On $J = [\alpha, \beta]$, the matrix $A(t)$ plays a central role. It will be useful to evaluate a norm for a matrix [1-5].

We define such a norm and illustrate its usefulness in estimation and continuous dependence of the solution of equation (4.1) on the initial data $x(t_0) = x_0$.

NORMS FOR MATRICES

Definition 4.1

The norm of matrix A with elements $[a_{ij}]$ will be defined as:

$$\|A\| = \max_{1 \le j \le n} \sum_{i}^{n} |a_{ij}| \tag{4.2}$$

Benjamin Oyediran Oyelami

i.e. the maximum of the sum of the absolute value of elements in each column. Many other definitions are available, for example, an alternative definition analogous to the norm of an operator is given as:

$$\begin{aligned}\|A\| &= \lim\sup_{\|x\|\le 1}\|Ax\| \\ &= \lim\sup_{\|x\|\ne 0}\frac{\|Ax\|}{\|x\|} \\ &= \inf\left\{k:\|Ax\|\le\|x\|\right\}\end{aligned} \quad \textbf{(4.3)}$$

i.e. the best upper bound for Ax .

At times, in some texts, one finds it being defined as:

$$\|A\| = \sum_{j=1}^{n}\sum_{i=1}^{n}|a_{ij}| \quad \textbf{(4.4)}$$

i.e. the sum of the absolute entries of matrix $[a_{ij}]$. It must be mentioned that each of these norms is equivalent to one another, the square of the norm $\|A\|$, *i.e*. $\|A\|^2$ in the equation. (4.3) is the maximum of each value of matrix A.

Norm of a Vector

The norm of a vector with components $e_1, e_2, ..., e_n$

is defined by $$\|E\| = \sum_{i=1}^{n}|e_i|$$

Note, $\min_{i,j}\|a_{ij}\|\|x\| \le \|Ax\| \le \max_{i,j}\|a_{ij}\|\|x\|$ (see [1,7,9-11])

Observe that $\|I\| = 1$.

Properties of Matrix Norm

Let A and B be square matrices and I is a column n-vector

Such that:

i. $\|A\| \geq 0,\ \|A\| = 0 \quad \Rightarrow A = [0],$ zero matrix

ii. $\|A + B\| \leq \|A\| + \|B\|$ (Triangular inequality)

iii. $\|\alpha A\| = |\alpha| \|A\|$ For every α complex number.

iv. $\|A\varepsilon\| \leq \|A\| \|\varepsilon\|$

The reader will quickly notice properties; (i) through (iii) as those of a norm.

Properties together with the vector space of square matrices constitutes a normed space**.**

Example 4.1

Prove the above-stated properties of the norm of a matrix.

1. $\|A\| = \max_{j \in \{1,2,3,\ldots,n\}} \sum_{i=1}^{n} |a_{ij}|$. Since $|a_{ij}| \geq 0$, $|a_{ij}| = 0$ if and only if $a_{ij} = 0$. Therefore,

$$\|A\| \geq 0, \|A\| = 0 \text{ if and only if } \quad A = [0]$$

$$\begin{aligned} \|A + B\| &= \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij} + b_{ij}| \\ &\leq \max_{i,j} \sum_{i,j} \left(|a_{ij}| + |b_{ij}| \right) \\ &= \|A\| + \|B\| \end{aligned}$$

2. $$\|AE\| = \max_{1 \leq j \leq n} \sum_{1 \leq i \leq n} |a_{ij}| \|E_j\| = \|A\| \|E\|$$

In the same vein,

$$\begin{aligned} \|AB\| &= \max \sum_{j=1}^{n} \sum_{i=1}^{n} |a_{ij} b_{ij}| \leq \max \sum_{j=1}^{n} \sum_{i=1}^{n} |a_{ij}| |b_{ij}| \\ &\leq \|A\| \|B\| \end{aligned}$$

CHAPTER 5

Canonical Transformation and Matrix Solutions of Differential Equations

Abstract: We consider canonical transformation for transforming scalar differential equations to matrix differential equations. We determine conditions for linear independence of solutions using the Wronskian method and use the Jordan canonical form to find bounds for solutions of ODES. Also considered are: the generalized eigenvectors method for obtaining matrix solutions to ODES and corresponding bounds for the autonomous differential equations, upper and lower bounds for solutions. Conditions for continuous dependence of solutions on initial data are formulated. Periodic systems are studied too with the application of the Floquet rule to finding solutions to some linear periodic systems. The Theorem on how to construct monodromy matrices is presented for the linear periodic systems together with some examples.

Keywords: Autonomous differential equations, Canonical transformation, Floquet rule, Jordan canonical forms, Matrix solutions, Monodromy matrices, ODES solutions, Periodic systems, Upper and lower bounds, Wronskian method.

INTRODUCTION

This chapter looks at the canonical transformation method, and transformations of scalar equations into vector forms [1-5]. The fundamental matrix, principal matrix, and adjoint to homogenous systems are revisited. The theory of autonomous linear homogeneous systems will be introduced together with the canonical transform process in Jordan Canonical form. We will construct fundamental matrix solutions using Sylvester's formula and derive upper and lower solution bounds to ODES. We will investigate how solutions of ODEs continuously depend on initial data. At the end of the chapter, linear periodic systems and applications are to be considered with examples given.

Canonical Transformation

Suppose we have a scalar differential equation:

$$D^n y + a_1(t)D^{n-1} + ... + a_n(t)y = g(t) \tag{5.1}$$

Benjamin Oyediran Oyelami

Our interest is to find an equivalent vector equation having the same solution as the equation (5.1).

Canonical transformation provides us with a methodology.

Let:

$$\left.\begin{array}{l} y = y_1 \\ Dy = Dy_1 = y_2 \\ D^2 y = D^2 y_1 = Dy_2 = y_3 \\ \vdots \\ \text{Then} \\ D^{n-1} y = -a_n y_1 - a_{n-1} y_2 - \cdots - a_1 y_2 + g(t) \end{array}\right\} \tag{5.2}$$

Let $x = (y_1, y_2, ..., y_n)^T$ then:

$$\dot{x} = A(t)x + h(t) \tag{5.3}$$

Where:

$$A(t) = \left[a_{ij}\right]_{i,j\in\{1,2,...,n\}} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & a_{n-2} & \cdots & -a_2 & -a_1 \end{bmatrix} \tag{5.4}$$

$h(t) = [0, 0, \cdots, g(t)]^T$

We recall the definition of Wronskian of $[\phi_1, \phi_2, \cdots, \phi_n]$ of (n) continuously differentiable functions:

$$W[\phi_1,\phi_2,\cdots,\phi_n]=\begin{vmatrix}\phi_1 & \phi_2 & \cdots & \phi_n\\ \dot{\phi}_1 & \dot{\phi}_2 & \cdots & \dot{\phi}_n\\ \vdots & \vdots & \vdots & \vdots\\ \phi_1^{(n-1)} & \phi_2^{(n-1)} & \cdots & \phi^{(n-1)}{}_n\end{vmatrix} \qquad \textbf{(5.5)}$$

We said in (Chapter 3 ,section 3) that $\phi_1,\phi_2,\cdots,\phi_n$ are linearly independent on the interval $[t_0,+\infty)$ if $W[\phi_1,\phi_2,\cdots,\phi_n]$ does not vanish for at least one $t\in[t_0,+\infty)$. Otherwise, triviality of the Wronskian $W[\phi_1,\phi_2,\cdots,\phi_n]$ implies the linear dependence of $[\phi_1,\phi_2,\cdots,\phi_n]$ (see [6, 8]).

Theorem 5.1

Let $\dot{x}(t)=A(t)x(t), x(t_0)=x_0$, $t\in[t_0,+\infty)$, where $A(t)$ is an n-square matrix function of t on the interval $J=[0,+\infty)$. By close analogy, the exponential matrix is similar to the scalar exponential function and shared same properties with the exception of commutativity which breaks down in the case of exponential matrices.

Let A,B be $n\times n$ matrices. Then the following are true:

$$1.\, e^{A+B}=e^{A}\bullet e^{B} \qquad \textbf{(5.6)}$$

$$2.\, (e^{A})^{-1}=e^{-A} \quad \text{(inverse)} \qquad \textbf{(5.7)}$$

Warning

No mistake should be committed in assuming commutativity for e^A and e^B. In general, it is not guaranteed. In 1927, N H Abel established a relation for obtaining the Wronskian of solutions of differential equations using the traces of matrices of second order equations. The ideal was later generalized by J. Liouville and M. V. Ostrogradsky to nth order equations [7].

Theorem 5.1

If $x_1,x_2,...,x_n$ are solutions of equation (5.3),

CHAPTER 6

Stability Theory

Abstract: Stability including its characterizations is considered using quantitative and qualitative theories. Stability criteria are discussed using the Routh-Hurwitz criterion and fundamental matrices. Stability is investigated for nonlinear systems through linearization and Lyaponov's methods. Applications are made to single and multi-species population models.

Keywords: Fundamental matrices, Linearization, Lyaponov's stability, Multi-species populations, Nonlinear systems, Routh-Hurwitz criteria, Stability criteria, Single species populations.

INTRODUCTION

Stability over the century has constituted the backbone of study for modern dynamical systems. Scientists and engineers often take this concept into consideration whenever a mechanism is to be designed [1,2,3,5].

Intuitively, the concept could be said to have evolved from the study of the behavior of a system when perturbed (disturbed) from its equilibrium (resting) positions when the motion would not radically deviate from resting positions; for example, the vibration of a simple pendulum, when displaced from the equilibrium position in such a way that the amplitude of the oscillation is small.

The motion of a ball on a smooth parabolic surface is a motion exhibiting stability phenomenon [10].

Stability Theorems

Consider the initial value problem (IVP):

$$\dot{x}(t) = f(t, x(t)), x(t_0) = x_0 \tag{6.1}$$

Where, $f \in C(I \times \Omega, E^n)$, Ω is an open and connected subset of E^n. Assuming f has the property that the solution $x(t, t_0, x_0)$ exists and is unique, for example, if

$x(t,t_0,x_0)$ is a solution and f satisfies the hypotheses of the Picard-Linderlof existence and unique theorem (see Chapter 10) .

The core or central question of stability is as follows: is there a special solution $\phi(t,t_0,x_0)$ existing on the interval $[t_0,+\infty)$ such that a small perturbation would result in small deviations from system behavior? If such a solution exists, we say such a system is stable, otherwise it is unstable [7-10].

Stability treatment covers a vast majority of phenomena. For the purpose of a comprehensive study, our scope of discussion will encompass three considerations, namely stability through $(\varepsilon-\delta)$ argument. The interpretation of this will be given when the fundamental definition of stability is stated. Secondly, we will consider stability *via* fundamental matrices and finally, by the use of a scalar function called the Lyapunov stability function [4-7]. This method is fundamentally based on the energy concept.

Stability and its Characterization by $(\varepsilon-\delta)$ Argument

Definition 6.1

Let $\phi(t)=\phi(t,t_0,y_0)$ be a solution of (6.1) such that $\phi(t_0)=y_0$. The solution $\phi(t)$ is said to be stable (in Lyapunov sense, L. S.), if given $\in>0, t_0\in E^1$, there exists $\delta(t,\in,t_0)>0$, such that $\|x_0-y_0\|<\delta$ implies that: $\|x(t,t_0,x_0)-\phi(t)\|<\in, t\geq t_0+T(\eta)$, for every $t\geq t_0$; otherwise unstable $\phi(t)$ is asymptotically stable if it is stable and in addition:

$$\|x(t,t_0,x_0)-\phi(t)\|\to 0 \text{ as } t\to\infty \qquad \textbf{(6.2)}$$

$\phi(t)$ is uniformly stable, if the choice of $\delta(t,t_0,\in)$ is independent of t_0 or equivalently, for every $\eta>0$, there exists $T(\eta)$ such that $\|x_0-y_0\|<\delta$ implies that $\|x(t,t_0,x_0)-\phi(t)\|<\in, t\geq t_0+T(\eta)$.

Fig. (**6.1**) gives a diagrammatic explanation of the Stability concept which shows that a system is stable if the trajectory of the solution remains in the circle $\boldsymbol{S_1}$ with a small radius $\boldsymbol{\epsilon>0}$ for a given circle $\boldsymbol{S_2}$ with radius $\boldsymbol{\delta>0}$ containing the initial data of the system. The trajectory will never penetrate the boundary of $\boldsymbol{S_1}$ ($\|\boldsymbol{x(t)}\| < \boldsymbol{\epsilon}, \boldsymbol{t} \geq \boldsymbol{t_0}$) for the finite value of norm of the initial data ($\|\boldsymbol{x_0}\| < \boldsymbol{\delta}$).

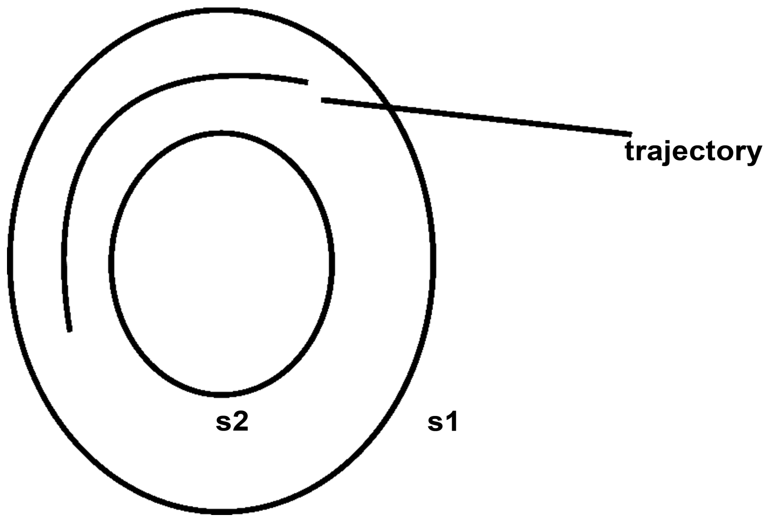


Fig. (6.1) Stability definition.

Geometric Interpretation

A solution $x(t,t_0,x_0)$ is called a trajectory of motion in Definition 6.1. This is interpreted as follows: The origin 0 is stable if given $\in > 0$ and:

$$S_1 = \{x \in E^n : \|x\| < \in\} \tag{6.3}$$

there exists a ball S_2 where:

$$S_2 = \{x \in E^n : \|x\| < \delta\},\ \delta < \in \text{ in } E^n, \tag{6.4}$$

such that the trajectory will never penetrate S_2.

For asymptotic stability, if conditions in Definition 6.1 hold, then $x(t,t_0,x_0) \to 0$ as $t \to \infty$. Stability can be geometrically interpreted in terms of trajectories remaining in the cylinder of infinite radius $\in$.

In Engineering, asymptotic stability is desirable since the solution eventually decays to zero.

CHAPTER 7

Stability of Perturbed Systems

Abstract: The stability property is investigated for perturbed linear autonomous and non-linear systems using linearization and Lyapunov's methods. Some examples are given on the stability of some nonlinear systems through eigenvalues of the linearized systems and coupled with the estimation of the norm of the error of approximation.

Keywords: Eigenvalues, Linearization, Lyapunov's method, Nonlinear systems, Non-linear systems, Perturbed linear autonomous, Stability properties.

INTRODUCTION

A system, from an energy perspective, could be said to have stable equilibrium if it is in the least energy state or when its energy is non-increasing. It could be said to be stable if its solution evolved in such a way that a small change in the equilibrium point will not lead to a radical change in the behavior of the system. If it does, it is said to be unstable [1-4].

This chapter, the stability of linear perturbed systems including periodic ones is considered. Moreover, Lyapunov stability technique which is also called the "Lyapunov second method" is introduced and it is a generalization of the energy concept. The emphasis is on how to qualitatively investigate the stability properties of equilibrium of a system. We will consider a linearizing procedure and establish the stability properties of nonlinear systems from the linearized ones.

Furthermore, we will construct Lyapunov functions and use them to obtain some stability criteria for the equilibrium points of some systems. Different types of stability will be studied using the Lyapunov second method [5-7, 12].

