Numerical Methods for Time Dependent Phenomena

William Layton

(WLayton) Departmet of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA

Current address, WLayton: Departmet of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA

E-mail address, W Layton: wjl@pitt.edu

URL: http://www.math.pitt.edu/~wjl

Dedicated to the hopes: "I rejoice in a belief that intellectual light will spring up in the dark corners of the earth; that freedom of inquiry will produce liberality of conduct; that mankind will reverse the absurd position that the many were made for the few." George Washington, Inaugural Address.

2000 *Mathematics Subject Classification.* Primary 05C38, 15A15; Secondary 05A15, 15A18

The Author thanks V. Exalted.

Abstract. "Prediction is hard, especially about the future." attributed widely including to both Niels Bohr and Yoggi Berra.

This document is a preliminary set of lecture notes developed by the above author for his students. It is evolving and at various stages of polish and completion. All rights are reserved by the author. In particular, do not make copies without the written permission of the author.

Contents

Preface

> “Prediction is hard, especially about the future.”
>
> -attributed to many people including Yogi Berra and Niels Bohr
>
> To display formal fireworks, which are so much in the centre of many mathematical treatises—perhaps as a status-symbol by which one gains admission to the august guild of mathematicians–was not the primary aim of the book. [Lanczos, in the preface of his book "Discourse on Fourier Series"]

This book is about using numerical methods to predict the future reliably and efficiently. **Reliability** means errors in the prediction are a central concern. **Efficiency** means that the second central concern is cost / turnaround time. / resources required to produce the prediction. The problem we consider, the **initial value problem** or **IVP**, arises as follows. Suppose the state of a system at time t can be characterized by a collection of N numbers (the vector $y(t)$) where N is often quite large. Suppose the state of the system is known today (taken to be $t = 0$) and finally that the laws governing the system are known: the way the system changes depends on the time and on the state of the system

$$\begin{aligned} y' &= f(t, y), \text{ for all time } t > 0 \\ y(0) &= y_0, \text{ at time } t = 0. \end{aligned} \tag{IVP}$$

The *initial value problem* is then to predict (reliably and efficiently) the future state of the system: find $y(t)$ for $t > 0$.

The most basic solution to this problem was devised by the great Leonard Euler. It (**Euler's method**) proceeds as follows: We pick a step size called $\triangle t$. The variables t_j and y_j denote $t_j = j\triangle t$ and y_j is the approximation we compute to $y(t_j)$:

$$\triangle t = \text{step size},\ t_j = j\triangle t = j^{th} \text{ time step},\ y_j \approx y(t_j).$$

Euler's method to find y_j is constructive. It is motivated as follows: Suppose we know $y(t_j)$ exactly and want $y(t_{j+1}) = y(t_j + \triangle t)$. Expanding y in a Taylor series at t_j gives:

$$y(t_{j+1}) = y(t_j) + y'(t_j)\triangle t + \frac{1}{2}y''(\xi)\triangle t^2 \text{ , for some } \xi,\ t_j < \xi < t_{j+1}.$$

Now the equation $y(t)$ satisfies is $y'(t_j) = f(t_j, y(t_j))$. Thus:

$$y(t_{j+1}) = y(t_j) + \triangle t f(t_j, y(t_j)) + \frac{1}{2}y''(\xi)\triangle t^2 \text{ , for some } \xi,\ t_j < \xi < t_{j+1}.$$

The last term, $\frac{1}{2}y''(\xi)\triangle t^2$, is "unknowable" since both y'' and the point ξ is unknown but it is small[1] if $\triangle t$ is small. Just dropping this last term is Euler's method:

$$\text{Given } y_j \text{ find } y_{j+1} \text{ by}$$

$$\text{(Euler)} \qquad y_{j+1} = y_j + \triangle t f(t_j, y_j) \text{ , for } j = 0, 1, 2, \cdots.$$

In theory this works: It is easy to program, constructive, convergent

$$y_n \to y(t_n) \text{ as } \triangle t \to 0.$$

In practice, Euler's[2] method is nearly completely inadequate. It has low accuracy: on many computers the small amount of roundoff errors always present accumulates fast enough to overwhelm the methods accuracy. Its error can grow exponentially fast as more steps are taken even in cases when the true solution does not. There are also cases where its predictions are fundamentally, qualitatively wrong. These are the central concerns in developing methods, inspired by Euler's method, of much greater accuracy. Reliability requires the methods function like expert systems in some respects and select their own time step to produce target accuracy with minimal work. This approach is called adaptivity and is a central contribution of modern numerical analysis and the heart of this book. Without efficient adaptive methods no further progress is possible as all other issues are limited by numerical errors. With adaptivity, other important issues can be addressed with hope.

Some of these other issues include the following:

1. Errors due to uncertain initial conditions or measured parameters in the model.

> Chaos- when the present determines the future but the *approximate* present does not determine the *approximate* future. - Edward Lorenz

If $y(0)$ or some parameter in $f(t, y)$ are known from measurements, they will only be known to a few significant digits

$$y(0) = y_0 \pm \varepsilon.$$

The correct way to treat this uncertainty is by statistical techniques. Operationally, a number of systems are solved with random perturbations of the initial conditions and the results are averaged to obtain the most likely scenario. The spread of forecasts the also gives a confidence interval for the influence of the initial error. To get an idea of the costs involved, consider a small problem. If the model has only 100 components in $y(t)$ and each component only 100 different perturbations are evaluated; this means $100^2 = 10,000$ different runs solving the IVP are required. If the uncertainty arises in $f(t, y)$ these $10,000$ new problems arise each time step.

2. Unknown initial data.

It is also quite common for a complete initial condition to be unknown. Thus suppose we only know $C_0 y(0) = x_0$ for C not of full rank. Somehow the missing

[1] In applied math, something is *small* usually means that $(\textit{something})^2$ is negligable.

[2] Adapted from the Wikipedia article on him:

Leonhard Euler (1707 – 1783) was a Swiss mathematician who made important discoveries in many branches of mathematics. He is also known for his work in mechanics, fluid dynamics, optics, astronomy, and music theory. Euler is held to be one of the greatest scientists in history.

components of $y(0)$ must be inferred from measurements / observations of the solution at later times. If we know components of $y(T)$, so for some other matrix C_T not of full rank, $C_T y(T) = y_{data}(T)$, the uncertain components of the initial data are filled in by solving (backwards in time) the problem

$$\begin{array}{ll} \text{minimize :} & |C_0 y(0) - x_0|^2 + |C_T y(T) - y_{data}(T)|^2 \\ \text{subject to:} & y' = f(t,y), 0 < t < T. \end{array}$$

3. Errors in the model due to measured parameters or to unrepresented processes.

These must be identified from extra solution measurements. Measurements or observations are averages. Thus, they containing necessarily less information than $y(t)$. (Otherwise, we would just use the observation as a new initial condition.) Thus, we have a matrix C which is not of full rank and observations $y_{data}(t)$ and want to minimize $y_{data}(t) - Cy(t)$. Thus the problem becomes:

$$\begin{array}{ll} \text{minimize :} & |y_{data}(t) - Cy(t)|^2 \\ \text{subject to:} & y' = f(t,y), t > 0, \ \& \ y(0) = y_0. \end{array}$$

Cases 2 and 3 above are optimization problems with IVP sitting at their center. Optimization problems are solved by iteration, (in simple form, given a guess of the unknown data, change it a bit and see if the quantity minimized goes up or down and use that information to improve the guess of the unknown data) which requires solving the IVP many times.

It should be clear by now that the cost of using numerical methods to understand phenomena (rather than just solve problems) can be very high when repeated solves are done to address all these other issues. This high cost has led to four great streams of ideas in the modern development of numerical methods:

- **Adaptivity,**
- **Parallelism**,
- **Modularity** and
- **Hierarchical computations.**

0.1. The discoverers. ADAPTED FROM WIKIPEDIA:

Edward Norton Lorenz (23 May 1917 – 16 April 2008) was an American mathematician, meteorologist, and a pioneer of chaos theory. He introduced the strange attractor notion and coined the term butterfly effect. Lorenz was born in West Hartford, Connecticut. He studied mathematics at both Dartmouth College in New Hampshire and Harvard University in Cambridge, Massachusetts. From 1942 until 1946, he served as a meteorologist for the United States Army Air Corps. After his return from World War II, he decided to study meteorology. Lorenz earned two degrees in the area from the Massachusetts Institute of Technology where he later was a professor for many years. He was a Professor Emeritus at MIT from 1987 until his death.

During the 1950s, Lorenz became skeptical of the appropriateness of the linear statistical models in meteorology, as most atmospheric phenomena involved in weather forecasting are non-linear. His work on the topic culminated in the publication of his 1963 paper "Deterministic Non-periodic Flow" in Journal of the Atmospheric Sciences, and with it, the foundation of chaos theory. He states in that paper:

Two states differing by imperceptible amounts may eventually evolve into two considerably different states ... If, then, there is any error whatever in observing the present state—and in any real system such errors seem inevitable—an acceptable prediction of an instantaneous state in the distant future may well be impossible....In view of the inevitable inaccuracy and incompleteness of weather observations, precise very-long-range forecasting would seem to be nonexistent.

His description of the butterfly effect followed in 1969. He was awarded the Kyoto Prize for basic sciences, in the field of earth and planetary sciences, in 1991, the Buys Ballot Award in 2004, and the Tomassoni Award in 2008. In his later years, he lived in Cambridge, Massachusetts. He was an avid outdoorsman, who enjoyed hiking, climbing, and cross-country skiing. He kept up with these pursuits until very late in his life, and managed to continue most of his regular activities until only a few weeks before his death. According to his daughter, Cheryl Lorenz, Lorenz had "finished a paper a week ago with a colleague." On April 16, 2008, Lorenz died at his home in Cambridge at the age of 90, having suffered from cancer.

Part 1

The First Part

"All great things are slow of growth." - Epictetus.

More quotes:

Chaos- when the present determines the future but the approximate present does not determine the approximate future." Edward Lorenz

......

Analysis and algebraic conditions: Theorem 2.2 [Dahlquist equivalence theorem] demonstrates a state of affairs that prevails throughout mathematical analysis. Thus, we desire to investigate an analytic condition, e.g. whether a differential equation has a solution, whether a continuous dynamical system is asymptotically stable, whether a numerical method converges. By their very nature, analytic concepts involve infinite processes and continua, hence one can expect analytic conditions to be difficult to verify, to the point of unmanageability. For all we know, the human brain (exactly like a digital computer) might be essentially an algebraic machine. It is thus an important goal in mathematical analysis to search for equivalent algebraic conditions. The Dahlquist equivalence theorem is a remarkable example of this: everything essentially reduces to determining whether the zeros of a polynomial reside in a unit disc, and this can be checked in a finite number of algebraic operations! In the course of this book we will encounter numerous other examples of this state of affairs. Cast your mind back to basic infinitesimal calculus and you are bound to recall further instances where analytic problems are rendered in an algebraic language. [Arieh Iserles AFCINADE2, p. 25]

....

Does anyone believe that the difference between the Lebesgue and Riemann integrals can have physical significance, and that whether say, an airplane would or would not fly could depend on this difference? If such were claimed, I should not care to fly in that plane. [Richard W. Hamming, in N. Rose's Mathematical Maxims and Minims]

It is frequently claimed that Lebesgue integration is as easy to teach as Riemann integration. This is probably true, but I have yet to be convinced that it is as easy to learn. [Thomas William Korner, A Companion to Analysis: A Second First and First Second Course in Analysis, p. 197]

"In the future, proponents of numerical fluid dynamics should explain the limitations (as well as statistical uncertainties)..." Garrett Birkhoff, p. 29 in: Numerical Fluid Dynamics, SIAM Review, 25(1983), 1-34.

....

... discrete mathematics is more difficult than continuous mathematics. If you look at formulas for derivatives of reciprocals and then finite differences for reciprocals, you see how things are more complicated in the discrete case. ... The main point in the theory of difference approximations is to prove stability. To prove stability is like getting an a priori estimate for the solution of the equation. But to get those estimates for difference approximations is much more sophisticated than to get them for a differential equation. [Peter Lax, MAA Focus (May/June 2005)]

"All great things are slow of growth." - Epictetus.

....

Since a priori estimates lie at the heart of most of his arguments, many of Leray's papers contain symphonies of inequalities; sometimes the orchestration is heavy, but the melody is always clearly audible. [Peter Lax on Jean Leray]

The equations at which we arrive must be such that a person of any nation, by substituting the numerical values of the quantities as measured by his own national units, would obtain a true result.

james clerk maxwell

....

Through all of scientific computing runs this common theme: Increase the accuracy at least to second order. What this means is: Get the linear term right. [GS Gilbert Strang , BAMS, 1993, Wavelet Transforms vs. Fourier Transforms]

....

The reason is not to glorify "bit chasing"; a more fundamental issue is at stake here: numerical subroutines should deliver results that satisfy simple, useful mathematical laws whenever possible . [...] Without any underlying symmetry properties, the job of proving interesting results becomes extremely unpleasant. The enjoyment of one's tools is an essential ingredient of successful work .

donald knuth

....

The purpose of computing is insight, not numbers. [Richard Hamming]

....

A computer lets you make more mistakes faster than any invention in human history - with the possible exceptions of handguns and tequila.

– Mitch Ratliffe

....

Too many people write papers that are very abstract and at the end they may give some examples. It should be the other way around. You should start with understanding the interesting examples and build up to explain what the general phenomena are. This was your progress from initial understanding to more understanding. This is both for expository purposes and for mathematical purposes. It should always guide you.

When someone tells me a general theorem I say that I want an example that is both simple and significant. It's very easy to give simple examples that are not very interesting or interesting examples that are very difficult. If there isn't a simple, interesting case, forget it. [Sir Michael Atiyah]

....

There are many methods for predicting the future. For example, you can read horoscopes, tea leaves, tarot cards, or crystal balls. Collectively, these methods are known as "nutty methods." Or you can put well-researched facts into sophisticated computer models, more commonly referred to as "a complete waste of time."

– Scott Adams, The Dilbert Future

....

Truth is treason in an empire of lies. [George Orwell, "1984"]

....

To think, you have to write. If you're thinking without writing, you only think you're thinking. [Leslie Lamport, Thinking for Programmers]

....

Writing is nature's way of letting you know how sloppy your thinking is. [Guindon]

....

To display formal fireworks, which are so much in the centre of many mathematical treatises—perhaps as a status-symbol by which one gains admission to the august guild of mathematicians–was not the primary aim of the book. [Lanczos, in the preface of his book "Discourse on Fourier Series"]

....

numerical analysis is very much an experimental science.

Peter Wynn. "On some recent developments in the theory and application of continued fractions". Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis Volume 1, pages 177–197.

....

"I rejoice in a belief that intellectual light will spring up in the dark corners of the earth; that freedom of inquiry will produce liberality of conduct; that mankind will reverse the absurd position that the many were made for the few." George Washington, Inaugural Address.

....

The cancellation in the subtraction only gives an indication of the unhappy consequence of a loss of information in previous steps, due to rounding of [at least] one of the operands, and is not the cause of the inaccuracy. [Dahlquist and Bjork, Numerical Methods in Scientific Computing, Volume 1, p. 17]

....

Most unfortunately, the habit in the numerical analysis literature is to speak not of the convergence of these magnificently efficient methods [Adams multistep integration methods, or other numerical methods for that matter], but of their error, or more precisely their discretization or truncation error as distinct from rounding error. This ubiquitous language of error analysis is dismal in tone, but seems ineradicable. [Llyod Nick Trefethen, Numerical Analysis, PCM]

Part 2

Initial value problems

> There are many methods for predicting the future. For example, you can read horoscopes, tea leaves, tarot cards, or crystal balls. Collectively, these methods are known as "nutty methods." Or you can put well-researched facts into sophisticated computer models, more commonly referred to as "a complete waste of time."
>
> – Scott Adams, The Dilbert Future

In an initial value problem, the state of a system at time t is described by a collection of numbers (i.e., a vector) that change with time:

$$\overrightarrow{y}(t) = (y_1(t), y_2(t), \cdots, y_N(t))^T.$$

If the state is one number it is simply $y(t)$ and if it consists of two numbers is is generally called $(x(t), y(t))$ instead of $(y_1(t), y_2(t))$. The state of the system is known at some starting time (almost always taken to be $t = 0$)

$$\overrightarrow{y}(0) = \overrightarrow{y}_0 \text{ (a known vector).}$$

The laws governing the system are also known and take the general form

$$\frac{d}{dt}\overrightarrow{y}(t) = \overrightarrow{f}(t, \overrightarrow{y}(t)) \text{ for } t > 0.$$

This simply says that the system changes ($\frac{d}{dt}\overrightarrow{y}(t)$) in response to its current condition ($\overrightarrow{f}(t, \overrightarrow{y}(t))$). This vector notation is shorthand for

$$\left\{\begin{array}{c} y_1'(t) = f_1(t, y_1, y_2, \cdots, y_N), \\ y_2'(t) = f_2(t, y_1, y_2, \cdots, y_N), \\ \cdots \\ y_N'(t) = f_N(t, y_1, y_2, \cdots, y_N) \end{array}\right., \text{ for } t > 0,$$

and

$$\left\{\begin{array}{c} y_1(0) = known, \\ y_2(0) = known, \\ \cdots \\ y_N(0) = known \end{array}\right..$$

Systems of ODEs often occur from reduction of higher order ODEs and higher order ODEs often occur through Newton's laws. For example, a particle's position $s(t)$ satisfies (via $f = ma$) a second order IVP:

$$\begin{array}{rcl} s''(t) &=& g(t, s(t), s'(t)), t > 0, \\ s(0) &=& s_0 \text{ and } s'(0) = s_1. \end{array}$$

This (and any higher order ODE) can be reduced to a first order system as follows. Let

$$y_1(t) = s(t) \text{ and } y_2(t) = s'(t).$$

The first equation is $y_1'(t) = y_2(t)$ and the second equation is

$$y_2'(t) = s''(t) = g(t, s(t), s'(t)) = g(t, y_1(t), y_2(t)).$$

Thus, this is equivalent to the IVP for system

$$\begin{array}{ccc} y_1' = y_2 & & \overrightarrow{y}' = \overrightarrow{f}(t, \overrightarrow{y}) \\ & \Leftrightarrow & \\ y_2'(t) = g(t, y_1(t), y_2(t)) & & \overrightarrow{y}(0) = (s_0, s_1)^T \end{array}$$

where $\overrightarrow{y} = (y_1, y_2)^T$ and

$$\overrightarrow{f}(t, \overrightarrow{y}) = \overrightarrow{f}(t, y_1, y_2) = \begin{bmatrix} y_2 \\ g(t, y_1(t), y_2(t)) \end{bmatrix}.$$

This illustrates that *any higher order IVP can be written as an IVP for a first order system.* Thus, *methods for solving IVPs for first order systems can be used to solve any IVP.*

Example 1 (Pendulum). *The pendulum equation*

$$\theta'' + \alpha\theta' + \frac{g}{L}\sin\theta = 0$$

naturally takes the form of a single second order equation. Here $\theta(t)$ is displacement from vertical, g is the constant of gravitational acceleration, L the pendulum length and α an air resistance parameter.

Example 2 (Particle paths). *The path $\overrightarrow{x}(t)$ taken by a particle in a flow starting at $\overrightarrow{x}(0)$ and with particle velocity $\overrightarrow{v}$ is known to satisfy the system*

$$\begin{aligned} \frac{d}{dt}\overrightarrow{v}(t) &= \frac{C_d}{St}\left(\overrightarrow{u}(\overrightarrow{x}(t), t) - \overrightarrow{v}(t)\right), \\ \frac{d}{dt}\overrightarrow{x}(t) &= \overrightarrow{v}(t). \end{aligned}$$

Here $\overrightarrow{u}(\overrightarrow{x}, t)$ is the velocity of the fluid (which is found by a separate calculation) and C_d, St are coefficients (the Stokes drag coefficient and the Stokes number respectively). Eliminating $\overrightarrow{v}(t)$ gives the second order system

$$\frac{d^2}{dt^2}\overrightarrow{x}(t) = \frac{C_d}{St}\left(\overrightarrow{u}(\overrightarrow{x}(t), t) - \frac{d}{dt}\overrightarrow{x}(t)\right).$$

Here is an application where the natural form is a first order system of 6 *equations that is equivalent to a second order system of* 3 *equations (one for each x, y, z component of position).*

The problem is to predict the future state:

find to high accuracy and with minimal cost $\overrightarrow{y}(t)$ for $t > 0$.

The initial value problem can be **scalar** (one equation only), a **system** (i.e., more than one equation such as

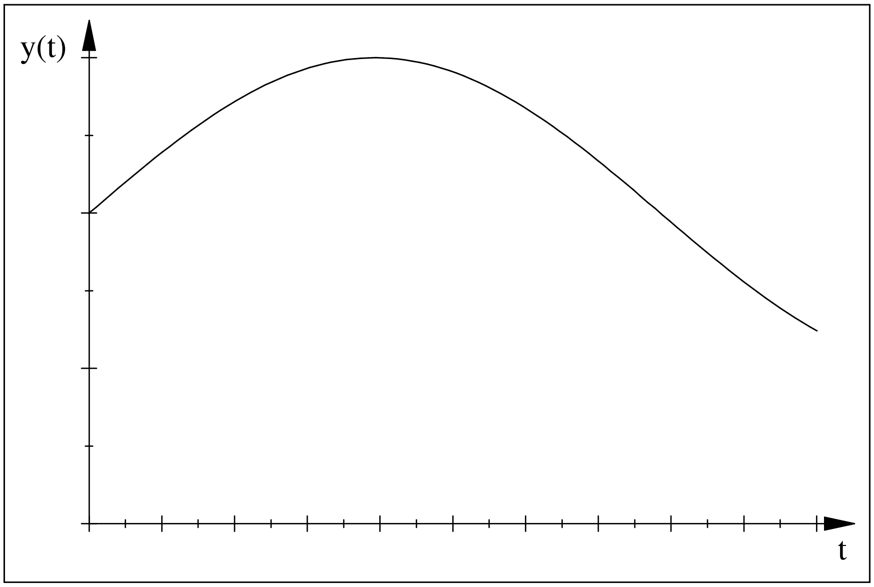
$$\begin{array}{lcl} x' = f(t, x, y) & & \text{for } \overrightarrow{y}(t) = (x(t), y(t))^T : \\ & \Leftrightarrow & \overrightarrow{y}'(t) = \overrightarrow{f}(t, \overrightarrow{y}(t)), \\ y' = g(t, x, y) & & \end{array}$$

first order ($y'(t) = f(t, y(t))$ as above), **second or even higher order** (such as the pendulum equation $\theta''(t) + \theta(t) = 0$), **linear, nonlinear** and so on. The types of initial value problems are as broad as the types of phenomena they describe.

We shall begin with the scalar case. Consider

$$\begin{aligned} y' &= f(t, y), t > 0, \\ y(0) &= y_0 \text{ (a known value)}. \end{aligned}$$

The solution is a curve beginning at the $y(0)$ value, e.g.,



The starting point for all numerical methods is Euler's method. It is given as follows. Select a small time step (denoted variously as $\triangle t$ or k or h) $k > 0$ (for example $k = 0.01$). Let $t_n = nk$ and let y_n be an approximation to $y(t_n)$. Euler's method is:

$$\text{Given } y_n \text{ find } y_{n+1} \text{ by solving}$$

$$\text{(Euler again)} \qquad \frac{y_{n+1} - y_n}{k} = f(t_n, y_n) \text{ , for } n = 0, 1, 2, \cdots.$$

Here "solving" is easy it means simply

$$y_{n+1} = y_n + kf(t_k, y_j) \ .$$

For this method and all methods the fundamental questions that need to be answered are:

- *Convergence: Does $y_n \to y(t_n)$ as $k \to 0$? Without convergence the above method is only a random number generator at best.*
- *Efficiency: Does $|y(t_n) - y_n| \to 0$ as fast as possible for the amount of work (time) required?*
- *Reliability: How do we calculate an approximation to $y(t_n)$ with a desired number of accurate significant digits (without knowing $y(t_n)$)?*

The ingredients we shall develop that answer these three meta questions for numerical ODEs are:

- *A theorem that states "stability + consistency => Convergence".*
- *A theory that evaluates stability for a method by a simple calculation.*
- *A theory that evaluates consistency for a method by another simple calculation.*
- *A algorithmic way to estimate local errors and adapt the time step to control local errors.*

EXERCISE 1. *The following, where $F = F(t), G = G(t)$, is a simplification of an equation arising in boundary layer theory. Write is an a first order system;*

$$\begin{aligned} F'' - F'G - F^2 + G^2 &= 0 \\ G'' - 2GF &= 0 \end{aligned}$$

1. Elements of the theory of Initial Value Problems

Consider the scalar IVP:

$$\frac{d}{dt}y(t) = f(t, y(t)) \text{ for } t > 0 \text{ and } y(0) = y_0 \text{ (a known vector).}$$

The first question that arises is whether a solution exists.

DEFINITION 1 (Lipschitz condition). *Let D be an open, connected set D in $\mathbb{R}^2$. $f(t, y)$ satisfies a **Lipschitz condition** on D if $f(t, x)$ is a continuous function on D and*

$$|f(t, y_1) - f(t, y_2)| \leq K|y_1 - y_2| \textit{ for all } y_1.y_2 \in D.$$

The mean value theorem implies

$$|f(t, y_1) - f(t, y_2)| \leq |\frac{\partial f}{\partial y}(t, \xi)||y_1 - y_2|, \text{ for some } \xi \text{ between } y_1, y_2.$$

Thus if $f(t, y)$ is C^1 it satisfies a Lipschitz condition with $K = max_D|\frac{\partial f}{\partial y}|$ if the latter is finite.

Concerning existence, Euler proved the following.

THEOREM 1 (Existence of solutions for an IVP). *Let $f(t, x)$ satisfy a Lipschitz condition on D, an open, connected set D in $\mathbb{R}^2$. Let $(0, y_0)$ be an interior point in D. Then, there is an $\varepsilon > 0$ such that a unique solution to the IVP exists for the time interval $-\varepsilon \leq t \leq +\varepsilon$.*

The idea Euler used in his proof was one of the first numerical methods for IVPs. He replaced $y'(t) = f(t, y(t))$ by a difference approximation:

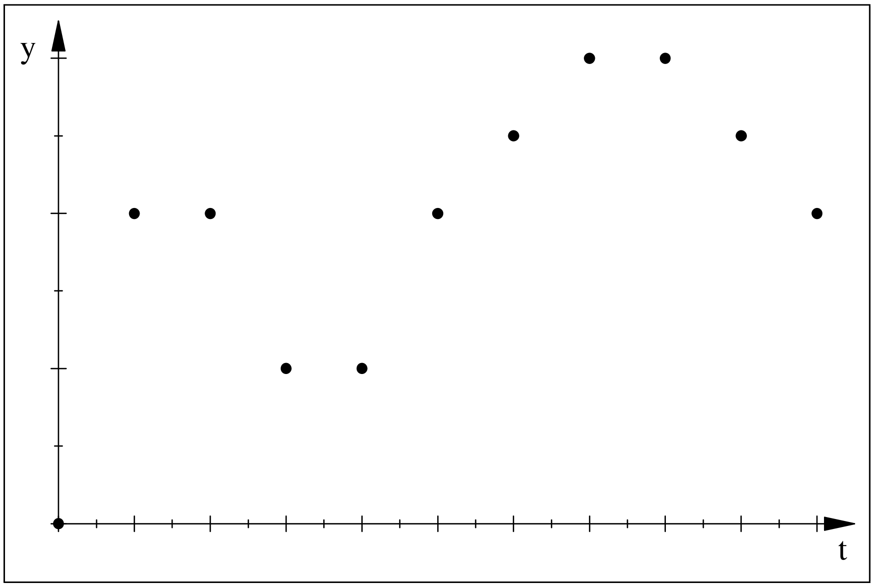
$$\frac{y_{j+1} - y_j}{\triangle t} = f(t_j, y_j) \text{ or, equivalently,}$$
$$y_{j+1} = y_j + \triangle t f(t_j, y_j) \text{ , for } j = 0, 1, 2, \cdots$$

where $\triangle t$ is the time step size, $t_j = j\triangle t$ is the j^{th} time and y_j is intended to be an approximation to the true $y(t)$ at $t = t_j$. Since y_0 is known, there is no problem of existence of these approximations. This gives a sequence of points

$$(t_j, y_j), \ j = 0, 1, 2, \cdots.$$

For example, the collection of points below

FIGURE 1



The dots are then connected to produce a continuous, piecewise linear curve:

This is then repeated for a sequence of timesteps $\triangle t \to 0$. Euler then showed that this sequence of curves converged and that the limit curve was a solution of the IVP. Euler's existence proof is constructive. The method of construction using $y_{j+1} = y_j + \triangle t f(t_j, y_j)$ is now known as "**Euler's method**".

The procedure Euler use was based on a truncated Taylor series. It will be important to note that since $y'(t) = f(t, y(t))$, all derivatives of y can be calculated by taking the total derivative of $f(t, y(t))$.

THEOREM 2 (Derivatives of the exact solution $y(t)$). *Let $y'(t) = f(t, y(t))$ and suppose $f(t, y)$ is smooth. Then higher derivatives of $y(t)$ are given by*

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y''(t) &= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t)) = f_t + f_y f \\ y'''(t) &= f_{tt} + 2 f_{ty} f + f_{yy} f + f_y f_t + f_y f_y f \end{aligned}$$

where all derivatives on the RHS are evaluated at $(t, y(t))$. Thus, for the true solution we have

$$y(t + \triangle t) = y(t) + \triangle t f(t, y(t)) + \frac{\triangle t^2}{2}\left[\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t))\right] +$$
$$\frac{\triangle t^3}{3!}\left[\begin{array}{c} f_{tt}(t, y(t)) + 2 f_{ty}(t, y(t)) f(t, y(t)) + \\ + f_{yy}(t, y(t)) f(t, y(t)) + f_y(t, y(t)) f_t(t, y(t)) + f_y^2(t, y(t)) f(t, y(t)) \end{array}\right]$$
$$+ O(\triangle t^4)$$

PROOF. As $y'(t) = f(t, y(t))$ we have $y''(t) = \frac{d}{dt} f(t, y(t))$. By the chain rule

$$\begin{aligned} y''(t) &= \frac{d}{dt} f(t, y(t)) = \\ &= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) y'(t) = \\ &= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t)) \end{aligned}$$

Similarly,

$$\begin{aligned} y'''(t) = \frac{d}{dt}\left(\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t))\right) = \\ = \frac{\partial^2 f}{\partial t^2}(t, y(t)) + \frac{\partial^2 f}{\partial t \partial y}(t, y(t)) y'(t) + \\ + \frac{\partial f}{\partial y}(t, y(t))\left(\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t))\right) + \\ + \left(\frac{\partial^2 f}{\partial t \partial y}(t, y(t)) + \frac{\partial^2 f}{\partial y^2}(t, y(t)) y'(t)\right) f(t, y(t)) = \\ = \frac{\partial^2 f}{\partial t^2}(t, y(t)) + \frac{\partial^2 f}{\partial t \partial y}(t, y(t)) f(t, y(t)) + \\ + \frac{\partial f}{\partial y}(t, y(t))\left(\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t))\right) \\ + \left(\frac{\partial^2 f}{\partial t \partial y}(t, y(t)) + \frac{\partial^2 f}{\partial y^2}(t, y(t)) f(t, y(t))\right) f(t, y(t)) \end{aligned}$$

Collecting terms gives the formula for $y'''(t)$. The formula $y(t + \triangle t) = y(t) + \cdots$ is simply Taylor's theorem using the formula for derivatives. □

Remark 1 (Cost of evaluating derivatives). *For the case of a system of equations the same formula holds (suitable interpreted). However, the scale of the computation changes dramatically. For example, evaluating* $\frac{\partial f}{\partial y}$ *for a scalar function is one function evaluation. For a system of* N *equations,* $\frac{\partial f}{\partial y}$ *is the* $N \times N$ *matrix*

$$\left(\frac{\partial f}{\partial y}\right)_{i,j\ entry} = \frac{\partial f_i}{\partial y_j}, i = 1, \cdots, N, j = 1, \cdots, N.$$

This requires N^2 *function evaluations. Similarly* $\frac{\partial^2 f}{\partial y^2}$ *requires* N^3 *function evaluations as it represents*

$$\frac{\partial^2 f_i}{\partial y_j \partial y_k}, i = 1, \cdots, N, j = 1, \cdots, N, k = 1, \cdots, N.$$

Exercise 2. *Find a formula for* $y''''(t)$*. For a system of* N *equations, how many function evaluations does it require to evaluate.*

1.1. The discoverers.

adapted from the Wikipedia article:

Rudolf Otto Sigismund Lipschitz (1832 – 1903) was a German mathematician who made contributions to mathematical analysis where he gave his name to the Lipschitz continuity condition.

2. Some test problems

> Too many people write papers that are very abstract and at the end they may give some examples. It should be the other way around. You should start with understanding the interesting examples and build up to explain what the general phenomena are. This was your progress from initial understanding to more understanding. This is both for expository purposes and for mathematical purposes. It should always guide you. When someone tells me a general theorem I say that I want an example that is both simple and significant. It's very easy to give simple examples that are not very interesting or interesting examples that are very difficult. If there isn't a simple, interesting case, forget it. -Sir Michael Atiyah

We record here some test problems[3] that are commonly used to compare one method against another. A good test problem makes many methods fail by a criteria that is non-arguable. Thus, solutions of a good test problem should have some easily observed qualitative features that are not so easily replicated under discretization. Some of the test problems are presented without exxplanation. These are ones that I have no experience with.

2.1. "The" test problem.

The simple IVP

$$\begin{aligned} y' &= \lambda y, t > 0 \\ y(0) &= 1, \\ solution &: y(t) = e^{\lambda t} \end{aligned}$$

has the property that

$$y(t) \to 0 \text{ as } t \to \infty \text{ provided } \operatorname{Re}(\lambda) < 0.$$

[3] There are several repositories on the web that collect test problems in a standard format, such as https://archimede.dm.uniba.it/~testset/testsetivpsolvers/.

Thus, one common test for a numerical method is whether computed solutions share this property:

$$\textit{Does } y_n \to 0 \textit{ as } n \to \infty \textit{ when } \mathrm{Re}(\lambda) < 0\textit{?}$$

If so then the approximate solution shares *asymptotic stability* with the true solution.

Also, if $Re(\lambda)$ >0 the solution grows no faster than exponential

$$|y(t)| \leq e^{\alpha t}, \alpha = \mathrm{Re}(\lambda)$$

so an approximate solution should as well. A method whose approximate solution has this property is called *0-stable.* There are many variations on what is meant by stability and each can be tested to see which methods satisfy the discrete version of the specific form of stability.

This becomes a much more interesting test if $\lambda = \lambda(t)$ as some methods are stable for constant but unstable for variable for some patterns of variability.