Stability of Linear Perturbed Systems

In this section, without loss of generality, the stability of perturbed linear autonomous systems will be closely studied, bearing in mind that the idea can be extended to non-autonomous systems.

Benjamin Oyediran Oyelami

Stability of homogeneous differential equations is a prototype of stability theory [1] and sometimes such a differential equation may contain a perturbation function. We intend to study stability and its characterization for linear perturbed systems. Jack [11] provided a bundle of theorems for this purpose, especially for non-autonomous systems.

Consider the linear autonomous system:

$$\dot{x} = Ax + f(t,x) \qquad \textbf{(7.1)}$$

where A is a square matrix and $f(t,x)$ is the perturbation function, which is continuous, such that:

$$\|f(t,x)\| \le m\|x\|,$$

for some positive constant m. We assert that the system is asymptotically stable [1, 12].

Using the above condition, it is not difficult to show that:

$$\|x\| \le c\exp-(\gamma - mc)(t-t_0)\|x_0\|, \qquad \textbf{(7.2)}$$

therefore, given $\eta > 0$, we can find $\sigma > 0$ such that $\|x_0\| \le \sigma \Rightarrow \|x(t)\| < \eta$

i.e. $\|x\| < c\exp-(\gamma - mc)(t-t_0)\sigma < \eta$,

for $t > t_0 + T(\eta)$, where $T(\eta) = \dfrac{1}{\gamma - mc}\ln\left(\dfrac{\eta}{c\sigma}\right)$ for $\gamma - mc > 0$ and $\in > c\delta$. Uniform stability is implied from the definition. We note that $\|x(t)\| \to 0$ as $t \to \infty$ thus asymptotic stability follows and hence the proof.

Remark 7.1

The above result holds for non-autonomous systems (Jack [11], pp.87).

Theorem 7.1

Let $\dot{x}(t) = A(t)x(t) + f(t, x(t))$ be a linear perturbed system where $A(t), f(t, x(t))$ satisfy conditions that allow solutions to exist and be uniquely determined in a given interval. Suppose also that $f(t, x(t))$ degenerates such that:

$$\| f(t, x(t))\| \leq \eta(t)\|x(t)\| \tag{7.3}$$

and $\int_{t_0}^{t} \eta(s)ds < m$ for some constant $m > 0$ and $t \geq t_0$.

Then the linear perturbed system is uniformly stable.

Proof

The proof is by a variation of constant parameter and estimation as follows:

$$\|x(t)\| \leq e^{At}\|x_0\| + c\int_{t_0}^{t} \eta(s)\|x(s)\|ds$$

$$\|x(t)e^{\alpha t}\| \leq c\|e^{\alpha t}x_0\| + c\int_{t_0}^{t} \eta(s)\|x(s)e^{\alpha t}\|ds$$

Let $V(t) = x(t)e^{\alpha t}$,thus by Gronwall's inequality [1, 9, 10] we have

$$\|v(t)\| \leq c\|v_0\| + \int_{t_0}^{t} c\eta(s)\|v(s)\|ds$$

$$\leq c\|v_0\|\exp(\int_{t_0}^{t} c\eta(s))ds$$

That is:

$$\|x(t)\| \leq c\|x_0\|\exp-(\alpha - cm)(t - t_0) \tag{7.4}$$

CHAPTER 8

Stability Property of Some Neural Firing and Avian Influenza Infection Models

Abstract: The stability property of equilibrium points of some neural network models is investigated. We have introduced different types of Lyapunov functions to carry out the investigation. The models considered are: Grossberg, Hopfield, Fitz-Nagomo and Fitzhugh models, respectively. The equilibrium points and stability conditions are obtained for the Avian influenza infection. The conditions for bio economic equilibrium points for the fish model were also obtained.

Keywords: Bio-economic equilibrium point, Equilibrium points, Fish model, Fitz-Nagomo models, Fitzhugh model, Grossberg, Hopfield, Lyapunov functions, Neural network models, Stability property.

INTRODUCTION

Neural network is primarily concerned with modelling the activity of the brain, its behavioral processes, and the application of these models to computers and related technologies [1,4,6,7,10]. The Areas where neural network find useful applications are neuroscience, artificial intelligence, vision and image processing, speech and language understanding, pattern recognition, parallel distributed processing, and so on (Gene *et al*. [4]; Hopfield [8]).

An Artificial neural network (ANN) is an information or signal processing entity containing elements, called artificial neurons, or sometimes referred to as nodes. The neurons are interconnected by direct links called connectives, which perform parallel distributed processing (PDP) in order to solve the desired biological or computational task [5,7,9, 20].

Artificial neural networks can be used effectively to solve many scientific and engineering problems that are formulated mainly as variational or optimization problems derived from the learning equation with or without a teacher [4].

This is a richly connected network of simple computational elements modelled as biological processes. Their origin can be traced back to the late nineteenth and early twentieth centuries when psychologists tried to identify the neural basis of intelligence [4].

Benjamin Oyediran Oyelami

MacCulloch initiated research on the central nervous system in 1943 and a neural network model which was published with Walter Pitts. In 1949, Donald Hebb proposed a model for learning in neural network and Dean Endermonds in 1951 proposed a similar model but on electromechanical learning machine which incorporated the ideas in a motor-driven memory with forty control knobs Cichocki and Unbehaven [1].

The discipline Artificial Intelligence was introduced in 1956 at the Dartmouth Conference where Anderson James presented a paper based upon his development on brain state in a box (BSB) (Nicholas [9]).

Grossberg developed a mathematical model in 1982, which encompassed a variety of neural network models as well as population biology and macro molecular evolution.

In this chapter, we consider four models of neural network types, namely; Grossberg, Hopfield, Fitzhugh-Nagumo and Fitzhugh models. Equilibrium points of the models are determined and consequently stability properties of systems investigated using a series of Lyapunov functions which we introduced. The fundamental problem we encountered was how to obtain suitable Lyapunov functions for the models. The reason why we opted for the stability of the models is that it offers a precondition for optimization of the models [11-15].

Preliminary Definitions

Neuron

This is the basic unit of the central nervous system (CNS) that sends signals through the neural network.

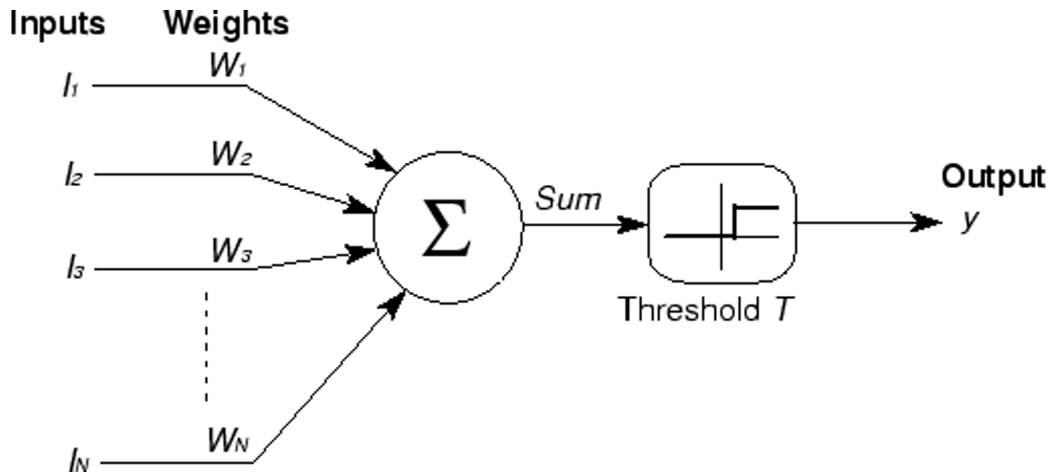
Synapse

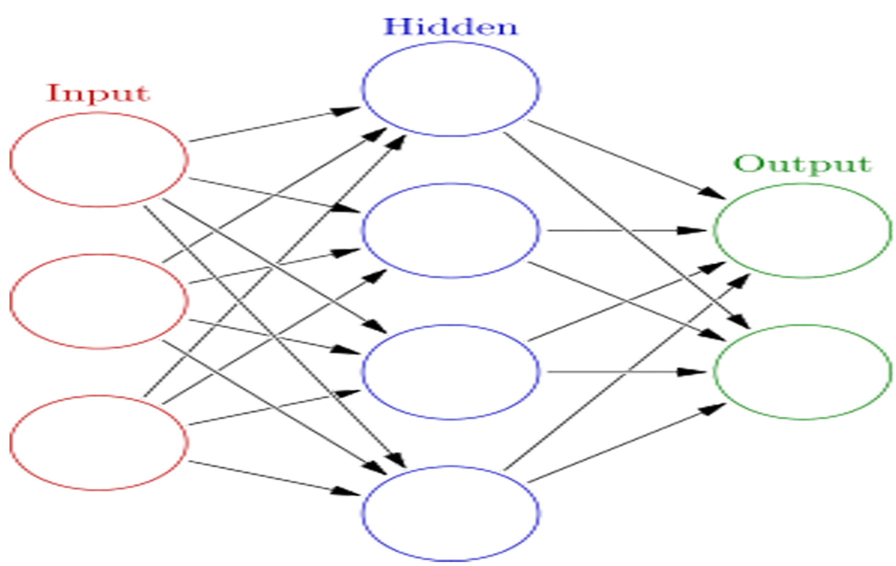
This is the connection junction between two neurons.

Neural networks are a class of models inspired by the neural circuitry in human and animal brains. They occupy a spectrum, ranging from artificial neural networks (ANNs) to biological neural networks (BNNs).

MACCULLOCH -PITT Model

The first real model of a nerve cell that could be simulated on a computer was developed by McCulloch and Pitts (Fig. **8.1**).





(Fig. 8.1) contd.....

CHAPTER 9

Numerical Solutions to Ordinary Differential Equations and Applications Using Maple

Abstract: Many complex nonlinear problems in science, economics, and engineering require computers to solve and simulate mathematical models describing them. Hence, it becomes extremely necessary to apply numerical methods to solve such problems. In this chapter, numerical methods for solving initial value problems are considered. Taylor series, Euler, Modified Euler's, Runge-Kutta, Adams-Bash forth-Moulton and Milne numerical methods are considered together with some Maple examples given. Numerical simulations are designed and implemented using Maple software for HIV/AIDS, Fitzhugh, and Fitzhugh-Nagumo, sickle cell anemia, zooplankton-fish, Gompertz tumor and neural firing models. The Explore facility in Maple 2022 is utilized to design sliders for investigating the behaviors of solutions of some ordinary differential equations subject to parameter change.

Keywords: Codes, Convergence, Fitzhugh, Fish models, Fitzhugh-Nagumo, Maple solution, Numerical, HIV/AIDS, Solutions, Stability, Sickle cell anemia, Tumor, Zooplankton.

INTRODUCTION

In recent times, there are several emerging complex nonlinear ordinary differential equations (ODEs) in science and technology. The needs for computer-based solution to those ODEs problems are becoming increasingly important. Applications of numerical methods and development of numerical simulations are everywhere present in most research works in engineering, economics and life science these days [2,3,4,14]. The reason d'état is not far from the fact that the analytic or exact solutions to most non-linear ordinary differential equations cannot be easily being found even with the applications of symbolic programing.

It is interesting to note that, sometimes, the symbolic programs may take several hours, or even days to generate symbolic solutions to a problem. The computer printout of the solution may run into several pages of paper and the result may appear to be meaningless at a glance [12]. Hence the use of numerical methods in solving applied problems in science, economics and engineering is becoming popular in the recent times [3, 4, 5,8,12, 13].

Benjamin Oyediran Oyelami

We note that, in the literature, much research has accumulated on numerical methods for solving ordinary differential equations. There are many physical systems with the governing equations, initial and boundary conditions, whose solution cannot be obtained with even some mathematical software in the market; yet the solution to the ODEs exist in some given interval. In this situation, numerical methods are often used to find the approximate solutions to the pig-headed ODE problems and numerical simulation now constitute the core of most researches on the behavior of the solution to the problems subject to parameter changes [2, 8, 12, 13].

Furthermore, the knowledge of numerical solution is very important to solve the differential equations [8]. We need efficient numerical methods in order to form algorithms for solving the problems. The algorithms must have desirable computation properties before being coded into computer programs. The programs are then implemented using higher level programming language to find the numerical solution to the differential equations. Maple software platform has given us the opportunity to gain understanding of the behavior of systems and discover laws underpinning them. It provides us with platforms to teach students how to find numerical solutions to ODEs and to develop numerical simulation to models [2, 5, 6, 9].

The numerical methods that we will consider are the Taylor series, Euler, Modified Euler's, Runge- Kutta, Adams-Bashforth, Adams-Bashforth-Moulton and Milne methods. It is worthy to note that each of these methods has some kind of complexities associated with it. These include: computer run time (time taken to run the program), computer memory (space occupied by the data generated from the numerical method); how fast the approximated solution tends to the analytic solution (convergence issue). The issue of consistency and stability of the methods are also paramount when considering numerical solutions [5,7,8,14]. Each of these computational properties will be discussed in our subsequent study on numerical solutions to ODEs.

Furthermore, in implementing numerical methods, a price must be paid which is associated with complexities in the numerical algorithms employed together with structuring programing language unitized to implement the numerical methods on the computer.

Finally, we will like to emphasize that numeric analysts would be interested in studying the numerical algorithms and the computational analytic concepts mentioned above, whereas other Scientists and Engineers will only be concerned with applying numerical methods to solve problems for as long as the solution generated from the numerical methods are accurate and satisfied the required computational properties.

Numerical Solution to Initial Value Problems

Let us consider an initial value problem (IVP):

$$\left.\begin{aligned} \frac{dy(t)}{dx} &= f(t,x(t)) \\ y(t_0) &= y_0 \end{aligned}\right\} \tag{9.1}$$

where $f(t,x(t))$ is a continuous function in the closed interval $[a,b]$ and differentiable in the open interval (a,b).Then by fundamental theorem of calculus the solution to the equation (9.1) is:

$$y(x) = y_0 + \int_{t_0}^{t} f(s,x(s))ds, t_0 = a, t \in [a,b] \tag{9.2}$$

Suppose the interval $[a,b]$ is partition into sub-intervals $[x_0,x_1),[x_1,x_2),[x_2,x_1),...[x_{k-1},x_k)$ such that: $0 \le x_0 < x_1 < x_2 < ... < x_k, x_{k+1} = x_k + h$, where h is constant.

We can find the Taylor series solution to the equation (9.1) as follows:

$$y(x) = y_0 + (x-x_0)y_0^{/} + (x-x_0)^2 y_0^{//} + (x-x_0)^3 y_0^{//} + ... + R_n \tag{9.3}$$

where $y_0 = y(x_0), y^{/}{}_0 = y^{/}(x_0),....,y_0^{(n)} = y^{(n)}(x_0)$.Therefore, the approximate solution to the differential equations can be generally be written as $y(x) = y(x_n) + \in_n = y_n + \in_n$, where y_n is the approximate solution to $y(x)$ (exact solution) and $\in_n$ is the error of approximation.

Later on, we will discover that numerical methods differ by the value of $y(x)$ and the corresponding error value $\in_n, n = 1,2,...$

APPENDIX A

Brief Highlights about Maple Software

Abstract: Many real-life problems can be solved through modeling and simulation and Maple 2022 is the world-leading software used by mathematicians, physicists, economists, engineers, and educators for the problem solving task. The power of Maple and the MapleSim software are exploited in this section. We present the starting process with the software and demonstrate the application of the software *via* some selected problems.

Keywords: 2D, 3D plots, Animation, C Codes, Hybrid computations, Maple, MapleSim, MapleSim, Monte Carlo Simulation, Numerical, Symbolic, Simulation.

A.1. POWER OF MAPLE

Maple has the most powerful Math engine, and smart document interface, along with Maple add-in and grid computing facilities for symbolic, numerical, and hybrid computation, sophisticated 2D, 3D plotting and animation, and document and word processing tools.

Furthermore, Maple T.A (Test and Assessment) has an E-learning solution. Maple T A is an easy-to-use web-based system for creating tests and assignments and automatically assessing students' responses and performance. It has Maple T.A. placement Test suite to deliver tests online which reduces the cost of administration and marking examinations using paper type.

A.1.1 Maple Net

- Maple Net: A facility that allows easy sharing of Maple documents, calculator and technical application. There is also MapleSim 2022 for the simulation of engineering and real-life processes.

A.1.2 Calculus Kits

- Calculus Kits are for students and teachers to interact with each other while solving mathematical problems.

A.1.3 Users

- Maple is software that can be used by mathematicians, physicists, engineers, chemists, social scientists and educators.

Benjamin Oyediran Oyelami

1.4 Maple Portal

- Maple makes use of what is called Portals. The Maple Portal is designed as a starting place for any Maple user. There are 3 types of portals in Maple that are related to our study in this textbook and these are:
- Portal for Engineers: which contains tools used by Engineers in solving mathematical problems. Engineering packages contain a dynamical system toolbox, scientific constants, scientific error analysis, tolerance and units.
- Portal for Students: Student packages are available for the following topics: Pre-calculus, calculus, vector calculus, differential equations, linear Algebra, and multivariate calculus.
- Portal for Math Educators: This portal contains information and tools for education, assessment, Maple Test and assessment of students. This portal contains student packages that allow instructors to deliver the course contents effectively; give students insight into understanding basic mathematical concepts and enhance their problem-solving practical skills. There is also a survival kit to enhance students' mathematical mastery of topics in the portal for students

A.1.5 Help Resources and Maple Tour

Maple also has Help Resources and Maple Tour to give tutorials on how to use the resources in Maple and Help system to help the users out of perceived problems and many examples on how to use maple resources. There is also a Quick reference card. This card gives vital information on how to make use of resources like the type of modes for creating documents in Maple. It also gives information on Toggle Math/Text entry mode, how to evaluate math expressions and display results in line; common operations available in the Maple in both document and worksheet Modes; 2-D math editing operations, keyboard shortcuts, and operations plotting and animation.

A.1.6 User Manuals and Web links

User Manuals: This manual gives comprehensive information about Maple, tutorials, and examples on Maple. The manual contains how to get started with maple toolboxes, the user manual and the programming guild.

Web links: This is the hyperlink to Maple soft Company, which is the developer and marketer of Maple software. The links provide information and registration of the company, and show how to register and take part on webinars, an online seminar series. It also provides information on how to get online resources on Maple.

A.2 Getting Started

A.2.1 Maple Tutorial

Maple tutorial helps to get started with the software, learn about the tools available in Maple, and lead you through a series of problems. It guides you on how to enter simple expression, functions, matrices, complex numbers, and evaluate expression and plotting functions.

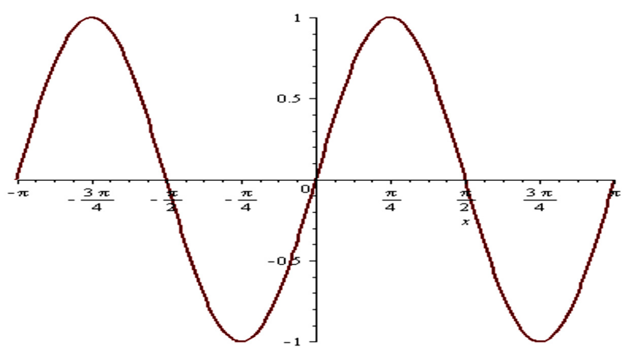
Maple has so many interesting modelling and simulation facilities as we are not going to make a discussion on them but we will demonstrate their applications in Maple Examples.

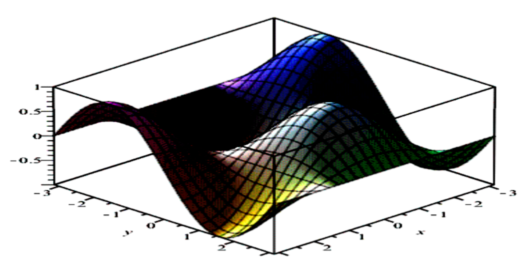
Examples on Graphs and Animations

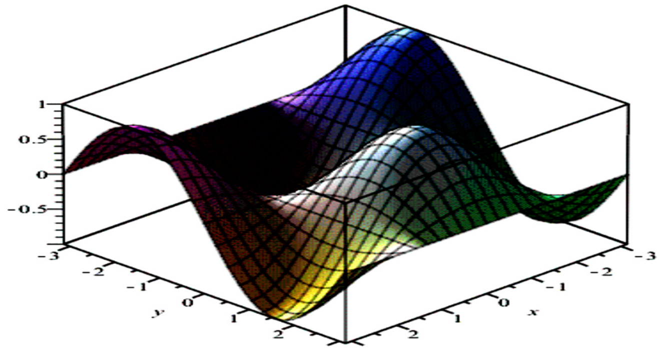
Example A1

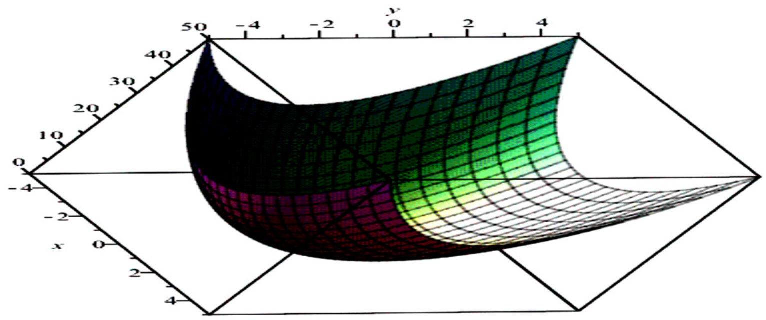
To plot the graph of sine function in the worksheet mode, type in the command:

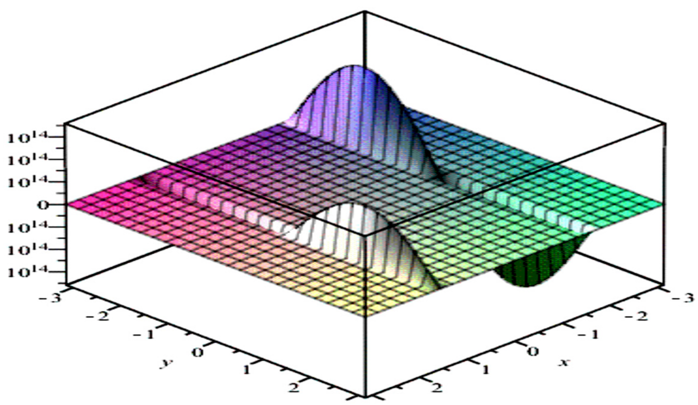
> Plot (sin (2*x),x =-Pi..Pi, thickness=2);

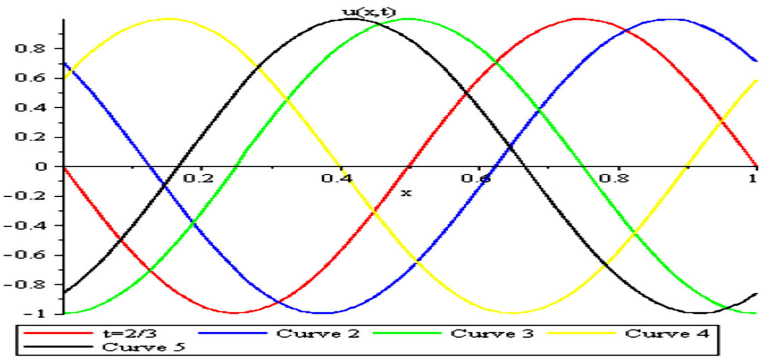


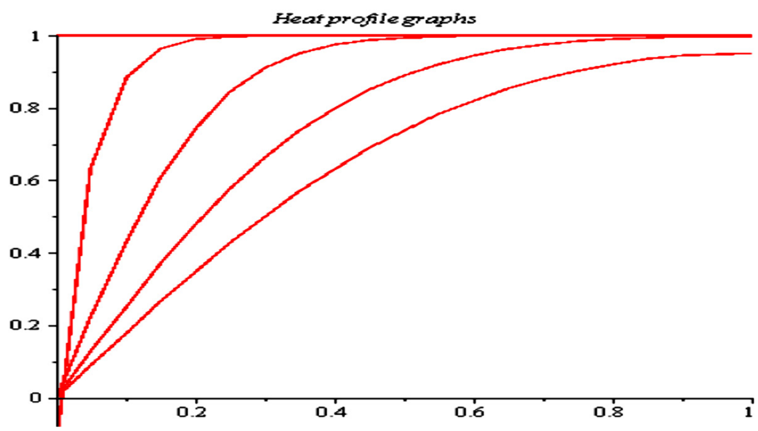



1
0.5
0
−0.5
−3
−2
−1
0
1
2
y
−3
−2
−1
0
1
2
x

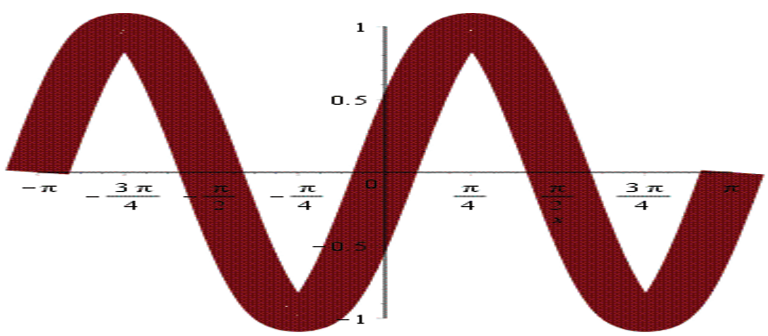

y
−4
−2
0
2
4
50
40
30
20
10
0
−4
−2
0
2
4
x


10^14
10^14
10^14
0
10^14
10^14
10^14
−3
−2
−1
0
1
2
y
−3
−2
−1
0
1
2
x





$plot(\sin(2 \cdot x), x = -\text{Pi}..\text{Pi}, thickness = 30);$

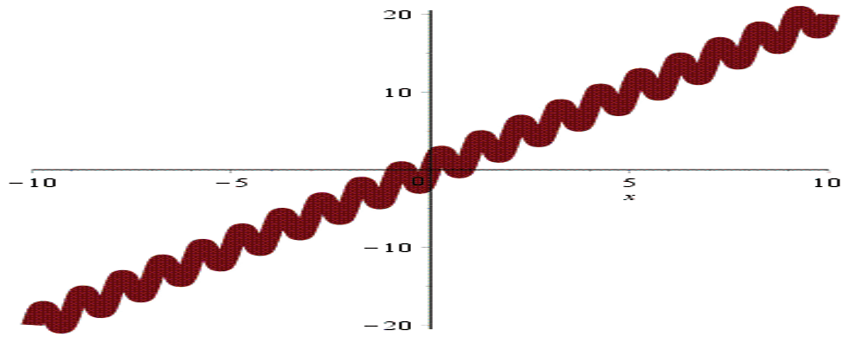


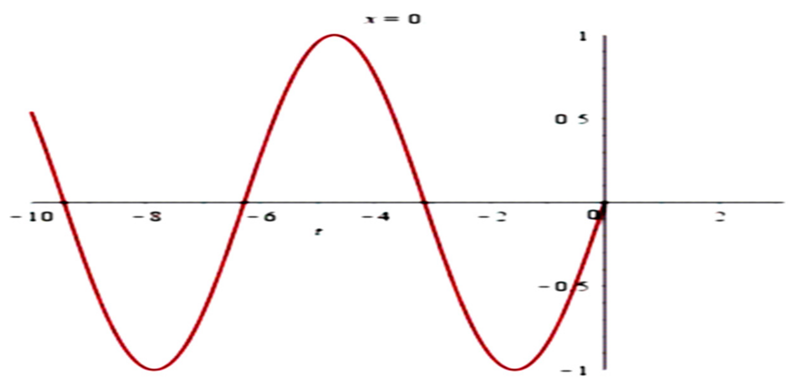
$> f := 2 \cdot \sin(2 \cdot \text{Pi} \cdot x) + 2 \cdot x;$

$f := 2 \sin(2 \pi x) + 2 x$

$> plot(f, x = -10..10, thickness = 10);$

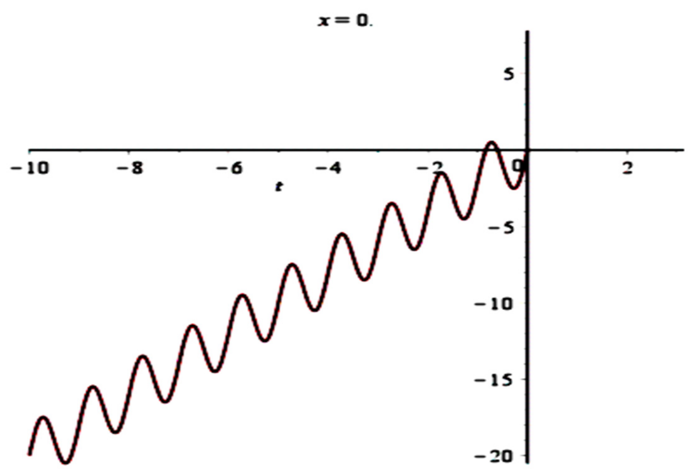
$>animate(plot, [\sin(t), t=-10..x], x=0..\mathrm{Pi});$





\>

$>animate(plot, [2\cdot\sin(2\cdot \mathrm{Pi}\cdot t) + 2\cdot \mathrm{t}, \mathrm{t}=-10..\mathrm{x}], x=0..\mathrm{Pi});$

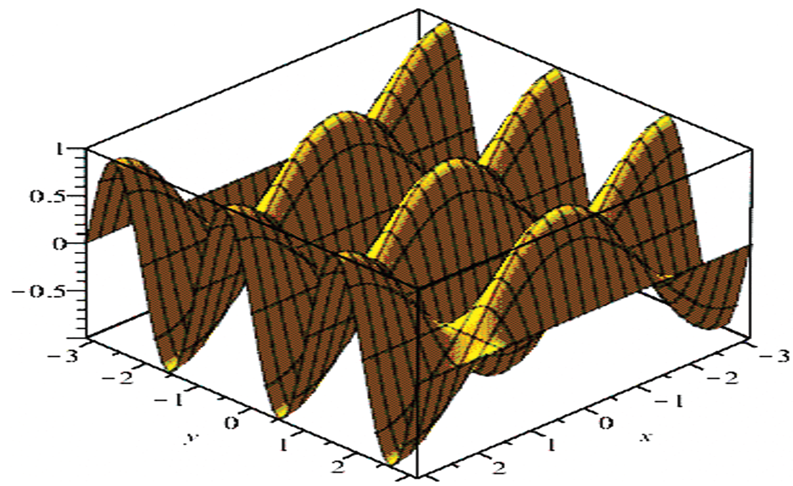


> *restart*

> *with(plots) :*

> $animate3d(\cos(t \cdot x) \cdot \sin(3 \cdot t \cdot y), x = -\mathrm{Pi}..\mathrm{Pi}, y = -\mathrm{Pi}..\mathrm{Pi}, t = 1..2);$

We can extend the plot to 3D using document mode: type in the following two dimension function w=w(x, y) and highlight the equation, right click the 3D plot, we have:



>

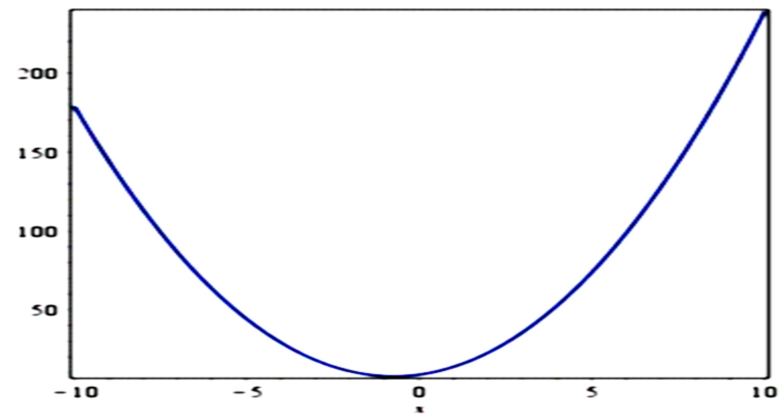
> *restart*

> *with(plots) :*

> *with(DEtools) :*

In the document mode, type the equation and highlight it and right click to select the 2D plot, then we have the plot:

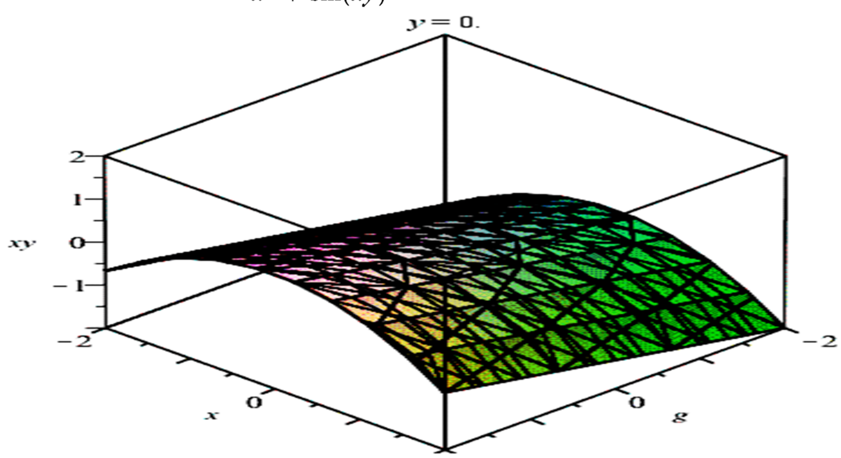
$y = 2x^2 + 3x + 9 \;\rightarrow$



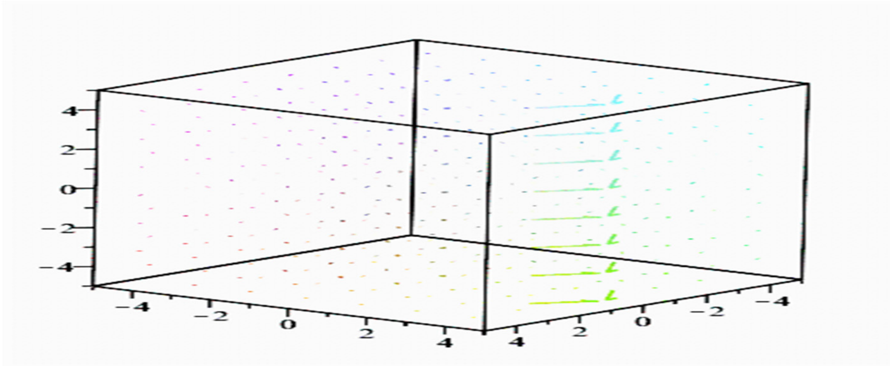
>

>We can also replicate the above plot using worksheet mode by typing in the equation and right- click and select 3D plot. You can also use the plot builder to have a variety of 3D-plots and even animate the plots too.

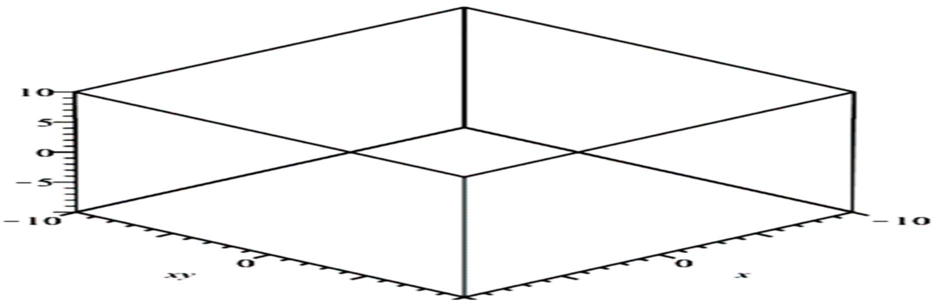
$$w = x^2 + y^2 + 3xy + \frac{2xy}{x^2 + \sin(xy)} \rightarrow$$



>**w=3x^2** $+ 3xy + y^2 + \frac{2xy}{x^2 + \sin(xy)};$



>

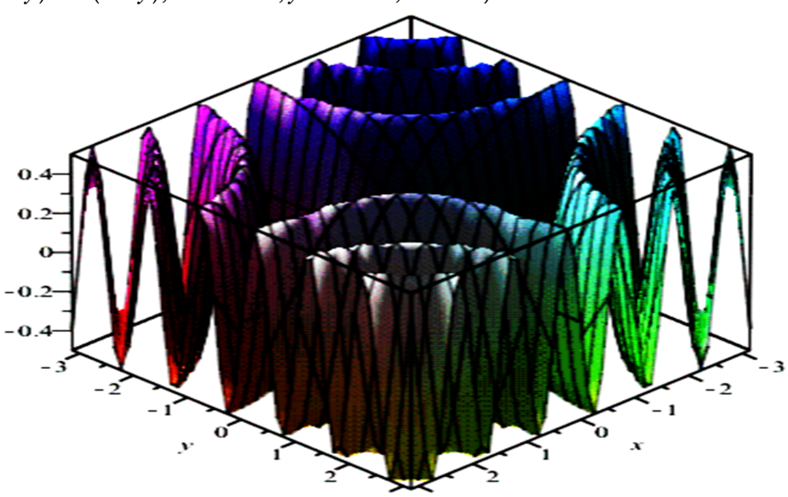


>

Using worksheet mode, you can animate a plot using the command with (plots) together with animate3d. For example, type in:

>*with*(*plots*) :

animate3d(sin(*t*·*x*·*y*) ·cos(*t*·*x*·*y*), *x* =−Pi ..Pi, *y* =−Pi ..Pi, *t* = 0 ..1)

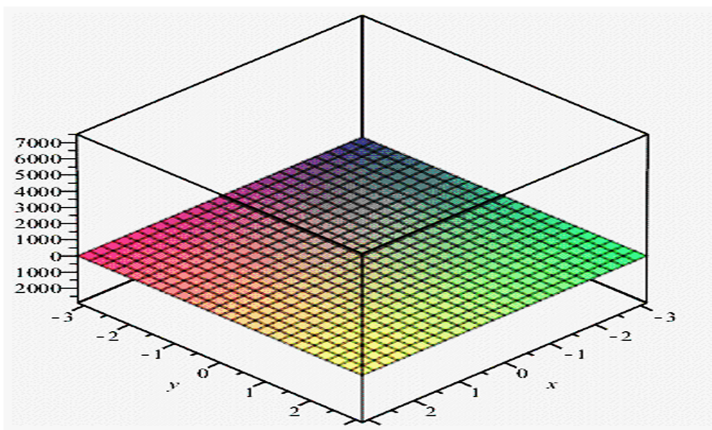


Highlight the 3D-plot and select the type of animation, whether short-time animation or continuous one. In Maple software, the memory can be cleared using 'restart'.

restart

with(*plots*) :

animate3d($\exp(t \cdot x \cdot y) \cdot \sin(t \cdot x \cdot y) \cdot \cos(t \cdot x \cdot y), x = -\mathrm{Pi}\,..\mathrm{Pi}, y = -\mathrm{Pi}\,..\mathrm{Pi}, t = 0\,..1$)



Maple contains several facilities for computation using Linear Algebra. Type in with (LineraAlgebra) with 'semicolon' to display the linear algebra facilities in the maple software. We can suppress this by using colon as usual.

> *with*(*LinearAlgebra*);

In the worksheet mode, a vector and a matrix can be typed in as follows:

x=Vector ([1, 0,-2, 3);

$$x := \begin{bmatrix} 1 \\ 0 \\ -2 \\ 3 \end{bmatrix}$$

$[\&x, Add, Adjoint, BackwardSubstitute, BandMatrix,$
$Basis, BezoutMatrix, BidiagonalForm, BilinearForm,$
$CARE, CharacteristicMatrix,$
$CharacteristicPolynomial, Column,$
$ColumnDimension, ColumnOperation, ColumnSpace,$
$CompanionMatrix, CompressedSparseForm,$
$ConditionNumber, ConstantMatrix, ConstantVector,$
$Copy, CreatePermutation, CrossProduct, DARE,$
$DeleteColumn, DeleteRow, Determinant, Diagonal,$
$DiagonalMatrix, Dimension, Dimensions,$
$DotProduct, EigenConditionNumbers, Eigenvalues,$
$Eigenvectors, Equal, ForwardSubstitute,$
$FrobeniusForm, FromCompressedSparseForm,$
$FromSplitForm, GaussianElimination,$
$GenerateEquations, GenerateMatrix, Generic,$
$GetResultDataType, GetResultShape,$
$GivensRotationMatrix, GramSchmidt, HankelMatrix,$
$HermiteForm, HermitianTranspose, HessenbergForm,$
$HilbertMatrix, HouseholderMatrix, IdentityMatrix,$
$IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar,$
$IsUnitary, JordanBlockMatrix, JordanForm,$
$KroneckerProduct, LA\_Main, LUDecomposition,$
$LeastSquares, LinearSolve, LyapunovSolve, Map,$
$Map2, MatrixAdd, MatrixExponential,$
$MatrixFunction, MatrixInverse,$
$MatrixMatrixMultiply, MatrixNorm, MatrixPower,$
$MatrixScalarMultiply, MatrixVectorMultiply,$
$MinimalPolynomial, Minor, Modular, Multiply,$
$NoUserValue, Norm, Normalize, NullSpace,$
$OuterProductMatrix, Permanent, Pivot, PopovForm,$
$ProjectionMatrix, QRDecomposition, RandomMatrix,$
$RandomVector, Rank, RationalCanonicalForm,$
$ReducedRowEchelonForm, Row, RowDimension,$
$RowOperation, RowSpace, ScalarMatrix,$
$ScalarMultiply, ScalarVector, SchurForm,$
$SingularValues, SmithForm, SplitForm,$
$StronglyConnectedBlocks, SubMatrix, SubVector,$
$SumBasis, SylvesterMatrix, SylvesterSolve,$
$ToeplitzMatrix, Trace, Transpose, TridiagonalForm,$
$UnitVector, VandermondeMatrix, VectorAdd,$
$VectorAngle, VectorMatrixMultiply, VectorNorm,$
$VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]$

>A:=Matrix([[1, 2, 0, 3], [0, 0, -1, 4], [0, 0, -3, 2], [2, 1, 0, 2]]);

$$A := \begin{bmatrix} 1 & 2 & 0 & 3 \\ 0 & 0 & -1 & 4 \\ 0 & 0 & -3 & 2 \\ 2 & 1 & 0 & 2 \end{bmatrix}$$

The element on the first row fourth colon can be displayed by typing in:

> $A[1, 4]$;

3

> $A[3, 3]$;

-3

A matrix A can be multiplied by itself using the code:

> $A.A$;

$$\begin{bmatrix} 7 & 5 & -2 & 17 \\ 8 & 4 & 3 & 6 \\ 4 & 2 & 9 & -2 \\ 6 & 6 & -1 & 14 \end{bmatrix}$$

Matrix A can be post multiplied using the vector x as follows:

> ***A.x***

$$\begin{bmatrix} 10 \\ 14 \\ 12 \\ 8 \end{bmatrix}$$

B: =Matrix ([**[1,2],[5,7],[3,5], [0,3]]);**

$$B := \begin{bmatrix} 1 & 2 \\ 5 & 7 \\ 3 & 5 \\ 0 & 3 \end{bmatrix}$$

> ***A.B;***

$$\begin{bmatrix} 11 & 25 \\ -3 & 7 \\ -9 & -9 \\ 7 & 17 \end{bmatrix}$$

> ***B.A;***

Error, (in LinearAlgebra:-Multiply) first matrix column dimension (2) <> second matrix row dimension (4)

> $h := i \rightarrow i^2;$

$$h := i \rightarrow i^2$$

> **y := Vector(8, h);**

$$y := \begin{bmatrix} 1 \\ 4 \\ 9 \\ 16 \\ 25 \\ 36 \\ 49 \\ 64 \end{bmatrix}$$

> *# generate the HilbertMatrix*

> **H:= Matrix(5,5, (i,j) -> 1/(i+j-1));**

$$H := \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

> $C := \langle\langle 1, 2, 3\rangle | \langle 0, 0, 1\rangle | \langle 0, 0, 1\rangle\rangle;$

$$C := \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 1 & 1 \end{bmatrix}$$

> *# Find the basis for A and C;*

> $NullSpace(A);$

$\{ \}$

> $NullSpace(C);$

$$\left\{ \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \right\}$$

> *# A has no basis;*

> $d := \langle 1, 2, 0, -1\rangle;$

$$d := \begin{bmatrix} 1 \\ 2 \\ 0 \\ -1 \end{bmatrix}$$

>z:=LinearSolve(A,d);

Warning, inserted missing semicolon at end of statement

$$z := \begin{bmatrix} -\frac{6}{5} \\ \frac{1}{5} \\ \frac{2}{5} \\ \frac{3}{5} \end{bmatrix}$$

> E:=IdentityMatrix(4);

Warning, inserted missing semicolon at end of statement

$$E := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

> $Determinant(A);$

-30

> $Rank(A);$

4

> $l2ip := (f, g) \rightarrow int(f(x) \cdot g(x), x = 0..1);$

$$l2ip := (f, g) \rightarrow \int_0^1 f(x)\, g(x)\, dx$$

>N:=f ->sqrt(%(f,f));

$$N := f \rightarrow \sqrt{\%(f, f)}$$

> $unassign('x');$

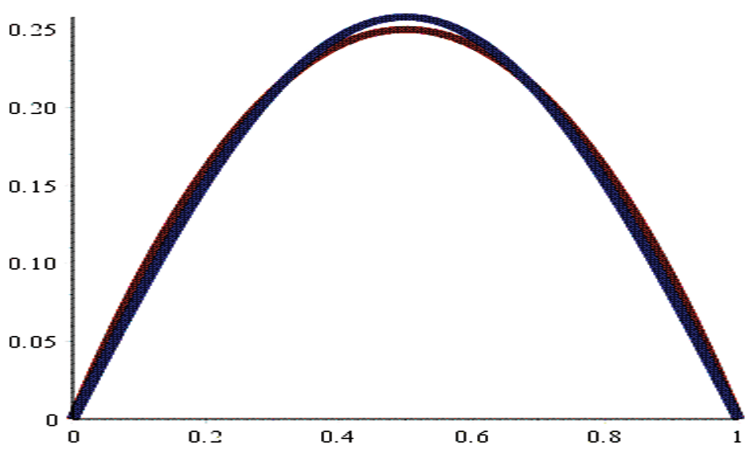
> $f := x \rightarrow x \cdot (1 - x);$

$$f := x \rightarrow x\,(1 - x)$$

> $g := x \rightarrow \frac{8}{\pi^3} \cdot \sin(\mathrm{Pi} \cdot x);$

$$g := x \rightarrow \frac{8 \sin(\pi x)}{\pi^3}$$

> plot({f(x),g(x)},x = 0 .. 1,thickness = 6);



>>> $edn := diff(y(x), x) = -\frac{x}{y};$

$edn := \frac{d}{dx}\, y(x) = -\frac{x}{y}$

> $with(plots)$

[*animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot, display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors, setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d, tubeplot*]

> $edn := diff(y(x), x) = -\frac{x}{y};$

$edn := \frac{d}{dx}\, y(x) = -\frac{x}{y}$

> $c := gradplot\left(-\frac{x}{y}, \mathrm{x} = -10\,..10, \mathrm{y} = -10\,..10\right)$