2.2. A special case of a Ricatti equation.

The test problem is

$$\begin{aligned} y' &= -\frac{1}{2}y^3, t > 0 \\ y(0) &= 1, \\ \textit{solution} &: \; y(t) = (t+1)^{-1/2}. \end{aligned}$$

This is a nonlinear analog of "the" test problem where

$$\lambda = -\frac{1}{2}y^2.$$

It is always interesting to see how much can be inferred from linear problems to nonlinear problems.

2.3. An oscillatory test problem.

Many physical problems have solutions that conserve some total energy. The simplest example of such a problem is the linear pendulum:

$$\theta'' + \omega^2\theta = 0, t > 0, \text{ where } \omega \text{ is real,}$$
$$\theta(0), \theta'(0) \text{ both specified.}$$

It is easy to check that

$$\frac{d}{dt}\left[\theta'(t)^2 + \omega^2\theta(t)^2\right] = 0$$

so that the energy $E(t) = \theta(t)^2 + \omega^2(\theta'(t))^2$ is exactly conserved:

$$\theta'(t)^2 + \omega^2\theta(t)^2 = \theta'(0)^2 + \omega^2\theta(0)^2 \text{ for all } t.$$

Written as a first order system in the usual way $(x(t) = \theta(t), y(t) = \theta'(t))$ this gives

$$x' = y, y' = -\omega^2 x$$

or

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}.$$

The eigenvalues of the above 2×2 matrix are easily found to be $\pm\omega i$ ($i = \sqrt{-1}$) so that the above system is equivalent to test problem 1 with $\lambda = \pm\omega i$.

This test problem is interesting because of its exact conservation properties. A perfect numerical solution to the above system should satisfy

$$y_n^2 + \omega^2 x_n^2 = y_0^2 + \omega^2 x_0^2 \text{ for all } n > 0.$$

Thus *any significant growth or decay is a numerical artifact.*

The system is also interesting because it reveals something about timesteps. If the initial data is $x(0) = 1, y(0) = 0$, the exact solution is

$$x(t) = \cos(\omega t), y(t) = \sin(\omega t).$$

For any p^{th} order method, we shall see that the local truncation error takes the form

$$\begin{aligned} LTE &= C\triangle t^{p+1} \frac{d^{p+1}}{dt^{p+1}} (\cos(\omega t), \sin(\omega t)) \text{ so that} \\ |LTE| &= C\left(|\omega|\triangle t\right)^{p+1}. \end{aligned}$$

Obviously, for ω large, $\triangle t$ must be small enough that

$$|\omega|\triangle t < 1$$

to hope for even a single digit of accuracy, regardless of the order of the method or its stability properties. This condition is often interpreted as saying:

$$\text{wave speed} \times \text{time step} < 1.$$

Amplitude and Phase Errors. Consider now Euler's method for

$$y' = i\omega y, y(0) = 1.$$

One step of the true solution and one step of Euler's method give respectively

$$\begin{array}{rc} \text{True soln:} & y(t_{n+1}) = e^{i\omega\triangle t} y(t_n) \\ \text{Euler Approx.:} & y_{n+1} = R y_n \end{array}$$

where

$$R = 1 + i\omega\triangle t.$$

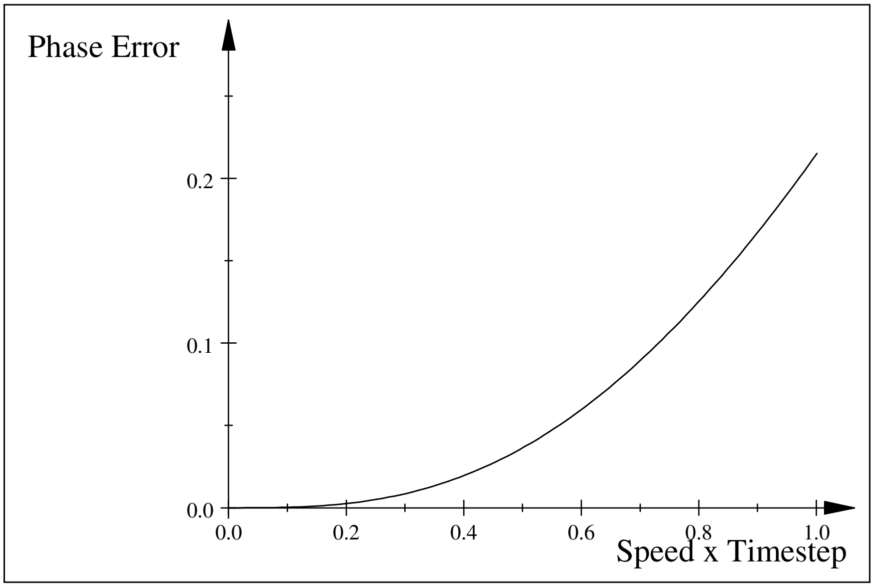
Decompose R as amplitude and phase by

$$\begin{aligned} R &= |R| e^{i\theta}, \\ |R| &= \sqrt{1 + (\omega\triangle t)^2}, \\ \tan\theta &= \omega\triangle t. \end{aligned}$$

The amplitude and phase error of 1 step of Euler's method are, respectively,

$$\begin{array}{rc} \text{Amplitude Error:=} & 1 - \sqrt{1 + (\omega\triangle t)^2} \\ \text{Phase Error:=} & \omega\triangle t - \arctan(\omega\triangle t). \end{array}$$

Note that $|R| > 1$ so Euler's method is asymptotically unstable. Since the interesting region to study the phase error is $|\omega|\triangle t < 1$ one can simply plot it:



Phase Error of Eulers Method

From this plot and Taylor's theorem numerous interesting properties can be inferred:

When the phase error is positive the method errs by accelerating waves/oscillations and when negative by slowing down waves.

One can also develop *methods with zero average phase error by taking combinations of methods with positive and negative phase errors*, an idea of Fromm. The phase error is also related to accuracy: we have the following.

PROPOSITION 1. *For Euler's method, over* $|\omega\triangle t| < 1$

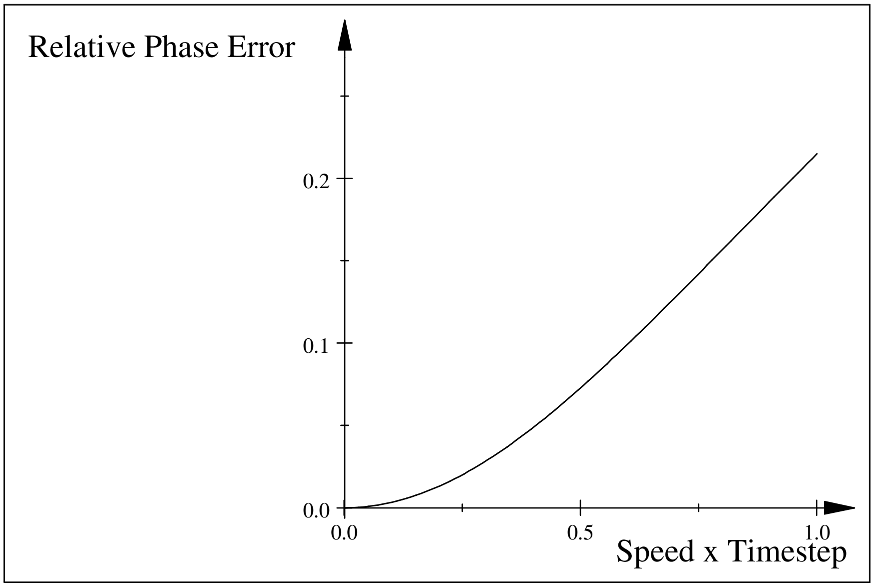
$$\begin{aligned} |Amplitude\_Error| &\leq C|\omega\triangle t|^2, \\ |Phase\_Error| &\leq C|\omega\triangle t|^3. \end{aligned}$$

The accuracy of a method is related to the order of contact of the phase error with the horizontal axis at the origin. Since the phase error is small there it is natural to study instead the relative phase error.

The relative phase error of Euler's method is

$$\text{Relative Phase Error:=} \quad \frac{\omega\triangle t - \arctan(\omega\triangle t)}{\omega\triangle t} = 1 - \frac{\arctan(\omega\triangle t)}{\omega\triangle t}.$$

The relative phase error of Euler's method is plotted below.



Relative Phase Error of Eulers Method

It is interesting to see what this means in as simple an example as possible. The following[4] is a plot of the Euler solution of pendulum equation with $\triangle t = 1/16$ and the true solution. In the plot it is clear that the error in the computer period is very small while the error in the amplitude is significant (and would take a much smaller time step to get a good answer). This is consistent with Euler's method having higher order phase accuracy than amplitude error.

Exercise 3. *Strong stability preserving methods take weighted averages of Euler steps with positive weights. The weight is used to increase accuracy. If the weights are positive then the averages preserve positivity if the Euler step preserves positivity. This is one new approach to higher accuracy plus positivity preservation. One example is as follows: given* y_n

$$\begin{aligned} y^1_{n+1} &= y_n + \triangle t f(t_n, y_n), \qquad \text{(Method 1)} \\ y^2_{n+1} &= y^1_n + \triangle t f(t_n + \triangle t, y^1_{n+1}) \\ y_{n+1} &= \frac{1}{2} y^1_{n+1} + \frac{1}{2} y^2_{n+1} \end{aligned}$$

NOTE: superscript 1 and 2 are NOT exponents. Apply this to the oscillatory test problem. Plot the amplitude error, phase error and relative phase error. Analyze all three: what is their order as $\triangle t\omega \to 0$*? What is the average phase error? Compare the above results to Euler's method. Consider the following method with more general weighted (weighting parameter =* θ*) averages*

$$\begin{aligned} y^1_{n+1} &= y_n + \triangle t f(t_n, y_n), \qquad \text{(Method 2)} \\ y^2_{n+1} &= y_n + \triangle t f(t_n + \frac{1}{2\theta} \triangle t, y_n + \frac{1}{2\theta} \triangle t f(t_n, y_n)) \\ y_{n+1} &= (1-\theta) y^1_{n+1} + \theta y^2_{n+1}. \end{aligned}$$

[4] Provided by Joseph Fiordilino.

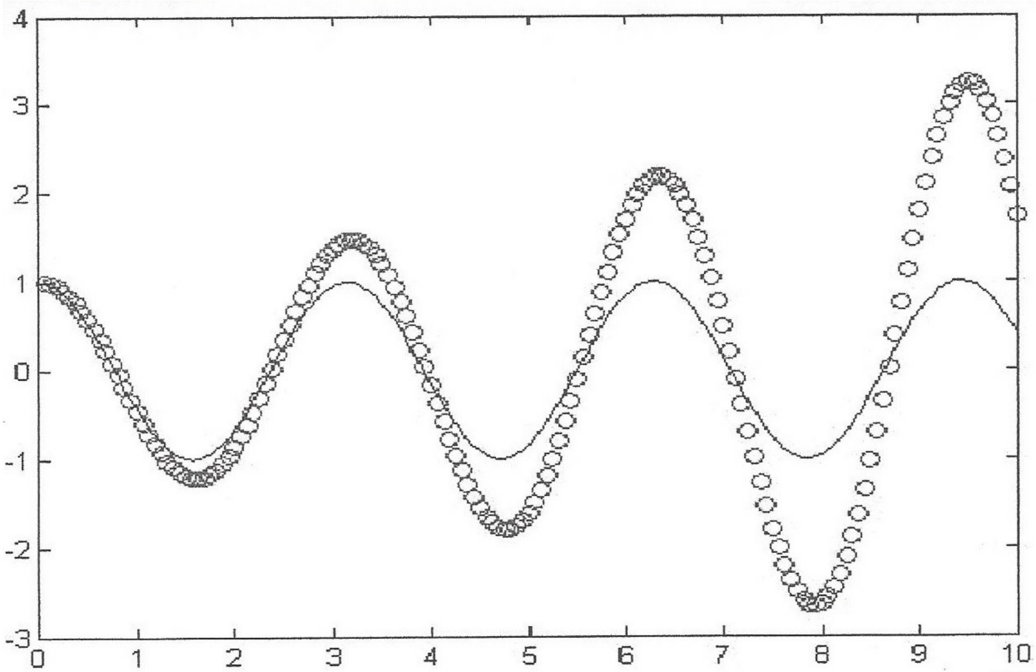


FIGURE 2. True solution & Euler approximation, $\triangle t = 1/16$

Repeat the analysis of problem 1 for this method. Here the phase error will be a function of 2 variables $x = \triangle t\omega$ and $y = \theta$ so the plots will be a surface. Take advantage of the fact that the weight may be chosen to ***derive a method with minimum phase error****. Pick one or more test problems with periodic solutions and a moderate timestep. Compare Euler, Method 1 and Method 2 over many periods. Draw conclusions. [NOTE: Your goal is NOT to make all the methods work but to make some work and some fail. The "Test problem", the mode of "Failure" and size of "moderate" will be up to you to choose. The choice will require some computational explorations.]*

2.4. The discoverers. J.E. Fromm was a researcher at IBM Research Laboratory. One example of this work

Fromm, J. E., Practical Investigation of Convective Difference Approximations of Reduced Dispersion, Physics of Fluids, 12, II-3-II-12 (1969), DOI:http://dx.doi.org/10.1063/1.1692465

2.5. Another oscillatory test problem. The previous problem becomes more ineresting when non-autonomous:

$$\theta'' + \omega(t)^2\theta = 0, t > 0, \text{ where } \omega(t) \text{ is real,}$$
$$\theta(0), \theta'(0) \text{ both specified.}$$

It is easy to check that

$$\frac{1}{2}\frac{d}{dt}\left[\theta'(t)^2+\omega(t)^2\theta(t)^2\right]-[\omega(t)\omega'(t)]\theta(t)^2=0$$

so that the energy $E(t)=\theta(t)^2+\omega^2(\theta'(t))^2$ is no longer exactly conserved:

$$\theta'(t)^2+\omega^2\theta(t)^2-2\int_0^t[\omega(s)\omega'(s)]\theta(s)^2ds=\theta'(0)^2+\omega^2\theta(0)^2 \text{ for all } t.$$

2.6. Yet another switching growth to decay test problem. The test problem is

$$\begin{aligned} y' &= \cos(t)y, t>0 \\ y(0) &= 1. \end{aligned}$$

This is a nonautonomous analog of "the" test problem where

$$\lambda=-\frac{1}{2}y^2.$$

It is always interesting to see how much can be inferred from autonomous problems to nonautonomous problems.

To be very direct. In such a test the goal is never to show that what holds for the simple case also holds for the complicated case. *The goal is always to find cases when what holds in the simple case fails in the complicated case and then to explain why and suggest what to do next.*

Another, 2D, version of this problem. If we let $x(t)=t$ then $x'=1$. Thus the following is mathematically equivalent but can produce different numerical results:

$$\begin{aligned} y' &= \cos(x)y, \\ x' &= 1 \\ y(0) &= 1, x(0)=0. \end{aligned}$$

Yet another, 3D, version of this problem. The following is mathematically equivalent but can produce different numerical results:

$$\begin{aligned} y' &= xy, \\ x' &= -z, \\ z' &= +x, \\ y(0) &= 1, x(0)=1, z(0)=0. \end{aligned}$$

2.7. The $\lambda-\omega$ System. This is an excellent test problem whose solution behavior changes depending on choice of the functions $\lambda(t),\omega(t)$:

$$\begin{bmatrix} x' \\ \\ y' \end{bmatrix} = \begin{bmatrix} \lambda(t) & +\omega(t) \\ \\ -\omega(t) & \lambda(t) \end{bmatrix}\begin{bmatrix} x \\ \\ y \end{bmatrix}$$
$$x(0), y(0) \text{ specified.}$$

Its solution can combine growth, decay and oscillation.

2.8. The Logistic or S-curve Model. The shape of the solution suggests small timesteps should be taken initially and then the timesteps should keep getting larger as the curve gets flatter (as t -> infinity):

$$\begin{aligned} y' &= \frac{1}{4}y(1-\frac{1}{20}y), t>0, \\ y(0) &= 1. \end{aligned}$$

2.9. A Linear (simplified) Chemical Reaction Model. The system is

$$\begin{aligned} x' &= -x+y, x(0)=2 \\ y' &= x-2y+z, y(0)=0, \\ z' &= y-z, z(0)=1. \end{aligned}$$

2.10. A Radioactive Decay Chain. Let A denote the $10x10$ matrix:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and let $y(t)$ denote a 10-vector. The radioactive decay chain test problem is

$$\begin{aligned} y' &= Ay, \\ y(0) &= [1,0,0,0,0,0,0,0,0,0]^T. \end{aligned}$$

2.11. Another Radioactive Decay Chain. Let A denote the $10x10$ matrix:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & -4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & -5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & -6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & -7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & -8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & -9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 \end{bmatrix}$$

and let $y(t)$ denote a 10-vector. The radioactive decay chain test problem is

$$\begin{aligned} y' &= Ay, \\ y(0) &= [1,0,0,0,0,0,0,0,0,0]^T. \end{aligned}$$

2.12. The Lorentz equations. The Lorentz[5] system have solutions that are contained in a bounded region but where small errors in initial conditions grow exponentially until "saturation" (until $\mathcal{O}(1)$ apart). Thus this system is very sensitive. The first and most famous of many Lorentz systems is

$$\begin{aligned} x' &= \sigma(y-x), \\ y' &= rx - y - xz, \\ z' &= xy - bx. \end{aligned}$$

Lorentz chose the initial conditions and parameters:

$$\begin{aligned} x(0) &= 0, y(0) = 1, z(0) = 0, \\ \sigma &= 10, b = 8/3, r = 28. \end{aligned}$$

Solving this system and plotting its shadow in the $x-z$ plane[6] produced the famous butterfly plot. Typically a method is used to solve the Lorentz system for constant time steps and the solution compared with a self-adaptive simulation to estimate the error in the first.

2.13. Robertson's equations. The following is a simplified chemical reaction system used[7] to test methods failure when solutions approach equilibrium very rapidly

$$\begin{aligned} x' &= -\alpha x + \beta yz, \\ y' &= \alpha x - \beta yz - \gamma y^2, \\ z' &= +\gamma y^2. \end{aligned}$$

Standard values taken in tests are

$$x(0) = 1,\ y(0) = 0,\ z(0) = 0,$$
$$\alpha = 0.04,\ \beta = 10^4,\ \gamma = 3 \times 10^7.$$

2.14. The van der Pol oscillator.

> The purpose of computing is insight, not numbers. [Richard Hamming]

The equation for the van der Pol oscillator is

$$\theta'' + 10\theta'(1-\theta) + \theta = 0, t > 0,$$
$$\theta(0) = 1, \quad \theta'(0) = 5.$$

[5] adapted from the Wikipedia article: Edward Norton Lorenz (1917 – 2008) was an American mathematician, meteorologist, and pioneer of chaos theory. His 1963 paper "Deterministic Nonperiodic Flow" states:

"Two states differing by imperceptible amounts may eventually evolve into two considerably different states ... If, then, there is any error whatever in observing the present state — and in any real system such errors seem inevitable — an acceptable prediction of an instantaneous state in the distant future may well be impossible....In view of the inevitable inaccuracy and incompleteness of weather observations, precise very-long-range forecasting would seem to be nonexistent."

[6] The solution curve parameterized by time is $(x(t), y(t), z(t))$, a curve in 3 dimensions. Its shadow in the $x-z$ plane can be easily plotted and is the planar curve $(x(t), z(t))$.

[7] U. Asher and L. Petzold, Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations, SIAM, Philadelphia, 1998.

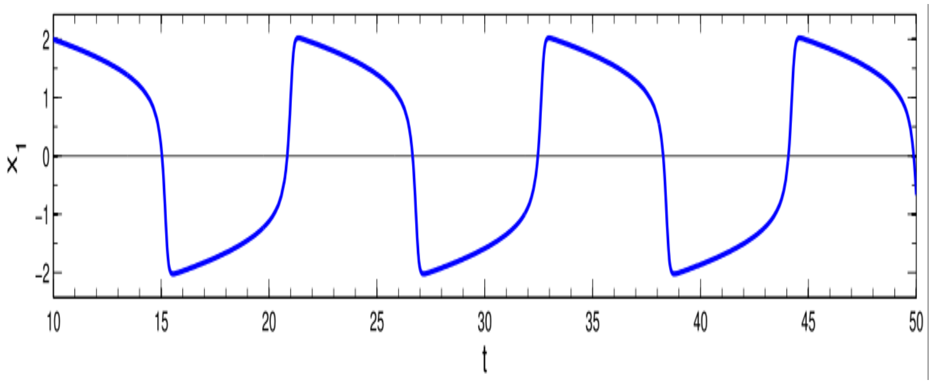


FIGURE 3. The van der Pol relaxation oscillation

This equation undergoes relaxation oscillations to a stable periodic solution. The solution has a pattern that resembles the teeth on a saw for $\theta(t)$ and a series of large spikes for $\theta'(t)$:[8]

This test problem is interesting in that without adaptivity, most methods require a small time step to capture the solution. Thus, methods can be compared based on how small the time step must be to give a faithful approximation. The van der Pol equation with forcing is also an interesting test problem:

$$\theta'' + (\theta^2 - 1)\theta' + \theta = 1.3\cos(0.2t),$$
$$\theta(0) = 1, \quad \theta'(0) = 5.$$

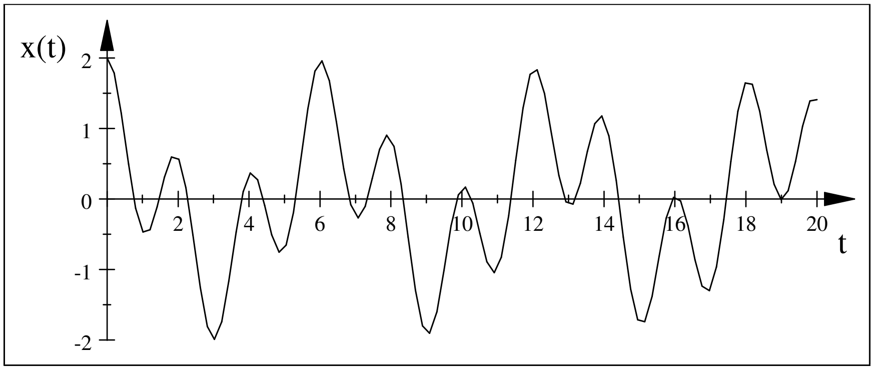
2.15. A Quasi-periodic oscillation problem. The is a simple test problem without complicated solutions or sharp fronts. The issue here for constant timestep methods is that for too lartge timesteps weird solutions result while for small enouh timesteps solutions have reasonable accuracy. Thus: How to select the timestep?

Solve the IVP below written as a first order system

$$x'''' + (\pi^2 + 1)x'' + \pi^2 x = 0, 0 < t < 20,$$
$$x(0) = 2, x'(0) = 0, x''(0) = -(1 + \pi^2), x'''(0) = 0.$$

This has exact solution $x(t) = cos(t) + cos(\pi t)$, the sum of two periodic functions with incommensurable periods, quasi-periodic. Start with timestep $k = 0.1$, tolerance $TOL = 0.1$.

[8] This figure is from https://en.wikipedia.org/wiki/File:Vanderpol_mu%3D5.svg

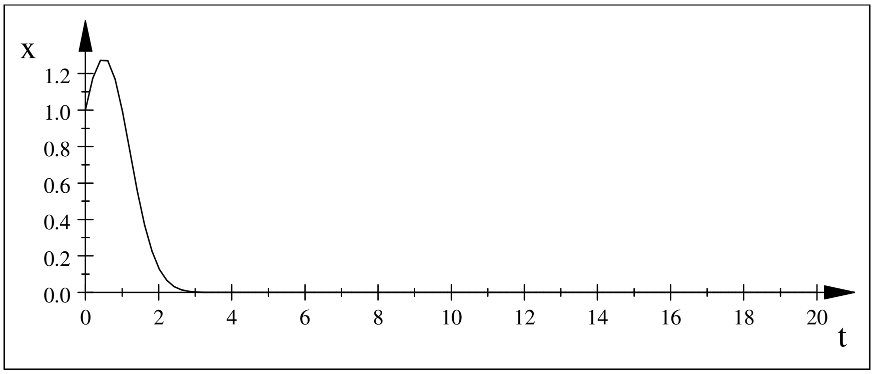


The solution

2.16. A problem with increasing stiffness. Solve over $0 < t < 20$

$$x' = (1 - 2t)x, x(0) = 1.$$

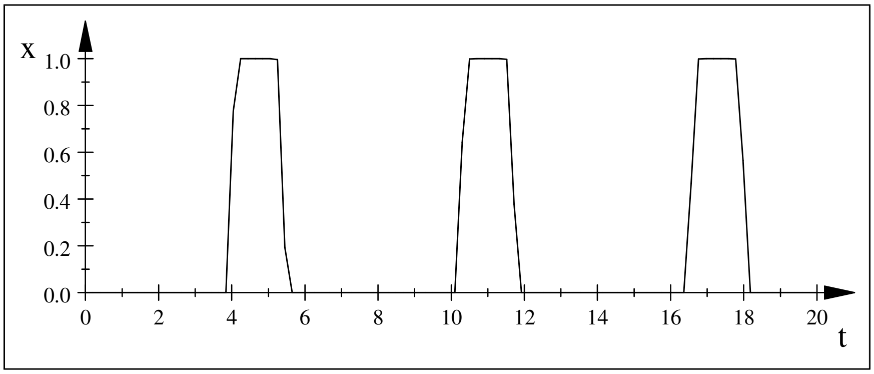
Take $TOL = 0.001$. It is useful to plot the solution $x(t) = exp\left(t - t^2\right)$



The solution

.

2.17. Sharp transition regions. Take $f(t) = \exp\left(-\left(4.0 + 4.0\sin(x)\right)^{10}\right)$



Function f(t)

Solve

$$x' = \lambda x + f(t), x(0) = 1, 0 < t < 20, \lambda = -1 \;\&\; \lambda = -1000.$$

2.18. A problem with an unstable limit cycle. This problem is almost impossible to solve correctly without adaptivity except for some odd cases where the discrere eequations have their own exact limit cycle. Solve $x(0) = 1, y(0) = 0, 0 < t < 20$

$$\begin{aligned} x' &= -x - y + x\sqrt{x^2+y^2},\ 0 < t < 20,\\ y' &= -y + x + y\sqrt{x^2+y^2}, 0 < t < 20,\\ x(0) &= 1,\\ y(0) &= 0. \end{aligned}$$

This has true solution

$$x(t) = cos(t), y(t) = sin(t)$$

that goes around and around the unit circle. Any perturbation from any source makes the solution diverge quickly from this exact solution.

2.19. The Kepler orbit equations. This is a system of 4 equations describing an orbit with eccentriocity e:

$$\begin{array}{rcr} y_1' = y_3 & , & y_1(0) = 1 - e\\ y_2' = y_4 & , & y_4(0) = 0\\ y_3' = -\frac{y_1}{\left(y_1^2+y_2^2\right)^{3/2}} & , & y_3(0) = 0\\ y_4' = -\frac{y_2}{\left(y_1^2+y_2^2\right)^{3/2}} & , & y_4(0) = \sqrt{\frac{1+e}{1-e}} \end{array}$$

2.20. The Brusselator. The Brusselator is a system of 2 equations proposed as a theoretical model for an autocatalytic reaction by (Nobel laureate) Ilya Prigogine:

$$\begin{aligned} y_1' &= A + y_1^2 y_2 - (B+1)y_1, y_1(0) = 1.5\\ y_2' &= By_1 - y_1^2 y_2, y_2(0) = 3. \end{aligned}$$

This has an equilibrium at $(A, B/A)$ which uis unstable for $B > 1 + A^3$. Typical choices are $A = 1, B = 3$ for which a stable limit cycle emerges.

2.20.1. *The discoverers.* Adapted from the Wikipedia article:

Balthasar van der Pol (1889 – 1959) was a Dutch physicist. He studied physics in Utrecht, and in 1920 he was awarded his doctorate. His main interests were in radio wave propagation, theory of electrical circuits, and mathematical physics. The van der Pol oscillator, one of the most widely used models of nonlinear self-oscillation, is named after him. Van der Pol became member of the Royal Netherlands Academy of Arts and Sciences in 1949.

2.21. Transport. Oscillations of a pendulum are not a compelling or high impact application (possibly aside from clock makers). However, the standard test problem for transport (when something is moved around by a liquid or gas) is

$$y' = \pm i\omega y, \omega \text{ a real number.}$$

To see why we briefly consider the simplest transport problem: for $u(x, t)$ a concentration of something that is moves to the right with speed $a > 0, u(x, t)$ satisfies the partial differential equation

$$\begin{aligned} \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} &= 0, -\infty < x < \infty, t > 0,\\ u(x, 0) &= f(x) \text{ , the concentration initially.} \end{aligned}$$

It is easy to check by direct substitution that the exact solution is

$$u(x,t) = f(x-at)$$

which is the profile $f(x)$ moving to the right with speed a. The simplest case is when $f(x)$ is one Fourier mode such as $f(x) = cos(nx) + sin(nx)$ and, as usual we shall do the calculation with $f(x) = e^{inx}$ because it is easier. Then write

$$u(x,t) = y(t)e^{inx}$$

substitute into $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$ and cancel gives

$$\frac{\partial}{\partial t}(y(t)e^{inx}) + a\frac{\partial}{\partial x}(y(t)e^{inx}) = 0 \Leftrightarrow$$
$$y'(t)e^{inx} + ay(t)ine^{inx} = 0 \Leftrightarrow$$
$$y'(t) = -i(na)y$$
$$\text{so}$$
$$\omega = na.$$

Similarly, if the transport is to the left we get $y'(t) = +i(an)y$. In all cases, faster transport speed (larger a) means larger $\omega = na$ in the test problem $y' = \pm i\omega y$.

2.22. A Second order IVP. The following second IVP seems inoffensive

$$y'' + 1001y' + 1000y = 0, t > 0$$
$$y(0) = 1 \text{ and } y'(0) = -1.$$

However, it has solution (which can be found by standard methods)

$$y(t) = C_1e^{-t} + C_2e^{-1000t},$$

where $C_{1,2}$ are determined by the initial conditions[9]. If we write the second order IVP as one for a first order system in the usual way ($y_1 = y$, $y_2 = y'$ etc.) we get

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1000 & -1001 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

The eigenvalues of the above 2×2 matrix are easily found to be $\lambda = -1$ & -1000. The stability region of RK4 shows that if this system is approximated by RK4, it will converge nicely if $\triangle t < 0.002$ but the approximate solution will blow up exponentially if $\triangle t \geq 0.003$.

This solution exhibits rate constants $\lambda = -1, -1000$ which begins to be stiff. Thus the effect of the e^{-1000t} mode dies out very fast and the solution looks like e^{-t}. Unfortunately, look what happens with Euler's method for such a problem.

2.23. Conduction / Diffusion. The IVP for heat conduction in a bar in its simplest form is a partial differential equation for the temperature $u(x,t)$ at the point x at time t. The initial temperature $u(x,0)$ and the temperature at both ends $u(0,t)$ and $u(1,t)$ are known and the internal temperature satisfies

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \text{ for } 0 < x < 1, t > 0.$$

[9] $y(0) = 1$ gives $C_1 + C_2 = 1$ and $y'(0) = -1$ gives $C_1(-1) + C_2(-1000) = -1$. This is a 2 by 2 linear system for C_1, C_2.

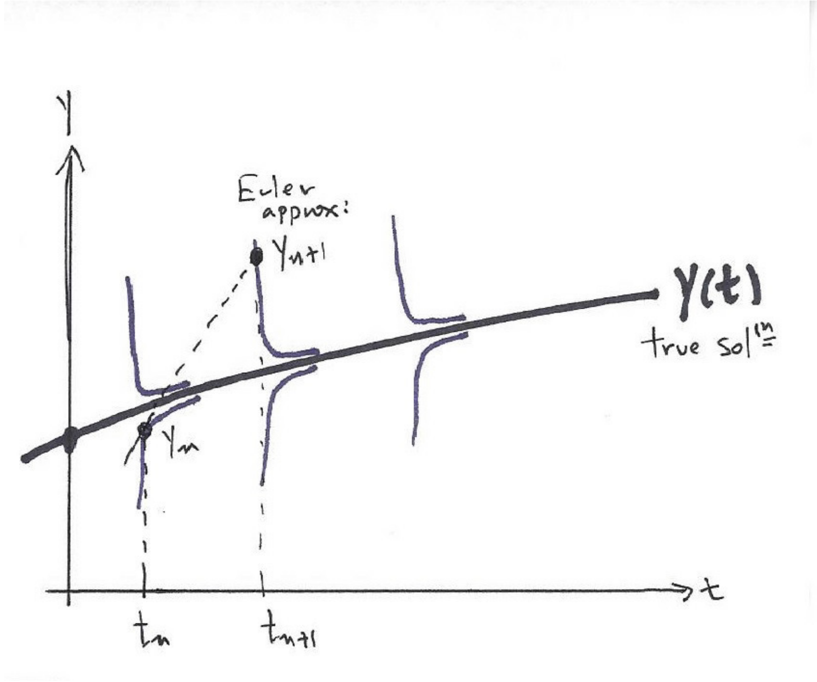


Figure 4. The problem of stiffness

To predict the temperature it is converted into an IVP for system of ODEs as follows. Pick a space mesh width $\triangle x = 1/(N+1)$ and let

$$x_j = j\triangle x \text{ and } u_j(t) = \text{ approximation to } u(x_j, t).$$

We approximate

$$\frac{\partial^2 u}{\partial x^2}(x_j, t) \simeq \frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{\triangle x^2} \text{ (which has error } O(\triangle x^2)).$$

We then have the system of equations for $u_j(t)$

$$\begin{aligned} u_1' &= \frac{-2u_1 + u_2}{\triangle x^2} \\ u_2' &= \frac{+u_1 - 2u_2 + u_3}{\triangle x^2} \\ &\quad \cdots\cdot \\ u_{N-1}' &= \frac{+u_{N-2} - 2u_{N-1} + u_N}{\triangle x^2} \\ u_N' &= \frac{+u_{N-1} - 2u_N}{\triangle x^2}. \end{aligned}$$

This is written in matrix form as

$$\frac{d}{dt}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \frac{1}{\triangle x^2}\begin{bmatrix} -2 & +1 & & \\ +1 & -2 & +1 & \\ & \searrow & \searrow & \searrow \\ & & +1 & +2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}.$$

The above matrix is denoted $tridiag(+1,-2,+1)$. We shall see that for this problem, RK2 is stable if and only if the timestep is very small $\triangle t \leq Const.\triangle x^2$, an issue related to *stiffness*.

2.24. A problem with some aspects of Fluid Flow. This is an example, capturing some aspects of transition to turbulence, from the thesis of Lionel Walker. For $R > 0$ large, $0 < t < 500$, $\delta > 0$ take initial condition

$$(x(0), y(0)) = \frac{\delta}{\sqrt{2}}(1,1)$$

The equation is

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} &= \begin{pmatrix} R^{-1} & 1 \\ 0 & R^{-1} \end{pmatrix} \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \\ &+\sqrt{x(t)^2+y(t)^2}\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix}. \end{aligned}$$

The solution is visualized by plotting the phase plane $(x(t), y(t))$.