$c := PLOT(\ldots)$

>

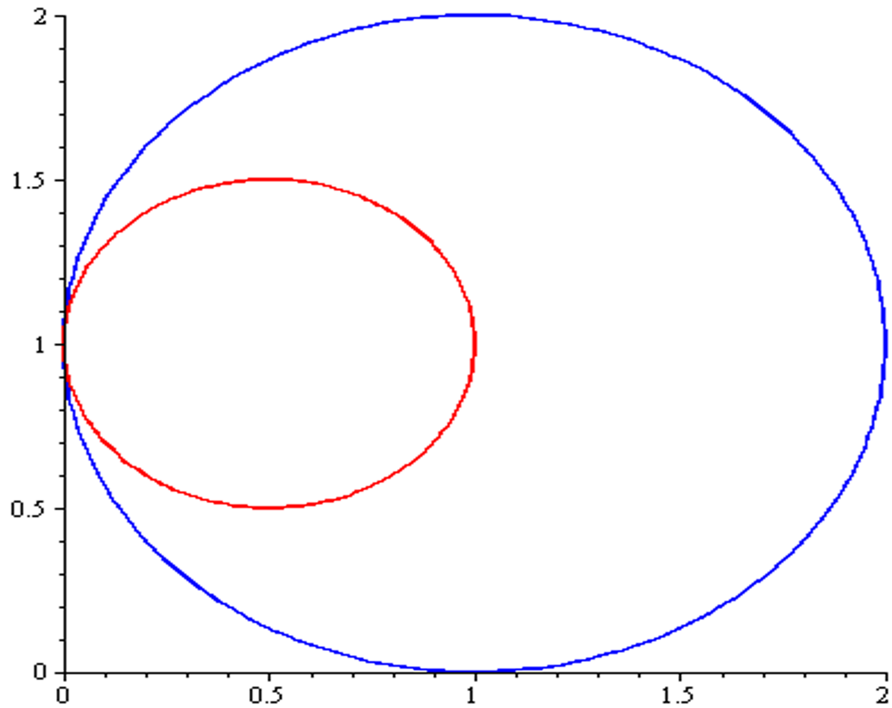
> with(plottools):

> with(plots):

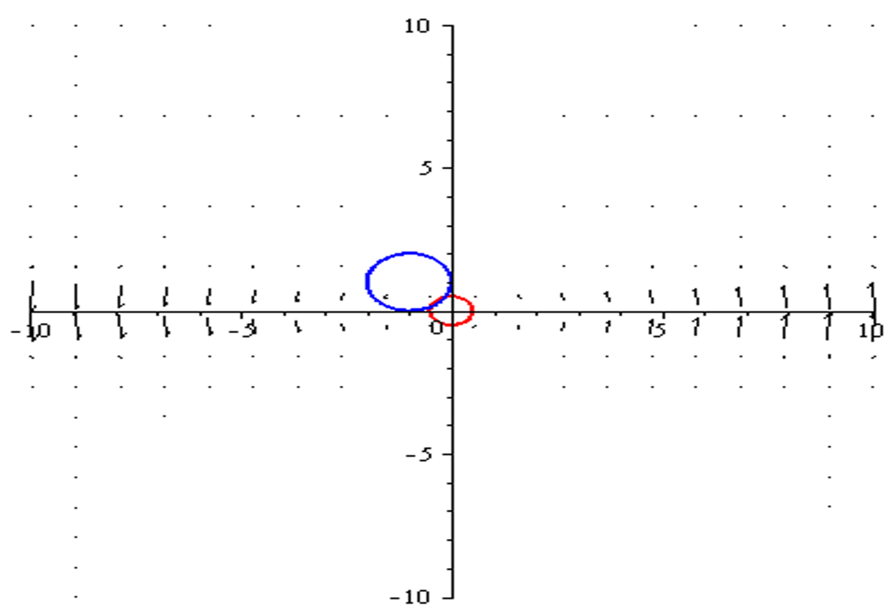
> c1:= circle([1,1], 1,color=blue):

> c2:=circle([1/2,1], 1/2,color=red):

> display([c1,c2);



> *display*(*c*, *c2*, *c1*); :



> $with(plots)$:

> $c := sphere([1, 1, 1], 3.3)$:

> $display(c, scaling = constrained, axes = boxed)$

> $d := sphere([1, 5, 1], 2)$:

> $display(c, d, scaling = constrained, style = patchnogrid)$

> $esphere([1, 5, 1], 2)$:

> $display(c, e, scaling = constrained, style = patchnogrid)$

>

> ;=

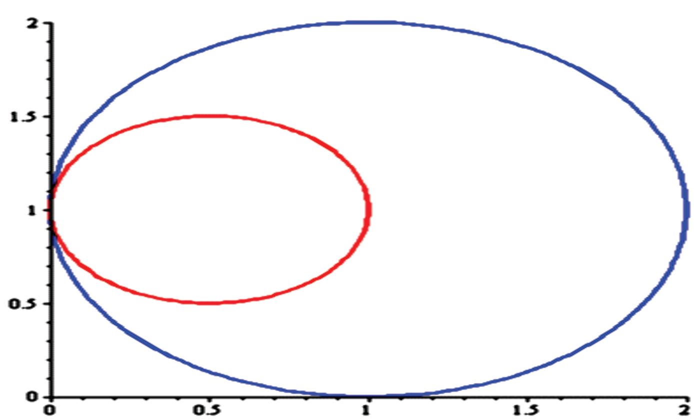
> with(plottools):

> with(plots):

> c1 := ellipse([1,1], 1, color=blue):

> c2 := circle([1/2,1], 1/2, color=red):

> display(c1,c2);



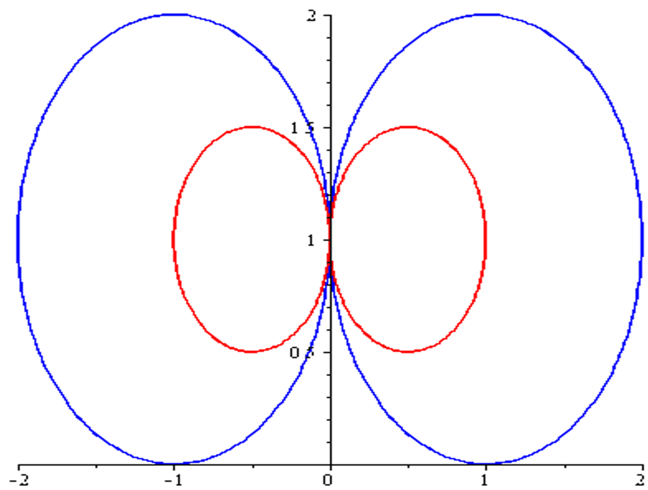
> c3:= ellipse([-1,1], 1,color=blue):

> c4:= circle([-1/2,1], 1/2,color=red):

> with(plottools):

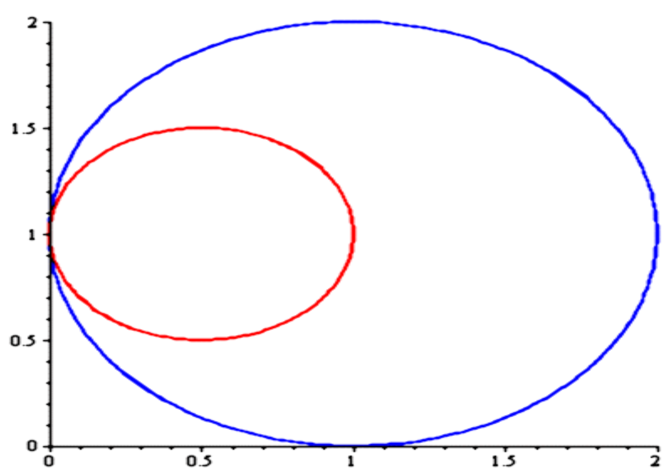
> with(plots):

> display(c1,c2,c3,c4);



> c2:=circle([1/2,1], 1/2,color=red):

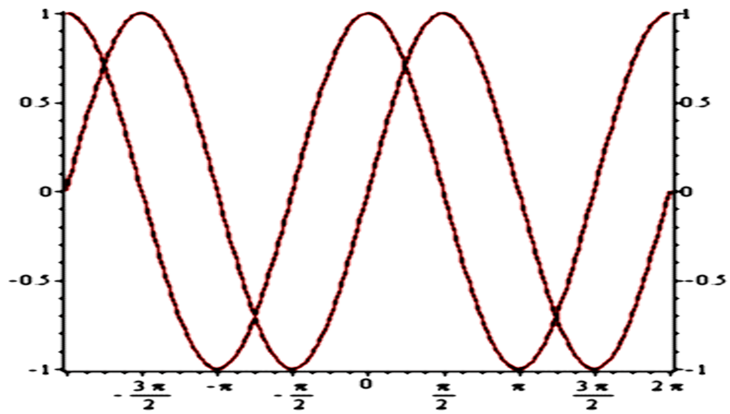
> display(c1,c2);



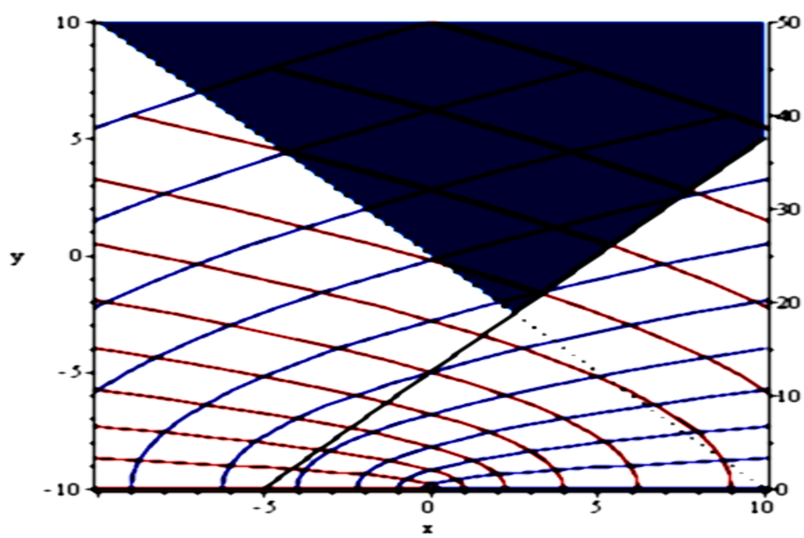
> *restart*

> *with*(*plots*) :

> *dualaxisplot*(*plot*(sin), *plot*(cos))

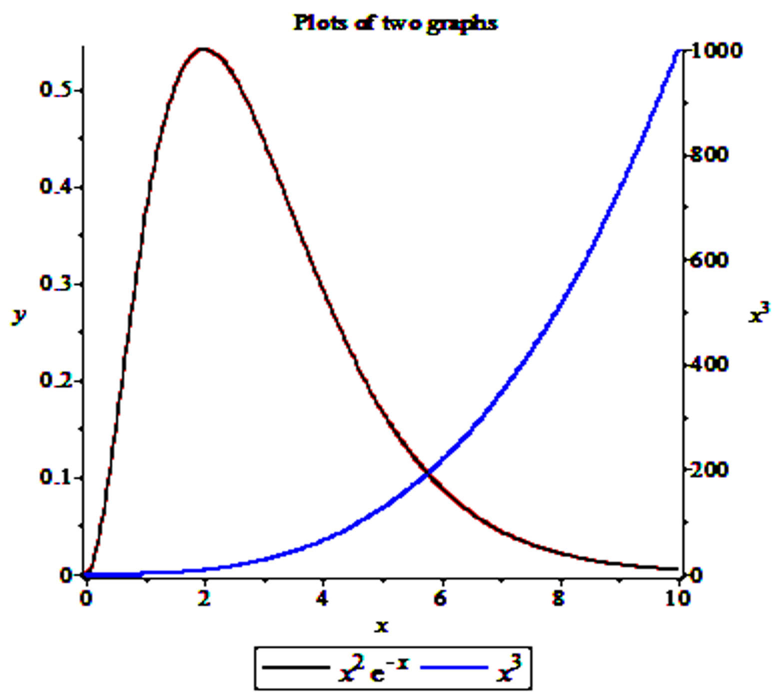


>
$dualaxisplot\left(inequal(\{x - y \le 5, 0 < x + y\}, x = -10..10, y = -10..10, optionsexcluded = (color = white)), conformal\left(z^2, z = 0..5 + 5I\right)\right)$



\>
$dualaxisplot\left(plot\left(x^2\cdot\exp(-x), x = 0..10, labels = [x, y], legend = x^2\cdot\exp(-x)\right), plot\left(x^3, x = 0..10, color = blue, labels = \left[x, x^3\right], legend = x^3\right), title = \text{" Plots of two graphs "}\right)$

\>
$dualaxisplot\left(animate\left(plot, \left[A\,x^3, color = blue, labels = \left[x, x^3\right]\right], A = 0..1\right), plot\left(x^2, labels = \left[x, x^2\right]\right)\right)$



\> *restart*

> with(plots,[pareto]):

>pdata:= 'Engine 1'=327,

\`Engine 2\`= 240,

\`Engine 3\`=176,

\`Wire 1\`=105,

`Wire 2`=43,

`Wire 3`=36,

Oil=33,

Coils=90,

`Gear Box`=61,

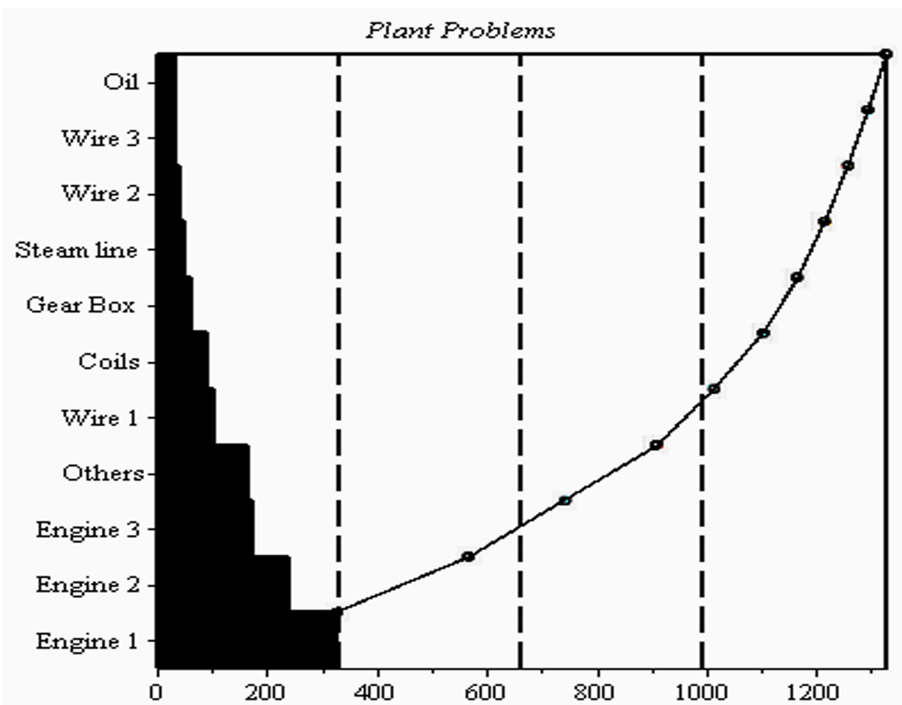
`Steam line`=50,

Others=166]:

>Fdata:=map(rhs,Pdata):

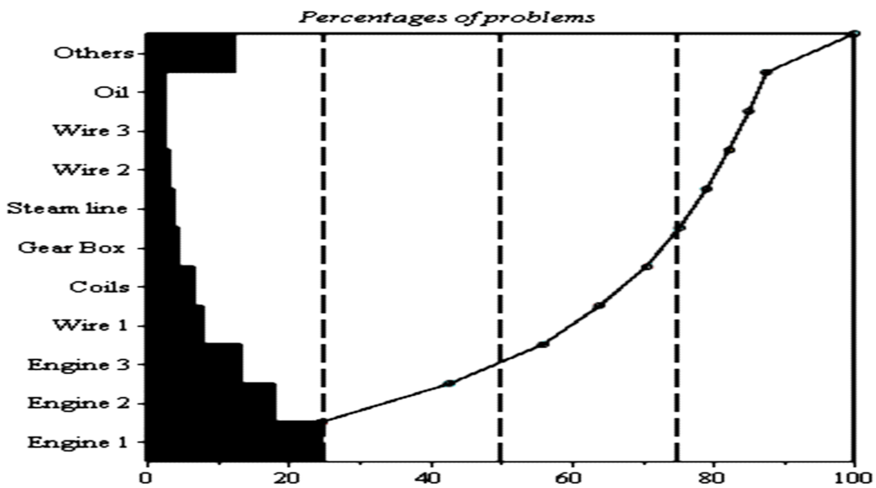
> Lab:=map(lhs,Pdata):