2.25. A test problem of Orszag related to turbulence. In his notes "*Statistical Theory of Turbulence*" S. Orszag suggested the following system of 5 ODEs as an interesting proxy of certain features of turbulence (at infinite Reynolds number):

$$\begin{aligned} \frac{d}{dt}x_i(t) &= x_{i+1}x_{i+2} + x_{i-1}x_{i-2} - 2x_{i+1}x_{i-1} \text{ for } t > 0 \text{ and} \\ x_i &= x_{i+5} \text{ for all i.} \end{aligned}$$

It is easy to verify that the solution satisfies

$$\frac{d}{dt}\sum_{i=1}^{5} x_i^2(t) = 0 \text{ for } t > 0.$$

He chose the initial conditions below and plotted the 2d shadow traced by $(x_1(t), x_2(t))$:

$$\begin{aligned} x_1(0) &= 0.540323 \\ x_2(0) &= -1.543569 \\ x_3(0) &= -0.680421 \\ x_4(0) &= -1.185361 \\ x_5(0) &= -0.676307. \end{aligned}$$

3. Stability of Initial Value Problems

There are many different stability concepts for IVPs. We give only a few. Local stability means simply continuous dependence.

Definition 2 (Local Stability / Continuous dependence). *The solution $y(t)$ of the IVP is **locally stable** if there is an $\varepsilon > 0$ such that for $||\delta(x)|| < \varepsilon$ and $|\widetilde{y}_0| < \varepsilon$ the solution $x(t)$ of the perturbed IVP*

$$\begin{aligned} \frac{d}{dt}x(t) &= f(t, x(t)) + \delta(x) \textit{ for } t > 0 \textit{ and} \\ x(0) &= y_0 + \widetilde{y}_0 \textit{ (a known vector).} \end{aligned}$$

satisfies

$$\max_{0\le t\le T} |x(t)-y(t)| \le C(T)\varepsilon.$$

EXERCISE 4. *Show that of $f(t,y)$ is Lipschitz continuous then the IVP is locally stable.*

Asymptotic stability describes long time behavior with perturbed initial conditions. For an asymptotically stable system, solutions squeezer together as $t \to \infty$.

DEFINITION 3 (Asymptotic Stability). *Let $x(t), y(t)$ satisfy*

$$\begin{aligned} \frac{d}{dt}x(t) &= f(t,x(t)) \textit{ for } t>0 \textit{ and} \\ x(0) &= x_0, \textit{ (a known vector)} \end{aligned}$$

and

$$\begin{aligned} \frac{d}{dt}y(t) &= f(t,y(t)) \textit{ for } t>0 \textit{ and} \\ y(0) &= y_0 \textit{ (a known vector).} \end{aligned}$$

Then the equation $y' = f(t,y)$ is ***asymptotically stable or A-stable*** *if, for any x_0, y_0,*

$$|x(t)-y(t)| \to 0 \textit{ as } t \to \infty.$$

For the simplest, linear, scalar equation

$$y' = ay + f(t)$$

subtraction shows that $w(t) := x(t) - y(t)$ satisfies

$$w' = aw$$

The solution is $w(t) = e^{at}w(0)$ and thus as $t \to \infty$

$$w(t) \to 0 \text{ if and only if } Re(a) < 0$$

Indeed, if $a = \alpha + i\beta$, we have

$$w(t) = e^{\alpha t}\left[\cos\beta t + i\sin\beta t\right]w(0).$$

and $w(t) \to 0$ as $t \to \infty$ if and only if $\alpha = \mathrm{Re}(\lambda) < 0$.

CHAPTER 1

Solving an IVP by Euler's method

One basic task of computational science is the following. Knowing the initial state

$$y(0)=y_0 \text{ (} y_0 \text{ is a known number)}$$

and the "laws" governing a system

$$y'(t)=f(t,y(t)), \text{ for } 0<t\leq T_{final}$$

predict the future! Specifically:

$$\text{find } y(t) \text{ for } t>0.$$

This could be a single equation ($y(t)$ is a scalar function, $y:[0,\infty)\rightarrow\mathbb{R}$), or system of equations (so $y(t)$ is a vector function of t, $y:[0,\infty)\rightarrow\mathbb{R}^N$, and $f(t,y):[0,\infty)\times\mathbb{R}^N\rightarrow\mathbb{R}^N$). The system can also involve higher derivatives. Since the ideas are all the same, we will first consider a scalar problem.

Naturally, we can't expect to "solve" for $y(t)$ in closed form. We pick a step size called variously[1] $\triangle t, k$ or h. The variables t_j and y_j denote $t_j=j\triangle t$ and y_j is the approximation we compute to $y(t_j)$:

$$\triangle t = \text{step size},\ \ t_j=j\triangle t=j^{th} \text{ time step},\ \ y_j\approx y(tj).$$

The simplest way to find y_j is a method used by Euler to prove that initial value problems have solutions (i.e., that the future exists!). It's constructive, so we can use Euler's method for calculations. It is motivated as follows: Suppose we know $y(tj)$ exactly and want $y(t_{j+1})=y(t_j+\triangle t)$. Expanding y in a Taylor series at t_j gives:

$$y(t_{j+1})=y(t_j)+y'(t_j)\triangle t+\frac{1}{2}y''(\xi)\triangle t^2$$
$$\text{for some } \xi,\ t_j<\xi<t_{j+1}.$$

Now the equation $y(t)$ satisfies is $y'(t_j)=f(t_j,y(t_j))$. Thus:

$$y(t_{j+1})=y(t_j)+\triangle t f(t_j,y(t_j))+\frac{1}{2}y''(\xi)\triangle t^2$$
$$\text{for some } \xi,\ t_j<\xi<t_{j+1}.$$

The last term, $\frac{1}{2}y''(\xi)\triangle t^2$, is "unknowable" but it is small if $\triangle t$ is small. Just dropping this last term is Euler's method:

$$\text{Given } y_j \text{ find } y_{j+1} \text{ by}$$
$$y_{j+1}=y_j+\triangle t f(t_j,y_j)\text{ , for } j=0,1,2,\cdots.$$

[1] We shall use $\triangle t$ in the text and h or H in the algorithms. In problems with derivatives in both space and time, the time step is often called k and the space step h.

It is worthwhile seeing how it works in a simple example.

EXAMPLE 3 (Euler's Method for $y' = y$). *Consider Euler's method for the simple IVP*

$$\begin{aligned} y'(t) &= y(t), \text{ for } t > 0, \\ y(0) &= 1. \end{aligned}$$

The exact solution is

$$y(t) = e^t.$$

Euler's method for this equation is

$$y_0 = 1 \text{ and } \frac{y_{n+1} - y_n}{\triangle t} = y_n \text{ for } n \geq 0.$$

This is

$$\begin{aligned} y_{n+1} &= (1 + \triangle t) y_n \\ &= (1 + \triangle t)(1 + \triangle t) y_{n-1} \\ &= \cdots = (1 + \triangle t)^{n+1}. \end{aligned}$$

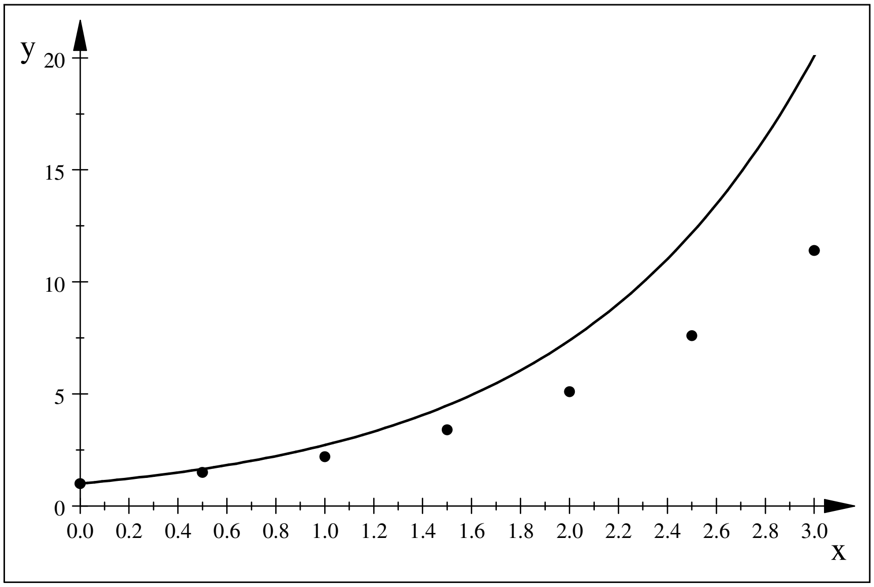
Note that this means the method converges for fixed t as $\triangle t \to 0$. Indeed, since $t_n = n\triangle t, n = t_n/\triangle t$

$$y_n = (1 + \triangle t)^n =$$
$$= \left[(1 + \triangle t)^{1/\triangle t}\right]^{t_n} \to e^{t_n} \text{ since}^2$$
$$(1 + \triangle t)^{1/\triangle t} \to e \text{ as } \triangle t \to 0.$$

If we take $\triangle t = 1/2$ (much too large for practical calculation) we get the following table

n	0	1	2	3	4	5	6
t_n	0	1/2	1	3/2	2	5/2	3
y_n	1	1.5	2.225	3.375	5.0625	7.59375	11.390625

This has the generally correct behavior (it does grow) but the error (the gap between the curve below and the points) increases as the calculation progresses. The next figure plots the true solution and the approximation.



True solution (curve) and Euler (points)

Notice that the error grows (exponentially). The ***relative error*** *actually decays as t increases (exercise below). Thus, while the error in Euler's method does grow for this problem, the number of significant digits of accuracy actually improves.*

Example 4. *Two more important tests. Broadly, there are three types of generic behavior of solitons to IVPs. Solutions can grow as in the last example, or decay or oscillate. We complement the last example with two more important tests of Euler's method for*

$$\begin{array}{rc} \textit{Decay:} & y' = -100y, y(0) = 1 \\ \textit{Oscillation:} & x' = y \textit{ and } y' = -x,\ x(0) = 1, y(0) = 0. \end{array}$$

Their exact solutions are

$$\begin{array}{rc} \textit{Decay Solution:} & y(t) = e^{-100t} \\ \textit{Oscillating solution:} & \left\{ \begin{array}{c} x(t) = \cos(t) \\ y(t) = -\sin(t) \end{array} \right. \end{array} .$$

Note that by doing Euler's method we commit an error $O(\triangle t^2)$ every step since

$$\begin{array}{cc} \textit{true} & y(t_{j+1}) = y(t_j) + \triangle t f(t_j, y(t_j)) + \frac{1}{2} y''(\xi) \triangle t^2 \\ \textit{Euler} & y_{j+1} = y_j + \triangle t f(t_j, y_j) \end{array} .$$

This is error committed each step called the "**local truncation error**" = the error in performing one step of the method starting exactly.

Definition 4 (Local Truncation Error). *The "****local truncation error****" of a method for solving $y'(t) = f(t, y(t))$ is the error in performing one step of the method starting exactly. In other words, it is the error := $y(t_{j+1}) - y_{j+1}$ provided the exact value of $y(t_{j+1})$ is used for the method.*

For Euler's method, the LTE is therefore

$$\begin{aligned} LTE &= y(t+\triangle t) - \left[y(t) + \triangle t f(t, y(t))\right], \text{ where} \\ y(t) &= \text{ exact solution at time } t, \\ y(t+\triangle t) &= \text{true/exact solution at } t+\triangle t, \\ y(t) + \triangle t f(t,y(t)) &= \text{one Euler with exact solution } y(t). \end{aligned}$$

We calculate the local truncation error of Euler's method using Taylor's theorem as follows:

$$\begin{gathered} \text{Local truncation error of Euler's method} = \\ = y(t_{j+1}) - (y(t_j) + \triangle t f(t_j, y(t_j))) = \\ = \frac{1}{2} y''(\xi) \triangle t^2 = \frac{1}{2} y''(t_j) \triangle t^2 + \mathcal{O}(\triangle t^3). \end{gathered}$$

By the equation $y'(t) = f(t, y(t))$ thus

$$\begin{aligned} y''(t) &= \frac{d}{dt}\left(f(t, y(t))\right) = \text{(by the chain rule)} = \\ &= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) y'(t) = \\ &= \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t)) \end{aligned}$$

Thus, we have shown.

Proposition 2. *The local truncation error of Euler's method is*

$$\begin{gathered} \textit{Local truncation error of Euler's method} = \\ = \frac{1}{2}\left[\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) f(t, y(t))\right] \triangle t^2 + \mathcal{O}(\triangle t^3). \end{gathered}$$

To calculate the solution from $t = 0$ to $t = T$ we take $J = T/\triangle t (= \mathcal{O}(\triangle t^{-1}))$ steps. If errors are additive we thus expect an error in $y(T)$ of $0(\triangle t)$. We shall prove this is true in a later section.

Programming Euler's method is very simple. Here is an algorithm for it:

Algorithm 1. *Euler's method for $y' = f(t,y)$ over $0 < t < T$ and $y(0) = y_0$.*

```
    Define the function $f(t,y)$
    Input $h$ = the step size
    Input $y_0$ = the initial condition
    Input $T$ = the final time
    $t_{OLD} = 0.0$
    $y_{OLD} = y_0$
(*) $t_{NEW} = t_{OLD} + h$
    $y_{NEW} = y_{OLD} + hf(t_{OLD}, y_{OLD})$
    If ($t_{NEW} > T$) STOP
    Else $y_{OLD} \Leftarrow y_{NEW}$
    $t_{OLD} \Leftarrow t_{NEW}$ pick a new timestep $h$ if necessary and Go To (*)
```

The main issues is doing better than Euler's method are the ones central to all numerical analysis:

- **Accuracy:** Euler's method is only $\mathcal{O}(\triangle t)$ accurate. Thus it is nearly impossible to get more than about two significant digits of accuracy with it in the presence of round off error.
- **Efficiency:** Can we obtain (greater) accuracy with less work?
- **Reliability:** Can the calculation be performed to have actual error within some preset tolerance?

Additional issues arise in the numerical solution of initial value problems and Euler's method. These include:

- **Physical behavior:** The types of behavior systems of ODEs posses are as varied as all the phenomena of nature. Thus, it is highly unlikely that one method or a small collection of methods would be reasonable choices for any system of ODEs.
- **Long time calculations:** Every step in Euler's method depends on all the results (and contains their accumulation of errors as well) of all the previous steps. If a calculation is to proceed over a long time interval, accumulation of inherited errors can result in the method being, in essence, a random number generator.
- **Parallelism:** It is not uncommon for a system to contain many millions of equations. To obtain accurate answers in a timely way often requires algorithms that can access parallel capabilities for storing data and computing approximations.
- **Instabilities:** Since every step depends on all the results in the previous steps, exponential instabilities can occur.
- **Conservation:** Some physical systems have an energy that is exactly conserved. It is often critical that an approximation exactly conserve some discrete version of the physical energy.
- **Software engineering aspects:** For complex problems, the very first step in the above algorithm, "*Define the function* $f(t, y)$", can have a variety of meanings. Often given t, y, the function value $f(t, y)$ is the result of running another program (often a legacy program) where t, y are inputs and $f(t, y)$ is the program's output. This setting yields restrictions on methods.

The error in Euler's method can (remarkably!) be calculated inside the algorithm. Recall that the local truncation error is:

$$\begin{aligned} LocalTruncationError &= \frac{1}{2}y''(t)\triangle t^2 + \mathcal{O}(\triangle t^3) \\ &= \frac{1}{2}\left[\frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t))f(t, y(t))\right]\triangle t^2 + \mathcal{O}(\triangle t^3) \end{aligned}$$

For very simple (scalar) problems, the global error can be estimated inside the algorithm by adding a few more lines (added in **BOLD** below) to the above algorithm.

ALGORITHM 2. *Euler's method for* $y' = f(t, y)$ *over* $0 < t < T$. *and* $y(0) = y_0$.

```
    Define the function $f(t, y)$
    **Define the functions**

        $FT(t, t) := f_t(t, y)$
```

```
                $FY(t,t) := f_y(t,y)$
                $YPP(t,y) := FT(t,y) + FY(t,y)f(t,y)$
          **Set:** $ERREST = 0.0$
Input $h =$ the step size
Input $y_0 =$ the initial condition
Input $T =$ the final time
```

```
ALGORITHM 3.        $t_{OLD} = 0.0$
        $y_{OLD} = y_0$
    (*) $t_{NEW} = t_{OLD} + h$
        $y_{NEW} = y_{OLD} + hf(t_{OLD}, y_{OLD})$
        ***Calculate:*** $LOCERR = |0.5 * h * h * YPP(t_{NEW}, y_{NEW})|$
                    **and** $ERREST \Leftarrow ERREST + LOCERR$
        If $(t_{NEW} > T)$ STOP
```

```
ALGORITHM 4.        *Else* $y_{OLD} \Leftarrow y_{NEW}$
        $t_{OLD} \Leftarrow t_{NEW}$,
          *pick a new timestep h if necessary and*
          *Go To* (*)
```

When we are solving a system of equations this is too expensive as FY is an $N \times N$ Jacobi[3] matrix. Thus, important questions include: How to calculate an estimate of the error more efficiently? How can it be used to improve the accuracy and efficiency of the program as the calculation progresses? The above also is based in the plausible assumption[4] that:

The global error is estimated by the sum of the local errors on all previous steps.

This is realized in the steps where the local error is estimated by LOCERR and then accumulated (added up) by ERREST <= ERREST + LOCERR.

> To think, you have to write. If you're thinking without writing, you only think you're thinking. [Leslie Lamport, Thinking for Programmers]
>
> Writing is nature's way of letting you know how sloppy your thinking is. [Guindon]

EXERCISE 5. *Write a program for Euler's method. Use it to solve an IVP with a known exact solution. Insert these statements inside the Euler's method algorithm at the correct locations and resolve. Compare the estimated error with the true errors. Next repeat using the alternate estimate of* YPP

$$YPP = \frac{y_{n+1} - 2y_n + y_{n-1}}{\triangle t^2}$$

and compare. Draw a conclusion as to which method of estimating the errors is more accurate and reliable.

[3] Adapted from the Wikipedia article:

Carl Gustav Jacob Jacobi (1804 – 1851) was a German mathematician, who made fundamental contributions to differential equations. Jacobi was the first Jewish mathematician to be appointed professor at a German university. One of his maxims was: 'Invert, always invert' ('man muss immer umkehren'), expressing that the solution of many hard problems can be clarified by re-expressing them in inverse form.

[4] This assumption is useful but incorrect as will be explained subsequently.

EXERCISE 6. *Consider Euler's method applied to $y' = y, y(0) = 1$. Show that the* ***relative error*** *actually decays as $t \to \infty$ as claimed.*

EXERCISE 7. *Consider the linear pendulum*

$$\theta'' \ + \theta = 0, \theta(0) = \pi/4, \theta'(0) = \pi/4.$$

If this is written as a first order system via

$$x(t) = \theta(t) \text{ and } y(t) = \theta'(t),$$

we obtain:

$$\begin{aligned} X' &= Y, X(0) = \pi/4 \\ Y' &= -X, Y(0) = \pi/4. \end{aligned}$$

Take $\triangle t = 1/3$ and compute an approximation to $\theta(1)$ and $\theta'(1)$ using Euler's method. Find the error. (Hint: The exact solution takes the form

$$\theta(t) = C_1 cos(t) + C_2 sin(t),$$

where $C_{1,2}$ depend on $\theta(0)$ and $\theta'(0)$.)

EXERCISE 8. *Repeat the above test for the nonlinear pendulum $\theta''(t) + sin\,(\theta(t)) = 0$. (The error cannot be calculated explicitly.) Write it as a first order system in the usual way [$x = \theta, y = \theta'$]. Show first that $G(x, y) = (1/2)y^2 - \cos x$ is constant along solutions. Plot $G(x, y)$ vs. t. Draw conclusions. Next try the equation $\theta'' + sign(\theta) = 0$. Again, pick the initial conditions so the solution has interesting behavior.*

EXERCISE 9. *Consider the leapfrog method for $y' = f(t, y)$ given by $y_{j+1} = y_{j-1} + 2\triangle t f(t_j, y_j)$. State the definition of "local truncation error". Analyze its LTE.*

1. The discoverers: Carl Gustav Jacob Jacobi

Adapted from the Wikipedia article:

CARL GUSTAV JACOB JACOBI (1804 – 1851) was a German mathematician, who made fundamental contributions to differential equations. Jacobi was the first Jewish mathematician to be appointed professor at a German university. One of his maxims was: 'Invert, always invert' ('man muss immer umkehren'), expressing that the solution of many hard problems can be clarified by re-expressing them in inverse form.

CHAPTER 2

More about Euler's method

> He calculated without any apparent effort, just as men breathe, as eagles sustain themselves in the air. - François Arago
>
> Read Euler: he is our master in everything. - Pierre-Simon Laplace
>
> He was later to write that he had made some of his best discoveries while holding a baby in his arms surrounded by playing children. -Richard Mankiewicz, in The Story of Mathematics (2000), p. 142

Euler's method is not used for practical calculations. However, it enables us to introduce in a context that is as simple and clear as possible techniques used for practical calculations for other methods. In this section we show that

- **Solving systems** of equations is as easy (in principle) as solving a single equation.
- **Roundoff error** can be controlled through a technique called 'partial double precision accumulation'.
- **Convergence** of the approximate solution to the true solution can be reduced to verifying two separate conditions of stability and consistency.
- **Stability** can be reduced to the problem of stability of the method for the scalar problem $y' = \lambda y$.

1. Systems of equations

Euler's method for a single scalar equation reads as follows.

ALGORITHM 5 (Euler for one equation). *Euler's method for $y' = f(t,y)$ over $0 < t < T$. and $y(0) = y_0$.*

```
        Define the function $f(t,y)$
          Input $h$ = the initial step size
        Input $y_0$ = the initial condition
        Input $T$ = the final time
        $t_{OLD} = 0.0$
        $y_{OLD} = y_0$
(*)     $t_{NEW} = t_{OLD} + h$
        $y_{NEW} = y_{OLD} + hf(t_{OLD}, y_{OLD})$
        If ($t_{NEW} > T$) STOP
        Else $y_{OLD} \Leftarrow y_{NEW}$
             $t_{OLD} \Leftarrow t_{NEW}$
               if desired pick a new stepsize $h$
             and Go To (*)
```

Remark 2 (Printing every step is not a good idea). *At some point the approximate solution needs to be sent to some sort of output to be viewed, analyzed or evaluated. The simplest is to just print the approximate solution at each step. However, putting a print statement like "PRINT t_{NEW}, y_{NEW} " inside the loop where the calculations are done (here and below) is not generally a good idea. Arithmetic occurs at the speed of electrons. On some systems, the program will stop at that step for the printer to print that line. If the timestep is small, this statement will also yield tens of thousands of lines of output. In a real program, only the solution at times where it is needed will be stored (at that line). After the calculations are done, a new loop occurs where the results are sent to the output device). The exact implementation depends on the operating system and what information is needed.*

Suppose we have a system of N equations. For the system, Euler's method is a minor change from the case of scalar equations.

Algorithm 6 (Euler's Method for Systems). *Euler's method for $\overrightarrow{y}' = \overrightarrow{f}(t, \overrightarrow{y})$ over $0 < t < T$. and $\overrightarrow{y}(0) = \overrightarrow{y}_0$.*

```
        Define the functions $f_j(t, y_1, y_2, \cdots, y_N), j = 1, \cdots, N$
        Input $h$ = the initial step size
        For $j = 1, \cdots, N$: Input $y_{j,0}$ = the initial condition
        Input $T$ = the final time
        $t_{OLD} = 0.0$
        For $j = 1, \cdots, N$: $y_{j,OLD} = y_{j,0}$
    (*) $t_{NEW} = t_{OLD} + h$
        For $j = 1, \cdots, N$:
                $y_{j,NEW} = y_{j,OLD} + hf_j(t_{OLD}, y_{1,OLD}, y_{2,OLD}, \cdots, y_{N,OLD})$
        If $(t_{NEW} > T)$ STOP
        Else: For $j = 1, \cdots, N$:   $y_{j,OLD} \Leftarrow y_{j,NEW}$
              $t_{OLD} \Leftarrow t_{NEW}$
               if desired pick a new stepsize $h$
              and Go To (*)
```

Sometimes a system means only two equations. For two equations we usually write the system as

$$\begin{aligned} x'(t) &= f(t, x(t), y(t)) \text{ for } t > 0 \text{ and } x(0) = x_0 \\ y'(t) &= g(t, x(t), y(t)) \text{ for } t > 0 \text{ and } y(0) = y_0. \end{aligned}$$

In that case Euler's method becomes the following.

Algorithm 7 (Eulers method for two equations). *Euler's method for*

```
        $x'(t) = f(t, x(t), y(t)), y'(t) = g(t, x(t), y(t)),$
           $y(0) = y_0, x(0) = x_0$
        Define the functions $f(t, x, y), g(t, x, y)$
        Input $h$ = the initial step size
        Input $x_0, y_0$ = the initial conditions
        Input $T$ = the final time
        $t_{OLD} = 0.0$
        $x_{OLD} = x_0$
        $y_{OLD} = y_0$
    (*) $t_{NEW} = t_{OLD} + h$
        $x_{NEW} = x_{OLD} + hf(t_{OLD}, x_{OLD}, y_{OLD})$
```

```
$y_{NEW} = y_{OLD} + hg(t_{OLD}, x_{OLD}, y_{OLD})$
If $(t_{NEW} > T)$ STOP
Else $x_{OLD} \Leftarrow x_{NEW}$ and $y_{OLD} \Leftarrow y_{NEW}$
          $t_{OLD} \Leftarrow t_{NEW}$
            if desired pick a new stepsize $h$
          and Go To $(*)$
```

2. Controlling roundoff by partial double precision accumulation

> A computer lets you make more mistakes faster than any invention in human history - with the possible exceptions of handguns and tequila. – Mitch Ratliffe
>
> The cancellation in the subtraction only gives an indication of the unhappy consequence of a loss of information in previous steps, due to rounding of [at least] one of the operands, and is not the cause of the inaccuracy. - Dahlquist and Bjork, Numerical Methods in Scientific Computing, Volume 1, p. 17.

Every step of **Euler's method** performs: given y_n,

Step 1:	Evaluate $f(t_n, y_n)$
Step 2:	Add $y_n + \triangle t f(t_n, y_n)$ to give y_{n+1}

Note that most of the **cost** occurs in Step 1's functions evaluation(s). However, in Step 2 we add $y_n = O(1)$ to $\triangle t f(t_n, y_n) = O(\triangle t)$, that is, **adding a large number to a small number**. Thus, most of the **roundoff error** occurs in Step 2. One standard tool to control roundoff error in calculations which have the structure that they can be *split into a numerically stable step where most of the computational work is performed followed by an inexpensive step where most of the roundoff error occurs* is called:

partial double precision accumulation.

It proceeds as follows:

(i) each y_n is stored in extended (e.g., double) precision.

(ii) $\triangle t f(t_n, y_n)$ is computed in lower precision then converted to extended precision.

(iii) The sum $y_{n+1} = y_n + \triangle t f(t_n, y_n)$ is performed in double precision.

This procedure is economical (costing only 1 double precision sum per step) and minimizes the **roundoff error** arising from **adding large to small** at every step.

3. Convergence of Euler's method

Consider Euler's method for the scalar IVP

$$\begin{aligned} y'(t) &= f(t, y), && \text{(IVP)} \\ \frac{y_{n+1} - y_n}{\triangle t} &= f(t_n, y_n). && \text{(Euler)} \end{aligned}$$

The following convergence theorem holds.

THEOREM 3 (Error estimate for Euler's method). *Suppose $y''(t)$ and $f_y(t, y(t))$ are bounded:*

$$\begin{array}{rcl} |y''(t)| & \leq & Y \ (< \infty) \\ \left| \dfrac{\partial f}{\partial y}(t, y) \right| & \leq & L \ (< \infty). \end{array}$$

Then the error in Euler's method satisfies

$$|y(t_n) - y_n| \leq \triangle t Y \frac{e^{Lt_n} - 1}{2L}.$$

This implies that **the error is $\mathcal{O}(\triangle t)$** and suggests that there are cases where **the error increases exponentially as t_n increases**. This last effect occurs but not always. We will give a detailed proof.

3.1. Proof of the convergence theorem. We give a detailed proof of the convergence theorem. It is based on two main ideas: consistency and stability.

Step 1: Use consistency to derive a difference equation for the error.

The true solution satisfies

$$y(t_{n+1}) = y(t_n) + \triangle t y'(t_n) + \frac{1}{2}\triangle t^2 y''(\xi), \text{ for some } \xi\text{:} \quad t_n < \xi < t_{n+1}.$$

As $y' = f(t, y)$ this means

$$y(t_{n+1}) = y(t_n) + \triangle t f(t_n, y(t_n)) + \frac{1}{2}\triangle t^2 y''(\xi), \text{ for some } \xi\text{:} \quad t_n < \xi < t_{n+1}.$$

REMARK 3 (An equivalent approach). *The systematic way to perform this step is to write the IVP as Euler's method plus a residual term then expand the residual term in a Taylor series as: Start with $y' = f(t, y)$. Rewrite it as Euler + Residual*

$$y(t_{n+1}) = y(t_n) + \triangle t f(t_n, y(t_n)) + R$$

$R =$ *whatever it needs to be for the above to be correct, i.e., reduce to* $0 = 0$,

$$R = y(t_{n+1}) - [y(t_n) + \triangle t f(t_n, y(t_n))]$$

Expanding R in a Taylor series we then find

$$\begin{array}{rcl} R & = & y(t_{n+1}) - [y(t_n) + \triangle t f(t_n, y(t_n))] = \\ & = & y(t_{n+1}) - [y(t_n) + \triangle t y'(t_n)] = \cdots = \dfrac{1}{2}\triangle t^2 y''(\xi) \end{array}$$

Now write the above and Euler's method and **subtract** to get an **exact difference equation for the error** $e_n := y(t_n) - y_n$

$$\begin{array}{cccc} y(t_{n+1}) = & y(t_n)+ & \triangle t f(t_n, y(t_n))+ & \frac{1}{2}\triangle t^2 y''(\xi) \\ y_{n+1} = & y_n+ & \triangle t f(t_n, y_n) & \\ ---- & ---- & --------- & ----- \\ e_{n+1} = & e_n+ & \triangle t\,[f(t_n, y(t_n)) - \triangle t f(t_n, y_n)] & +\frac{1}{2}\triangle t^2 y''(\xi) \end{array}$$

Step 2. Convert the difference equation to a linear, constant coefficient difference inequality.

We begin with

$$e_{n+1} = e_n + \triangle t\,[f(t_n, y(t_n)) - \triangle t f(t_n, y_n)] + \frac{1}{2}\triangle t^2 y''(\xi).$$

From the assumptions of the theorem we have

$$\begin{aligned}|y''(\xi)| &\leq Y \text{ and}\\ |f(t_n,y(t_n))-f(t_n,t_n)| &\leq L|f(t_n,y(t_n))-f(t_n,y_n)| = L|e_n|\end{aligned}$$

Thus, we have the difference inequality

$$|e_{n+1}| \leq (1+\triangle tL)|e_n| + \frac{1}{2}\triangle t^2 Y$$

Indeed, step by step gives

$$e_{n+1} = e_n + \triangle t\left[f(t_n,y(t_n)) - \triangle t f(t_n,y_n)\right] + \frac{1}{2}\triangle t^2 y''(\xi)$$

$$\Rightarrow$$

$$|e_{n+1}| \leq |e_n| + \triangle t|f(t_n,y(t_n)) - f(t_n,y_n)| + \frac{1}{2}\triangle t^2|y''(\xi)| \leq |e_n| + \triangle tL|e_n| + \frac{1}{2}\triangle t^2 Y$$

$$|e_{n+1}| \leq (1+\triangle tL)|e_n| + \frac{1}{2}\triangle t^2 Y.$$

REMARK 4 (What this looks like). *This difference inequality can be rewritten as*

$$\frac{|e_{n+1}|-|e_n|}{\triangle t} \leq L|e_n| + \frac{1}{2}\triangle tY$$

which resembles Euler's method for

$$y' = Ly + \frac{1}{2}\triangle tY$$

The next step will be to analyze stability or Euler's method for this constant coefficient IVP.

Step 3: Use stability to bound $|e_n|$ by $O(\triangle t)$ terms.

We have

$$\begin{aligned}|e_0| &= 0 \quad \text{and}\\ |e_{n+1}| &\leq (1+\triangle tL)|e_n| + \frac{1}{2}\triangle t^2 Y.\end{aligned}$$

There are two methods to use stability to bound $|e_n|$.

Method 1: Direct Assault! Backsolving the difference inequality gives

$$\begin{aligned}|e_{n+1}| &\leq (1+\triangle tL)|e_n| + \frac{1}{2}\triangle t^2 Y \quad \text{and}\\ |e_n| &\leq (1+\triangle tL)|e_{n-1}| + \frac{1}{2}\triangle t^2 Y, \text{ thus}\\ |e_{n+1}| &\leq (1+\triangle tL)\left[(1+\triangle tL)|e_{n-1}| + \frac{1}{2}\triangle t^2 Y\right] + \frac{1}{2}\triangle t^2 Y.\end{aligned}$$

Simplifying the last line gives

$$|e_{n+1}| \leq (1+\triangle tL)^2|e_{n-1}| + \left[1+(1+\triangle tL)\right]\frac{1}{2}\triangle t^2 Y$$

As $|e_{n-1}| \leq (1+\triangle tL)|e_{n-2}| + \frac{1}{2}\triangle t^2 Y$ we can continue one more step backwards

$$|e_{n+1}| \leq (1+\triangle tL)^3|e_{n-2}| + \left[1+(1+\triangle tL)+(1+\triangle tL)^2\right]\frac{1}{2}\triangle t^2 Y.$$

This is repeated[1] all the way down to $n = 0$, giving

$$|e_{n+1}| \leq (1 + \triangle tL)^{n+1}|e_0| + \\ + \left[1 + (1 + \triangle tL) + (1 + \triangle tL)^2 + \cdots + (1 + \triangle tL)^n\right] \frac{1}{2}\triangle t^2 Y.$$

Since $|e_0| = 0$ and shifting back one we then have

$$|e_n| \leq \frac{1}{2}\triangle t^2 Y \sum_{l=0}^{n-1}(1 + \triangle tL)^l$$

Notice that the sum is a geometric series and will be summed exactly.

Method 2: Majorization. This gives the same bound as Method 1 so we will sketch the steps. Viewing the difference inequality

$$|e_{n+1}| \leq (1 + \triangle tL)|e_n| + \frac{1}{2}\triangle t^2 Y$$

suggests we consider the difference *equation*:

$$\begin{aligned} \phi_{n+1} &= (1 + \triangle tL)\phi_n + \frac{1}{2}\triangle t^2 Y, n \geq 0, \\ \phi_0 &= |e_0| \end{aligned}$$

This is **exactly Euler's method** for the linear, constant coefficient IVP

$$\phi'(t) = L\phi(t) + \frac{1}{2}\triangle tY, t > 0, \text{ and } \phi(0) = |e_0|.$$

Next we prove that the solution to the difference equality majorizes the difference inequality solution.

LEMMA 1. *We have*

$$\phi_n \geq |e_n|. \textit{ for every } n.$$

PROOF. This is a very simple induction argument. □

Now all that remains is to solve the difference equality. Its solution is

$$\phi_n = (1 + \triangle tL)^n|e_0| + \frac{1}{2}\triangle t^2 Y \sum_{l=0}^{n-1}(1 + \triangle tL)^l = \frac{1}{2}\triangle t^2 Y \sum_{l=0}^{n-1}(1 + \triangle tL)^l.$$

Thus, as before,

$$|e_n| \leq \phi_n = \frac{1}{2}\triangle t^2 Y \sum_{l=0}^{n-1}(1 + \triangle tL)^l.$$

Step 4: Put the error estimate in a form that is easy to understand. Consider the error bound

$$|e_n| \leq \frac{1}{2}\triangle t^2 Y \sum_{l=0}^{n-1}(1 + \triangle tL)^l$$

The geometric series can be summed exactly to give

$$|e_n| \leq \frac{1}{2}\triangle t^2 Y \left(\frac{(1 + \triangle tL)^n - 1}{\triangle tL}\right) = \triangle t\frac{Y}{2L}\left((1 + \triangle tL)^n - 1\right).$$

[1] A stickler for formal proof would insert an induction argument at this point.

There is a standard way to make the term $(1+\triangle tL)^n$ more comprehensible. Recall that the Taylor series

$$e^{\triangle tL} = 1 + \triangle tL + \frac{(\triangle tL)^2}{2!} + \frac{(\triangle tL)^3}{3!} + \cdots$$

has all positive terms. Thus, dropping (positive) terms gives

$$\begin{aligned} e^{\triangle tL} &\geq 1 + \triangle tL \text{ which implies} \\ (1+\triangle tL)^n &\leq \left(e^{\triangle tL}\right)^n = e^{(n\triangle t)L} = e^{Lt_n}. \end{aligned}$$

This gives

$$|e_n| \leq \frac{1}{2}\triangle tY\left(\frac{e^{Lt_n}-1}{L}\right),.$$

which completes the proof.

Remark 5. *Let us consider again the difference equation*

$$\phi_{n+1} = (1+\triangle tL)\phi_n + \frac{1}{2}\triangle t^2 Y$$

The following can be shown.

Theorem 4. *The general solution to the above difference equation can be written*

$$\phi_n = \phi_n^G + \phi_n^P$$

where ϕ_n^G is the general solution to the homogeneous difference equation

$$\phi_{n+1} = (1+\triangle tL)\phi_n$$

and ϕ_n^P is and particular solution to the inhomogeneous difference equation

$$\phi_{n+1} = (1+\triangle tL)\phi_n + \frac{1}{2}\triangle t^2 Y.$$

Remark 6. *We can find ϕ_n^P by guessing that is $\phi_n^P = K$ constant (independent of n). The motivation for this guess is that $\frac{1}{2}\triangle t^2 Y$ is independent of n. This guess gives and equation for K:*

$$\begin{aligned} K &= (1+\triangle tL)K + \frac{1}{2}\triangle t^2 Y, \text{ and thus} \\ \phi_n^P &= K = \frac{1}{2\triangle tL}\triangle t^2 Y. \end{aligned}$$

The general solution to $\phi_{n+1} = (1+\triangle tL)\phi_n$ is found by guessing $\phi_n^G = CR^n$. This gives

$$R^{n+1} = (1+\triangle tL)R^n \text{ so } R = (1+\triangle tL).$$

This yields the solution

$$\phi_n = C(1+\triangle tL)^n + \frac{\triangle tY}{2L}.$$

Exercise 10. *Adapt the 6 steps of the proof to the backward Euler method and prove a convergence theorem for it.*

CHAPTER 3

A General Theory

> "In theory there is no difference between theory and practice. In practice there is." - Yogi Berra

The theorem on convergence of Euler's method is a special case of a general theory in numerical ODEs that uncouples convergence into two separate conditions of *stability* and *consistency* that are more easily verified. The general theorem below is that stability plus consistency implies convergence. We begin by filling in the definitions needed

1. Consistency

> Through all of scientific computing runs this common theme: Increase the accuracy at least to second order. What this means is: Get the linear term right. [G.S. Gilbert Strang , BAMS, 1993, Wavelet Transforms vs. Fourier Transforms]

Suppose we have a general $k-$step method (suppressing t dependence) of the form

$$y_{n+1} = \Phi(y_{n+1}, y_n, \cdots, y_{n-k}; \triangle t). \qquad \text{(GENERAL K-STEP)}$$

Let the IVP have a smooth solution. Then the method (GENERAL K-STEP) is consistent of order l if, inserting the true solution $y(t)$ into the method, we have

$$y(t_{n+1}) - \Phi(y(t_{n+1}), y(t_n), \cdots, y(t_{n-k}); \triangle t) = \mathcal{O}(\triangle t^{l+1}).$$

EXAMPLE 5 (**Consistency is evaluated by Taylor series**). *For example, for Euler's method*

$$\frac{y_{n+1} - y_n}{\triangle t} = f(t_n, y_n)$$

*first rewrite it in the above form (*GENERAL K-STEP*)*

$$y_{n+1} = y_n + \triangle t f(t_n, y_n)$$

so that

$$\Phi(y_{n+1}, y_n, \cdots, y_{n-k}; \triangle t) = y_n + \triangle t f(t_n, y_n).$$

now insert the true solution. Expand all in a Taylor series and cancel terms

$$y(t_{n+1}) - (y(t_n) + \triangle t f(t_n, y(t_n))) = O(\triangle t^2).$$

EXERCISE 11. *Find the methods of the following form that have minimum local truncation error:*

$$\begin{aligned} ay_{n+1} + by_n + cy_{n-1} &= \triangle t f(t_{n+1}, y_{n+1}), \\ ay_{n+1} + by_n + cy_{n-1} &= \triangle t f(t_n, y_n), \\ ay_{n+1} + by_n + cy_{n-1} &= \triangle t f(t_{n-1}, y_{n-1}). \end{aligned}$$

Exercise 12 (Consistency error in BDF2). *Consider the difference approximation used in BDF2*

$$D_2 y_n := \frac{3y_n - 4y_{n-1} + y_{n-2}}{2\triangle t}.$$

Use Taylor's theorem with integral remainder to show that

$$\begin{aligned} D_2 y(t_n) &= y'(t_n) + \\ &+ \frac{1}{2\triangle t}\int_{t_{n-2}}^{t_n}\left[2(t-t_{n-1})_+^2 - \frac{1}{2}(t-t_{n-2})^2\right] y'''(t)dt, \\ \textit{where} &: (t-t_{n-1})_+ := \max\{(t-t_{n-1}), 0\}. \end{aligned}$$

Use this formula to estimate the consistency error in BDF2.

Exercise 13 (Consistency error in AB2). *Consider the extrapolation used in AB2*

$$E(y) := 2y_{n-1} - y_{n-2}.$$

Use Taylor's theorem with integral remainder to show that

$$\begin{aligned} E(y(t)) &= y(t_n) + \\ &+ \frac{1}{2\triangle t}\int_{t_{n-2}}^{t_n}\left[2(t-t_{n-1})_+ - (t-t_{n-2})\right] y''(t)dt, \\ \textit{where} &: (t-t_{n-1})_+ := \max\{(t-t_{n-1}), 0\}. \end{aligned}$$

Use this formula to show that the extrapolation error is $O(\triangle t^2)$.

1.1. Brook Taylor FRS (18 August 1685 – 29 December 1731). Adapted from Wikipedia:

Brook Taylor FRS (18 August 1685 – 29 December 1731) was an English mathematician who is best known for Taylor's theorem and the Taylor series. He entered St. John's College, Cambridge, as a fellow-commoner in 1701, and took degrees of LL.B. and LL.D. in 1709 and 1714, respectively. Having studied mathematics under John Machin and John Keill, in 1708 he obtained a remarkable solution of the problem of the "centre of oscillation," which, however, remained unpublished until May 1714, when his claim to priority was disputed by Johann Bernoulli. Taylor's Methodus Incrementorum Directa et Inversa (1715) added a new branch to higher mathematics, now called the "calculus of finite differences". Among other ingenious applications, he used it to determine the form of movement of a vibrating string, by him first successfully reduced to mechanical principles. The same work contained the celebrated formula known as Taylor's formula, the importance of which remained unrecognized until 1772, when J. L. Lagrange realized its powers and termed it "the main foundation of differential calculus".

In his 1715 essay Linear Perspective, Taylor set forth the true principles of the art in an original and more general form than any of his predecessors; but the work suffered from the brevity and obscurity which affected most of his writings, and needed the elucidation bestowed on it in the treatises of Joshua Kirby (1754) and Daniel Fournier (1761).

Taylor was elected a fellow of the Royal Society early in 1712, and in the same year sat on the committee for adjudicating the claims of Sir Isaac Newton and Gottfried Leibniz, and acted as secretary to the society from 13 January 1714 to 21 October 1718. As a mathematician, he was the only Englishman after Sir Isaac

Newton and Roger Cotes capable of holding his own with the Bernoullis, but a great part of the effect of his demonstrations was lost through his failure to express his ideas fully and clearly.

2. $0-$Stability

> Analysis and algebraic conditions: Theorem 2.2 [Dahlquist equivalence theorem] demonstrates a state of affairs that prevails throughout mathematical analysis. Thus, we desire to investigate an analytic condition, e.g. whether a differential equation has a solution, whether a continuous dynamical system is asymptotically stable, whether a numerical method converges. By their very nature, analytic concepts involve infinite processes and continua, hence one can expect analytic conditions to be difficult to verify, to the point of unmanageability. For all we know, the human brain (exactly like a digital computer) might be essentially an algebraic machine. It is thus an important goal in mathematical analysis to search for equivalent algebraic conditions. The Dahlquist equivalence theorem is a remarkable example of this: everything essentially reduces to determining whether the zeros of a polynomial reside in a unit disc, and this can be checked in a finite number of algebraic operations! In the course of this book we will encounter numerous other examples of this state of affairs. Cast your mind back to basic infinitesimal calculus and you are bound to recall further instances where analytic problems are rendered in an algebraic language. - Arieh Iserles

There are many different types of stability. The most basic (without which the numerical method is nonsense) is **zero stability / 0-stability**.

Definition 5 (0-Stability). *The method (*General k-step*) is **0-stable** if, when it is applied to*

$$y' = Ly, \text{ with } L \text{ constant,}$$

the solution satisfies

$$|y_n| \le e^{Ct_n}|y_0|,$$

where C is a constant independent of $t_n, \triangle t$ but dependent on L.

The connection with the analysis of Euler's method is that if a method is $0-$stable for $y' = L$ then it can be shown that it is also $0-$stable for $y' = Ly + F$, as required in the convergence analysis of Euler's method. This is quite general; if a linear multistep method is stable for $F = 0$ then it is stable for any F non-zero.

We shall see examples how stability is analyzed for methods in a section to come.

Theorem 5 (Stability+Consistency⇒Convergence). *For a numerical method for a well posed IVP, **0-stability** (over $0 < t \le T(< \infty)$) plus **consistency** (local truncation error of order $l > 1$) implies **convergence** as $\triangle t \to 0$ over $0 < t \le T$.*

Analysis of $0-$stability for a 1 step method is particularly easy. Any one step method applied to $y' = Ly$ yields a difference equation of the form

$$\begin{aligned} y_{n+1} + a(\triangle tL)y_n &= 0 \text{ whence} \\ y_n &= R^n y_0, \text{ with } R = a(\triangle tL). \end{aligned}$$

Therefore the following holds.

PROPOSITION 3. *A* $1-$*step method* $y_{n+1}+a(\triangle tL)y_n=0$ *is zero stable if and only if* $R=a(\triangle tL)$ *satisfies*

$$|R|\leq 1+\alpha\triangle t$$

for some $\alpha>0$ *independent of* $\triangle t$ *(but dependent on* L*).*

PROOF. We prove the "if" part. If $|R|\leq 1+\alpha\triangle t$ then

$$\begin{aligned} y_n &= |R^n y_0|\leq(1+\alpha\triangle t)^n|y_0| \\ &\leq \left(1+\alpha\triangle t+\frac{1}{2!}(\alpha\triangle t)^2+\frac{1}{3!}(\alpha\triangle t)^3+...\right)^n|y_0| \\ &\leq \left(e^{\alpha\triangle t}\right)^n|y_0|=e^{\alpha n\triangle t}|y_0|=e^{\alpha t_n}|y_0|. \end{aligned}$$

The "only if" part is an exercise in calculus inequalities. □

EXAMPLE 6 (The midpoint rule). *The midpoint rule is: given* y_n

$$\begin{aligned} k_1 &= \triangle tf(t_n,y_n) \\ k_2 &= \triangle tf(t_n+\frac{1}{2}\triangle t,y_n+\frac{1}{2}k_1), \\ y_{n+1} &= y_n+k_2. \end{aligned}$$

Set $f(y)=Ly$. *Then, simplifying we have*

$$y_{n+1}-y_n=\triangle tL(1+\triangle tL)y_n$$

Thus

$$R=1+\triangle tL+\frac{1}{2}(\triangle tL)^2$$

so that we have

$$\begin{aligned} |y_n| &= |y_0|\left(1+\triangle tL+\frac{1}{2}(\triangle tL)^2\right)^n\leq|y_0|\left(e^{\triangle tL}\right)^n \\ &\leq |y_0|e^{Lt_n} \end{aligned}$$

and the midpoint rule is 0-stable.

EXAMPLE 7. ***Oscillatory problems*** *have consistency restrictions. To illustrate, consider the linear pendulum:*

$$\theta''+\omega^2\theta=0,t>0,\ \textit{where } \omega \textit{ is real},$$
$$\theta(0),\theta'(0)\ \textit{both specified.}$$

Written as a first order system in the usual way $(x(t)=\theta(t),y(t)=\theta'(t))$ *this gives*

$$x'=y,y'=-\omega^2x$$

or

$$\frac{d}{dt}\begin{bmatrix}x\\y\end{bmatrix}=\begin{bmatrix}0&1\\-\omega^2&0\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}.$$

The eigenvalues of the above 2×2 *matrix are easily found to be* $\pm\omega i$ $(i=\sqrt{-1})$ *so that the above system is equivalent to* $y'=\lambda y$ *with* $\lambda=\pm\omega i$. *If the initial data is* $x(0)=1,y(0)=0$, *the exact solution is*

$$x(t)=\cos(\omega t),y(t)=\sin(\omega t).$$

For any p^{th} order method, the local truncation error takes the form

$$\begin{aligned} LTE &= C\triangle t^{p+1}\frac{d^{p+1}}{dt^{p+1}}(\cos(\omega t),\sin(\omega t)) \text{ so that} \\ |LTE| &= C\left(|\omega|\triangle t\right)^{p+1}. \end{aligned}$$

Obviously, for ω large, $\triangle t$ must be small enough that

$$|\omega|\triangle t < 1$$

to hope for even a single digit of accuracy, regardless of the order of the method or its stability properties. This condition is often interpreted as saying:

$$\textit{wave speed} \times \textit{time step} < 1.$$

Exercise 14. *Prove the "only if" part by filling in the following steps. First show that it is equivalent to showing*

$$e^x \leq 1 + ax$$

for $0 < x \leq \alpha$ and for some a that can depend on α. Next, use calculus to analyze the sign of the function $f(x) = 1 + ax - e^x$. In all such calculus inequalities, sketching $f(x)$ is very helpful.

3. Solving difference equations

> . . . discrete mathematics is more difficult than continuous mathematics. If you look at formulas for derivatives of reciprocals and then finite differences for reciprocals, you see how things are more complicated in the discrete case. . . . The main point in the theory of difference approximations is to prove stability. To prove stability is like getting an a priori estimate for the solution of the equation. But to get those estimates for difference approximations is much more sophisticated than to get them for a differential equation. - Peter Lax, MAA Focus (May/June 2005).

Next, consider multi-step methods. A $k-$step method applied to $y' = Ly$ leads to a linear, constant coefficient, homogeneous difference equation of the general form

$$y_{n+k} + a_1 y_n + a_2 y_{n-1} + \cdots + a_k y_n = 0,$$

where each coefficient depends on the product $\triangle tL$. To seek solutions we set $y_n = R^n$. This is a solution if and only of R is a root of the characteristic equation

$$R^k + a_1 R^{k-1} + a_2 R^{k-2} + \cdots + a_k = 0.$$

This has k roots counting multiplicity, $R_1, \cdots, R_k$. The general solution when these roots are distinct is

$$y_n = C_1 R_1^n + \cdots + C_k R_k^n.$$

Thus one simple conclusion is as follows.

Proposition 4. *If the characteristic equation*

$$R^k + a_1 R^{k-1} + a_2 R^{k-2} + \cdots + a_k = 0.$$

has k distinct roots and each root satisfies

$$|R_i| \leq 1 + \alpha\triangle t, i = 1, ..., k,$$

then the method is zero stable.

There are cases when the general solution

$$y_n = C_1 R_1^n + \cdots + C_k R_k^n.$$

needs to be modified or supplemented. The most common two cases a

Case 1: A complex conjugate pair of roots: $R_{1,2} = \alpha \pm \beta i$. In this case

$$R_{1,2} = \alpha \pm \beta i = re^{\pm i\theta},$$

where

$$r = \sqrt{\alpha^2 + \beta^2}, \theta = \arctan(\frac{\beta}{\alpha}),$$
$$e^{\pm i\theta} = \cos\theta \pm i \sin\theta.$$

Then

$$\begin{aligned} C_1 R_1^n + C_2 R_2^n &= r^n \left(C_1[\cos n\theta + i\sin n\theta] + C_2[\cos n\theta - i \sin n\theta]\right) \\ &= \widetilde{C}_1 r^n \cos n\theta + \widetilde{C}_2 r^n \sin n\theta. \end{aligned}$$

Thus, the complex powers $C_1 R_1^n + C_2 R_2^n$ and be replaced by $\widetilde{C}_1 r^n \cos n\theta + \widetilde{C}_2 r^n \sin n\theta$ which is real if $\widetilde{C}_1, \widetilde{C}_2$ are real.

Case 2: Multiple roots. If $R_1 = R_2$ is a double root, then $C_1 R_1^n + C_2 R_2^n$ does not contain two linearly independent constants. It can be easily verified that in this case R_1^n, nR_1^n are linearly independent solutions of the difference equation. therefore a double root contributes terms

$$C_1 R_1^n + C_2 n R_1^n$$

to the general solution.

Other root configurations are elaborations of these cases and handled in the expected way.

4. 0−Stability of Linear Multistep Methods

Dahlquist gave a simple and powerful condition for 0−stability of linear multistep methods.

Definition 6 (linear multi-step method). *The general $k-$step method*

$$y_{n+1} = \Phi(y_{n+1}, y_n, \cdots, y_{n-k}; \triangle t)$$

is a linear multi-step method if $\Phi(y_{n+1}, y_n, \cdots, y_{n-k}; \triangle t)$ is a linear combination of $f(t_j, y_j)$ and y_j.

We begin with an example. The linear multi-step method BDF2 is

$$3y_{n+1} - 4y_n + y_{n-1} = 2\triangle t f(t_{n+1}, y_{n+1}).$$

When this is applied to $y' = Ly$ we get

$$3y_{n+1} - 4y_n + y_{n-1} = 2\triangle t L y_{n+1}.$$

Seeking a solution $y_n = R^n$ yields the quadratic equation

$$3R^2 - 4R + 1 = 2\triangle t L R$$

which has a part (the LHS) independent of $\triangle t L$ and another (the RHS) with a multiplier $\triangle t L$. This pattern holds generally. The characteristic equation of a linear multistep method always takes the form

$$\rho(R) = \triangle t L \sigma(R)$$

where $\rho(R), \sigma(R)$ are polynomials independent of $\triangle tL$. Dahlquist[1] proved the following simple necessary and sufficient condition for 0−stability.

Theorem 6 (The root condition for 0-stability). *A linear multistep method is zero stable if and only $\rho(R)$ (the characteristic polynomial when $\triangle t = 0$) satisfies **the root condition** which is:*
1. All roots of $\rho(R) = 0$ satisfy $|R| \leq 1$, and
2. If a root satisfies $|R| = 1$ then R is a simple root.

Proof. We will not prove this theorem but only indicate the idea. The idea is the Newton-Puiseux theorem. □

Theorem 7. Proof. **Theorem.** [**Newton-Puiseux theorem**] Let R be a root of $p(x) = 0$ where the coefficients of $p(x)$ depend analytically upon some small parameter (such as h) called ε so that $R = R(\varepsilon)$ depends on ε as well. If R is a simple root then R(ε) is an analytic function of ε. If R is a double root then R(ε) is an analytic function of $\sqrt{\varepsilon}$ (and so on). □

Proof. The idea of the proof is now this. To prove 0-stability it is necessary only to show that the roots of $\rho(R) = \triangle tL\sigma(R)$ satisfy $|R| \leq 1 + c\triangle t$ for $\triangle t$ small enough. The roots of $\rho(R) = 0$ are the roots R($\triangle t$)$|_{\triangle t=0}$ of $\rho(R) = \triangle tL\sigma(R)$. In the above cases 1 and 2, if the root of |R|<1 of $\rho(R) = 0$ then clearly by the above theorem the roots of $\rho(R) = \triangle tL\sigma(R)$ satisfy $|R| \leq 1 + c\triangle t$ for $\triangle t$ small enough by analyticity. □

We consider 3 examples. In these examples note that for a 2 step method

$$\rho(R) = (R - R_1)(R - R_2) = R^2 - (R_1 + R_2)R + R_1R_2.$$

Thus if

$$\begin{aligned} \rho(R) &= R^2 - AR + B \\ A &= R_1 + R_2 \\ B &= R_1R_2 \end{aligned}$$

Example 8. *The method*

$$y_{n+1} + 4y_n - 5y_{n-1} = \triangle t\,(4f(t_{n+1}, y_{n+1}) + 2f(t_n, y_n))$$

is not 0-stable.

Indeed, we find

$$\begin{aligned} \rho(R) &= R^2 - (-4)R + (-5) \\ -5 &= R_1R_2 \end{aligned}$$

so one of them must be larger than 1 in magnitude,

[1] Adapted from Wikipedia:

Germund Dahlquist (1925 – 2005) was a Swedish mathematician known for his contributions to the theory of numerical analysis of differential equations.

Dahlquist studied mathematics at Stockholm University in 1942 at the age of 17, where the Danish mathematician Harald Bohr was a profound influence. He then worked with Carl-Gustaf Rossby on early numerical weather forecasts.

Dahlquist completed his Ph.D., “Stability and Error Bounds in the Numerical Solution of Ordinary Differential Equations" in 1958. In 1959 he moved to the Royal Institute of Technology (KTH) and became Sweden’s first Professor of Numerical Analysis in 1963.

Example 9. *The method (known as an Adams[2]-Bashforth[3] method)*

$$y_{n+1} - y_n = \triangle t \left(2f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right)$$

is 0-stable.

Indeed,

$$\begin{aligned} \rho(R) &= R^2 - R \\ R_1 &= 0, R_2 = 1. \end{aligned}$$

Example 10. *The leapfrog method*

$$y_{n+1} - y_{n-1} = 2\triangle t f(t_n, y_n)$$

is 0-stable.

Indeed,

$$\begin{aligned} \rho(R) &= R^2 - 1 \\ R_1 &= -1, R_2 = +1. \end{aligned}$$

Dahlquist also proved the following which is the first of the Dahlquist barriers.

Theorem 8 (A Dahlquist barrier). *The local truncation error $O(\triangle t^{p+1})$ of a $0-$stable linear multi step method with k steps is limited to*
(i) $p \le k+2$ if k is even.
(ii) $p \le k+1$ if k is odd.
(iii) $p \le k$ if the method is explicit.

Exercise 15. *Consider the leapfrog method for $y' = f(t, y)$ given by $y_{j+1} = y_{j-1} + 2\triangle t f(t_j, y_j)$. Show that Leapfrog is $0-$stable. Classify its approximate solution behavior as $t_n \to \infty$.*

Exercise 16. *Find the method of minimum consistency error of the form*

$$a_1 y_{n+1} + a_2 y_n + a_3 y_{n-1} = \triangle t \left(b_1 f(t_{n+1}, y_{n+1}) + b_2 f(t_n, y_n) + b_3 f(t_{n-1}, y_{n-1})\right).$$

Analyze its 0-stability.

[2] Adapted from Wikipedia:

John Couch Adams FRS (1819 – 1892) was a British mathematician and astronomer. His most famous achievement was predicting the existence and position of Neptune, using only mathematics. The calculations were made to explain discrepancies with Uranus's orbit and the laws of Kepler and Newton. At the same time, but unknown to each other, the same calculations were made by Urbain Le Verrier. He was Lowndean Professor in the University of Cambridge from 1859 until his death. Neptune's outermost known ring and the asteroid 1996 Adams are named after him. He was "extraordinarily uncompetitive, reluctant to publish imperfect work to stimulate debate or claim priority, averse to correspondence about it, and forgetful in practical matters".

[3] Adapted from Wikipedia:

Francis Bashforth (1819 - 1912) was a British applied mathematician. Between 1864 and 1880 he undertook some systematic ballistics experiments that studied the resistance of air. He also studied liquid drops and surface tension. The Adams–Bashforth method was used the method to study drop formation in 1883.

4.1. The discoverers. Adapted from Wikipedia:

Germund Dahlquist (1925 – 2005) was a Swedish mathematician known for his contributions to the theory of numerical analysis of differential equations.

Dahlquist studied mathematics at Stockholm University in 1942 at the age of 17, where the Danish mathematician Harald Bohr was a profound influence. He then worked with Carl-Gustaf Rossby on early numerical weather forecasts. Dahlquist completed his Ph.D., "Stability and Error Bounds in the Numerical Solution of Ordinary Differential Equations" in 1958. In 1959 he moved to the Royal Institute of Technology (KTH) and became Sweden's first Professor of Numerical Analysis in 1963.

John Couch Adams FRS (1819 – 1892) was a British mathematician and astronomer. His most famous achievement was predicting the existence and position of Neptune, using only mathematics. The calculations were made to explain discrepancies with Uranus's orbit and the laws of Kepler and Newton. At the same time, but unknown to each other, the same calculations were made by Urbain Le Verrier. He was Lowndean Professor in the University of Cambridge from 1859 until his death. Neptune's outermost known ring and the asteroid 1996 Adams are named after him. He was "extraordinarily uncompetitive, reluctant to publish imperfect work to stimulate debate or claim priority, averse to correspondence about it, and forgetful in practical matters".

Francis Bashforth (1819 - 1912) was a British applied mathematician. Between 1864 and 1880 he undertook some systematic ballistics experiments that studied the resistance of air. He also studied liquid drops and surface tension. The Adams–Bashforth method was used the method to study drop formation in 1883.

CHAPTER 4

It is easy to generate new methods for IVPs

> "Imagination is everything. It is the preview of life's coming attractions." ■ Albert Einstein

Part of the richness of the theory of numerical methods for IVP comes from the complexity of the phenomena the methods are used to understand. Part comes from the great diversity of methods. We illustrate next that is it very easy to generate new methods for the IVP

$$y' = f(t,y), \text{ for } t > 0 \text{ and } y(0) = y_0. \tag{0.1}$$

New methods means only new ways to take the IVP and generate a table of numbers. The central question is whether these numbers are faithful representations of the IVP's solution? Existence of hundreds of methods for IVPs also means that the role of a useful theory must be to find the simplest possible problems that separates (and explains) the behavior of the different methods. Such a theory will answer the question of how to pick a method for a given IVP that will solve it accurately.

In describing methods, it is useful to have a taxonomy. Consider a general method of the form

$$y_{n+1} = \Phi(\triangle t, t_n, y_{n+1}, y_n, y_{n-1}, \cdots, y_{n-k}). \tag{0.2}$$

Definition 7. *The method (0.2) is*

- *a one step method if* $\Phi = \Phi(\triangle t, t_n, y_{n+1}, y_n)$,
- *a multi-step method if* $\Phi = \Phi(\triangle t, t_n, y_{n+1}, y_n, y_{n-1}, \cdots, y_{n-k})$ *where* $k \geq 1$,
- *explicit if* $\Phi = \Phi(\triangle t, t_n, y_n, y_{n-1}, \cdots, y_{n-k})$, *i.e.* Φ *is independent of* y_{n+1},
- *implicit if* $\Phi = \Phi(\triangle t, t_n, y_{n+1}, y_n, y_{n-1}, \cdots, y_{n-k})$, *i.e.* Φ *depends on* y_{n+1}.

As examples of each we have the following.

Example 11. ***Euler***: $y_{n+1} = y_n + \triangle t f(t_n, y_n)$ *is a one step method since* $\Phi = y_n + \triangle t f(t_n, y_n)$ *only depends on* y_{n+1}, y_n *and not previous values.*

Example 12. ***Leapfrog:*** $y_{n+1} = y_{n-1} + 2\triangle t f(t_n, y_n)$ *is a multistep method with* $k = 2$ *(a 2 step method to be precise) since* $\Phi = y_{n-1} + 2\triangle t f(t_n, y_n)$, *depends on* y_{n+1}, y_n *and* y_{n-1}.

Example 13. ***Backward Euler:*** $y_{n+1} = y_n + \triangle t f(t_{n+1}, y_{n+1})$ *is an implicit method since* $\Phi = y_n + \triangle t f(t_{n+1}, y_{n+1})$ *depends on* y_{n+1}. *Each step requires solving a nonlinear equation (or system) for the new value.*

Example 14. ***Euler:*** $y_{n+1} = y_n + \triangle t f(t_n, y_n)$ *is also an explicit method since* $\Phi = y_n + \triangle t f(t_n, y_n)$ *does not depend on* y_{n+1}. *Each step requires only one function evaluation for the new value.*

In deriving methods, we shall see there are twin goals of accuracy and stability. Accuracy is measured by the local truncation error and the LTE of a method is evaluated by Taylor series.

DEFINITION 8 (Local Truncation Error). *Let $y(t)$ be the exact solution to*

$$y' = f(t,y)$$

and consider the method (0.2). The Local Truncation Error of (0.2) is the residual of the true solution in the discrete method:

$$LTE := y(t_{n+1}) - \Phi(\triangle t, t_n, y(t_{n+1}), y(t_n), y(t_{n-1}), \cdots, y(t_{n-k})).$$

If the local truncation error is $O(\triangle t^{p+1})$ as $\triangle t \to 0$ then the method is said to be a p^{th} order method.

1. The Taylor series of the true solution gives methods

The true solution of the IVP satisfies

$$y(t+\triangle t) = y(t) + \triangle t f(t,y(t)) + \frac{\triangle t^2}{2}\left[\frac{\partial f}{\partial t}(t,y(t)) + \frac{\partial f}{\partial y}(t,y(t))f(t,y(t))\right] +$$
$$\frac{\triangle t^3}{3!}\left[\begin{array}{c} f_{tt}(t,y(t)) + 2f_{ty}(t,y(t))f(t,y(t)) + \\ +f_{yy}(t,y(t))f(t,y(t)) + f_y(t,y(t))f_t(t,y(t)) + f_y^2(t,y(t))f(t,y(t)) \end{array}\right]$$
$$+\mathcal{O}(\triangle t^4).$$

Denote by

$$f^n \quad : \quad = f(t_n, y_n), f_t^n := \frac{\partial f}{\partial t}(t_n,y_n), f_y^n := \frac{\partial f}{\partial y}(t_n,y_n)$$

and so on for higher derivatives.

Then we have the following first three examples of the family of **Taylor series methods**:

$$\begin{aligned} y_{n+1} &= y_n + \triangle t f^n, && (1^{st} \text{ order TS = Euler}) \\ y_{n+1} &= y_n + \triangle t f^n + \frac{\triangle t^2}{2}\left[f_t^n + f_y^n f^n\right], && (2^{nd} \text{ order TS method}) \\ y_{n+1} &= y_n + \triangle t f^n + \frac{\triangle t^2}{2}\left[f_t^n + f_y^n f^n\right] && (3^{rd} \text{ order TS method}) \\ & +\frac{\triangle t^3}{3!}\left[f_{tt}^n + 2f_{ty}^n f^n + f_{yy}^n f^n + f_y^n f_t^n + f_y^n f_y^n f^n\right], \end{aligned}$$

and so on to all orders.

Taylor series methods are not used because they are far too expensive for systems of equations. Better methods of comparable accuracy are available.

EXERCISE 17. *Write out explicitly the second order Taylor series method for the 2×2 system:*

$$\begin{aligned} x'(t) &= f(t,x(t),y(t)) \text{ for } t > 0 \text{ and } x(0) = x_0 \\ y'(t) &= g(t,x(t),y(t)) \text{ for } t > 0 \text{ and } y(0) = y_0. \end{aligned}$$

EXERCISE 18. *For a system of N equations, count one scalar function evaluation and one work unit. Calculate the work involved per step for TS methods of order 1,2 and 3 as a function of N.*

1.1. The discoverers.

Adapted from Wikipedia:

Brook Taylor FRS (1685 – 1731) was an English mathematician best known for Taylor's theorem and the Taylor series. Taylor's Methodus Incrementorum Directa et Inversa (1715) added the "calculus of finite differences". He used it to determine the form of movement of a vibrating string. The same work contained the celebrated formula known as Taylor's formula, the importance of which remained unrecognized until J. L. Lagrange realized its powers and termed it "the main foundation of differential calculus".

2. Implicit methods are quite acceptable

> Julius Sextus Frontinus
>
> Inventions have long since reached their limit, and I see no hope for further development.
>
> –Highly regarded engineer in Rome, 1st century A.D.

For examples, consider the **Backward or Implicit Euler Method**:

$$y'(t_{n+1}) = \frac{y(t_{n+1}) - y(t_n)}{\triangle t} + \mathcal{O}(\triangle t)$$

this gives

$$\frac{y_{n+1} - y_n}{\triangle t} = f(t_{n+1}, y_{n+1}).$$

and all the BDF methods. These methods are *implicit*, meaning *every step requires solving a nonlinear equation.*

Definition 9 (Implicit and Explicit methods). *An* ***implicit method*** *for $y' = f(t, y)$ is one wherein every step requires solving a nonlinear equation of system. Equivalently, it is one that has y_{n+1} as an argument of $f(\cdot, \cdot)$ somewhere. An* ***explicit method*** *is one that is not implicit; calculating y_{n+1} requires only function evaluations.*

Generally, solving nonlinear equations for each step is not so difficult for initial value problems. For example, for the backward Euler method, given y_n, we must solve for y_{n+1} :

$$y_{n+1} - \triangle t f(t_{n+1}, y_{n+1}) = y_n.$$

The simplest way to solve is by simple iteration taking advantage of the time step being small and requiring only repeated evaluations of $f(t, y)$:

$$\begin{aligned} \text{Guess:} \quad & y^{old} = y_n \\ \text{Until satisfied:} \quad & y^{new} - \triangle t f(t_{n+1}, y^{old}) = y_n \\ \text{When satisfied set:} \quad & y^{n+1} = y^{new}. \end{aligned}$$

"Satisfaction" is measured as usual by small residual

$$|y^{new} - \triangle t f(t_{n+1}, y^{old}) - y_n| \text{ small}$$

and small update

$$|y^{new} - y^{old}| \text{ small.}$$

Convergence of the simple iteration for small enough time step follows from the contraction mapping theorem[1].