> > *pareto*(*Fdata*, *tags* = *Lab*, *title* = `*Plant Problems*`);



> Fdata_norm:=map((x,s) -> 100*x/s, Fdata, `+`(op(Fdata))):

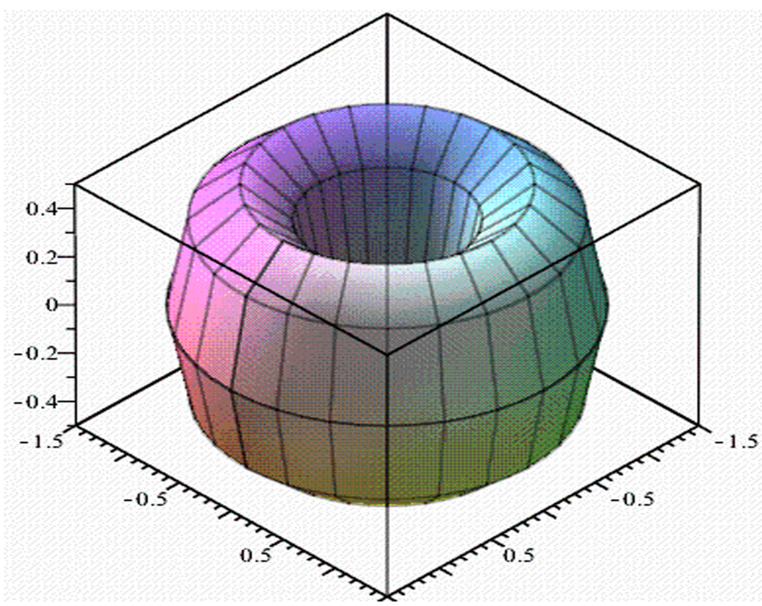
> pareto(Fdata_norm, tags=Lab, misc=Others, title=`Percentages of problems`);



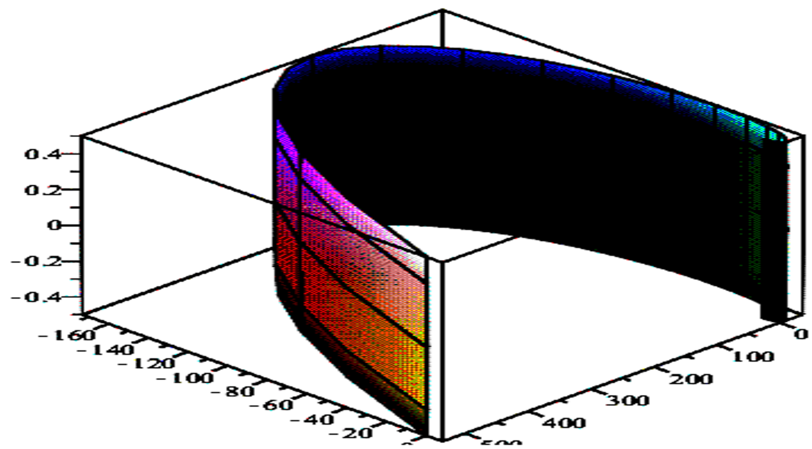
> *restart*

> *with*(*plots*) :

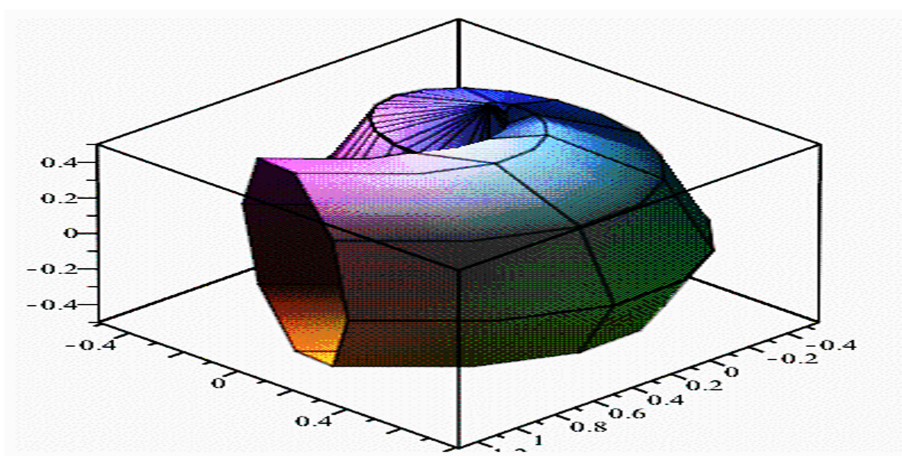
> $tubeplot([\cos(t), \sin(t), 0], t = 0 ..2\pi, radius = 0.5)$



> $tubeplot([\exp(t)\cdot\cos(t), \exp(t)\cdot\sin(t), 0], t = 0..2\pi, radius = 0.5)$



> $tubeplot([\exp(-t)\cdot\cos(t), \exp(-t)\cdot\sin(t), 0], t = 0..5\pi, radius = 0.5)$

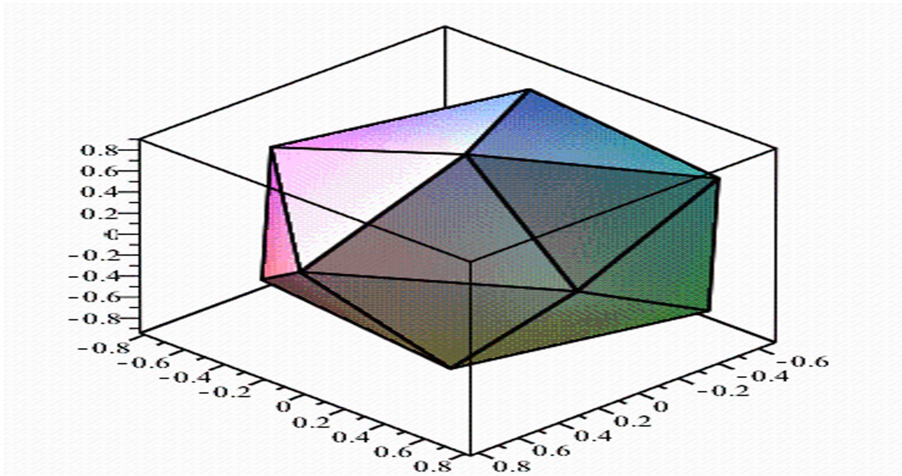


> $restart$

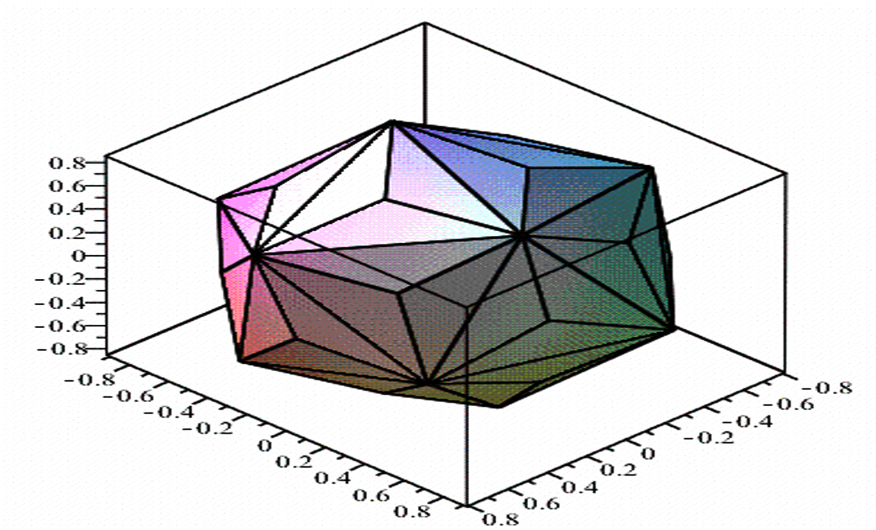
> $with(plots):$

> $polyhedra\_supported():$

> $polyhedraplot([0, 0, 0], polytype = GyroelongatedPentagonalPyramid, scaling = constrained)$



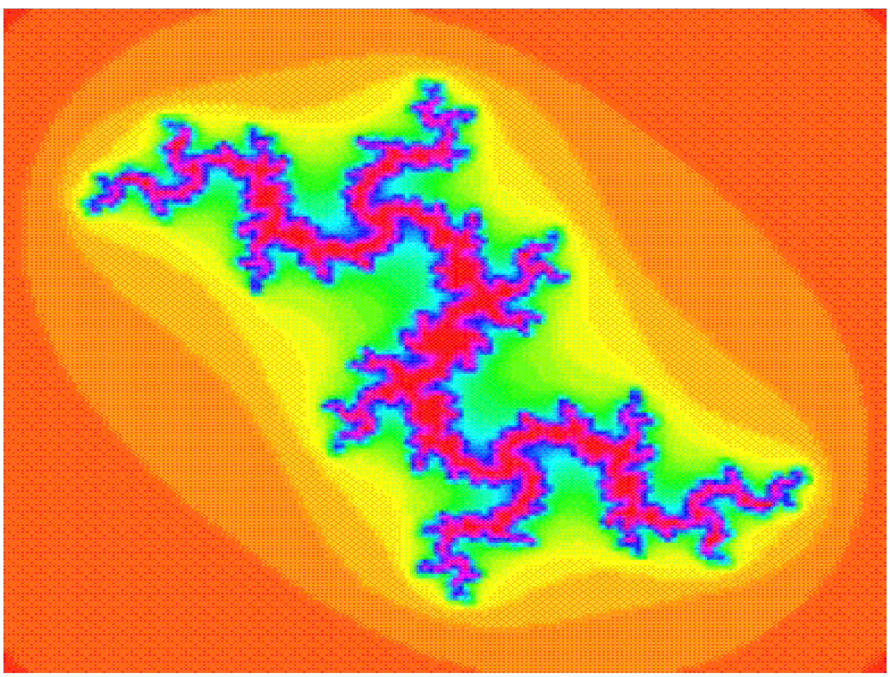
> *polyhedraplot*([0, 0, 0], *polytype* = *TriakisIcosahedron*, *scaling* = *constrained*)



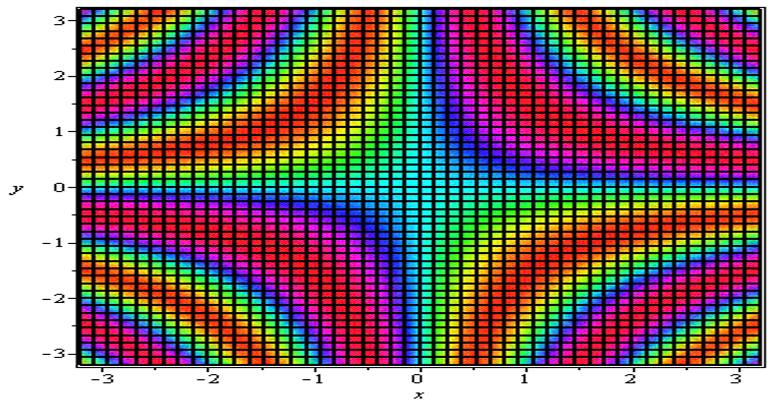
```
>JuliaSet:= proc(a,b)

 local z1, z2, z1s, z2s,m;

 (z1, z2): = (a,b):

 z1s:= z1^2:

 z2s: = z2^2;
```

```
 for m to 30 while z1s+z2s < 4 do
  (z1, z2):= (z1s-z2s, 2*z1*z2) + (0, 0.75);
  z1s:=z1^2;
  z2s:= z2^2;
 end do;
m;
end proc:
```

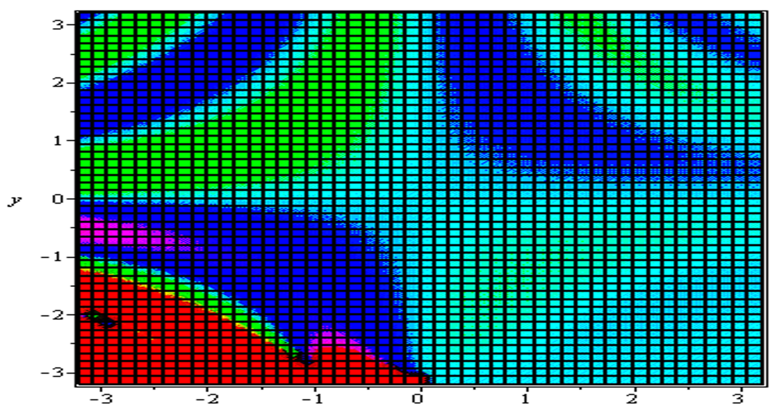
>
densityplot(*JuliaSet*, −1.5 ..1.5, −1.4 ..1.4, *colorstyle* = *HUE*, *grid* = [150, 150], *style* = *patchnogrid*, *axes* = *none*)



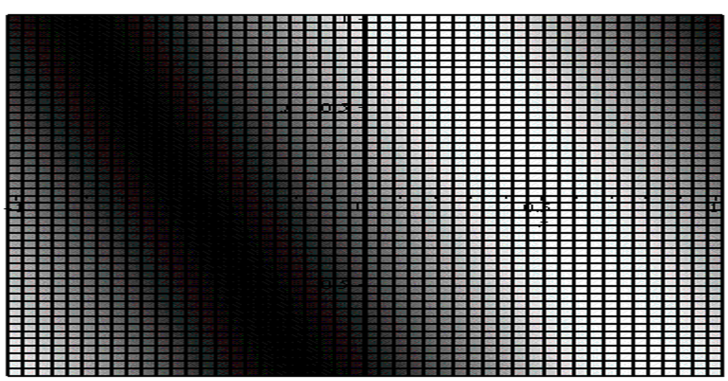
> *densityplot*$(\sin(x\,y), x = -\pi\,..\pi, y = -\pi\,..\pi,$ *axes* = *boxed*, *colorstyle* = *HUE*$)$



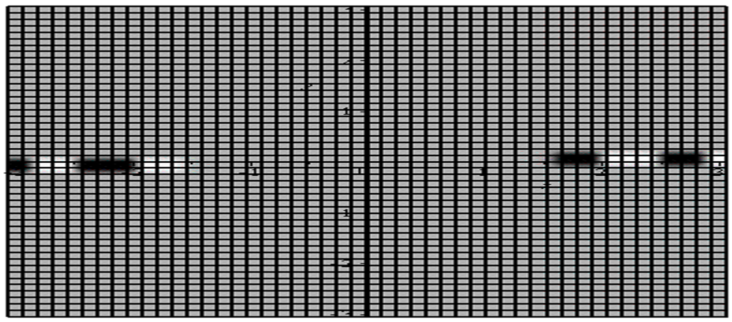
> $densityplot\left(\exp\left(-\frac{x}{y+\mathrm{Pi}}\right)\cdot\sin(x\,y), x=-\pi\,..\pi, y=-\pi\,..\pi, axes=boxed, colorstyle=HUE\right)$



> $densityplot(\sin(\mathrm{Pi}\cdot x+y), x=-1\,..1, y=-1\,..1)$



$> \mathit{densityplot}\left(\exp\left(\frac{x}{y}\right)\cdot\cos(\mathrm{Pi}\cdot x - y)\cdot\sin(\mathrm{Pi}\cdot x + y), x = -3\,..3, y = -3\,..3\right)$



$> \mathit{SpaceCurve}\left(\langle \mathrm{e}^{-t}\cos(t), \mathrm{e}^{-t}\sin(t)\rangle, t = 4\,..8\right)$