[1] **Contraction Mapping Theorem.** If $y^* = G(y^*)$, $G(y)$ is C^1 and $|G'(y^*)| < 1$ then the simple iteration $y^{new} = G(y^{old})$ converges locally to y^*.

PROPOSITION 5. *Let $f(t,y), f_y(t,y)$ both be continuous. If $\triangle t|f_y(t_{n+1},y_{n+1})| < 1$ then the simple iteration $y^{new} - \triangle t f(t_{n+1},y^{old}) = y_n$ converges to y_{n+1} provided the initial guess is close enough.*

3. Any finite difference approximation to y' gives a method

We give a few examples.

3.1. Euler's Method. Euler's method arises from the difference approximation

$$y'(t_n) = \frac{y(t_{n+1}) - y(t_n)}{\triangle t} + \mathcal{O}(\triangle t)$$

this gives

$$\frac{y_{n+1} - y_n}{\triangle t} = f(t_n, y_n) \text{ or}$$
$$y_{n+1} = y_n + \triangle t f(t_n, y_n).$$

This is an *explicit, single step* method.

DEFINITION 10 (Single step and multi-step methods). *A **single step method** is one wherein the formula for y_{n+1} involves only the y value y_n . A **multi-step method** is one wherein the formula for y_{n+1} involves more past y values than just y_n .*

3.2. BDF2 (backward differentiation formula, order 2) Method. The second order backward approximation to $y'(t)$ is

$$y'(t_{n+1}) = \frac{3y(t_{n+1}) - 4y(t_n) + y(t_{n-1})}{2\triangle t} + \mathcal{O}(\triangle t^2)$$

This gives

$$\frac{3y_{n+1} - 4y_n + y_{n-1}}{2\triangle t} = f(t_{n+1}, y_{n+1}) \tag{BDF2}$$

Euler's method is an explicit single step method. In contrast, **BDF2** is an implicit **multi-step method**. The formula for y_{n+1} involves more y values than just y_n and a nonlinear system must be solved each step too get the new approximation.

3.3. BDF3 (backward differentiation formula, order 3) Method. This is based on the difference quotient

$$y'(t_{n+1}) = \frac{y(t_{n+1}) - (18/11)y(t_n) + (9/11)y(t_{n-1}) - (2/11)y(t_{n-2})}{(6/11)\triangle t} + \mathcal{O}(\triangle t^3).$$

This gives

$$\frac{y_{n+1} - (18/11)y_n + (9/11)y_{n-1} - (2/11)y_{n-2}}{(6/11)\triangle t} = f(t_{n+1}, y_{n+1}) \tag{BDF3}$$

In this way **BDF methods** of any order can be generated. The BDF methods of order 4,5,6 are:

$$\text{(BDF456)}\quad BDF4: y_{n+4}-(48/25)y_{n+3}+(36/25)y_{n+2}-(16/25)y_{n+1}+(3/25)y_n$$
$$=(12/25)\triangle t f(t_{n+4},y_{n+4}) \qquad (3.1)$$
$$BDF5: insert: y_{n+5}-()y_{n+4}+()y_{n+3}-()y_{n+2}+()y_{n+1}+()yn$$
$$=()\triangle t f(t_{n+5},y_{n+5}) \qquad (3.2)$$
$$BDF6: insert: y_{n+6}-()y_{n+5}-()y_{n+4}+()y_{n+3}-()y_{n+2}+()y_{n+1}+()yn$$
$$=()\triangle t f(t_{n+6},y_{n+6}) \qquad (3.3)$$

Those of order ≤ 6 are very good methods and used frequently. Those with order > 6 are unstable and not used.

Exercise 19. *Find the precise timestep condition required for simple iteration to converge for the nonlinear equation arising from BDF2 and BDF3.*

Exercise 20. *Analyze 0-stability of BDF2 and BDF3.*

3.4. Leapfrog Method. The LF method is

$$y'(t_n)=\frac{y(t_{n+1})-y(t_{n-1})}{2\triangle t}+\mathcal{O}(\triangle t^2)$$

this gives

$$\frac{y_{n+1}-y_{n-1}}{2\triangle t}=f(t_n,y_n).$$

The leapfrog method is useful for a few very specific problems of the form

$$y'+\Lambda y=f(t) \text{ where } \Lambda \text{ is skew symmetric}$$

but is otherwise unstable. It also has issues with non-autonomous systems and variable timesteps.

To see why it is useful for those specific systems, consider

$$\begin{aligned} x' &= y \\ y' &= -x \end{aligned}$$

This has the exact conservation property that

$$x^2(t)+y^2(t)=x^2(0)+y^2(0).$$

The leapfrog approximation has a similar and related exact conservation property that

$$x_n^2+y_n^2+x_{n-1}^2+y_{n-1}^2+\triangle t\,(x_ny_{n-1}-x_{n-1}y_n) = x_1^2+y_1^2+x_0^2+y_0^2+\triangle t\,(x_1y_0-x_0y_1)$$

for all n>1.

Exercise 21. *Prove the claimed exact conservation property for Leapfrog.*

3.4.1. *Variable time-step extensions of Leapfrog.* For variable time-steps stability of LF is not clear and has been proven to fail in several realizations of the variable step scheme. Indeed, work of Calvo and Sanz-Serna 1993 and Skeel and Gear 1992 concluded that

"Variable time-steps seriously degrades symplectic integrators."

Several ideas have been proposed. Consider

$$y' + \Lambda y = f(t) \text{ where } \Lambda \text{ is skew symmetric.}$$

Let

$$\begin{aligned} k_{n+1} &: = t_{n+1} - t_n, \\ k_n &: = t_n - t_{n-1}, \text{ and} \\ \omega_{n+1} &= \frac{t_{n+1} - t_n}{t_n - t_{n-1}} = \frac{k_{n+1}}{k_n}. \end{aligned}$$

1. Huang and Leimkuhler, 1997, proposed adapting by stretching the time-step. They apply variants of LF to the coupled system

$$\begin{aligned} \frac{dy}{ds} &= \frac{1}{R(y)} \Lambda y \text{ and} \\ \frac{dt}{ds} &= \frac{1}{R(y)} \text{ where} \\ 0 < \min \triangle t &\leq R(y) \leq \max \triangle t < \infty. \end{aligned}$$

2. Centering the scheme at $t^* = (t_{n+1} + t_{n-1})/2$ we have another variable step CNLF method

$$\begin{aligned} \frac{y^{n+1} - y^{n-1}}{k_{n+1} + k_n} &= \Lambda \left(\frac{1+\omega}{2} y^n + \frac{1-\omega}{2} y^{n-1} \right) \\ \text{where } \omega &= \omega_{n+1}. \end{aligned}$$

Here y^n is an approximation to $y(t_n)$ and not at t^*.

3. Note that

$$\begin{aligned} \frac{1}{k_{n+1}} \left[\frac{1}{1+\omega} y^{n+1} - (1-\omega) y^n - \frac{\omega^2}{1+\omega} y^{n-1} \right] &= \\ = \frac{\frac{1}{\omega} y^{n+1} + (1 - \omega - \frac{1}{\omega}) y^n - \omega y^{n-1}}{k_{n+1} + k_n} &= \\ = y'(t_n) + O(k^2). \end{aligned}$$

LF can be centered at t_n by

$$\frac{1}{k_{n+1}} \left[\frac{1}{1+\omega} y^{n+1} - (1-\omega) y^n - \frac{\omega^2}{1+\omega} y^{n-1} \right] = \Lambda \left(y^n \right)$$
$$\text{where } \omega = \omega_{n+1}.$$

4. The simple but first order accurate extension:

$$\frac{y^{n+1} - y^{n-1}}{k_{n+1} + k_n} = \Lambda \left(y^n \right) \ .$$

This is first order accurate when the stepsize varies.

5. The Variable Step CNLF of Wang[2005]. In his MS thesis, equation (2.17) page 8, in 2005 Dong Wang derived the variable step CNLF scheme:

$$\frac{1}{k_{n+1}} \left[\frac{1}{1+\omega} y^{n+1} - (1-\omega) y^n - \frac{\omega^2}{1+\omega} y^{n-1} \right] = \Lambda \left(y^n \right)$$
$$\text{where } \omega = \omega_{n+1}.$$

The analysis of any of these extensions seems to be an open problem.

3.5. A Method for second order IVPs.

Consider the second order IVP:

$$\begin{aligned} s''(t) &= g(t, s(t), s'(t)), t > 0 \\ & s(0), s'(0) \text{ known.} \end{aligned}$$

This can be solved by writing it as a first order system but also as a single, scalar second order equation using the difference approximations:

$$\begin{aligned} s''(t_n) &= \frac{s(t_{n+1}) - 2s(t_n) + s(t_{n-1})}{\triangle t^2} + O(\triangle t^2), \\ s'(t_n) &= \frac{s(t_n) - s(t_{n-1})}{\triangle t} + O(\triangle t). \end{aligned}$$

This gives the method: given s_0, s_1 for $n \geq 1$:

$$\frac{s_{n+1} - 2s_n + s_{n-1}}{\triangle t^2} = g(t_n, s_n, \frac{s_n - s_{n-1}}{\triangle t})$$

To solve it as a first order system we write

$$\begin{aligned} x &= s \\ y &= s' \end{aligned}$$

so that $x' = y, y' = s''$ which is known from the equation. This gives the system

$$\begin{aligned} x' &= y, \\ y' &= g(t, x, y) \\ & x(0), y(0) \text{ known.} \end{aligned}$$

Euler's method for the first order system would then be

$$\begin{aligned} \frac{x_{n+1} - x_n}{\triangle t} &= y_n, \\ \frac{y_{n+1} - y_n}{\triangle t} &= g(t_n, x_n, y_n). \end{aligned}$$

Exercise 22. *In the above Euler method for the equivalent first order system*

$$\begin{aligned} \frac{x_{n+1} - x_n}{\triangle t} &= y_n, \\ \frac{y_{n+1} - y_n}{\triangle t} &= g(t_n, x_n, y_n). \end{aligned}$$

Eliminate y_n by replacing it in the second equation everywhere by $\frac{x_{n+1}-x_n}{\triangle t}$. Simplify and compare the resulting method to

$$\frac{s_{n+1} - 2s_n + s_{n-1}}{\triangle t^2} = g(t_n, s_n, \frac{s_n - s_{n-1}}{\triangle t})$$

4. Any numerical integration formula gives a method

If we integrate $y' = f(t, y)$ over some time interval $[a, b]$ we get

$$y(b) - y(a) = \int_a^b f(t, y(t))dt.$$

A numerical integration method replaces the RHS by a weighted sum of function values. There results a method to advance in time from $t = a$ to $t = b$. A few examples are given next.

4.1. Revisiting Leapfrog. The midpoint rule (the 1 point Gauss rule) yields the leapfrog method. Indeed,

$$\begin{aligned} y(t_{n+1}) - y(t_{n-1}) &= \int_{t_{n-1}}^{t_{n+1}} f(t, y(t))dt \\ &\simeq f(t_n, y(t_n)) \cdot (t_{n+1} - t_{n-1}) + \mathcal{O}(\triangle t^2). \end{aligned}$$

which yields, as $t_{n+1} - t_{n-1} = 2\triangle t$, the leapfrog method

$$\text{(LeapFrog)} \qquad \frac{y_{n+1} - y_{n-1}}{2\triangle t} = f(t_n, y_n).$$

4.2. The trapezoid rule. The trapezoid rule yields an especially interesting method. We approximate the integral with the usual trapezoid rule by

$$\begin{aligned} y(t_{n+1}) - y(t_n) &= \int_{t_n}^{t_{n+1}} f(t, y(t))dt \\ &\simeq (t_{n+1} - t_n)\frac{f(t_{n+1}, y(t_{n+1})) + f(t_n, y(t_n))}{2} + \mathcal{O}(\triangle t^2). \end{aligned}$$

This yields, as $t_{n+1} - t_n = \triangle t$, the method know by various names including the trapezoid rule, trapezium rule, the $1-1$ Padé method, the Crank-Nicolson method[2] ...

$$\text{(Trapezoid Rule)} \qquad \frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_n, y_n).$$

4.2.1. *About the discoverers.* from : http://www-history.mcs.st-andrews.ac.uk/Biographies/Pade.html

Henri Padé was born in Abbeville which is a town northwest of Amiens in the Picardy region of northern France. He attended school in his home town and obtained his baccalaureate in 1881 at the age of seventeen. He then went to Paris to continue his education at the Lycée St. Louis where he spent two years preparing to sit the university entrance examinations.

From http://www-history.mcs.st-andrews.ac.uk/Obits2/Crank_Telegraph.html:

Professor John Crank, who died on October 3 aged 90, was a leading figure in computational mathematics and mathematical modelling, best-known for his work with Phyllis Nicolson on the numerical solution of the heat equation.

By the time he went to Brunel in 1957 he was already a recognized expert in the numerical solution of partial differential equations, particularly the heat equation, which stretches back two centuries to J.B.J. Fourier, one of Napoleon's mathematicians.

In the 1940s the calculations required to solve this, the most common of partial differential equations, were carried out on simple mechanical desk machines, and required an enormous amount of the most exacting work. Crank said that to "burn a piece of wood numerically" in those days – without computers – could take a week.

His work with Phyllis Nicolson, a near contemporary of his as a student at Manchester University, on the numerical solution of the heat equation sprang from a method for solving this problem which had been proposed by LF Richardson in 1910.

[2] from: J Crank and P Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Proc. Cambridge Philos. Soc. 43 (1947). 50-67.

Richardson's method yielded a numerical solution which was very easy to compute, but which was numerically unstable and thus useless. The instability was not recognized until lengthy numerical computations were carried out by Crank, Nicolson, and others. Crank and Nicolson devised a method which is numerically stable and which turned out to be so fundamental and useful that it is a cornerstone of every discussion of the numerical solution of partial differential equations.

Since its inception, it has been used routinely in computer codes, with applications ranging from options pricing and oceanography to pattern formation and petrology.

John Crank was born on February 6 1916 at Hindley, Lancashire, the only son of a carpenter's pattern-maker. He studied at Manchester University, where he gained his B.Sc. and M.Sc. At Manchester he was a student of the physicist Lawrence Bragg, the youngest-ever winner of a Nobel prize, and of Douglas Hartree, a leading numerical analyst.

Crank was seconded to war work during the Second World War, in his case to work on ballistics. This was followed by employment as a mathematical physicist at Courtaulds Fundamental Research Laboratory from 1945 to 1957. He was then, from 1957 to 1981, professor of mathematics at Brunel University (initially Brunel College in Acton).

Crank published only a few research papers, but they were seminal. Even more influential were his books. His work at Courtaulds led him to write The Mathematics of Diffusion, a much-cited text that is still an inspiration for researchers who strive to understand how heat and mass can be transferred in crystalline and polymeric material. He subsequently produced Free and Moving Boundary Problems, which encompassed the analysis and numerical solution of a class of mathematical models that are fundamental to industrial processes such as crystal growth and food refrigeration.

As a specialist in numerical mathematics, Crank was a figure of particular importance at a time when that area was often regarded by the mathematical establishment as being rather slight, and he attracted a cadre of devoted students and young collaborators. He was a founder member of the Institute of Mathematics and its Applications, and a key player in the setting up of the Royal Institution Mathematics programme.

Crank was a fine raconteur and a good listener, with a kindly sense of humour, admired and respected by his colleagues and loved by his many students.

He met his wife, Joan, to whom he was married for 63 years, on a Holiday Fellowship walking holiday. They retained an enthusiasm for walking and were also keen gardeners.

His retirement gift to Brunel was a garden; and recently the university named a building after him. Joan Crank died in 2005; he is survived by their two children.

From http://www-history.mcs.st-andrews.ac.uk/Biographies/Nicolson.html:

Phyllis Nicolson's maiden name was Lockett. She was educated at Stockport High School and received the degrees of B.Sc. (1938) and M.Sc. (1939) and Ph.D. in Physics (1946) from Manchester University and was a research student (1945-46) and research fellow (1946-49) at Girton College, Cambridge. In 1942 she married Malcolm Nicolson. She had a strong wish to have her first child before reaching thirty, and she achieved this ambition with a day to spare. After her husband's untimely death in a train crash in 1952, she was appointed to fill his lectureship in

Physics at Leeds University. In 1955 she married Malcolm McCaig, who was also a physicist.

During the period 1940-45 she was a member of a research group in Manchester University directed by Douglas Hartree, working on wartime problems for the Ministry of Supply, one being concerned with magnetron theory and performance. Phyllis Nicolson is best known for her joint work with John Crank on the heat equation, where a continuous solution $u(x,t)$ is required which satisfies the second order partial differential equation

$u_t - u_{xx} = 0$

for $t > 0$, subject to an initial condition of the form $u(x,0) = f(x)$ for all real x. They considered numerical methods which find an approximate solution on a grid of values of x and t, replacing $u_t(x,t)$ and $u_{xx}(x,t)$ by finite difference approximations. One of the simplest such replacements was proposed by L. F. Richardson in 1910. Richardson's method yielded a numerical solution which was very easy to compute, but alas was numerically unstable and thus useless. The instability was not recognized until lengthy numerical computations were carried out by Crank, Nicolson and others. Crank and Nicolson's method, which is numerically stable, requires the solution of a very simple system of linear equations (a tridiagonal system) at each time level.

Nicolson died of breast cancer in 1968

5. Methods can be combined in different ways to give new methods

A few examples.

5.1. Extrapolation can be used to generate new methods.

For example, linear extrapolation to the new time is

$$f(t_{n+1}) = 2f(t_n) - f(t_{n-1}) + O(\triangle t^2).$$

If the trapezoid rule is used, a nonlinear equation must be solved every time step. If the above extrapolation is used to replace $f(t_{n+1}, y_{n+1})$ for the RHS we get

$$\begin{aligned}\frac{y_{n+1} - y_n}{\triangle t} &= \frac{1}{2}f(t_{n+1}, y_{n+1}) + \frac{1}{2}f(t_n, y_n)\\ &\simeq \frac{1}{2}\left(2f(t_n, y_n) - f(t_{n-1}, y_{n-1})\right) + \frac{1}{2}f(t_n, y_n).\end{aligned}$$

This gives the method known as **AB2** = second order **Adams-Bashforth**

$$\text{(AB2)} \qquad \frac{y_{n+1} - y_n}{\triangle t} = \frac{3}{2}f(t_n, y_n) - \frac{1}{2}f(t_{n-1}, y_{n-1}).$$

There exists a whole family of **AB methods** of different orders.

Since extrapolation is easily done on non-uniform points, AB2 has an easy extension to variable timestep:

(Variable Step AB2)

$$y_{n+1} = y_n + \frac{\triangle t_n}{2}\left[(2 + \frac{\triangle t_n}{\triangle t_{n-1}})f(t_n, y_n) - \frac{\triangle t_n}{\triangle t_{n-1}}f(t_{n-1}, y_{n-1})\right].$$

5.2. CN-AB2. AB2 is commonly used with the trapezoid rule. This combination (another IMEX = Implicit-Explicit method) is used in some applications when the system takes the form

$$y' = f(t, y(t)) + g(t, y(t)).$$

The combination is called CN-AB2 and not TR-AB2 because in the applications where it is commonly used the trapezoid rule is called the CN = Crank-Nicolson method. CN-AB2 is then

$$\frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_{n-1}, y_{n-1}) + \frac{3}{2} g(t_n, y_n) - \frac{1}{2} g(t_{n-1}, y_{n-1}).$$

5.3. CNLF. Combining the Trapezoid rule[3] with doubled timestep with Leapfrog. This combination (known as an IMEX = Implicit-Explicit method) is commonly used in some applications when the system takes the form

$$\overrightarrow{y}' = \overrightarrow{f}(t, \overrightarrow{y}(t)) + \Lambda \overrightarrow{y}(t),$$

where Λ is skew symmetric, i.e., $\Lambda^T = -\Lambda$.

The combination CNLF is then

$$\text{(CNLF)} \qquad \frac{y_{n+1} - y_{n-1}}{2\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_{n-1}, y_{n-1}) + \Lambda y_n.$$

5.4. Predictor-Corrector methods. For example, the trapezoid rule reads

$$\frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_n, y_n).$$

Each step requires the solution of a nonlinear system since $f(t_{n+1}, y_{n+1})$ contains y_{n+1}. If y_{n+1} is replaced by its value predicted by Euler's method we have

$$\text{(Heun's Method)} \qquad \begin{array}{lr} \text{Predict:} & \frac{y^P_{n+1} - y_n}{\triangle t} = f(t_n, y_n) \\ \text{Correct:} & \frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y^P_{n+1}) + \frac{1}{2} f(t_n, y_n) \end{array}$$

In this form is is sometimes called Heun's method. If y^P_{n+1} is eliminated (replace y^P_{n+1} by its value $y^P_{n+1} = y_n + \triangle t f(t_n, y_n)$) we get

$$\text{(RK2)} \qquad \frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_n + \triangle t f(t_n, y_n)) + \frac{1}{2} f(t_n, y_n),$$

which is the second order Runge-Kutta method (RK2).

If the correction is performed twice,

$$\begin{array}{lr} \text{Predict:} & \frac{y^P_{n+1} - y_n}{\triangle t} = f(t_n, y_n) \\ \text{Correct twice:} & \begin{array}{r} \frac{y^C_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y^P_{n+1}) + \frac{1}{2} f(t_n, y_n) \\ \frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y^C_{n+1}) + \frac{1}{2} f(t_n, y_n) \end{array} \end{array}$$

instead of once as above, then a new method results. *Changing the number of correction steps gives a new method.*

[3] It is called CNLF and not TRLF because in the applications where it is commonly used the trapezoid rule is called the CN = Crank-Nicolson method.

5.4.1. *About the discoverers.* Adapted from Wikipedia:

Karl Heun (1859 - 1929) was a German mathematician who introduced Heun's equation, Heun functions, and Heun's method. He received his Habilitierung in 1886 in Munich with the thesis Über lineare Differentialgleichungen zweiter Ordnung, deren Lösungen durch den Kettenbruchalgorithmus verknüpft sind.

Carl David Tolmé Runge (1856–1927) was a German mathematician, physicist, and spectroscopist. He was codeveloper of the Runge–Kutta method. In 1880, he received his Ph.D. in mathematics. His interests included mathematics, spectroscopy, geodesy, and astrophysics.

Martin Wilhelm Kutta (1867 – 1944) was a German mathematician. Kutta was born in Pitschen, Upper Silesia (today Byczyna, Poland). In 1901, he codeveloper the Runge-Kutta method. He is also remembered for the Zhukovsky-Kutta airfoil, the Kutta-Zhukovsky theorem and the Kutta condition in aerodynamics.

5.5. Weighted averages of methods give new methods. For example, for forward and backward Euler

$$\begin{aligned}\frac{y_{n+1}-y_n}{\triangle t} &= f(t_n,y_n), &\text{(Forward Euler)}\\ \frac{y_{n+1}-y_n}{\triangle t} &= f(t_{n+1},y_{n+1}). &\text{(Backward Euler)}\end{aligned}$$

Pick θ between 0 and 1 and take the weighted average $\theta \times Euler + (1-\theta) \times (Backward - Euler)$ gives the method known as the $\theta-$Method

$$\frac{y_{n+1}-y_n}{\triangle t} = \theta f(t_n,y_n) + (1-\theta) f(t_{n+1},y_{n+1}).$$

Parameters can be picked for accuracy, stability or other reasons.

5.6. Other combinations are possible. For example, for the system of two equations

$$\begin{aligned} x' &= f(t,x,y),\\ y' &= g(t,x,y),\end{aligned}$$

Combinations of Euler and backward Euler can be used for equations 1 and 2: given x_n, y_n

$$\begin{aligned}\frac{x_{n+1}-x_n}{\triangle t} &= f(t_n,x_n,y_n), &\text{(Forward Euler)}\\ \frac{y_{n+1}-y_n}{\triangle t} &= g(t_{n+1},x_{n+1},y_n). &\text{(Backward Euler)}\end{aligned}$$

This combination is explicit when done in the above order. This sort of stepping through the individual equations in a system can be done in many different ways. For example, the above combination of FE BE can be performed in both orders of $x-y$ equations giving two different approximations to each variable that are then averaged.

6. Runge Kutta Methods

Runge-Kutta methods are among the most successful general purpose methods so we shall devote considerable time to their development. The first RK methods are (surprisingly) due to Heun who viewed them as extending Simpson's rule from $y' = f(t)$ so that

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t)dt$$

to $y' = f(t, y(t))$. Runge had a different and clearer development for scalar problems while Kutta extended the methods to systems and to include the methods Heun was developing. The general idea is that, just as evaluating $f(t, y)$ gives y', evaluating $f(\cdot, \cdot)$ at several points gives information on how y' changes, i.e., on higher derivatives of $y(t)$.

Undetermined parameters can be inserted in various places in a formula. The formula expanded in a Taylor series which is matched as far as possible to the Taylor series of the true solution. This is a simple process but more detailed. As an example, the **RK2 method** can be derived as follows. Introduce 4 free parameters, a, b, α, β and write

$$y_{n+1} = y_n + \triangle t \left[af(t_n, y_n) + bf(t_n + \alpha\triangle t, y_n + \beta\triangle t f(t_n, y_n))\right].$$

The **local truncation error** (**LTE**) of the above method is

$$\mathbf{LTE} := y(t+\triangle t) - (y(t) + \triangle t\left[af(t, y(t)) + bf(t+\alpha\triangle t, y(t) + \beta\triangle t f(t, y(t)))\right]),$$

where $y(t)$ is the true solution. The 4 free parameters in the scheme are chosen to maximize accuracy (by minimizing the local truncation error). The above RHS can be expanded in a Taylor series and the free parameters chosen to minimize the methods local truncation error.

6.1. Derivation of RK2. Indeed, we have previously calculated the Taylor series of the true solution

$$y(t+\triangle t) = y(t) + \triangle t f(t, y(t)) + \frac{\triangle t^2}{2}\left[f_t(t, y(t)) + f_y(t, y(t))f(t, y(t))\right] +$$

(Exact TS)

$$\frac{\triangle t^3}{3!}\left[\begin{array}{c} f_{tt}(t, y(t)) + 2f_{ty}(t, y(t))f(t, y(t)) + \\ +f_{yy}(t, y(t))f(t, y(t)) + f_y(t, y(t))f_t(t, y(t)) + f_y^2(t, y(t))f(t, y(t)) \end{array}\right]$$
$$+\mathcal{O}(\triangle t^4).$$

Another Taylor expansion gives

$$f(t+\alpha\triangle t, y(t) + \beta\triangle t f(t, y(t))) = f(t, y(t))$$
$$+\alpha\triangle t f_t(t, y(t)) + \beta\triangle t f(t, y(t)))f_y(t, y(t)) +$$
$$\triangle t^2\left[\frac{\alpha^2}{2}f_{tt}(t, y(t)) + \alpha\beta f(t, y(t)))f_{ty}(t, y(t)) + \frac{\beta^2 f(t, y(t))^2}{2}f_{yy}(t, y(t))\right]$$
$$+O(\triangle t^3)$$

Inserting this expansion in the RHS of the methods LTE gives

$$\text{LTE} := y(t+\triangle t) - (y(t) + \triangle t\,[af(t,y(t)) + bf(t+\alpha\triangle t, y(t)+\beta\triangle t f(t,y(t)))]) =$$

$$= y(t+\triangle t) - \left(y(t) + \triangle t \left[af(t,y(t)) + b \left\{ \begin{array}{c} f(t,y(t)) + \alpha\triangle t f_t(t,y(t)) + \\ +\beta\triangle t f(t,y(t)))f_y(t,y(t)) + \\ +\frac{\alpha^2\triangle t^2}{2} f_{tt}(t,y(t)) + \\ +\alpha\triangle t\beta\triangle t f(t,y(t)))f_{ty}(t,y(t)) + \\ +\frac{\beta^2(\triangle t f(t,y(t))))^2}{2} f_{yy}(t,y(t)) + \\ +O(\triangle t^3) \end{array} \right\} \right] \right)$$

Collecting terms and simplifying where possible (including suppressing all the arguments $(t, y(t))$ of f and all its partial derivatives)gives

$$\text{LTE} :=$$

$$= y(t+\triangle t) - y(t) - \left[\begin{array}{c} (a+b)\triangle t f + b\triangle t^2(\alpha f_t + \beta f f_y) + \\ +b\triangle t^3(\frac{\alpha^2}{2} f_{tt} + \alpha\beta f f_{ty} + \frac{\beta^2}{2} f_{yy} f)) + O(\triangle t^4) \end{array} \right]$$

The LTE can be minimized by picking the 4 free parameters so that the above expansion matches the Taylor expansion of the true solution ((Exact TS) above) as far out as possible. We compare his expansion to the expansion of the true solution below:

$$\begin{array}{cc} \begin{array}{c}\text{True}\\ \text{Solution:}\end{array} & y(t+\triangle t) - y(t) - \left[\begin{array}{c} \triangle t f + \frac{\triangle t^2}{2}\left(f_t + f_y f\right) + \\ \frac{\triangle t^3}{3!}\left(f_{tt} + 2f_{ty} f + f_{yy} f + f_y f_t + f_y^2\right) + \mathcal{O}(\triangle t^4) \end{array} \right] \\ \begin{array}{c}\text{Local}\\ \text{Truncation}\\ \text{error:}\end{array} & y(t+\triangle t) - y(t) - \left[\begin{array}{c} (a+b)\triangle t f + b\triangle t^2(\alpha f_t + \beta f f_y) + \\ b\triangle t^3(\frac{\alpha^2}{2} f_{tt} + \alpha\beta f f_{ty} + \frac{\beta^2}{2} f_{yy} f)) + O(\triangle t^4) \end{array} \right] \end{array}$$

To minimize the local truncation error (maximize accuracy) we must choose a, b, α, β to satisfy

$$\text{(Order Conditions)} \qquad a + b = 1 \text{ and } b\alpha = \frac{1}{2} \text{ and } b\beta = \frac{1}{2}.$$

This is three equations for four variables. There are an infinite number of solutions and *any solution is an RK2 method*. One commonly use solution is

$$a = b = \frac{1}{2} \text{ and } \alpha = \beta = 1.$$

These values are so commonly used that it is often (incorrectly) called "*the*" RK2 method and also Heun's method.

6.2. The standard RK2 method. This **standard RK2 method** (with $a = b = \frac{1}{2}$ and $\alpha = \beta = 1$) is usually written in stages (as it is programmed) as:

$$\begin{array}{rcl} given & : & y_n \\ k_1 & = & \triangle t f(t_n, y_n) \\ k_2 & = & \triangle t f(t_n + \triangle t, y_n + k_1) \\ y_{n+1} & = & y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2. \end{array} \qquad \text{(RK2 in stages)}$$

RK2 is second order accurate (the LTE is $O(\triangle t^3)$), explicit and costs only 2 function evaluations per step.

One commonly use solution is

$$a = b = \frac{1}{2} \text{ and } \alpha = \beta = 1.$$

These values are so commonly used that it is often (incorrectly) called "*the*" RK2 method.

6.3. The Ralston rule. The **Ralston Rule RK2 method** is another solution of the RK2 equations. Ralston[4] derived it by finding an estimate of the error in the general RK2 method and then minimizing it with respect to the method's parameters. His analysis gave a concrete method and numerical tests cobnfirmed that it is indeed generally the most accurate RK2 method, sometimes by a little and *sometimes by a lot*. Here is a simplified idea of hios approach. Suppose we apply the general RK2 method to the IVP

$$\begin{aligned} y' &= t^2, y(0) = 1 \\ \text{true solution} &: y(t) = \frac{1}{3}t^3. \end{aligned}$$

The general RK2 method (where $a + b = 1$ and $b\alpha = \frac{1}{2}$ and $b\beta = \frac{1}{2}$) is:

$$\begin{aligned} \text{given} &: y_n \\ k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \alpha \triangle t, y_n + \beta k_1) \\ y_{n+1} &= y_n + a k_1 + b k_2. \end{aligned}$$

with $f(t) = t^2$

$$\begin{aligned} \text{given} &: y_0 = 1, t_0 = 0, f(t, y) = t^2 \\ k_1 &= \triangle t \cdot t_0^2 = 0 \\ k_2 &= \triangle t (t_0 + \alpha \triangle t)^2 = \triangle t^3 \alpha^2 \\ y_1 &= y_0 + a k_1 + b k_2 = 0 + 0 + b \triangle t^3 \alpha^2. \end{aligned}$$

If we pick the parameters so that $y_1 = true - value = \frac{1}{3}(\triangle t)^3$ we must have

$$\begin{aligned} b\alpha^2 &= \frac{1}{3} \text{ in addition to} \\ a + b &= 1 \text{ and } b\alpha = \frac{1}{2} \text{ and } b\beta = \frac{1}{2}. \end{aligned}$$

The solution to these equations is $a = \frac{1}{4}, b = \frac{3}{4}$ and $\alpha = \beta = \frac{2}{3}$. This gives the **Ralston rule for RK2**.

Written in stages it is:

$$\begin{aligned} \text{given} &: y_n \qquad \text{(Ralston Rule in stages)} \\ k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \frac{2}{3} \triangle t, y_n + \frac{2}{3} k_1) \\ y_{n+1} &= y_n + \frac{1}{4} k_1 + \frac{3}{4} k_2. \end{aligned}$$

[4] A. Ralston, Runge-Kutta methods with minimum error bounds, Math. Comp., 16 (1962), 431-437

The Ralston rule corresponds to the solution:

$$a = \frac{1}{4}, b = \frac{3}{4} \text{ and } \alpha = \beta = \frac{2}{3}.$$

These values do indeed satisfy the second order conditions:

$$\text{(Ralston rule Order Conditions)} \qquad a + b = 1 \text{ and } b\alpha = \frac{1}{2} \text{ and } b\beta = \frac{1}{2},$$

$$a + b = \frac{1}{4} + \frac{3}{4} = 1 \text{ and } b\alpha = \frac{3}{4}\frac{2}{3} = \frac{1}{2} \text{ and } b\beta = \frac{3}{4}\frac{2}{3} = \frac{1}{2}.$$

Interestingly, in his 1960 paper Ralston also derives maximally accurate RK3 and RK4 methods but the above simple, second order one seems to be the one that persisted in the literature attached to his name.

Exercise 23. *Do 1 step with the general RK4 method (with $a + b = 1$ and $b\alpha = \frac{1}{2}$ and $b\beta = \frac{1}{2}$) for the problem*

$$\begin{aligned} y' &= f(t), f(t) = 6t^5, \\ y(0) &= 0, \\ sol &: y(t) = t^6. \end{aligned}$$

Find the values of the RK4 parameters that give the exact answer after 1 step. Check Ralston's paper and see if your parameters agree with the ones he found.

6.4. The general RK method. An **s-step RK method** takes the general form: given y_n,

$$k_i = \triangle t f(t_n + c_i \triangle t, y_n + \sum_{j=1}^{s} a_{ij} k_j) \text{ for } i = 1, \cdots, s.$$

$$y_{n+1} - y_n = \triangle t \sum_{i=1}^{s} b_i k_i$$

Thus, an RK method is determined by specifying the parameters b_i, c_i, a_{ij}. This was Heun's great idea: to determine the unknown parameters a_{ij}, c_i, b_i to maximize accuracy. He was motivated by Gauss's idea for numerical integration where free parameters are inserted and then optimized to derive the Gauss rules.

These parameters are determined by the twin constraints of high consistency and desired stability. RK methods are thus codified by presenting these parameters are an array called the "Butcher array" or "Butcher tableau" due to the work of Butcher in 1964:

$$\begin{bmatrix} \overrightarrow{c} & | & A \\ - & | & \\ & | & \overrightarrow{b}^T \end{bmatrix} = \begin{bmatrix} c_i & | & A_{ij} \\ - & | & \\ & | & b_j^T \end{bmatrix}.$$

As an example, Heun's standard RK2, above, is

$$\begin{aligned} b_1 &= b_2 = \frac{1}{2}, \\ c_1 &= 0, c_2 = 1 \\ a_{11} &= 0, a_{12} = 0 \\ a_{21} &= 1, a_{22} = 0. \end{aligned}$$

This corresponds to the Butcher array

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

Indeed, rewriting RK2 as follows (and using subscripts to indicate where in the array each number goes) gives:

$$\begin{aligned} k_1 &= \triangle t f(t_n + 0_{(c_1)}\triangle t, y_n + \big(0_{(A_{1,1})}k_1 + 0_{(A_{1,2})}k_2\big)) \\ k_2 &= \triangle t f(t_n + 1_{(c_2)}\triangle t, y_n + \big(1_{(A_{2,1})}k_1 + 0_{(A_{2,2})}k_2\big)) \\ y_{n+1} &= y_n + \left(\frac{1}{2}\right)_{(b_1)} k_1 + \left(\frac{1}{2}\right)_{(b_2)} k_2. \end{aligned}$$

Written this way, it is clear that in k1, if $A_{1,1} \neq 0$ one must solve a nonlinear equation or system of equations for k_1 and similarly for solving for k_2 in step 2. If $A_{1,2} \neq 0$ then the nonlinear system is twice as large as the nonlinear equations for k_1 and k_2 are coupled.

6.4.1. *About the discoverers.* Adapted from Wikipedia:

John Charles Butcher ONZM (born 1933) is a New Zealand mathematician who is a leader in the development of numerical methods for the solution of ordinary differential equations. Butcher works Runge-Kutta and general linear methods. The Butcher group and the Butcher tableau are named after him. Butcher was awarded the Jones Medal from the Royal Society of New Zealand in 2010, for his "*exceptional lifetime work on numerical methods for the solution of differential equations and leadership in the development of New Zealand mathematical sciences.*"

Adapted from Wikipedia:

Karl Heun (born 3 April 1859, Wiesbaden; died 10 January 1929, Karlsruhe) introduced Heun's equation, Heun functions, and Heun's method. He studied mathematics in Göttingen and Halle.

Adapted from: http://history.computer.org/pioneers/ralston.html; see also http://history.siam.org/oralhisto

Anthony Ralston (born 1930, New York City) received his PhD in mathematics from MIT in 1956. He worked at Bell Labs, the University of Leeds in England, the American Cyanamid Corporation, Stevens Institute of Technology and (for most of his career) the State University of New York at Buffalo. Ralston was the first chair of the Committee on Scientific Freedom and Human Rights of the ACM.

6.5. The explicit midpoint method. The explicit midpoint method is a second order RK method. If we choose $a = 0, b = 1, \alpha = \beta = 1/2$ the method also satisfies the Order Conditions for second order accuracy

$$a + b = 1$$

$$b\alpha = b\beta = \frac{1}{2}.$$

It becomes

$$\begin{aligned} given &: y_n && \text{(Explicit Midpoint)}\\ k_1 &= \triangle t f(t_n, y_n)\\ k_2 &= \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}k_1)\\ y_{n+1} &= y_n + k_2. \end{aligned}$$

This is known as the explicit midpoint method.

Obviously, matching more terms simply requires more parameters. Thus, by including enough free parameters, RK methods of every order of accuracy can be constructed.

6.6. The implicit midpoint method. The implicit midpoint method is an implicit, second order RK method. It is

$$\begin{aligned} given &: y_n \text{ solve for } y_{n+1} && \text{(Implicit Midpoint)}\\ y_{n+1} - y_n &= \triangle t f(\frac{t_n + t_{n+1}}{2}, \frac{y_n + y_{n+1}}{2}). \end{aligned}$$

This is known by many names including the implicit midpoint method, the one leg trapezoid rule.

6.7. An RK3 method. There is also an infinite family of RK3 methods. One popular RK3 method (part of the Bogacki-Shampine embedded RK pair) is

$$\begin{aligned} given &: y_n && \text{(RK3 in stages)}\\ k_1 &= \triangle t f(t_n, y_n)\\ k_2 &= \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}k_1)\\ k_3 &= \triangle t f(t_n, y_n + \frac{3}{4}k_2)\\ k_4 &= \triangle t f(t_n + \frac{2}{9}\triangle t, y_n + \frac{2}{9}k_1 + \frac{1}{3}k_2 + \frac{4}{9}k_3)\\ y_{n+1} &= y_n + \frac{7}{24}k_1 + \frac{1}{4}k_2 + \frac{1}{3}k_3 + \frac{1}{8}k_4. \end{aligned}$$

EXERCISE 24. *Write the Butcher tableau for the explicit midpoint rule, the implicit midpoint method and RK3.*

6.8. The Calahan DIRK method. RK methods can also be implicit. One common example is the Calahan Diagonally Implicit Runge Kutta, DIRK, method. The Calahan DIRK is given by the Butcher tableau

$$\begin{array}{c|cc} \alpha & \alpha & 0\\ 1-\alpha & \underline{1} - 2\alpha & \underline{\alpha}\\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

The Calahan DIRK.

This method is A_0 stable and generally second order accurate. For the special value

$$\alpha = \frac{3+\sqrt{3}}{6}$$

it is third order accurate. Written out in stages it is:

(Calahan DIRK)
$$\begin{gathered} given : y_n \\ \text{Solve nonlinear eqn. for } k_1\text{:} \\ k_1 = \triangle t f(t_n + \alpha\triangle t, y_n + \alpha k_1) \\ \text{Solve nonlinear eqn. for } k_2\text{:} \\ k_2 = \triangle t f(t_n + (1-)\triangle t, y_n + (1-2\alpha)k_1 + \alpha k_2) \\ y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2. \end{gathered}$$

6.9. The Fourth Order Runge-Kutta Method. One commonly used RK method is the fourth order method called **RK4**, given by

$$\begin{aligned} given &: y_n \\ k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}k_1) \\ k_3 &= \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}k_2) \\ k_4 &= \triangle t f(t_n + \triangle t, y_n + k_3) \\ y_{n+1} &= y_n + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \end{aligned} \qquad \text{(RK4 in stages)}$$

Like RK2, there are an infinite number of RK4 methods. The above is simply the one with simple (easy to remember) parameter values. **RK4 is fourth order accurate** (LTE is $O(\triangle t^5)$), explicit and **costs 4 function evaluations per step**.

EXERCISE 25. *Consider the linear pendulum*

$$\begin{gathered} \theta''(t) + \theta(t) = 0, t > 0 \\ \theta(0) = \pi/4, \theta'(0) = \pi/4. \end{gathered}$$

For the linear pendulum the following is constant

$$\frac{1}{2}\left[\theta'(t)^2 + \theta^2(t)\right] = \frac{1}{2}\left[\theta'(0)^2 + \theta^2(0)\right] \textit{ for all time.}$$

Write as an IVP for a first order system of two equations. Solve using every explicit method introduced. Take $\triangle t = 1/10, 1/20$ and $1/30$. Take the final time long enough to see 10 complete periods. For each calculate the above (appropriate) invariant and see if it grows or decays. Compare and draw conclusions.

EXERCISE 26. *Repeat the past problem for the nonlinear pendulum IVPs*

$$\begin{gathered} \theta''(t) + \sin\theta(t) = 0, t > 0 \\ \theta(0) = \pi/4, \theta'(0) = \pi/4. \end{gathered}$$

For it, the following is constant

$$\frac{1}{2}\left[\left(\theta'(t)\right)^2 - \cos\theta(t)\right] = \frac{1}{2}\left[\left(\theta'(0)\right)^2 - \cos\theta(0)\right] \textit{ for all time.}$$

Write as an IVP for a first order system of two equations. Solve using every explicit method introduced. Take $\triangle t = 1/10, 1/20$ and $1/30$. Take the final time long enough to see 10 periods. For each calculate the above (appropriate) invariant and see if it grows or decays. the linear pendulum. Compare and draw conclusions

Exercise 27. *Pick some method to adapt the time step in Euler's method. Repeat problem 1 for Euler's method with adaptive timestep selection. Also look at what happens to the timestep size as the calculation progresses.*

Exercise 28. *Solve the equation below with RK2 and RK4 [pick initial conditions so something interesting happens in the solution]. Plot and draw conclusions*

$$\theta'' + (\theta^2 - 1)\theta + \theta = 1.3\cos(0.2t).$$

Exercise 29. *Analyze the accuracy and stability of the explicit midpoint method*

$$y_{n+1} = y_n + \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}\triangle t f(t_n, y_n)).$$

Exercise 30. *Compare the explicit midpoint method with RK2. You may choose the test problem and criteria for comparison but pick one that will show a difference between the 2 methods.*

Exercise 31. *If the* 2*d wave equation under zero BCs on the unit square is discretized in space by the usual method for the discrete Laplacian on $N \times N$ mesh, one arrives at a system of second order ODEs to be solved in time:*

$$\frac{d^2}{dt^2}\overrightarrow{u}(t) = -c^2 A\overrightarrow{u}(t), \quad A = \textit{matrix from usual discrete } -\triangle^h$$

a. What are the eigenvalues of A? [No proof necessary-just look up the formula]. Methods for IVPs start by writing it as a first order system in time by $\overrightarrow{v}(t) = d/dt\overrightarrow{u}(t)$ then

$$\text{(ODEsystem)} \qquad \frac{d}{dt}\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -c^2A & 0 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}, \textit{ let } \mathcal{A} := \begin{pmatrix} 0 & 1 \\ -c^2A & 0 \end{pmatrix}$$

Find the block matrix eigenvalues $\lambda(\mathcal{A})$ explicitly (using those of the 2, 1 *block $\lambda(A)$). b. Pick an appropriate ODE method to be used to solve (ODEsystem) based on its stability and your answer to part a. Explain your choice briefly. c. For your chosen method, can the artificial variable v be eliminated to get a method for the original second order in time ODE system? Explain.*

6.10. Strong Stability Preserving Methods. The Idea of SSP methods is as follows. Suppose stability in a very strong sense in some important norm $|| * ||$:

$$||y^{n+1}|| \leq ||y^n||$$

can be proven for Euler's method

$$\frac{x_{n+1} - x_n}{\triangle t} = f(t_n, x_n)$$

under some timestep condition

$$\triangle t \leq C_{critical}.$$

The idea is to take a weighted combination with no negative weights of Euler steps and pick the weights to maximize accuracy. Doing so means the higher order methods, so constructed, will preserve (under the same timestep condition) the same strong stability property.

These were developed by Gottleib and Shu. We give a few examples for an autonomous equation $x' = f(x)$:

A two stage, second order SSP method: Given x_n,

$$\begin{aligned} x^1_{n+1} &= x_n + \triangle t f(x_n), \\ x^2_{n+1} &= x^1_{n+1} + \triangle t f(x^1_{n+1}) \\ x_{n+1} &= \frac{1}{2}(x_n + x^2_{n+1}). \end{aligned}$$

A 3 stage, third order SSP method: Given x_n,

$$\begin{aligned} x^1_{n+1} &= x_n + \triangle t f(x_n), \\ x^2_{n+1} &= \frac{3}{4}x_n + \frac{1}{4}\{x^1_{n+1} + \triangle t f(x^1_{n+1})\} \\ x^3_{n+1} &= x^2_{n+1} + \triangle t f(x^3_{n+1}) \\ x_{n+1} &= \frac{1}{3}x_n + \frac{2}{3}x^3_{n+1}. \end{aligned}$$

A 4 stage, third order SSP method of Ruuth and Spiteri: Given x_n,

$$\begin{aligned} x^1_{n+1} &= x_n + \frac{\triangle t}{2} f(x_n), \\ x^2_{n+1} &= x^1_{n+1} + \frac{\triangle t}{2} f(x^1_{n+1}) \\ x^3_{n+1} &= x^2_{n+1} + \frac{\triangle t}{2} f(x^2_{n+1}) \\ x^4_{n+1} &= \frac{2}{3}x_n + \frac{1}{3}x^3_{n+1} \\ x_{n+1} &= x^4_{n+1} + \frac{\triangle t}{2} f(x^4_{n+1}). \end{aligned}$$

These schemes are as good or as bad as the explicit Euler step. Thus, the formulation of the problem must be made at the very start so that the explicit Euler method's solution preserves the required stability property.

CHAPTER 5

Adapting the timestep

Truth is treason in an empire of lies. -George Orwell, 1984

The goal of adaptivity is for the algorithm to function nearly as an expert system so as to compute a solution whose accuracy is within a user-supplied prescribed tolerance at minimal or near minimal work. Adapting the timestep is built on the 4 pillars:

1. An assumption that allows localization of errors.

The most basic assumption used is that

$$\text{Global error} = \sum_{all\_previous\_steps} \text{Local error.}$$

This assumption leads to the local error condition: Adapt time step to make

$$\frac{|local\_error|}{\triangle t} < global\_error\_tolerance.$$

2.A method for estimating the local error committed going from t_n to t_{n+1}.

For the estimation step we say that an estimator is **reliable** if

$$|TrueError| \leq EST$$

and **pessimistic** if reliability is obtained by having at times

$$|TrueError| << EST$$

. A pessimistic estimator results in an adaptive method that is not **efficient**.

3. A strategy for changing the timestep in response to the estimated error.

Multi-step methods require a third pillar for adaptivity:

4. Interpolation or restarting with a 1 step method to provide the unknown previous values y_{n-j} when changing the timestep $\triangle t$.

1. Estimating local errors

> Knowledge has three degrees—opinion, science, and illumination. The means or instrument of the first is sense; of the second, dialectic; of the third, intuition. This last is absolute knowledge founded on the identity of the mind knowing with the object known. - Plotinus

We give two methods for estimating local errors. The error estimators we present are easy to implement, inexpensive and work reasonably well.

1.1. Method 1: Timestep halving and doubling. As a concrete example, consider Euler's method. We do one step from t_n to t_{n+1} giving $y_{n+1}^{low\_order}$ then halve $\triangle t$ and do two steps giving y_{n+1}. This gives two approximations of different accuracy. Their difference must be rescaled appropriately and is taken as the estimator. To see why re-scaling is needed, suppose a method of accuracy $\mathcal{O}(\triangle t^p)$ is used. The local error (starting exactly from the previous step) is then $\mathcal{O}(\triangle t^{p+1})$. If we:

assume the error at the previous step is under good control,

i.e., assume that the approximate value there is essentially exact $y_n \equiv y(t_n)$, this means that the error in the next step is exactly the local truncation error of that step. This assumption localizes the problem of estimating errors since it means

$$y_{n+1}^{low\_order} = y(t_{n+1}) + \tau_n \triangle t^{p+1} + \mathcal{O}(\triangle t^{p+2})$$
$$y_{n+1} = y(t_{n+1}) + 2\tau_n \left(\frac{\triangle t}{2}\right)^{p+1} + \mathcal{O}(\left(\frac{\triangle t}{2}\right)^{p+2}).$$

The leading order error term (which is the term we need to estimate to get the first significant digit in the error correctly) is $2\tau_n\left(\triangle t/2\right)^{p+1}$. Thus, we subtract and solve for this term:

$$y_{n+1}^{low\_order} - y_{n+1} = [2^p - 1]2\tau_n \left(\frac{\triangle t}{2}\right)^{p+1} + \mathcal{O}(\triangle t^{p+2}).$$

Thus, the leading order error term, $2\tau_n\left(\triangle t/2\right)^{p+1}$, is (to leading order)

$$\text{local error} = \frac{\left|y_{n+1}^{low\_order} - y_{n+1}\right|}{2^p - 1} + \textit{ Higher Order Terms.}$$

For Euler's method, $p = 1$, so $2^p - 1 = 1$. If a second order method is used, we have $2^p - 1 = 3$. Thus, simply using $\left|y_{n+1}^{low\_order} - y_{n+1}\right|$ overestimates the error by a factor of 3. This leads to doing 3× too much work!

For Euler's method, error estimation proceeds by:

$$\begin{array}{rcl}
\text{Set:} & & p = 1 \\
y_{n+1}^{low\_order} & = & y_n + \triangle t f(t_n, y_n) \\
Then & : & \\
y_{n+\frac{1}{2}} & = & y_n + \dfrac{\triangle t}{2} f(t_n, y_n) \\
y_{n+1} & = & y_{n+\frac{1}{2}} + \dfrac{\triangle t}{2} f(t_n, y_{n+\frac{1}{2}}) \\
Then & : & \\
EST & = & |y_{n+1} - y_{n+1}^{low\_order}|/(2^p - 1)
\end{array}$$

This strategy works for all methods.

1.2. Method 2: Two methods of different accuracy. Suppose we have methods of order $\mathcal{O}(\triangle t^p)$ and $\mathcal{O}(\triangle t^q)$ where $q > p$. Then

$$\begin{array}{rcl}
y_{n+1}^{low\_order} & = & y(t_{n+1}) + \tau_n \triangle t^{p+1} + \mathcal{O}(\triangle t^{p+2}) \\
y_{n+1} & = & y(t_{n+1}) + \widetilde{\tau}_n \triangle t^{q+1} + \mathcal{O}(\triangle t^{q+2}).
\end{array}$$

Subtraction gives

$$\begin{aligned} y_{n+1}^{low\_order} - y_{n+1} &= \left[\tau_n - \widetilde{\tau}_n \triangle t^{q-p}\right] \triangle t^{p+1} + \mathcal{O}(\triangle t^{p+2}) \\ &= \tau_n \triangle t^{p+1} + \textit{ Higher Order Terms.} \end{aligned}$$

Thus,

$$EST = |y_{n+1}^{low\_order} - y_{n+1}|$$

is an accurate estimator for the lower order approximation. To get an accurate estimator for the higher order approximation we redo the calculation:

$$y_{n+1}^{low\_order} - y_{n+1} = \left[\frac{\tau_n \triangle t^{-(q-p)} - \widetilde{\tau}_n}{\widetilde{\tau}_n}\right] \widetilde{\tau}_n \triangle t^{q+1} + \mathcal{O}(\triangle t^{p+2}).$$

An accurate estimator would then be

$$EST = \frac{|y_{n+1}^{low\_order} - y_{n+1}|}{\left[\frac{\tau_n \triangle t^{-(q-p)} - \widetilde{\tau}_n}{\widetilde{\tau}_n}\right]}$$

This obviously requires considerable detailed information about the methods local truncation error. Nevertheless, it has been made to work. Given the two alternatives, often the simpler estimator is used for the more accurate approximation and the lack of efficiency accepted as the cost of obtaining reliability.

Example 15. *The most economical estimator of this type is the combination of Euler and RK2 since the function evaluations needed for RK2 include the one needed for Euler's method:*

$$\text{(Euler and RK2)} \qquad \begin{aligned} &given: y_n \\ k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \triangle t, y_n + k_1) \\ y_{n+1}^{Euler} &= y_n + k_1 \\ y_{n+1} &= y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2, \\ EST &= |y_{n+1} - y_{n+1}^{Euler}| \\ \triangle t &= \textit{adapted based on } EST \textit{ and proceed.} \end{aligned}$$

Example 16. *With BDF2 and BDF3 this strategy would proceed by:*

$$\begin{gathered} \textit{Solve for } y_{n+1}^{BDF2}: \\ y_{n+1}^{BDF2} - \frac{2}{3}\triangle t f(t_{n+1}, y_{n+1}^{BDF2}) = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} \\ \textit{Then : Solve for } y_{n+1}: \\ y_{n+1} - \frac{6}{11}\triangle t f(t_{n+1}, y_{n+1}) = \frac{18}{11}y_n - \frac{9}{11}y_{n-1} + \frac{2}{11}y_{n-2} \\ \textit{Then : } EST = |y_{n+1} - y_{n+1}^{BDF2}| \end{gathered}$$

2. Stepsize control

Local adaptivity in IVPs is usually[1] built on a "*spherical cow assumption*[2]" that makes everything afterward simple:

(Spherical Cow) Error at t_n = sum of local errors on previous steps.

With this assumption, making the error at the time t_N smaller than the tolerance ε requires making the sum of the local error/unit step smaller than $N\varepsilon$. Indeed, this is

$$\sum \frac{local\_error}{\triangle t} \leq N\varepsilon \ .$$

Thus, the strategy is to keep the estimate of the local error per unit stepsize below some preset tolerance by cutting the timestep if EST is too big and increasing it (for greater efficiency) is EST is too far below the set tolerance.

The simplest implementation is by mesh halving and doubling. Suppose we are using a p^{th} order method so the local error is $\mathcal{O}(\triangle t^{p+1})$. Thus, when the mesh is halved or doubled, the local error is changed by $1/2^{p+1}$ and 2^{p+1}, respectively. To avoid flip flopping (time step halve then double then halve etc.) the upper and lower decision points must thus be set at least 2^{p+2} apart. Given a preset, user-supplied tolerance TOL, we seek to maintain

$$\frac{TOL}{2^{p+2}} < \frac{EST}{\triangle t} < TOL.$$

Given, y_n an adaptive algorithm computes y_{n+1} and from that EST, an estimate of the local error at that step. There is then three cases:

Case 1: Error just right:

$$\frac{TOL}{2^{p+2}} < \frac{EST}{\triangle t} < TOL.$$

In this case we accept the more accurate approximation y_{n+1}, keep the same stepsize $\triangle t$ and move to the next step.

Case 2: Error too big:

$$\frac{EST}{\triangle t} \geq TOL.$$

The error is too large. In this case we return to (t_n, y_n), cut $\triangle t$ in half, $\triangle t \Leftarrow \triangle t/2$, and recompute y_{n+1}.

Case 3: Error too small:

$$\frac{EST}{\triangle t} < \frac{TOL}{2^{p+2}}.$$

The error is much smaller than the sought accuracy. Thus the program is doing much more work then necessary. In this case we accept y_{n+1} but double $\triangle t$, $\triangle t \Leftarrow 2\triangle t$, for the next step.

For the Euler-RK2 pair the full algorithm is as follows.

[1] Note the word "usually". Since this is a heuristic it is modified when more is known about the problem. For example, if it is known that the solution approaches and equilibrium value rapidly as $t \to \infty$ then it is usually modified to be based on the assumption that errors do not accumulate. Thus it is assumed that *Error at a step = local error at that step.* The algorithmic realization is to adapt to make: $\frac{local\_error}{\triangle t} \leq \varepsilon$.

[2] There is a classic science/math joke with many variations whose punchline is "Assume a spherical cow".

```
            Input : $TOL$ ; Set : $p = 2$
            $given : y_n$
                $k_1 = \triangle t f(t_n, y_n)$
                $k_2 = \triangle t f(t_n + \triangle t, y_n + k_1)$
(Adaptive RK2)
            $y_{n+1}^{Euler} = y_n + k_1$
            $y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2,$
            $EST = |y_{n+1} - y_{n+1}^{Euler}|$
      IF $TOL/2^{p+2} < EST/\triangle t < TOL$ THEN proceed to next step
        IF $EST/\triangle t \geq TOL$ THEN $\triangle t \Leftarrow \triangle t/2$ and recompute this step
        IF $EST/\triangle t < TOL/2^{p+2}$ THEN $\triangle t \Leftarrow 2\triangle t$ and proceed to next step
```

EXERCISE 32. *Suppose in Case 2 and one is willing to change $\triangle t$ more flexibly than by halving and doubling. Show that picking $\triangle t$ to match $EST/\triangle t = TOL$ leads, to leading order terms, to*

$$\triangle t_{new} = \triangle t_{old} \left[\triangle t_{old} \frac{TOL}{EST_{old}} \right]^{1/p}.$$

EXERCISE 33. *Reformulate the decision tree to control the relative error rather than the absolute error. [Many believe that relative error should be the target quantity.]*

EXERCISE 34. *Update your program for Euler's method to incorporate adaptivity. Use it to solve the linear pendulum*

$$\begin{aligned} \theta'' + \theta &= 0, t > 0, \\ \theta(0) &= \pi/4, \\ \theta'(0) &= \pi/4 \end{aligned}$$

written as a first order system via

$$x(t) = \theta(t) \text{ and } y(t) = \theta'(t).$$

Euler's method is slightly unstable for this for fixed timestep in that its approximate solution grows slowly as more timesteps are taken. See if adaptivity saves Euler's method from its instability. Estimate the extra cost in saving Euler's method.

EXERCISE 35. *Repeat the last problem. Calculate the true error and compare it to the estimated error. Draw conclusions.*

EXERCISE 36. *Repeat the adaptive calculation for the nonlinear pendulum $\theta''(t) + sin(\theta(t)) = 0$. Calculate the invariant of the nonlinear pendulum and see how close to conserved it is with the adaptive method.*

EXERCISE 37. *Program adaptive Euler-RK2. Take $TOL = 0.001$ and $0 < t < 100$. Consider the nonlinear pendulum equation:*

$$\theta'' + \sin\theta = 0, 0 < t < 100,$$

$$\theta(0) \& \theta'(0) \text{ given[you pick]}.$$

Write it as a first order system in the usual way [$x = \theta, y = \theta'$]. Show first that

$$G(x,y) = (1/2)y^2 - \cos x$$

is constant along solutions. Solve the problem with Euler, RK2 and adaptive RK2. Plot $G(x,y)$ vs. t. Draw conclusions. Next try the equation

$$\theta'' + sign(\theta) = 0.$$

Again, pick the initial conditions so the solution has interesting behavior.

Exercise 38. *Solve the equation below both adaptively and non adaptively [You pick initial conditions so something interesting happens in the solution]. Plot and draw conclusions*

$$\theta'' + (\theta^2 - 1)\theta + \theta = 1.3\cos(0.2t).$$

Exercise 39. *Consider the method with $a = 0, b = 1, \alpha = \beta = 1/2$*

$$y_{n+1} = y_n + \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}\triangle t f(t_n, y_n))$$

Derive an adaptive algorithm for the explicit midpoint method. Compare it with adaptive RK2. You may choose the test problem and criteria for comparison but pick one that will show a difference between the two methods.

Exercise 40. *Literature search: Find out the precise description of "Milne's device" in numerical OEDs. Relate it to the adaptive algorithms presented.*

3. Embedded Runge-Kutta pairs

Adaptivity for single step methods has two aspects:

- How to estimate the local error at each step?
- How to change the stepsize given an estimate of the local error?

For multi-step methods, every time the stepsize is changed data is missing that is needed to proceed after the change in stepsize. Thus, adaptivity for multi-step methods has a third aspect:

- How to provided (by, e.g., interpolation or restarting with a single step method) the values needed for the multi-step method but missing after the step size change?

We have seen that the answer to the second question is universal and that an answer to the first question can be obtained (at extra cost) by halving and doubling. It is a remarkable feature of some RK methods that **estimation of local errors can be done at essentially no extra cost!** The idea behind development of these, so called, **embedded RK pairs** is completely explained by reconsidering **RK2**:

$$\begin{aligned} given &: y_n \\ k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \triangle t, y_n + k_1) \\ y_{n+1} &= y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2. \end{aligned}$$

After the first stage the approximation of Euler's method (which is $RK1$) can be computed:

$$y_{n+1}^{Euler} = y_n + k_1$$

This is a first order method and RK2 is a second order method. The most natural (but conservative) estimation of the error in RK2 is simply to say the digits of agreement between the two approximations are to be trusted. This can be a sharp estimate for the Euler approximation and is thus a *reliable*[3] *but pessimistic estimator for RK2*. We thus have the scheme[4]:

$$\begin{aligned}
given &: y_n \\
k_1 &= \triangle t f(t_n, y_n) \\
k_2 &= \triangle t f(t_n + \triangle t, y_n + k_1) \\
y_{n+1}^{Euler} &= y_n + k_1 \\
y_{n+1} &= y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2, \\
EST &= |y_{n+1} - y_{n+1}^{Euler}| \\
\triangle t &= \text{adapted based on } EST \text{ and proceed.}
\end{aligned} \tag{RK2 \& error estimator}$$

That EST is can be an overestimate is more than counter balanced by the fact that it is obtained at no extra cost!

The idea of embedded RK methods is to use the fact that there are infinitely many RK methods of every order to derive RK pairs with the property that the function evaluations needed to take a lower order RK step are repeated to take a higher order RK step. Thus, the two RK steps difference becomes a reliable error estimator. To our knowledge, this brilliant but simple idea was due to Felhberg. Different realizations[5] of it have been developed.

We present examples that work well for non-stiff problems.

[3] Recall that an estimator is **reliable** if $|TrueError| \leq EST$ and **pessimistic** if reliability is obtained by having at times $|TrueError| << EST$.

[4] This is sometimes rewritten as:

$$\begin{aligned}
k_1 &= \triangle t f(t_n, y_n) \\
y_{n+1}^{Euler} &= y_n + k_1 \\
k_2 &= \triangle t f(t_{n+1}, y_{n+1}^{Euler}) \\
y_{n+1} &= y_{n+1}^{Euler} + \frac{1}{2}(k_2 - k_1) \\
EST &= |\frac{1}{2}(k_2 - k_1)|.
\end{aligned}$$

[5] Bogacki, Przemyslaw; Shampine, Lawrence F. (1989), "A 3(2) pair of Runge–Kutta formulas", Applied Mathematics Letters 2 (4): 321–325, doi:10.1016/0893-9659(89)90079-7, ISSN 0893-9659

Dormand, J. R.; Prince, P. J. (1980), "A family of embedded Runge-Kutta formulae", Journal of Computational and Applied Mathematics 6 (1): 19–26, doi:10.1016/0771-050X(80)90013-3

Erwin Fehlberg (1969). Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems. NASA Technical Report 315.

Erwin Fehlberg (1970). "Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme," Computing (Arch. Elektron. Rechnen), vol. 6, pp. 61–71. doi:10.1007/BF02241732

Hairer, Ernst; Nørsett, Syvert Paul; Wanner, Gerhard (2008), Solving ordinary differential equations I: Nonstiff problems, Berlin, New York: Springer-Verlag, ISBN 978-3-540-56670-0.

3.1. The Bogacki-Shampine embedded RK 2-3 pair.

There is an infinite family of RK3 methods. exploiting this, The popular Bogacki-Shampine embedded RK2-3 pair is

$$\begin{aligned}
given &: y_n && \text{(RK2/3 pair)}\\
k_1 &= \triangle t f(t_n, y_n)\\
k_2 &= \triangle t f(t_n + \frac{1}{2}\triangle t, y_n + \frac{1}{2}k_1)\\
k_3 &= \triangle t f(t_n, y_n + \frac{3}{4}k_2)\\
k_4 &= \triangle t f(t_n + \frac{2}{9}\triangle t, y_n + \frac{2}{9}k_1 + \frac{1}{3}k_2 + \frac{4}{9}k_3)\\
y_{n+1}^{low} &= y_n + \frac{2}{9}k_1 + \frac{1}{3}k_2 + \frac{4}{9}k_3,\\
y_{n+1} &= y_n + \frac{7}{24}k_1 + \frac{1}{4}k_2 + \frac{1}{3}k_3 + \frac{1}{8}k_4,\\
EST &= |y_{n+1} - y_{n+1}^{low}|.
\end{aligned}$$

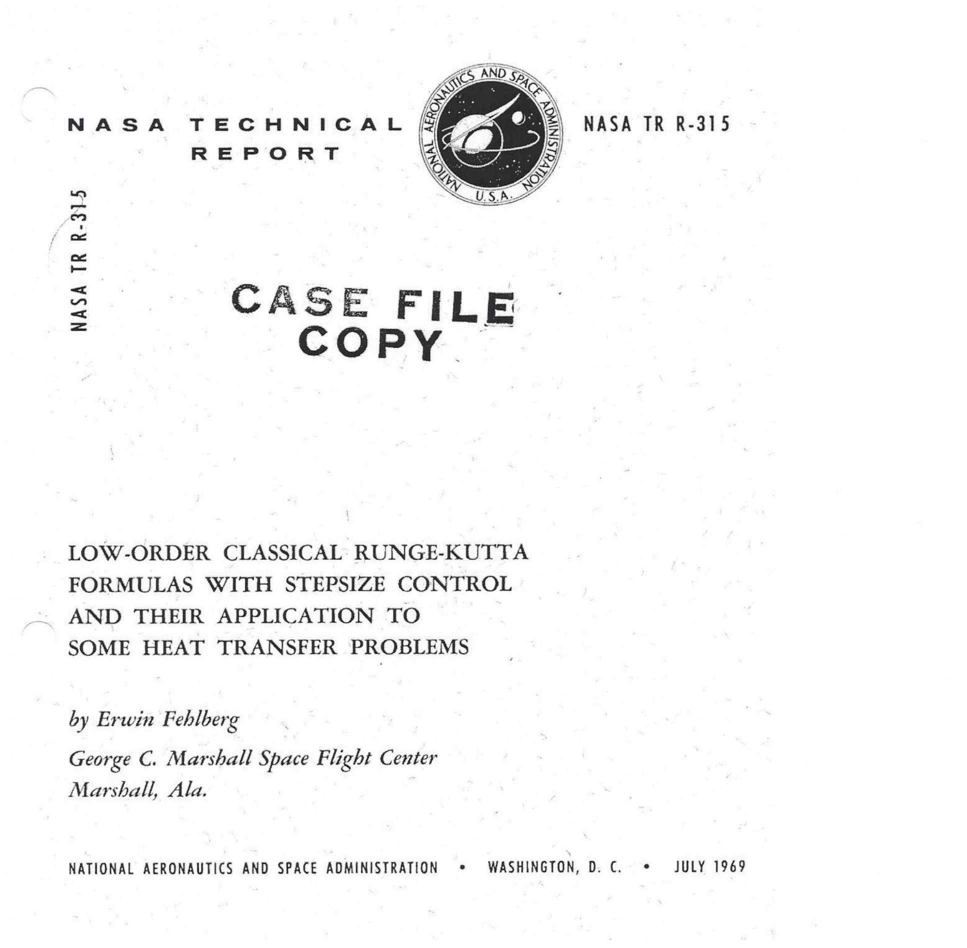
3.1.1. *About the discoverers.* Przemyslaw Bogacki is a Professor at Old Dominion University and Lawrence F. Shampine is a Professor Emeritus at Southern Methodist University.