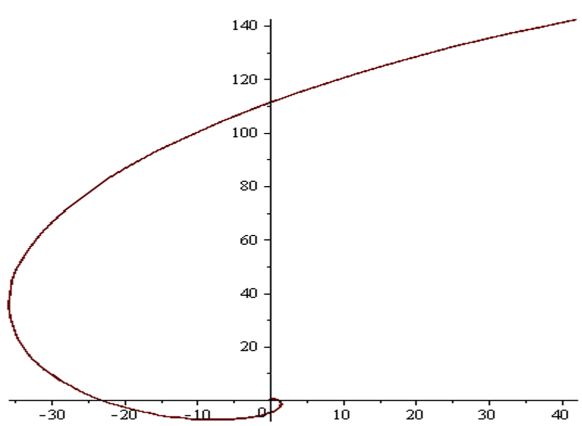
$$\mathit{SpaceCurve}\left(\begin{bmatrix} \mathrm{e}^{-t}\cos(t) \\ \mathrm{e}^{-t}\sin(t) \end{bmatrix}, t = 4\,..8\right)$$

$> \mathit{SpaceCurve}(\langle \cos(t), \sin(t), t\rangle, t = 1\,..9)$

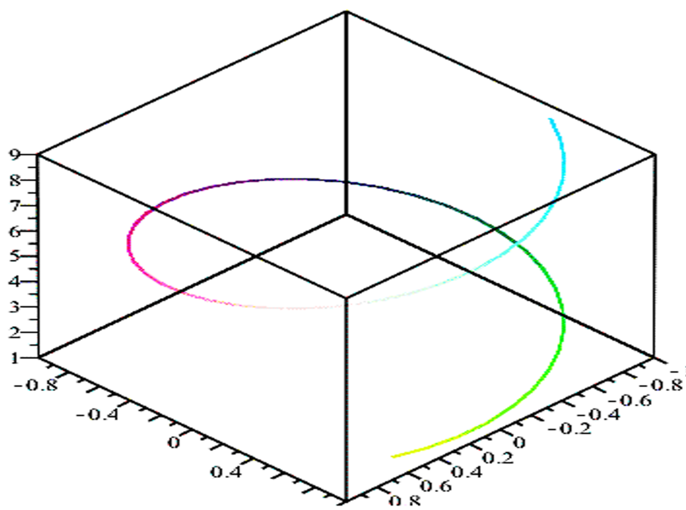
$$\mathit{SpaceCurve}\left(\begin{bmatrix} \cos(t) \\ \sin(t) \\ t \end{bmatrix}, t = 1\,..9\right)$$

$> \mathit{with}(\mathit{VectorCalculus}):$

$> \mathit{SpaceCurve}\left(\langle \mathrm{e}^{-t}\cos(t), \mathrm{e}^{-t}\sin(t)\rangle, t = -5\,..5\right)$



> $SpaceCurve(\langle \cos(t), \sin(t), t\rangle, t = 1\,..9)$



N-order Nuclear Reactor Process

$with(ODETools)$:

> $with(plots)$:

> $eqn1 := diff(N(t), t) = \frac{(k-1)\cdot N(t)}{l} - \frac{\text{beta}\cdot N(t)}{l} + sum(\text{lambda}[i]\cdot r[i](t), i = 1\,..m);$

$$eqn1 := \frac{\mathrm{d}}{\mathrm{d}t} N(t) = \frac{(k-1)\,N(t)}{l} - \frac{\beta\,N(t)}{l} + \sum_{i=1}^{m} \lambda_i r_i(t)$$

> $eqn2 := diff(r[i](t), t) = \frac{\text{beta}\cdot N(t)}{l} - \text{lambda}[i]\cdot r[i](t);$

$$eqn2 := \frac{\mathrm{d}}{\mathrm{d}t} r_i(t) = \frac{\beta\,N(t)}{l} - \lambda_i r_i(t)$$

> $wih(PDETools, dchange)$:

> $chngV1 := \{N(t) = sum(N[j]\cdot \exp(\text{omega}[j](t)), j = 0\,..m)\};$

$$chngV1 := \left\{ N(t) = \sum_{j=0}^{m} N_j \, e^{\omega_j(t)} \right\}$$

> $chngV2 := \{r[i](t) = sum(r[i,j] \cdot \exp(\text{omega}[j](t)), j = 0..m)\};$

$$chngV2 := \left\{ r_i(t) = \sum_{j=0}^{m} r_{i,j} \, e^{\omega_j(t)} \right\}$$

> $P1 := op(factor(combine(expand(chngV1), power)))$

$$P1 := N(t) = \sum_{j=0}^{m} N_j \, e^{\omega_j(t)}$$

> $P2 := op(factor(combine(expand(chngV2), power)))$

$$P2 := r_i(t) = \sum_{j=0}^{m} r_{i,j} \, e^{\omega_j(t)}$$

> $evalf(\{eqn1, eqn2\});$

$$\left\{ \frac{d}{dt} N(t) = \frac{(k - 1.)\, N(t)}{l} - \frac{1.\, \beta\, N(t)}{l} + \sum_{i=1}^{m} \lambda_i r_i(t), \frac{d}{dt} r_i(t) = \frac{\beta\, N(t)}{l} - 1.\, \lambda_i r_i(t) \right\}$$

> $subs([\text{P1}, \text{P2}], [eqn1, eqn2]);$

$$\left[\frac{\partial}{\partial t} \left(\sum_{j=0}^{m} N_j \, e^{\omega_j(t)} \right) = \frac{(k-1) \left(\sum_{j=0}^{m} N_j \, e^{\omega_j(t)} \right)}{l} - \frac{\beta \left(\sum_{j=0}^{m} N_j \, e^{\omega_j(t)} \right)}{l} + \sum_{i=1}^{m} \lambda_i \left(\sum_{j=0}^{m} r_{i,j} \, e^{\omega_j(t)} \right), \right.$$
$$\left. \frac{\partial}{\partial t} \left(\sum_{j=0}^{m} r_{i,j} \, e^{\omega_j(t)} \right) = \frac{\beta \left(\sum_{j=0}^{m} N_j \, e^{\omega_j(t)} \right)}{l} - \lambda_i \left(\sum_{j=0}^{m} r_{i,j} \, e^{\omega_j(t)} \right) \right]$$

> $p := simplify(\%);$

$$p := \left[\sum_{j=0}^{m} N_j \left(\frac{d}{dt} \omega_j(t) \right) e^{\omega_j(t)} = -\frac{\beta \left(\sum_{i=0}^{m} N_i e^{\omega_i(t)} \right) - \left(\sum_{i=0}^{m} N_i e^{\omega_i(t)} \right) k - \left(\sum_{i=1}^{m} \lambda_i \left(\sum_{j=0}^{m} r_{i,j} e^{\omega_j(t)} \right) \right) l + \sum_{i=0}^{m} N_i e^{\omega_i(t)}}{l}, \sum_{j=0}^{m} r_{i,j} \left(\frac{d}{dt} \omega_j(t) \right) e^{\omega_j(t)} = -\frac{\lambda_i \left(\sum_{j=0}^{m} r_{i,j} e^{\omega_j(t)} \right) l - \beta \left(\sum_{j=0}^{m} N_j e^{\omega_j(t)} \right)}{l} \right]$$

> *map*(*simplify*, **(28)**, '*assume* = *nonnegative*')

$$\left[\sum_{j=0}^{m} N_j \left(\frac{d}{dt} \omega_j(t) \right) e^{\omega_j(t)} = -\frac{\beta \left(\sum_{i=0}^{m} N_i e^{\omega_i(t)} \right) - \left(\sum_{i=0}^{m} N_i e^{\omega_i(t)} \right) k - \left(\sum_{i=1}^{m} \lambda_i \left(\sum_{j=0}^{m} r_{i,j} e^{\omega_j(t)} \right) \right) l + \sum_{i=0}^{m} N_i e^{\omega_i(t)}}{l}, \sum_{j=0}^{m} r_{i,j} \left(\frac{d}{dt} \omega_j(t) \right) e^{\omega_j(t)} = -\frac{\lambda_i \left(\sum_{j=0}^{m} r_{i,j} e^{\omega_j(t)} \right) l - \beta \left(\sum_{j=0}^{m} N_j e^{\omega_j(t)} \right)}{l} \right]$$

Examples on image processing

> *with*(*ImageTools*) :

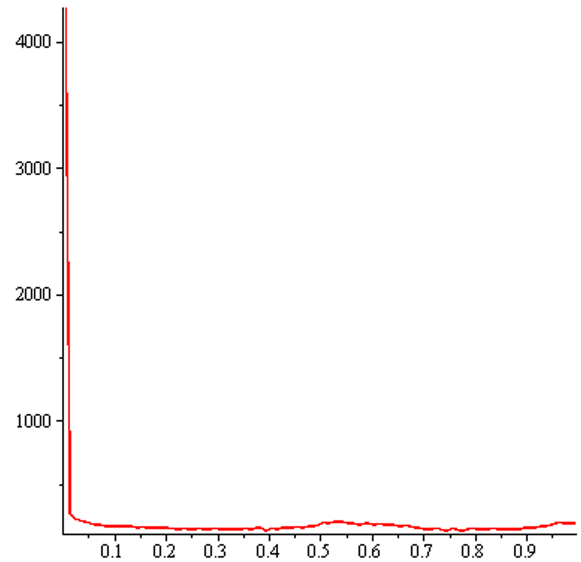
> $img1 := Create\left(100, 200, (r, c) \rightarrow evalf\left(0.5 \sin\left(\frac{r}{50}\right) + 0.5 \sin\left(\frac{c}{30}\right)\right)\right)$:

> *img2* := *Complement*(*img1*) :

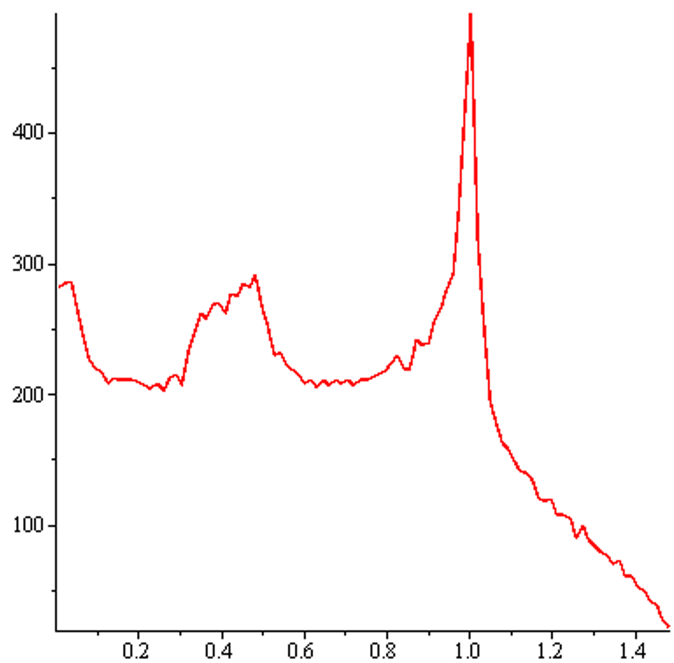
> *View*(*img1*)

> *View*([*img1*, *img2*])

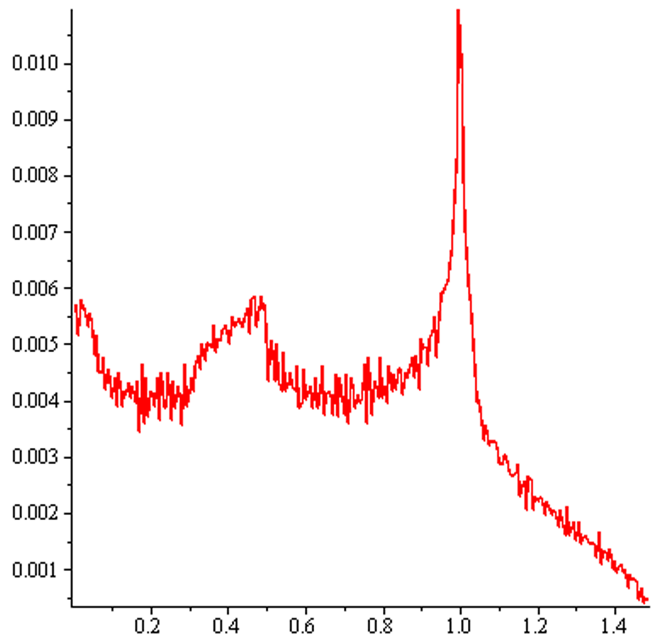
> $PlotHistogram(img1, 100)$



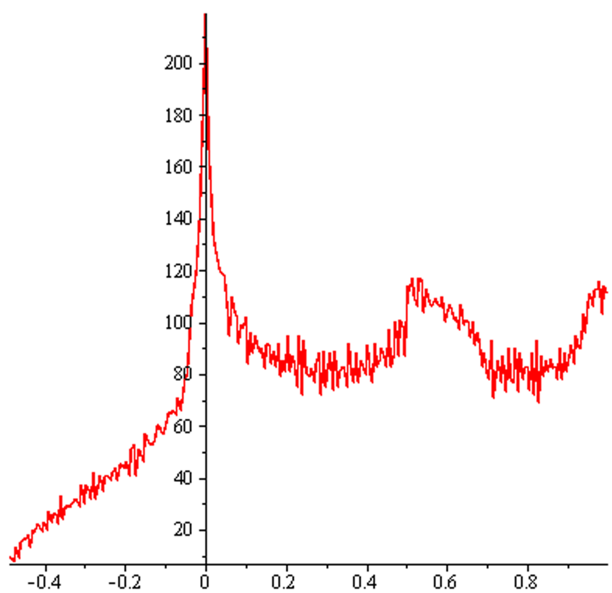
> $PlotHistogram(img2, 100, autorange)$



> $PlotHistogram(img2, autorange, normalized)$

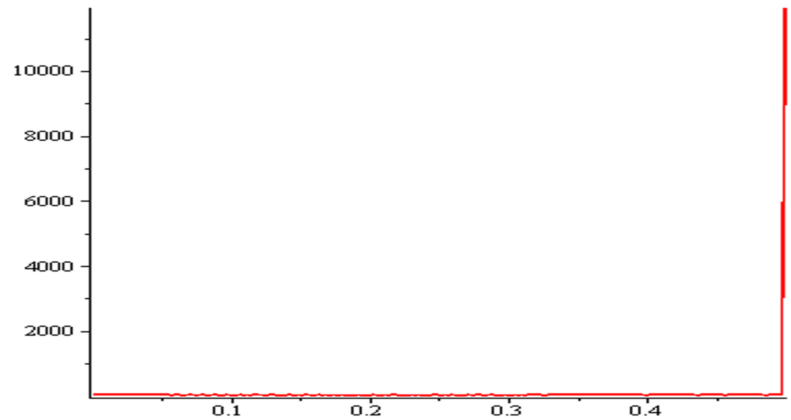


> $PlotHistogram(img1, autorange)$



> $PlotHistogram(img2, range = 0..0.5)$

Example for fitting experiments



> $restart$

> $with(Statistics):$

> $X := Vector([1, 2, 3, 4, 5, 6], datatype = float):$

> $Y := Vector([2, 5.6, 8.2, 20.5, 40.0, 95.0], datatype = float):$

> $ExponentialFit(X, Y, v)$

$1.01888654495804\,\mathrm{e}^{0.746236510177018\,v}$

> $W := Vector([1, 1, 1, 2, 5, 5], datatype = float):$

> $ExponentialFit(X, Y, weights = W)$

$$\begin{bmatrix} 0.989482297469512 \\ 0.752656239139387 \end{bmatrix}$$

> $LinearFit\left([1, t, t^2], X, Y, t\right)$

$24.6000000000000 - 23.9892857142857\,t + 5.79642857142857\,t^2$

> $LinearFit(a + b\,t + c\,t^2, X, Y, t)$

$$24.6000000000000 - 23.9892857142857\,t + 5.79642857142857\,t^2$$

Consider now an experiment where quantities x, y and z are quantities influencing a quantity w according to an approximate relationship

$$w = a\,x + \frac{b\,x^2}{y} + c\,y\,z$$

with unknown parameters a, b, and c. Six data points are given by the following matrix, with respective columns for x, y, z, and w.