3.2. The Runge-Kutta-Fehlberg RKF 4-5 Method.

The RKF4-5 pair is the original embedded RK pair developed by Erwin Fehlberg.

The RFK methods are still considered accurate and reliable. The RKF45 method proceeds as follows.

$$\begin{aligned}
given &: y_n\\
k_1 &= \triangle t f(t_n, y_n)\\
k_2 &= \triangle t f(t_n + \frac{1}{4}\triangle t, y_n + \frac{1}{4}k_1)\\
k_3 &= \triangle t f(t_n + \frac{3}{8}\triangle t, y_n + \frac{3}{32}k_1 + \frac{9}{32}k_2)\\
k_4 &= \triangle t f(t_n + \frac{12}{13}\triangle t, y_n + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3)\\
k_5 &= \triangle t f(t_n + \triangle t, y_n + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4)\\
&\quad and && \text{(RKF4/5 pair)}\\
k_6 &= \triangle t f(t_n + \frac{1}{2}\triangle t, y_n - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 - \frac{1859}{4104}k_4 - \frac{11}{40}k_5)\\
y_{n+1}^{LowOrder} &= y_n + \left(\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5\right)\\
y_{n+1} &= y_n + \left(\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\right)\\
EST &= |y_{n+1} - y_{n+1}^{LowOrder}|
\end{aligned}$$



NASA TECHNICAL REPORT

NASA TR R-315

NASA TR R-315

CASE FILE COPY

LOW-ORDER CLASSICAL RUNGE-KUTTA FORMULAS WITH STEPSIZE CONTROL AND THEIR APPLICATION TO SOME HEAT TRANSFER PROBLEMS

by Erwin Fehlberg

George C. Marshall Space Flight Center
Marshall, Ala.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JULY 1969

FIGURE 1. The paper of Fehlberg

EXERCISE 41. *Consider the following predator-Prey system for population levels of rabbits* ($R(t)$) *and Foxes* ($F(t)$)*:*

$$\begin{aligned} R' &= R - RF, R(0) = 3 \\ F' &= -F + RF, F(0) = 1. \end{aligned}$$

The solution is periodic. Solve this system over $0 < t < 20$ *with Euler, RK2 and adaptive RK2. Assume (a spherical cow) that the adaptive RK2 solution is exact and use it to compute the error in both Euler and ((non adaptive) RK2. Study how the error grows as* t *increases.*

Exercise. Below you will find a 'legacy' program in low level, FORTRAN of the sort that 98% of scientific programs in current use are built from. The exercise is:

Write a [low level] conversion of it to MatLab. If you want to improve the method that is fine but improvements must be documented. Pick a test problem among the many in the notes and compare adaptive vs. nonadaptive solution. If you get more or less that same result, try again! The goal of tests is to find where a distinction exists.

```
  PROGRAM ADAPT
C THIS PROGRAM SOLVES
C X'=F(T,X,Y), Y'=G(T,X,Y)
C ADAPTIVELY
C
F(T,X,Y)=Y
G(T,X,Y)=-1.0*X
C
C If you know the true solutions Xtru(T), YTRU(T),
C then fill in the RHS of the 2 lines below as nonzero.
C
XTRU(T)=(ATAN(1.0))*COS(T)
YTRU(T)=(ATAN(1.0))*SIN(T)
C
C If you know a first integral, FINT(X,Y), you can modify the
C RHS of the next line to the correct function.
C
FINT(X,Y)=(X*X+Y*Y)/(ATAN(1.0)**2)
TOL=0.01
C
C Note: you can turn off adaptivity by setting TOL
C to be very large (e.g. 100.0)
C
PI=ATAN(1.0)*4.0
TZERO=0.0
TFINAL=30.0
XZERO=PI/4.0
YZERO=0.0
C YOU MUST FILL IN THE ABOVE LINES TO INITIALIZE THE PROGRAM
C If the problem changes
C these lines MUST!!! be changed.
 ICOUNT=0
 XOLD=XZERO
 YOLD=YZERO
 TOLD=TZERO
 H=0.001
 HMIN=0.0001
 HP=0.5
 ERRTOT=0.0
 TRUERR=0.0
TP=TZERO+HP
 NSTEP=0
C
C YOU CAN EXPERIMENT BY ALTERING SOME OF THESE TOO. NOTE THAT HMIN
```

```
  C THE SMALLEST MESHWIDTH ALLOWED, MUST BE RELATED TO
TOL.
   10 TNEW=TOLD+H
     NSTEP=NSTEP+1
  C
  C NEXT TEST IF ITS TIME TO PRINT AND FLAG IT BY IP=1 IF SO
  C
  IP=0
  IF(TNEW.GE.TP) THEN
  TNEW=TP
  TP=TP+HP
  H=TNEW-TOLD
  IP=1
  ENDIF
  C
  C COMPUTE APPROXIMATIONS WITH EULER AND RK2
  C
  XK1=H*F(TOLD,XOLD,YOLD)
  YK1=H*G(TOLD,XOLD,YOLD)
  ICOUNT=IOUNT+2
  XK2=H*F(TOLD+H,XOLD+XK1,YOLD+YK1)
  YK2=H*G(TOLD+H,XOLD+XK1,YOLD+YK1)
  ICOUNT=ICOUNT+2
  XEULER=XOLD+XK1
  YEULER=YOLD+YK1
  XNEW=XOLD+(XK1+XK2)/2.0
  YNEW=YOLD+(YK1+YK2)/2.0
  C
  C NEXT COMPUTE AN ESTIMATE FOR THE LOCAL ERROR=EST
  C
  EST=ABS(XEULER-XNEW)+ABS(YEULER-YNEW)
  C
  C TEST IF:
  C TOL/32<EST/H< TOL
  C AND CHANGE "H" ACCORDINGLY
  C
  IF(EST/H.GE.TOL) THEN
  H=H/2.0
  IF(H.LE.HMIN) THEN
  H=HMIN
  GO TO 15
  ENDIF
  GO TO 10
  ENDIF
  C
  C NOW ACCEPT THE APPROXIMATION BUT
  C TEST IF ITS TOO ACCURATE
  C
```

```
IF(EST/H.LE.TOL/32.0) H=H*2.0
15 ERRTOT=ERRTOT+EST
C If you know the true solutions, you can compute true
C errors by deleting the comment characters in the next 2 statements.
C
c TRUERR = ABS(XTRU(TNEW)-XNEW)+ABS(YTRU(TNEW)-YNEW)
c FI=FINT(XNEW,YNEW)
c
C TEST IF ITS TIME TO PRINT
C
IF(IP.EQ.1) THEN
PRINT*, 'NSTEP=',NSTEP
PRINT*,'T=',TNEW,' X=',XNEW,' Y=',YNEW
PRINT*, 'Estimated error=',ERRTOT,'True Error=',TRUERR
PRINT*,'First Integrals value is:',FI
ENDIF
TRUERR=0.0
C
C TEST IF WEVE PASSED TFINAL YET
C
IF(TNEW.GE.TFINAL) GO TO 100
20 TOLD=TNEW
XOLD=XNEW
YOLD=YNEW
GO TO 10
100 PRINT*,"total number of function evaluations= ",ICOUNT
STOP
END
```

3.2.1. *About the pioneers.* Adapted from Wikipedia:

Erwin Fehlberg (1911 - 1990) was a German mathematician. His most important merit is the development of step-size control for Runge-Kutta methods for the numerical solution of ordinary differential equations (by today Runge-Kutta Fehlberg method). Fehlberg developed numerical solution methods for ordinary differential equations. Since 1960, when the Marshall Space Flight Center opened, he developed Runge-Kutta formula pairs. Their difference represents the numerical error. In 1969, Erwin Fehlberg received among others the "Exceptional Scientific Achievement Medal" of NASA.

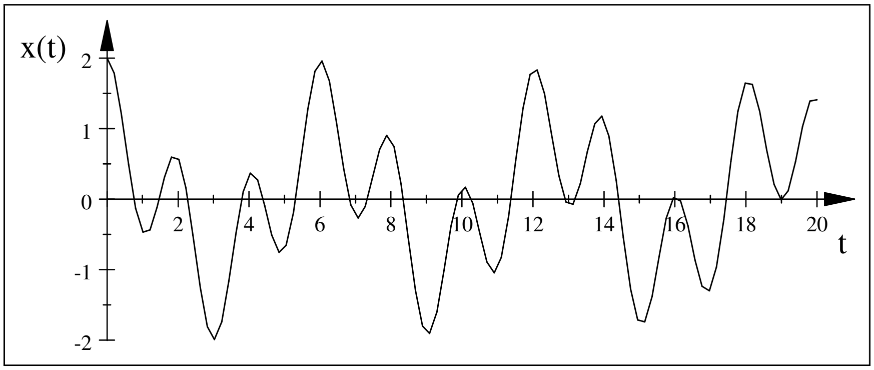
4. Some examples and test problems for adaptivity

This section presents a few classic test problems for adaptive methods. Several examples are given about how RK12 solves these problems. For some examples, the solution is very good while for others the adaptive method hits a stability issue for RK2. For these an error that is too large is not caused by the true solution doing something interesting that the timestep must resolve. It is caused by RK2 not being the appropriate method. Adaptivity then tries to make RK2 overcome instability by reducing the timestep until, eventually, it is small enough to be stable. We shall see that this is typical behavior for an adaptive but ill-chosen method confronting a stiff problem.

Test problem 1: The is a simple test problem without complicated solutions or sharp fronts. The issue here for constant timestep methods is How to select the timestep? The IVP below is solved after being written as a first order system

$$x'''' + (\pi^2 + 1)x'' + \pi^2 x = 0, 0 < t < 20,$$
$$x(0) = 2, x'(0) = 0, x''(0) = -(1 + \pi^2), x'''(0) = 0.$$

This has exact solution $x(t) = cos(t) + cos(\pi t)$, the sum of two periodic functions with incommensurable periods, quasi-periodic. Start with timestep $k = 0.1$, tolerance $TOL = 0.1$.

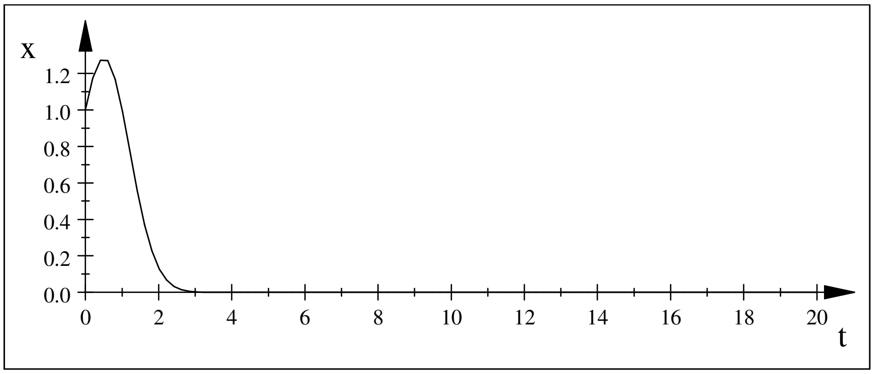


The solution

Test Problem 2a: This is solved over $0 < t < 20$

$$x' = (1 - 2t)x, x(0) = 1.$$

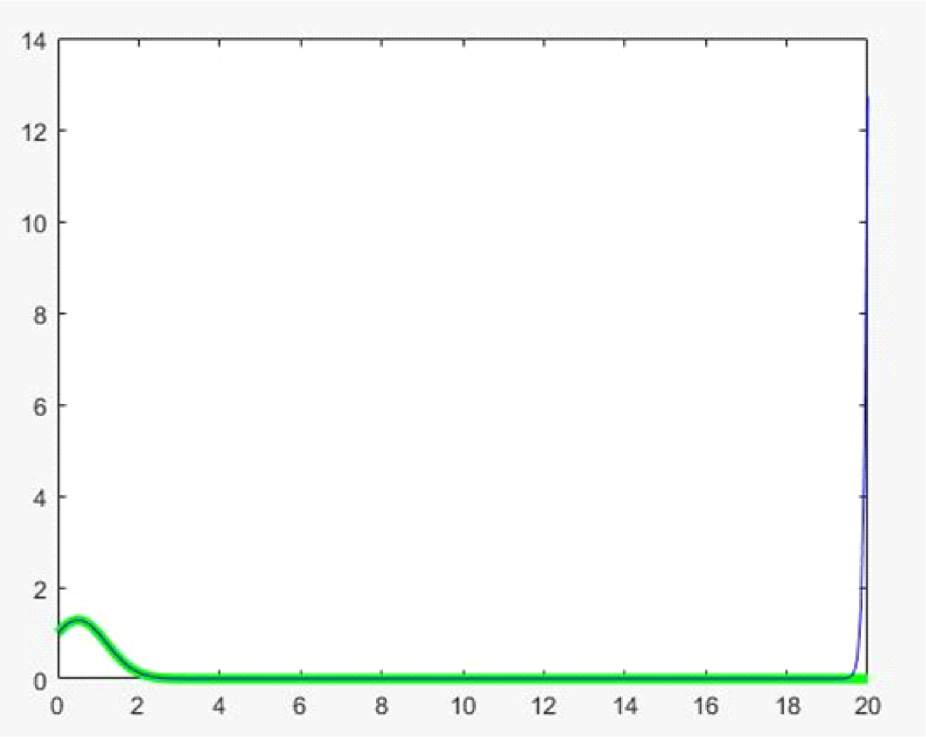
We take TOL = 0.001. It is useful to plot the solution $x(t) = exp\left(t - t^2\right)$



The solution

Clearly, nothing interesting happens in the solution after about $t = 5$. A good method would steadily increase the timestep thereafter.

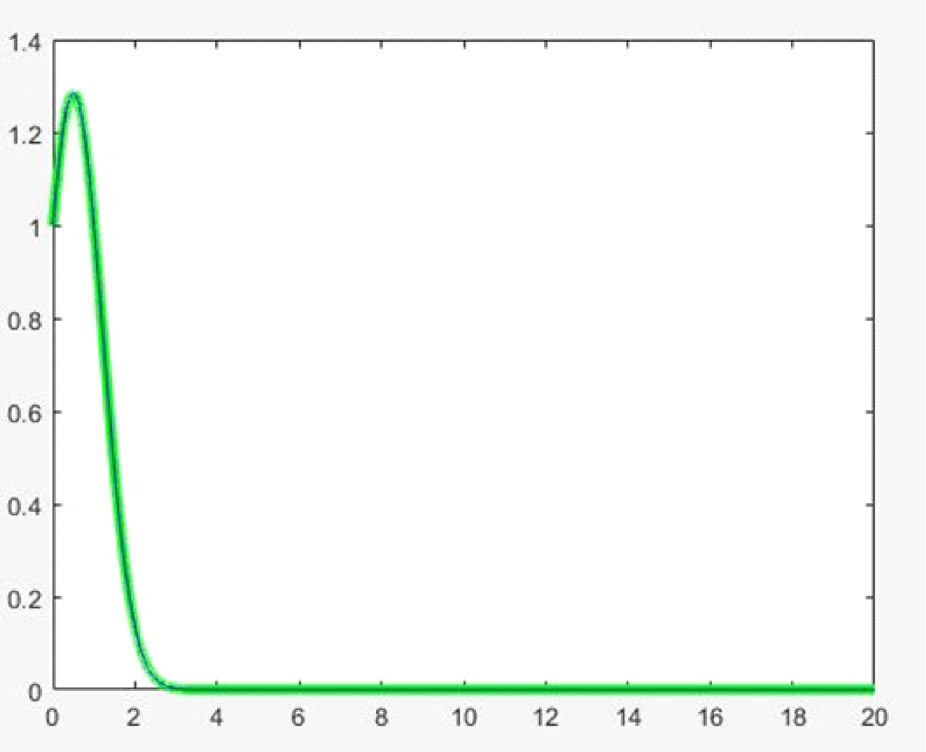
Testing non-adaptive RK2 with timestep 0.1 we see the following approximate solution:



A test of RK2 by Winlong Pei

Notice the blow up around $t = 19$. Even though the solution is nearly identically zero and applying RK2 with zero data yields zero, there is enough numerical noise to be amplified by the instability. If the base state were different from y=0 the blowup would occur much faster.

Adaptive RK2 starting with timestep 0.1 yields the good solution:



An adaptive test of Winlong Pei

The difference is seen in the cost of the two solutions also

$$\begin{aligned} Non - Adaptive &: \text{ 200 steps} \\ Adaptive &: \text{ 3739 steps} \end{aligned}$$

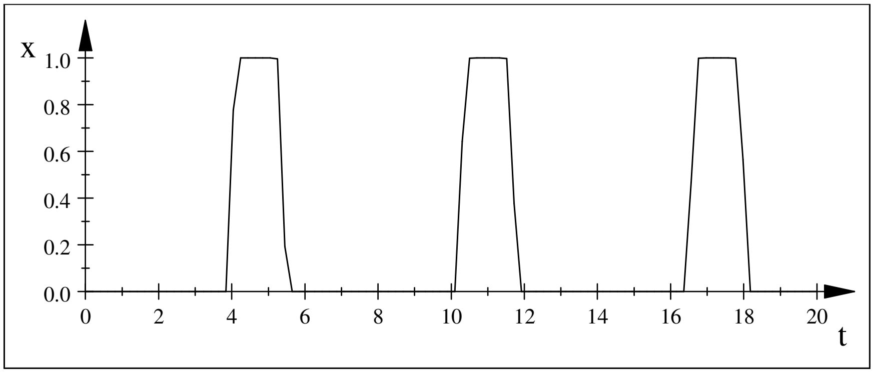
Clearly, adaptivity is not resolving solution behavior but controlling the instability of the method by taking timestep small enough to make that step fit within the RK2 stability region.

Test Problem 2b: This is solved over $0 < t < 20$

$$x' = (1 - 2t)(x - 2 - \cos(t)) - \sin(t), x(0) = 1.$$

Take T$OL = 0.001$. This test problem behaves like the previous one except $x(t) \rightarrow 2 + \cos t$ as $t \uparrow$.

Test Problem 3a: Take $f(t) = \exp\left(-(4.0 + 4.0\sin(x))^{10}\right)$



Function f(t)

Solve

$$x' = \lambda x + f(t), x(0) = 1, 0 < t < 20, \lambda = -1 \ \& \ \lambda = -1000.$$

This is an interesting problem because the solution does do interesting things. A good method would cut the timestep when the true solution is changing to accommodate the rapid changes of f(t) but then increase the timestep again when $f(t)$ is nearly constant.

Test problem 3b: This is a suggestion of Gear. We alter test problem 3a so that f(t) is the true solution. Solve

$$x' = \lambda(x - f(t)) + f'(t), x(0) = 1, 0 < t < 20, \ \lambda = -1000.$$

where, as in 3a,

$$f(t) = \exp\left(-\left(4.0 + 4.0 \sin\left(x\right)\right)^{10}\right).$$

Test Problem 4: Solve $x(0) = 1, y(0) = 0, 0 < t < 20$

$$x' = -x - y + x\sqrt{x^2 + y^2}, \ y' = -y + x + y\sqrt{x^2 + y^2}.$$

This has true solution $x(t) = cos(t), y(t) = sin(t)$ that goes around and around the unit circle. This is an interesting problem because it is unstable; any perturbation from the very simple solution grows rapidly.

Test Problem 5: The Lorenz system is

$$\frac{dX}{dt} = 10(Y - X), \frac{dY}{dt} = -XZ + 28X - Y, \frac{dZ}{dt} = XY - \frac{8}{3}Z.$$

The above uses the original parameter values of Lorenz. These produce a chaotic system. It must be noted that chaotic test problems tend to exaggerate differences between methods. The initial conditions are $(X_0, Y_0, Z_0) = (0, 1, 0)$. The system is solved over the time interval $[0, 5]$.

Test Problem 6: Van der Pol's equation is a classic test problem:

$$\begin{aligned} x'' - \mu(1 - x^2)x' + x &= 0 \\ x(0) &= 2 \\ x'(0) &= 0 \end{aligned}$$

The van der Pol equation with parameter $\mu = 1000$ is a common test problem for stiff solvers. take tolerance 10^{-4} and 10^{-6}. and plot the approximate solutions, the time step evolutions and the total number of halving, doubling and the same steps.

5. The Spherical Cow Assumption

> Milk production at a dairy farm was low, so the farmer wrote to the local university, asking for help from academia. A multidisciplinary team of professors was assembled, headed by a theoretical physicist, and two weeks of intensive on-site investigation took place. The scholars then returned to the university, notebooks crammed with data, where the task of writing the report was left to the team leader. Shortly thereafter the physicist returned to the farm, saying to the farmer, "I have the solution, but it only works in the case of spherical cows in a vacuum". - https://en.wikipedia.org/wiki/Spherical_cow

The assumption that the global error is the sum of local errors is very useful. In this section we shall examine it more carefully.

CHAPTER 6

Asymptotic Stability

Zero stability ensures that the numerical approximation grows no faster than exponential. There are (at least) 2 cases where restriction to exponential growth is insufficient:

(1) The true model may have a conserved energy and the calculation is over a long time interval. In this case it is critical that the numerical method exactly conserve a system energy related to the physical energy. Methods that do this are called symplectic methods.

(2) The true solution may be asymptotically stable and approach some steady state / equilibrium state as t→ ∞. In this case having an approximation that grows is not good. The numerical method used needs to be asymptotically stable as well.

The situation at this point is as follows.

- *There are very many methods (and it is easy to generate yet more methods). Thus it is critically important that some collection of simple criteria be developed to separate methods and pick the right method for the right application.*
- *While every reasonable method is zero stable*[1]*, when applied to problems whose solution is bounded or even decays to zero, some methods produce solutions that grow as more time steps are taken and other methods produce solutions that better resemble the true solution's qualitatively behavior for large t.*
- *Adaptivity makes all methods better. However, if a method produces a solution that (incorrectly) grows when the true solution decays, adaptivity tries to save the method by cutting the timestep until the calculation is no longer possible within time and resource constrains.*

The solution of all three of these issues is the theory of stability in numerical ODEs pioneered by G. Dahlquist. Since there are very many different kinds of stability, we must be precise.

Definition 11 (Asymptotic stability). *Consider the IVP*

$$y'(t) = f(t, y(t)) \quad \textit{for } t > 0, \quad \textit{and} \quad y(0) \textit{ given.}$$

[1] Recall that a method is **0-stable** if, when applied to $y' = Ly + F$, with L, F constants, the solution satisfies $|y_n| \leq C_1 e^{C_2 t_n}(|y_0| + |F|)$, where C_1, C_2 are constants independent of t_n, h but possibly dependent on L, F. It is known that if this estimate holds with $F = 0$ then it holds with nonzero F.

This IVP is (globally) ***Asymptotically Stable*** *if any two solutions $x(t), y(t)$ corresponding to any two different initial conditions satisfies*

$$|x(t) - y(t)| \to 0 \text{ as } t \to \infty.$$

Dahlquist studied the question:

If a method is applied to an asymptotically stable IVP, under what considerations does it produce an approximation that is asymptotically stable?

We shall develop the resulting stability theory in this section.

1. Stability Regions

We begin with one very simple but critically important example.

EXAMPLE 17. *Let λ denote a fixed/selected complex number. Consider the linear, scalar IVP*

$$y'(t) = \lambda y(t) + f(t), \text{ and } y(0) \text{ given.}$$

If $x(t), y(t)$ are two solutions with different initial conditions (e.g., $y(0) = y_0$ and $x(0) = x_0$) their difference $v(t) = x(t) - y(t)$ satisfies

$$v'(t) = \lambda v(t), \text{ and } v(0) = x_0 - y_0 \text{ given.}$$

Thus, for linear problems, the nonhomogeneous problem is asymptotically stable if and only if the homogeneous problems has solutions $v(t) \to 0$ as $t \to \infty$. Thus we consider the following which is the standard test problem for numerical methods for IVPs:

$$\text{(Model Problem)} \qquad y'(t) = \lambda y(t), \text{ and } y(0) \text{ given.}$$

Let $\lambda = \alpha + \beta i$, where α, β are real and $i = \sqrt{-1}$. Then, the solution to this problem is

$$y(t) = e^{\lambda t} y(0) = e^{\alpha t}\left[\cos(\beta t) + i \sin(\beta t)\right] y(0).$$

We observe that:

$$\begin{gathered} \text{The linear, constant model problem (Model Problem)} \\ \text{is asymptotically stable if and only if } \operatorname{Re}(\lambda) < 0. \end{gathered}$$

If $x(t), y(t)$ are numbers that are the amount of "something" then they have units of "something". For example, if $x(t), y(t)$ are distance travelled then they have units of length. The units of dx/dt are therefore "something"/time. Since $x' = \lambda x$ the units of the LHS and the RHS must also be equal. Thus, just by writing this equation we must have

$$units[\lambda] = 1/Time$$

so $1/\lambda$ has an interpretation of a relaxation or growth time.

From the calculation presented in the last example, we say that:

The ***stability region*** *of the standard test problem $y'(t) = \lambda y(t)$ is the left half-plane in $\mathbb{C}$: $\{z \in \mathbb{C} : \operatorname{Re}(z) < 0\}$.*

Every good theory is built from a calculation and begins with a collection of examples. Based on the above calculation and example we now give the definition.

Definition 12. *Consider a numerical method for IVPs applied to the standard test problem*

$$y'(t) = \lambda y(t), y(0) = 1.$$

A point $z = \triangle t\lambda$ is in the stability region of if the approximations y_n produced by the method for that value of $\triangle t\lambda$ satisfy $y_n \to 0$ as $n \to \infty$.

*A method is **A-stable** if its stability region includes the entire left half-plane, $\{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$.*

Obviously, when a method is applied to the standard test problem, a linear, homogeneous difference equation results. Since that can be solved exactly, stability regions can be calculated and plotted. We will now consider some examples of stability regions for numerical methods approximating the standard test problem.

Example 18 (No explicit method can be A-stable). *No explicit s-step method can be A-stable. In this case we have*

$$y^{n+1} = a(\triangle t\lambda)y_n \textit{ where } a(z) = \textit{ polynomial in } z.$$

Since any polynomial satisfies

$$|a(z)| \to \infty \textit{ as } |z| \to \infty$$

we cannot have A-stability.

Example 19 (Stability region of Euler's method). *Euler's method reads*

$$\frac{y_{n+1} - y_n}{\triangle t} = \lambda y_n \quad or \quad y_{n+1} = (1 + \triangle t\lambda)y_n.$$

This means that

$$y_n = (1 + \triangle t\lambda)^n y_0 \quad \textit{and thus } y_n \to 0 \textit{ if and only if } |1 + \triangle t\lambda| < 1.$$

Let $\lambda = \alpha + \beta i$ so that

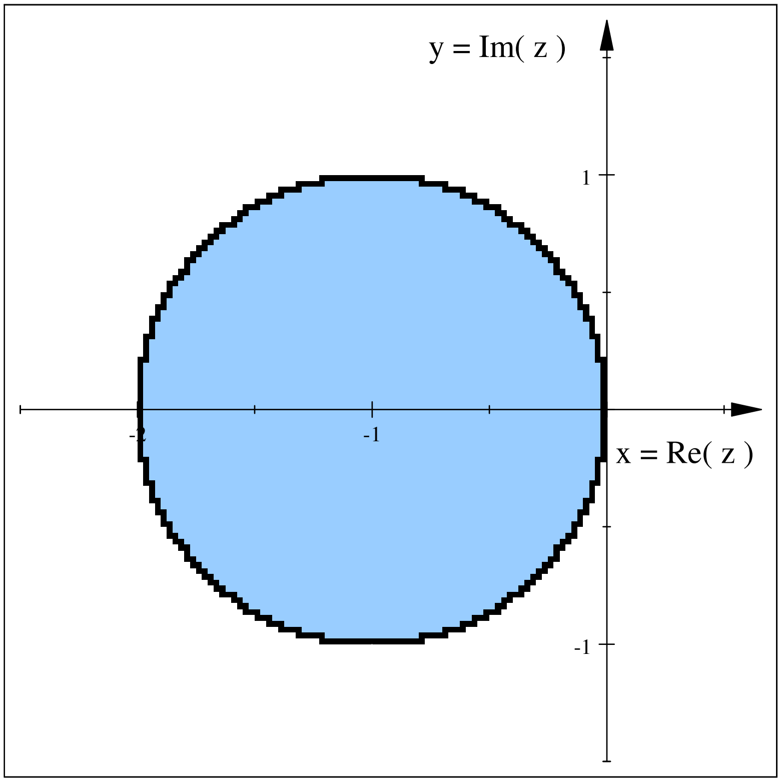
$$\begin{aligned} |1 + \triangle t\lambda|^2 &= |1 + \triangle t(\alpha + \beta i)|^2 = |(1 + \triangle t\alpha) + i(\triangle t\beta)|^2 \\ &= (1 + \triangle t\alpha)^2 + (\triangle t\beta)^2. \end{aligned}$$

Thus, the stability region is:

$$\{z = x + iy \in \mathbb{C} : (1 + x)^2 + y^2 < 1\}.$$

If $\triangle t\lambda$ lies in this region in $\mathbb{C}$ the approximate solution will decay to zero exponentially fast as t increases (correctly). If $\triangle t\lambda$ lies outside of this region, the approximate solution will blow up exponentially as $t \to \infty$. If it lies on the boundary of this region, (for this case[2] of Euler's method) its magnitude will be constant. This is the interior of a circle centered at -1 with radius 1:

[2] In general, if $h\lambda$ lies on the boundary of the stability region the approximate solution can be bounded or have polynomial growth.



Stability region of Euler's method

When stability regions are symmetric about the real axis, often only the top half is given in a figure.

Examining the stability region, we see that

- Euler's method is not A-stable.
- If one solves $y' = -10,000y$ by Euler's method, the approximate solution will blow up if $\triangle t > 2/10,000$ and will decay to zero only if $\triangle t < 2/10,000$.
- If one solves the pendulum equation written as a first order system, Euler's method will be unstable for all $\triangle t > 0$.

It is worthwhile examining the last claim in some detail. The pendulum equation (when $g/L = 1$) as a first order system is

$$\begin{aligned} x' &= y, y' = -x \\ &\textit{equivalently} \\ \frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}. \end{aligned}$$

The eigenvalues of the above 2×2 matrix are easily found to be $\lambda = \pm i$. Thus, for the pendulum equation, the relevant test problem is $y' = \pm iy$.

The last example (of the pendulum) illustratyes that sometimes stability is more important than accuracy. For example, Euler's method

$$\frac{y_{n+1} - y_n}{\triangle t} = f(t_n, y_n)$$

is first order accuracte but not appropriate for c9onsevative systems. The explicit method AB2, given by

$$\frac{y_{n+1}-y_n}{\triangle t} = \frac{3}{2}f(t_n,y_n) - \frac{1}{2}f(t_{n-1},y_{n-1})$$

is second order accurate but also not stable for the conservative systems for any timestep. Its stability region, like that of BE, does not include any of the imaginary axis.

An un-named, first order method. However, the first order method (which to our knowledge does not have a name)

$$\begin{aligned}\frac{y_{n+1}-y_n}{\triangle t} &= 2f(t_n,y_n) - 1f(t_{n-1},y_{n-1}), \text{ for constant time step , and}\\ \frac{y_{n+1}-y_n}{k_n} &= (1+\frac{k_n}{k_{n-1}})f(t_n,y_n) - \frac{k_n}{k_{n-1}}f(t_{n-1},y_{n-1}), \text{ for variable time step.}\end{aligned}$$

has the stability region belowThis includes a piece of the imkaginary axis and can thus be used for conservative systems under a time-step condition.

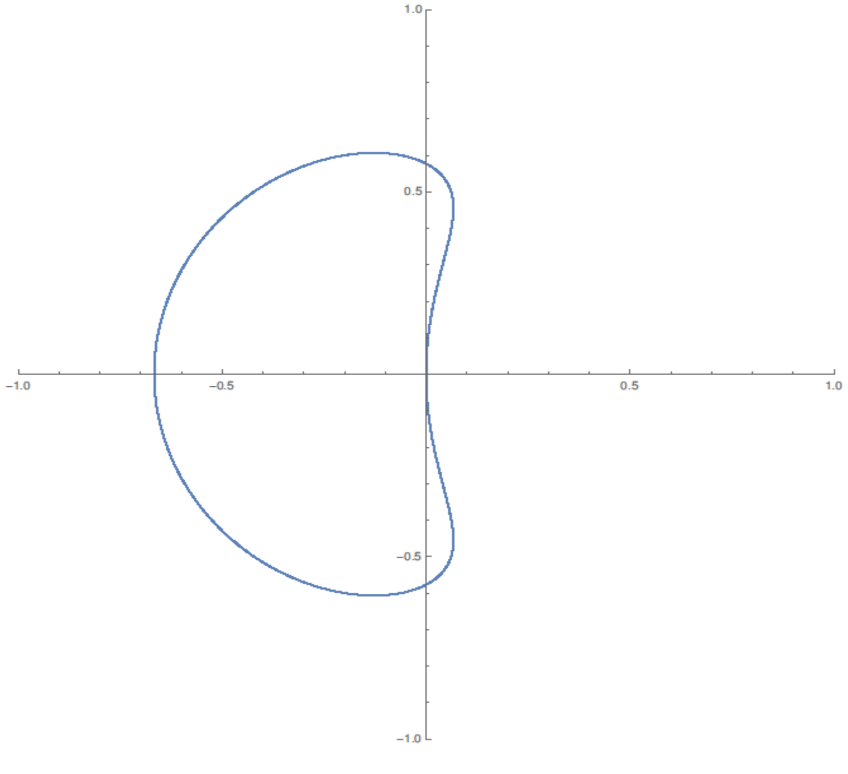


FIGURE 1. Stabilty region of $\frac{y_{n+1}-y_n}{\triangle t} = 2f_n - f_{n-1}$

EXAMPLE 20 (Transport). *Oscillations of a pendulum are not a compelling or high impact application (possibly aside from clock makers). However, the standard test problem for transport (when something is moved around by a liquid or gas) is*

$$y' = \pm i\omega y, \omega \text{ a real number.}$$

To see why we briefly consider the simplest transport problem: for $u(x,t)$ a concentration of something that is moves to the right with speed $a > 0, u(x,t)$ satisfies the partial differential equation

$$\begin{aligned} \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} &= 0, -\infty < x < \infty, t > 0, \\ u(x,0) &= f(x) \text{ , the concentration initially.} \end{aligned}$$

It is easy to check by direct substitution that the exact solution is

$$u(x,t) = f(x - at)$$

which is the profile $f(x)$ moving to the right with speed a. The simplest case is when $f(x)$ is one Fourier mode such as $f(x) = cos(nx) + sin(nx)$ and, as usual we shall do the calculation with $f(x) = e^{i\omega x}$ because its easier. Then write

$$u(x,t) = y(t)e^{inx}$$

substitute into $\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$ and cancel gives

$$\begin{gathered} \frac{\partial}{\partial t}(y(t)e^{inx}) + a\frac{\partial}{\partial x}(y(t)e^{inx}) = 0 \Leftrightarrow \\ y'(t)e^{nx} + ay(t)ine^{inx} = 0 \Leftrightarrow \\ y'(t) = -i(na)y. \end{gathered}$$

Similarly, if the transport is to the left we get $y'(t) = +i(an)y$. In all cases, faster transport speed (larger a) means larger $\omega = na$ in the test problem $y' = \pm i\omega y$.