> $ExperimentalData := \langle\langle 1, 1, 1, 2, 2, 2\rangle|\langle 1, 2, 3, 1, 2, 3\rangle|\langle 1, 2, 3, 4, 5, 6\rangle|\langle 0.531, 0.341, 0.163, 0.641, 0.713, -0.040\rangle\rangle$

$$ExperimentalData := \begin{bmatrix} 1 & 1 & 1 & 0.531 \\ 1 & 2 & 2 & 0.341 \\ 1 & 3 & 3 & 0.163 \\ 2 & 1 & 4 & 0.641 \\ 2 & 2 & 5 & 0.713 \\ 2 & 3 & 6 & -0.040 \end{bmatrix}$$

> $LinearFit\left(\left[x, \frac{x^2}{y}, y\,z\right], ExperimentalData, [x, y, z]\right)$

$$0.823072918385878\,x - \frac{0.167910114211606\,x^2}{y} - 0.0758022678386438\,y\,z$$

> $NonlinearFit(a + b\,v + e^{c\,v}, X, Y, v)$

$$2.15979247107424 - 1.22391291112346\,v + e^{0.766784080984173\,v}$$

> $NonlinearFit\left(x^a + \frac{b\,x^2}{y} + c\,y\,z, ExperimentalData, [x, y, z], initialvalues = [a = 2, b = 1, c = 0], output = [leastsquaresfunction, residuals]\right)$

$$\left[x^{1.14701973996968} - \frac{0.298041864889394\,x^2}{y} - 0.0982511893429762\,y\,z,\right.$$
$$[0.0727069457676300, 0.116974310183398, -0.146607992383251,$$
$$\left.-0.0116127470057686, -0.0770361532848388, 0.0886489085642805]\right]$$

APPENDIX B

Introduction to MapleSim Software

Abstract: MapleSim is a modelling environment for creating and simulating complex multi-domain physical systems. It allows building component diagrams that represent physical systems in the graphical form. MapleSim automatically generates model equations from the component diagrams using symbolic and numerical approaches and runs very highly accurate simulations.

MapleSim modelling environment combines components from different engineering domains such as mechanical, electrical, and multi-body for building and exploring realistic designs and for studying the system level.

Keywords: Maple, MapleSim, Monte Carlo Simulation, Numerical, Symbolic, Simulation

INTERACTIONS

In MapleSim environment

- Models' system level can be easily assessed to demonstrate concepts such as parameter optimization, sensitivity analyses, and interactions.

- Mathematical equations can be defined for new components from the first principle.
- Simulation can be carried out to investigate a much larger result of conditions that is possible. By testing with hardware alone, with no risk of damage to the equipment and for less cost.
- Allows export from MapleSim to C code, simulation, Labview, and other tools where it can be incorporated with a physical prototype.

Features in MapleSim

- MapleSim have facilities for visualization in 3D and animation of multibody systems, full playback, and cameral control in 3D visualization.
- Interface and modelling: It contains drag-and-drop block diagrams in modelling environment, modelling diagrams, and 3-D model construction of multibody systems, data import, and export.
- Block Library: MapleSim contains both physical component and signal-flow blocks. The physical component blocks have different formalities for many domains.

Benjamin Oyediran Oyelami

- Analysis and documentation: extract, view, and manipulation of the system equations for a model l, and parameter optimization. Simulation and parameter swaps including related files in a MapleSim model for easy documentation management and sharing.

Linear, nonlinear, continuous and discrete, SISO, MIMOS and hybrid systems parameter set managing and deployment to popular platforms from Mathword. MapleSim Connect can connect with Simulink.

B1. Design of Simulation using the MapleSim

B1.1 Code Generation

Code generation can handle all systems modeled in MapleSim, including hybrid systems with defined signal input (RealInput) and signal output (RealOutput) ports (MapleSim).

The source code in MapleSim is designed to interface with Maple, in the sample code; all inputs are assigned the value of 0. For more information about the available Code Generation command, see the GetCompiledProc help topic in MapleSim.

C Code Generation

For C code generation, select the attachment of the generation of code from the MapleSim.

Step 1: Subsystem Selection

Click the button:

Load Selected Subsystem

Step 2: Inputs/Outputs and Parameter Management

Inputs:

	Input Variables	Change Row
1		

Outputs:

Toggle Export Column

	Output Variables	Export	Change Row
1			
2	`Main.'output 1'.T`(t)	"X"	
3	`Main.output2.T`(t)	"X"	
4	`Main.output3.T`(t)	"X"	

☐ Add an additional output port for subsystem state variables

Parameters:

Click:

Toggle Export Column

Then, the parameters used in the model would be generated as:

	Parameters	Value	Export	Change Row
1			"X"	
2	HC1_C	15.	"X"	
3	HC2_C	15.	"X"	
4	HC3_C	15.	"X"	
5	TC1_G	0.1e2	"X"	
6	TC2_G	8.	"X"	

The C code for the modeling program can be generated using various solvers by selecting optimization optional and the max mean projection iteration

Step 3: C Code Generation Options

Solver Options:

Fixed step solver: ( ) Euler ( ) RK2 ( ) RK3 (•) RK4 ( ) Implicit Euler

Optimization Options:

Level of code optimization (0=None, 3=Full): 0 1 2 3

Constraint Handling Options:

Maximum number of projection iterations: 3

Error tolerance: 0.1e-4

[x] Apply projection during event iterations

Event Handling Options:

Maximum number of event iterations: 10

Width of event hysteresis band: 0.1e-9

Baumgarte Constraint Stabilization:

[ ] Apply Baumgarte constraint stabilization [x] Export Baumgarte parameters

Alpha: 10

Beta: 2

Step 4: Generate C Code

Target directory:

C:\Users\Prof B O Oyelami Browse

C-File:

MsimModel

Click to generate the C code:

Generate C code

Step 5: View C Code

```
/***************************************************
 * Automatically generated by Maple.
 * Created On: Fri Jun 12 04:49:35 2015.
***************************************************/
#ifdef WMI_WINNT
#define EXP __declspec(dllexport)
#else
#ifdef X86_64_WINDOWS
#define EXP __declspec(dllexport)
#else
#define EXP
#endif
#endif
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#ifdef FROM_MAPLE
#include <mplshlib.h>
static MKernelVector kv;
EXP ALGEB M_DECL SetKernelVector(MKernelVector kv_in, ALGEB args) { kv=kv_in; ret
#else
#ifdef WMI_WINNT
#define M_DECL __stdcall
#else
#define M_DECL
```

Monte Carlo Simulation

Author : Benjamin O Oyelami

Date:14 October 2022

Model Description

Monte Carlo simulation (MCS) can be made on a MapleSim model. To generate MCS, you define a random distribution for a parameter and you can run multiple simulations using this distribution. Note that the properties that are plotted are defined by the probes in the MapleSim model.

Monte-Carlo Simulation

To start, click **Load System**.

Load System

Parameter Distribution

Select the parameter you want to vary, and then choose an appropriate distribution and distribution parameters.

Parameter HC3.C

Nominal Value 15

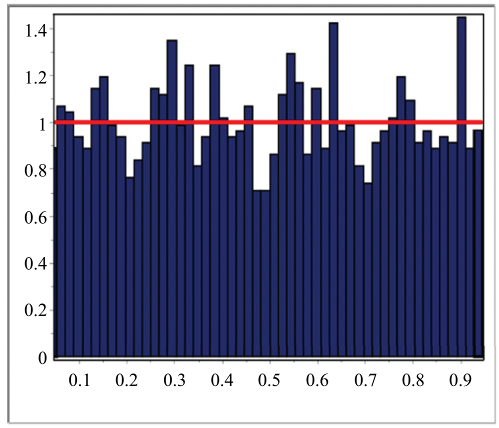
Distribution Uniform

Choose the parameter a and b, the Help

uniform distribution as follows:

a 0

b 1



Monte-Carlo Simulation

Enter the number of simulation runs and the number of bins in the simulation, and then click the **Run Simulation** button to create and display the simulation plots.

Number of simulations run (including nominal value) 6

To plot variation, in the boxes click:

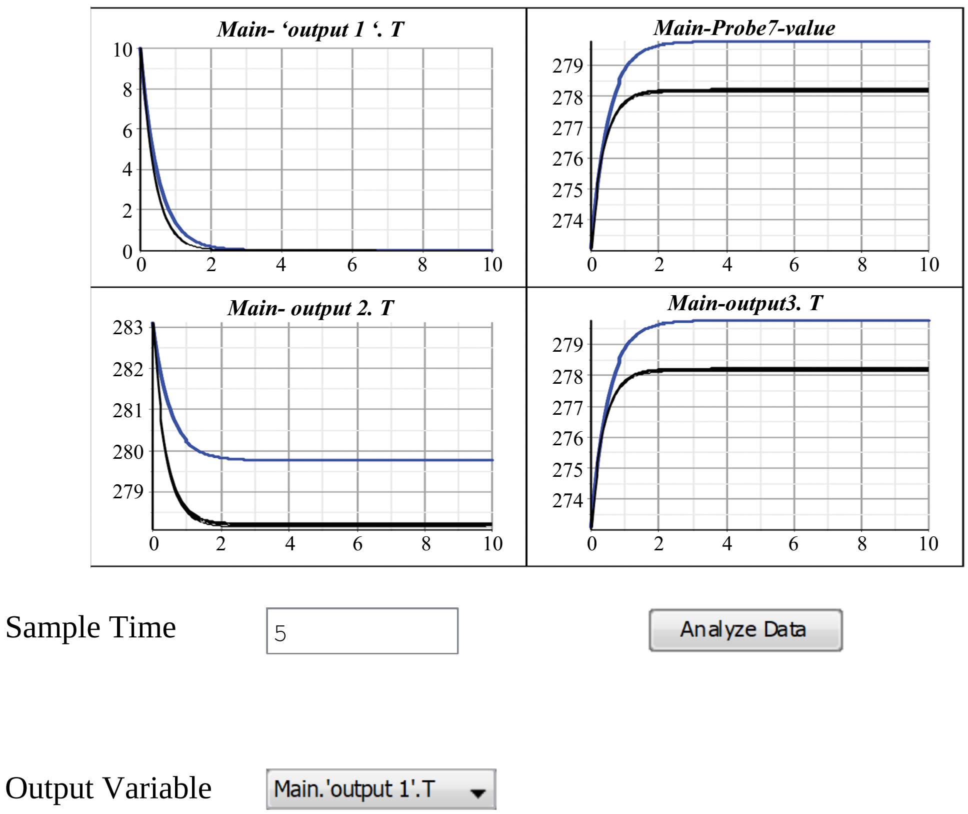
Run Simulation for All probes Plot variances in boxes

Click Run simulation for the given problem and number of bins and the probe plots are displayed above: 12 **Number of bins**

Note: The blue line corresponds to the nominal values.

Data Analysis

Specify a time value below (any float value between 0 and tf s), choose an output variable in the list, and then click **Analyze Data**. Statistics quantities will be displayed on a data set of 5 points, with each point corresponding to one of the simulations, not including the nominal. More information on the quantities displayed and plotted; see Statistics in the Maple Help. The data on which the quantities are computed and plotted are stored as a list of Matrices in the variable _data . The first element corresponds to the nominal value (which is not used in the statistics). Select the desired sample of output variables and click analyze data and the statistics quantities displayed as follows:



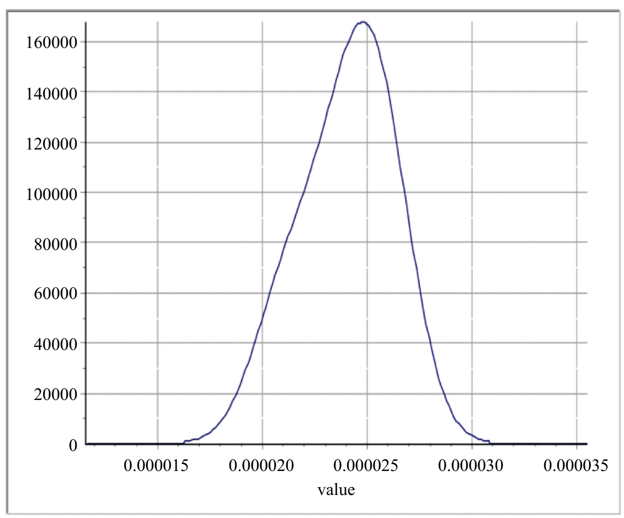
Statistics Quantities

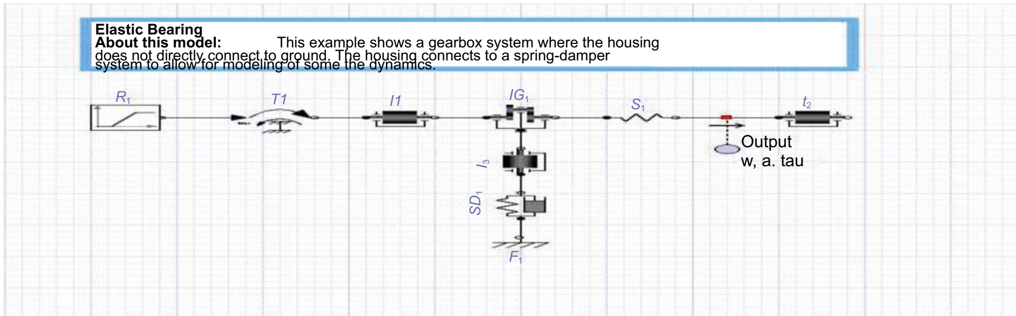
```
0.000005
Skewness
-0.517357
Standard Deviation
0.000002
Variance
0.000000
Variation
0.084149
```

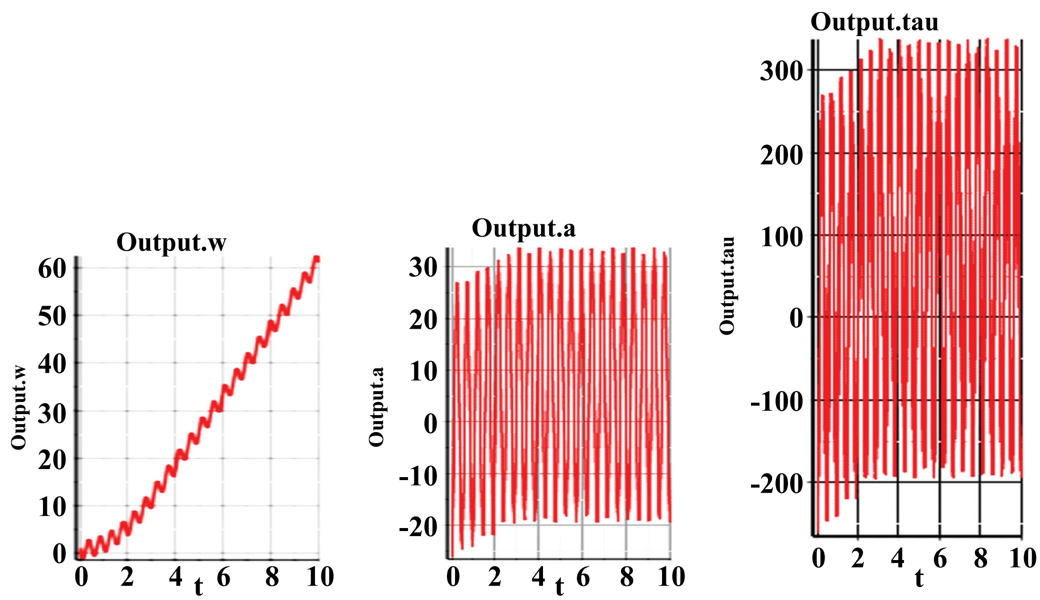
Select the type of plot you desire and click on the example,

Choose the Kernel/density plot and the plot displayed as follows:

KernelDensityPlot







Save this worksheet in Maple and then save the **msim** file to which this worksheet is attached in MapleSim.

SUBJECT INDEX

A

B

C

Benjamin Oyediran Oyelami

F

G

H

I

J