Example 21 (Stability region of the backward Euler method). *The backward Euler method reads*

$$\frac{y_{n+1} - y_n}{\triangle t} = \lambda y_{n+1} \quad \text{or} \quad y_{n+1} = (1 - \triangle t\lambda)^{-1} y_n.$$

This means that

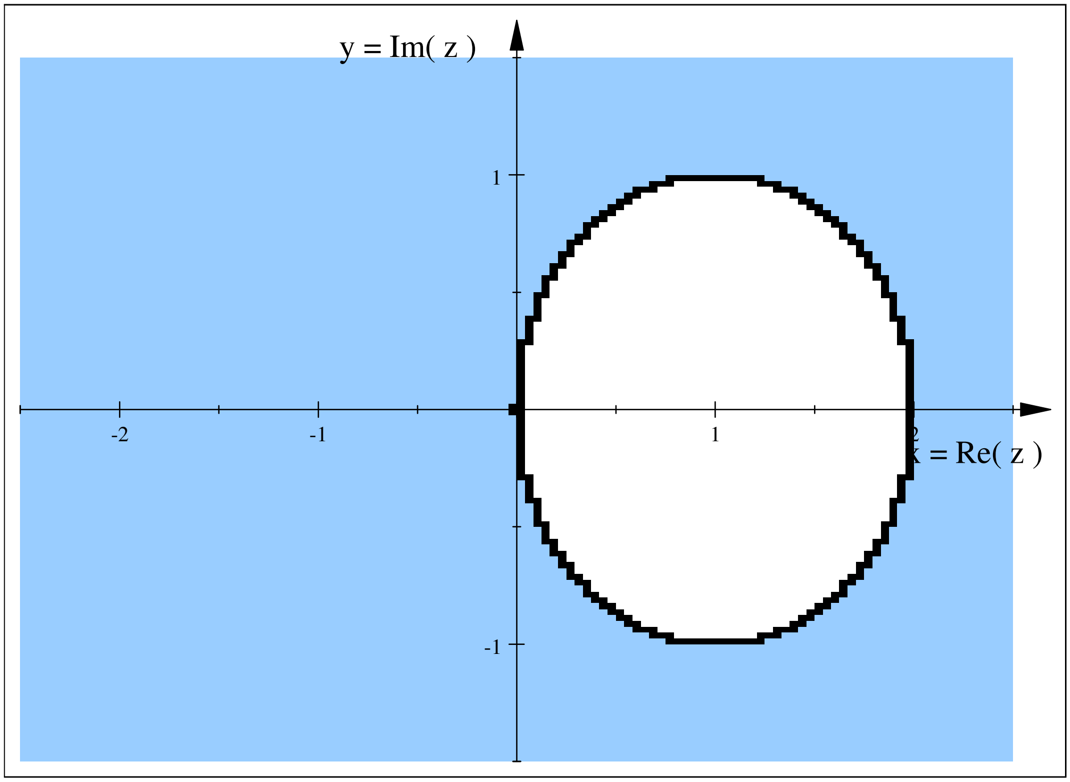
$$y_n = (1 - \triangle t\lambda)^{-n} y_0 \quad \text{and thus } y_n \to 0 \text{ if and only if } |1 - \triangle t\lambda| > 1.$$

Let $\lambda = \alpha + \beta i$ so that

$$|1 - \triangle t\lambda|^2 = (1 - \triangle t\alpha)^2 + (\triangle t\beta)^2.$$

Thus, the stability region is the exterior of the circle of radius 1 centered at 1

$$\{z = x + iy \in \mathbb{C} : (1 - x)^2 + y^2 > 1\}.$$



Stability region of backward Euler

If $\triangle t\lambda$ lies in this (shaded) region the approximate solution will decay to zero exponentially fast as t increases (correctly). If $\triangle t\lambda$ lies outside of this region, the approximate solution will blow up exponentially as $t \to \infty$.

Examining the stability region, we see that

- The backward Euler method is A-stable.
- If one solves $y' = -10,000y$ by backward Euler, the approximate solution will be stable for any $\triangle t > 0$.
- If one solves the pendulum equation written as a first order system, the backward Euler method will produce an approximate solution that is over damped.

Let us consider the Trapezoid rule.

Example 22 (Stability region of the Trapezoid rule). *The trapezoidal method reads*

$$\frac{y_{n+1} - y_n}{\triangle t} = \lambda \frac{y_{n+1} + y_n}{2} \quad or \quad y_{n+1} = \frac{1 + \frac{1}{2}\triangle t\lambda}{1 - \frac{1}{2}\triangle t\lambda} y_n.$$

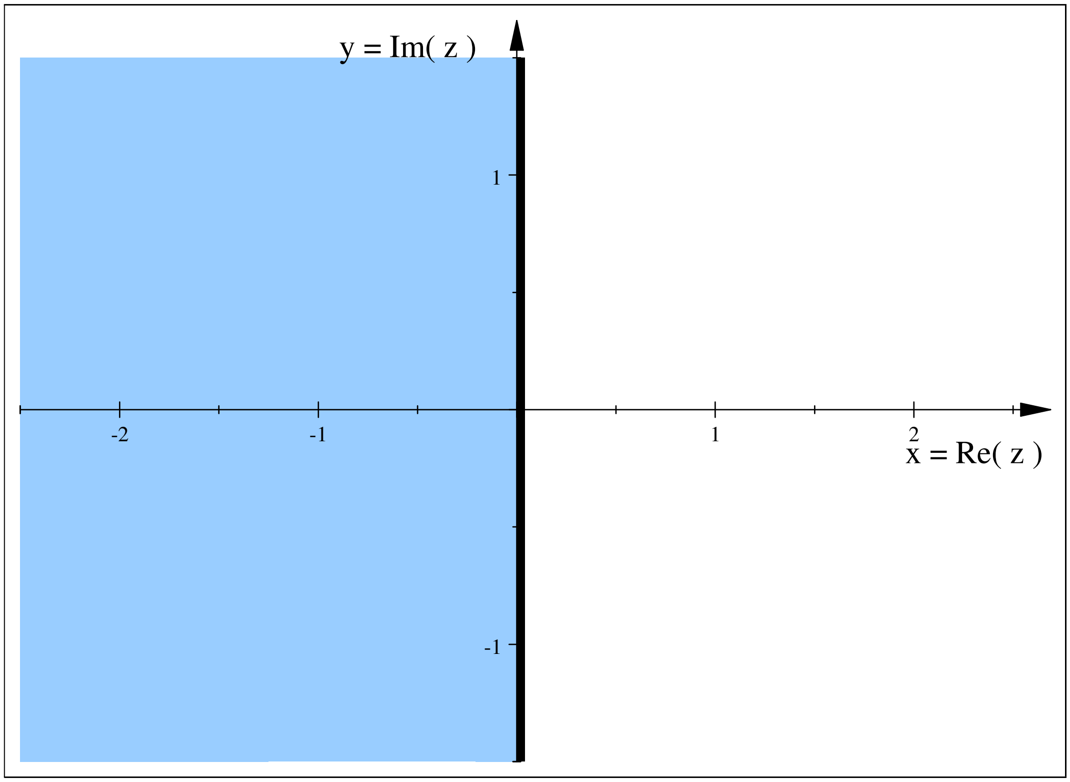
This means that[3]

$$y_n = \left(\frac{1+\frac{1}{2}\triangle t\lambda}{1-\frac{1}{2}\triangle t\lambda}\right)^n y_0 \quad \textit{and thus } y_n \to 0 \textit{ if and only if}$$

$$\left|\frac{1+\frac{1}{2}\triangle t\lambda}{1-\frac{1}{2}\triangle t\lambda}\right| < 1 \textit{ or} : |1+\frac{1}{2}\triangle t\lambda| < |1-\frac{1}{2}\triangle t\lambda|.$$

We proceed as above. Let $\lambda = \alpha + \beta i$ and calculate both sides. After some calculations, the result is the stability region is exactly the left half plane

$$\{z = x + iy \in \mathbb{C} : \mathrm{Re}(z) < 0\}.$$



Stability region of the trapezoid rule

Clever methods have been developed to calculate the stability region of a method. We give next some of the stability regions for popular methods produced by these techniques. (These regions are copied from http://www.mathworks.com.)

The Adams-Bashforth methods. These are explicit multistep methods. They are not A-stable and their stability regions generally get smaller as accuracy increases. One point to notice is that AB3's stability region includes an interval

[3] Observe that $\left|\frac{1+\frac{1}{2}\triangle t\lambda}{1-\frac{1}{2}\triangle t\lambda}\right| = \left|\frac{1+z}{1-z}\right|$ where $z = \frac{1}{2}\triangle t\lambda$. The function $\frac{1+z}{1-z}$ is a classic fractional linear transformation studied in complex analysis.

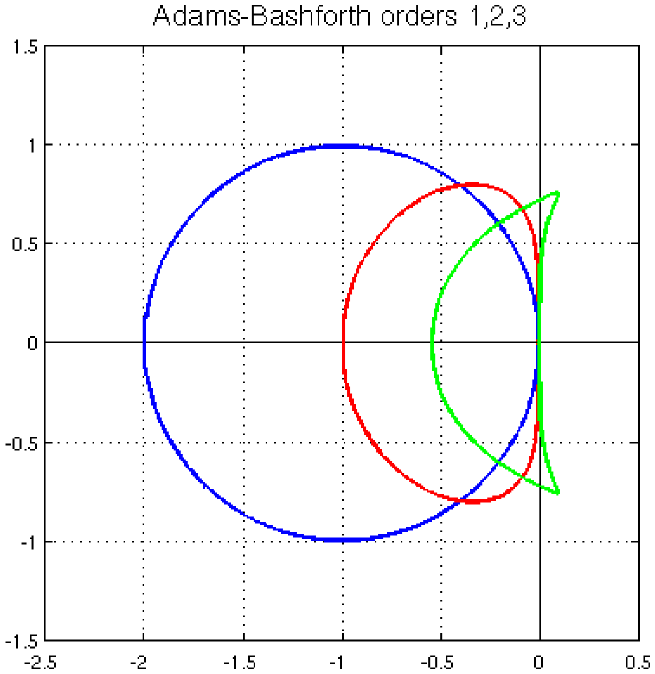


FIGURE 2. AB1,2,3 stability regions - interiors of figures

of the imaginary axis. Thus AB3 is stable (for $\triangle t$ small enough) for pendulum equations.

The Runge-Kutta methods. The explicit RK methods are also not A-stable. Their stability regions do increase as the methods accuracy increases.

The BDF Methods. The BDF methods are implicit. Their stability regions are the *exteriors* of the regions plotted below. Observe that BDF1 (backward Euler) and BDF2 are both A-stable and BDF3 is nearly A stable.

EXERCISE 42. *Analyze asymptotic stability for the Leapfrog method. Show that approximate solutions never* $\to 0$ *as* $t \to \infty$.

EXERCISE 43. *Verify for Euler's method that if* $\triangle t\lambda$ *lies outside of the stability region, the approximate solution will blow up exponentially as* $t \to \infty$. *If it lies on the boundary of this region, its magnitude will be constant.*

EXERCISE 44. *Consider the wave equation (which is the equation for sound propagation in a fluid at rest). Repeat the analysis performed for the transport problem and show that the appropriate test problem is also* $y' = \pm i\omega y$.

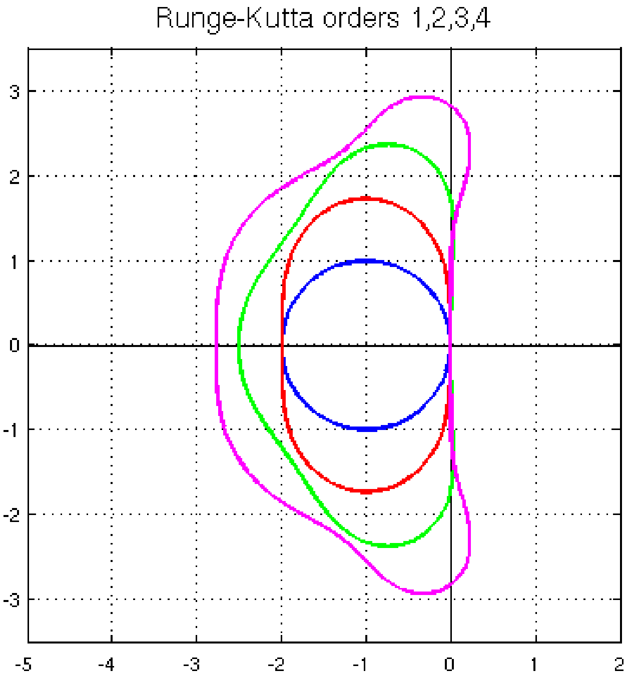


FIGURE 3. RK1 to 4 stability regions- interiors of curves

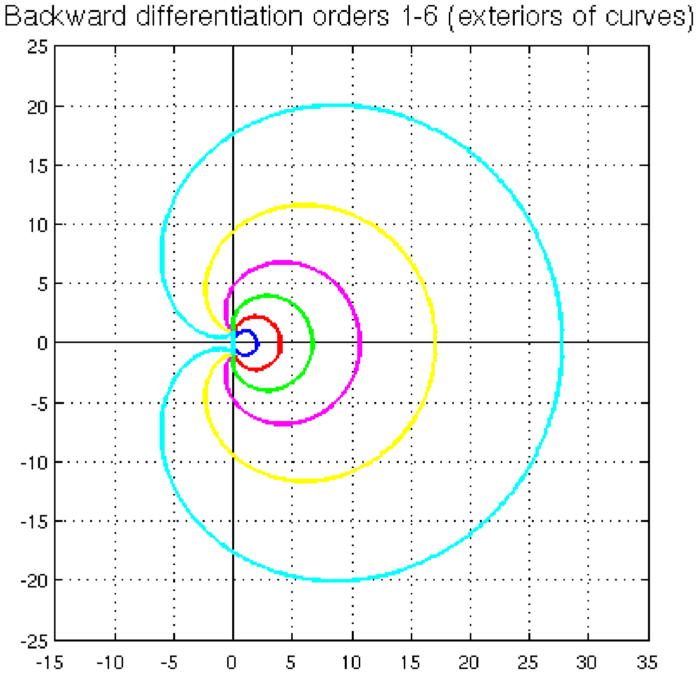


Figure 4. BDF methods stability regions

CHAPTER 7

The Dahlquist theory

1. The Dahlquist barriers

Recall that an A-stable method is one that is asymptotically stable for any $\triangle t$ for any (model test) problem that is itself asymptotically stable:

DEFINITION 13. *Consider a numerical method for the IVP for the standard test problem $y'(t) = \lambda y(t)$. A point $z = \triangle t\lambda$ is in the stability region of if the approximations y_n produced by the method for that value of $\triangle t$ and that value of λ satisfy $y_n \to 0$ as $n \to \infty$. A method is A-stable if its stability region includes the entire left half-plane.*

With an A-stable method the time step can be adapted strictly to resolve solution behavior for accuracy. With a method that is not A-stable there will be cases where adaptivity will function to produce a stable approximation rather than to obtain desired accuracy. It thus seems clear that , all things being equal, an A–stable method is to be preferred over one that is not A stable. Unfortunately, Dahlquist that A-stability presents three fundamental barriers to methods.

THEOREM 9 (Three Dahlquist barriers). *There are no A-stable explicit single step or linear multi-step methods.*

An implicit single or multi-step method that is A-stable can have at most second order accuracy.

The second order A-stable method with the greatest accuracy (in the sense of smallest local truncation error[1]) is the trapezoid rule. For the trapezoid rule the local truncation error is $\tau = -\frac{1}{12}\triangle t^3 y''(t_n) + O(\triangle t^4)$.

This landmark result had a number of important consequences. First there was intensive study of problems for which there was nothing better than just use an implicit method and solve the nonlinear system at every time step. These types of IVPs systems are now called *stiff systems*. Second, the trapezoid rule has been the subject of intense study and many small tweaks of it have been developed. Third, there has been an intense study of alternative stability concepts to see if requesting a form of stability in between $0-$stability and $A-$stability can both be useful for some applications and break one of the above Dahlquist barriers. Some alternative stability theories are summarized below.

DEFINITION 14 (Different stability concepts). *Let a method's stability region be $\mathcal{R}$.*

- *A method is **A-stable** if $\mathcal{R} \supset \{z : \mathrm{Re}(z) < 0\}$.*
- *A method is A_α**-stable** if $\mathcal{R} \supset \{z : \mathrm{Re}(z) < 0$ and $|\arg(z)| < \alpha\}$.*

[1] The LTE for the trapezoid rule is $LTE = -\frac{1}{12}y''(t_n)\triangle t^3 + \mathcal{O}(\triangle t^4)$. Thus here 'most accurate' means the constant mulitplier 1/12 is minimal.

- *A method is $A(0)$-**stable** if it is A_α-stable for some $\alpha > 0$ (however small).*
- *A method is A_0-**stable** if $\mathcal{R} \supset \{z : \mathrm{Re}(z) < 0$ and $\mathrm{Im}(z) = 0\}$.*
- *A method is **stiffly stable** if $\mathcal{R} \supset \mathcal{R}_1 \cup \mathcal{R}_2$ where*

$$\begin{aligned}\mathcal{R}_1 &= \{z : \mathrm{Re}(z) < -a < 0 \text{ for some } a > 0\} \text{ and}\\ \mathcal{R}_2 &= \{z : -a \leq \mathrm{Re}(z) < 0, -c \leq \mathrm{Im}(z) \leq +c \text{ for some } a > 0, c > 0\}.\end{aligned}$$

- *A method is L-**stable (or strongly A-stable)** if it is $A-$stable and its approximation y_n satisfies $y_n \to 0$ for n fixed but as $\lambda \to -\infty$.*

The motivation for L-stability is that it captures some aspect not in A-stability of the true solution of $y' = \lambda y$. Namely, the solution has the property that

$$y(t) \to 0 \text{ as } \mathrm{Re}(\lambda) \to -\infty \text{ for fixed } t.$$

2. Two step Methods

Since accuracy is limited, the most commonly used A stable methods are 1 and 2 step methods. The most general, second order 1 step method is the θ-method interpolating between the Trapezoid rule and implicit Euler.

THEOREM 10 (A-stable 1 step methods). *All 1 step, A stable methods of order ≥ 1 (i.e., LTE=$O(\triangle t^2)$ and higher) take the form*

$$\begin{aligned}\frac{y_{n+1} - y_n}{\triangle t} &= \theta f(t_{n+1}, y_{n+1}) + (1-\theta) f(t_n, y_n),\\ \frac{1}{2} &\leq \theta \leq 1.\end{aligned}$$

These are A stable for $\frac{1}{2} \leq \theta \leq 1$ but second order (i.e., LTE=$O(\triangle t^3)$) only for $\frac{1}{2} = \theta$.

Two step methods

$$\frac{\alpha_2 y_{n+1} + \alpha_1 y_n + \alpha_0 y_{n-1}}{\triangle t} = \beta_2 f(t_{n+1}, y_{n+1}) + \beta_1 f(t_n, y_n) + \beta_0 f(t_{n-1}, y_{n-1})$$

can also be useful for their other properties. The above 2 step method is consistent if

$$\begin{aligned}\alpha_2 + \alpha_1 + \alpha_0 &= 0,\\ 2\alpha_2 + 1\alpha_1 + 0\alpha_0 &= 0,\\ \beta_2 + \beta_1 + \beta_0 &= 1.\end{aligned}$$

Thus the general method has 6 parameters but must satisfy 3 conditions so 3 free parameters remain. Rewriting the method in terms of the remaining free parameters gives

$$\frac{(1+\xi) y_{n+1} - (1+2\xi) y_n + \xi y_{n-1}}{\triangle t} = \theta f(t_{n+1}, y_{n+1}) + (1-\theta+\phi) f(t_n, y_n) - \phi f(t_{n-1}, y_{n-1}).$$

The LTE is easily calculated to be

$$LTE = (\phi - \xi + \theta - \frac{1}{2}) \triangle t^2 y''(t_n) + \mathcal{O}(\triangle t^3)$$

so that the method is first order accurate except when the extra condition holds that

$$\phi - \xi + \theta - \frac{1}{2} = 0.$$

If this is true the LTE is then

$$LTE = (-\xi + 2\theta - \frac{5}{6})\triangle t^3 y'''(t_n) + \mathcal{O}(\triangle t^4),$$
$$\text{when } \phi - \xi + \theta = \frac{1}{2}.$$

Thus the following characterization (from Dahlquist [D78]) of them is helpful.

Theorem 11 (A-stable 2 step methods). *Concerning 2 step methods*

$$\frac{\alpha_2 y_{n+1} + \alpha_1 y_n + \alpha_0 y_{n-1}}{\triangle t} = \beta_2 f(t_{n+1}, y_{n+1}) + \beta_1 f(t_n, y_n) + \beta_0 f(t_{n-1}, y_{n-1})$$

These have order ≥ 2 *(i.e., LTE=*$O(\triangle t^3)$*) if*

$$\begin{array}{rr} \alpha_0 = -1 + \alpha_2, & \alpha_1 = 1 - 2\alpha_2, \\ \beta_0 = \frac{1}{2} - \alpha_2 + \beta_2, & \beta_1 = \frac{1}{2} + \alpha_2 - 2\beta_2. \end{array}$$

They are A stable if

$$\alpha_2 \geq \frac{1}{2}, \beta_2 \geq \alpha_2/2$$

and are L stable if

$$\alpha_2 > \frac{1}{2}, \beta_2 > \alpha_2/2.$$

Alternately, a consistent 2 step method

$$\frac{(1+\xi)y_{n+1} - (1+2\xi)y_n + \xi y_{n-1}}{\triangle t} = \theta f(t_{n+1}, y_{n+1}) + (1-\theta+\phi) f(t_n, y_n) - \phi f(t_{n-1}, y_{n-1}).$$

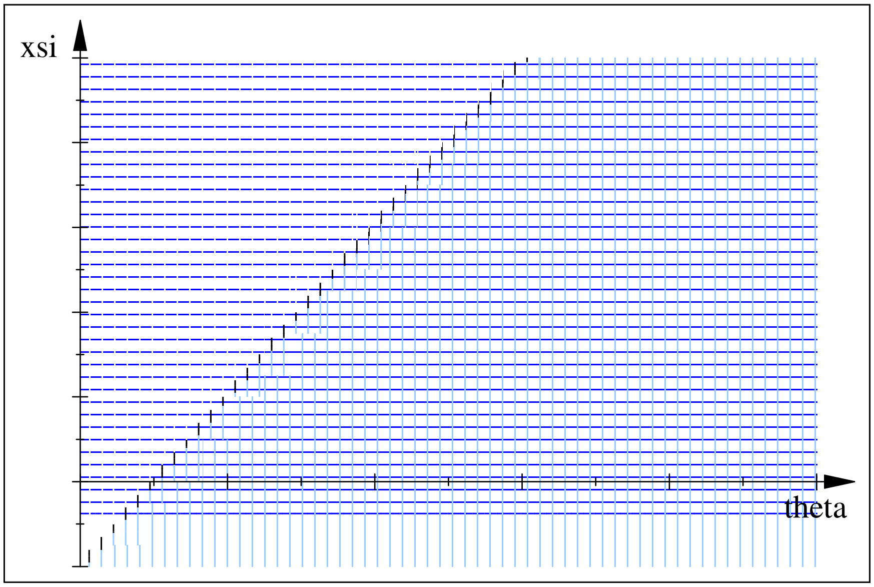
is A stable if and only if

$$\begin{array}{rcl} \theta & \geq & \phi + 1/2, \\ \xi & \geq & -1/2, \\ \xi & \leq & \theta + \phi - 1/2. \end{array}$$

For second order A-stable methods the conditions can be written in terms of 2 parameters and become

$$\xi \geq -1/2 \ \text{ and } \xi \leq 2\theta - 1.$$

This region $\xi \geq -1/2$ and $\xi \leq 2\theta - 1$ is visualized below.



Crosshatched A-stable θ, ξ values

EXERCISE 45. *Show that the trapezoid rule is not L−stable.*

EXERCISE 46. *Consider the 2 step methods below. Using the results in this section, analyze their accuracy and stability:*

$$\begin{array}{rl} BDF2 & \theta = 1, \xi = 1/2, \phi = 0 \\ Contractive - Adams & \theta = 3/4, \xi = 0, \phi = -1/4 \end{array}$$

EXERCISE 47. *Analyze stability of the following proposed by Durran:*

$$\frac{y_{n+1} - y_n}{\triangle t} = \frac{3}{4} f(t_{n+1}, y_{n+1}) + \frac{1}{4} f(t_{n-1}, y_{n-1}). \tag{TR}$$

EXERCISE 48. *Show that BDF2 approximation satisfies $y_n \to 0$ for n fixed and $\lambda \to -\infty$ and thus BDF2 is L−stable.*

CHAPTER 8

Stiffness and Implicit Methods

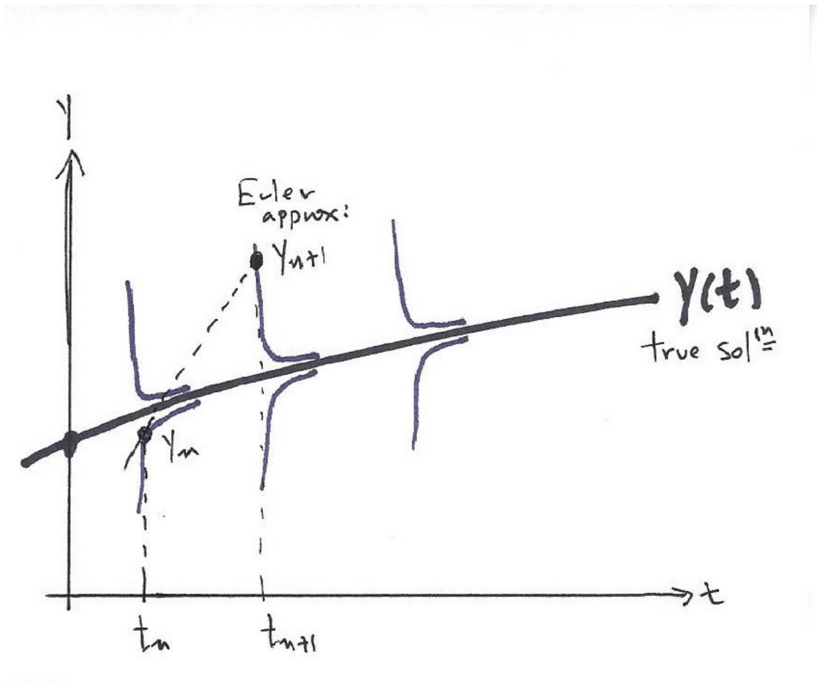
A stiff linear system is a system

$$\frac{d}{dt}\overrightarrow{y} = A\overrightarrow{y}$$

where the matrix A has some eigenvalues that are negative but very large in absolute value and others of moderate size. For such systems the amount of stiffness is often quantified by the stiffness ratio defined to be

$$\text{Stiffness ratio } := \frac{\max|\lambda|}{\min|\lambda|}$$

The dynamics of a stiff system are simple: the slowest decay solutions dominate and the faster decay modes quickly damp out. The next figure shows a depiction of this and the bad result for approximating a stiff system with Euler's method.



Depiction of stiffness and how Euler's method goes wrong

One definition of stuffiness is that:

Stiff systems are those for which the solution sought is slowly varying but perturbations of the solution are damped out at a much faster rate.

Here "*much faster*" means that the system cannot be solved within time and resource constraints by explicit methods, even adaptive explicit methods. Thus, while the stiffness ratio has a precise mathematical definition, "stiffness" only as meaning with respect to how people want to use the results of a simulation including how fast they need the result, how miuch computer resources are available and how much cost to generate the result is acceptable. For example, Google gives the following possible definitions of stiffness.

Definitions of 'stiff system'

In Civil Engineering:

Stiffness is the rigidity of an object — the extent to which it resists deformation in response to an applied force. The complementary concept is flexibility or pliability: the more flexible an object is, the less stiff it is.

Wikipedia:

... a stiff equation is a differential equation for which certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small. It has proven difficult to formulate a precise definition of stiffness, but the main idea is that the equation includes some terms that can lead to rapid variation in the solution.

J. D. Lambert:

If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use in a certain interval of integration a steplength which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be stiff in that interval.

paraphrased often to read:

If a numerical method is forced to use, in a certain interval of integration, a step length which is excessively small in relation to the smoothnessgif of the exact solution in that interval, then the problem is said to be stiff in that interval.

DM Thomas:

"In applications you usually find out if your ODE is stiff by numerically integrating it and watching Runge Kutta fall apart. Then you find the eigenvalues of the linearized DE and realize after the fact you have a stiff system. Then you grumble as you program in your stiff numerical solvers."

CW Gear:

Although it is common to talk about "stiff differential equations," an equation per se is not stiff, a particular initial value problem for that equation may be stiff, in some regions, but the sizes of these regions depend on the initial values and the error tolerance. (C. W. Gear (1982): Automatic detection and treatment of oscillatory and/or stiff ordinary differential equations. In: Numerical integration of differential equations, Lecture notes in Math., Vol. 968, p. 190-206.)

From 'Glossary of Meteorological Terms':

A system of differential equations with solutions that contain a rapidly damping component (as would describe the displacement of a stiff spring when stretched and then released).

Germund Dahlquist quoted in Exercise 9.1 of Shampine (1994):

"The stiffness ratios used by some authors ... may be of use when one estimates the amount of work needed, if a stiff problem is to be solved with an explicit method, but they are fairly irrelevant in connection with implicit methods...."

From around the web:

We say that a problem is stiff if the following conditions are fulfilled. A) No solution component is unstable, or equivalently, no eigenvalue of the Jacobian matrix has a real part which is at all large and positive, and at least some component is very stable, that is, at least one eigenvalue has a negative part which is negative and large. B) The solution is slowly varying with respect to the negative real part of the eigenvalues.

Stiff if eigenvalues of the Jacobi matrix df/dy are negative and large in magnitude.

Example 23. *As a concrete example, the problem*

$$y' = -100y + 101t + 101, y(0) = 1$$

has general solution and true solution

$$\begin{aligned} y_{general}(t) &= 1 + t + Ce^{-100t} \text{ and since } y(0) = 1 : \\ y(t) &= 1 + t. \end{aligned}$$

Perturbations of the solution, such as by discretizations are damped rapidly like e^{-100t}. The system is stiff means that perturbations are small but their derivatives are very large. This makes explicit method crash, as depicted above.

Thus, we take the model problem

$$y'(t) = \lambda y(t) \text{ where } \lambda < 0 \text{ and } |\lambda| \text{ is very large.}$$

"*Very large*" simply means so large that explicit methods cannot be used to get the solution within time and resource constraints due to the accompanying timestep restriction for stability. The solution of this problem is $y(t) = e^{\lambda t}y(0) \to 0$ very fast as $t \to \infty$. Although the behavior is not exotic, there is a natural time scale associated with this decay: the half life:

$$\frac{1}{2}\text{life} = \frac{\ln(2)}{|\lambda|}.$$

The more negative λ is, the shorter the half-life of the solution and (in that sense) the faster things happen in the IVP.

We look at a few examples to see possible sources of stiffness.

Example 24 (A Second Order IVP). *The following second IVP seems inoffensive*

$$y'' + 1001y' + 1000y = 0, t > 0$$
$$y(0) = 1 \quad \textit{and} \quad y'(0) = -1.$$

However, it has solution (which can be found by standard methods)

$$y(t) = C_1e^{-t} + C_2e^{-1000t},$$

where $C_{1,2}$ are determined by the initial conditions[1]. *This solution exhibits rate constants $\lambda = -1, -1000$ which begins to be stiff. The so called stiffness ratio of this problem is*

$$\textit{Stiffness ratio} := \frac{\max|\lambda|}{\min|\lambda|} = 1000.$$

[1] $y(0) = 1$ gives $C_1 + C_2 = 1$ and $y'(0) = -1$ gives $C_1(-1) + C_2(-1000) = -1$. This is a 2 by 2 linear system for C_1, C_2.

If we write the second order IVP as one for a first order system in the usual way ($y_1 = y$, $y_2 = y'$ *etc.) we get*

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1000 & -1001 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

The eigenvalues of the above 2×2 *matrix are easily found to be* $\lambda = -1$ *&* -1000. *The stability region of RK4 shows that if this system is approximated by RK4, it will converge nicely if* $\triangle t < 0.002$ *but the approximate solution will blow up exponentially if* $\triangle t \geq 0.003$.

Example 25 (Heat Conduction). *The IVP for heat conduction in a bar in its simplest form is a partial differential equation for the temperature* $u(x,t)$ *at the point* x *at time* t. *The initial temperature* $u(x,0)$ *and the temperature at both ends* $u(0,t)$ *and* $u(1,t)$ *are known and the internal temperature satisfies*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \text{ for } 0 < x < 1, t > 0.$$

To predict the temperature it is converted into an IVP for system of ODEs as follows. Pick a space mesh width $\triangle x = 1/(N+1)$ *and let*

$$x_j = j\triangle x \text{ and } u_j(t) = \text{ approximation to } u(x_j, t).$$

We approximate

$$\frac{\partial^2 u}{\partial x^2}(x_j, t) \simeq \frac{u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)}{\triangle x^2} \text{ (which has error } O(\triangle x^2)).$$

We then have the system of equations for $u_j(t)$

$$\begin{aligned} u_1' &= \frac{-2u_1 + u_2}{\triangle x^2} \\ u_2' &= \frac{+u_1 - 2u_2 + u_3}{\triangle x^2} \\ &\cdots\cdot \\ u_{N-1}' &= \frac{+u_{N-2} - 2u_{N-1} + u_N}{\triangle x^2} \\ u_N' &= \frac{+u_{N-1} - 2u_N}{\triangle x^2}. \end{aligned}$$

This is written in matrix form as

$$\frac{d}{dt}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \frac{1}{\triangle x^2}\begin{bmatrix} -2 & +1 & & \\ +1 & -2 & +1 & \\ & \searrow & \searrow & \searrow \\ & & +1 & +2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}.$$

The above matrix is denoted tridiag$(+1, -2, +1)$. *Its structure is so regular that an explicit formula exists for its eigenvalues. Even without an explicit formula the stiffness ratio could be estimated based on the (plausible and correct) assumption that the eigenvalues of* $\triangle x^{-2}$*tridiag*$(+1, -2, +1)$ *approximate the first* N *eigenvalues of the continuous problem:*

$$\begin{aligned} -\phi_n''(x) &= \lambda_n \phi_n(x), 0 < x < 1, \\ \phi_n(0) &= 0, \phi_n(1) = 0. \end{aligned}$$

These are easily calculated. For the matrix we have the following.

THEOREM 12 (Eigenvalues of tridiag(1,-2,1)). *'The eigenvalues of the* $N \times N$ *matrix* $\frac{1}{\triangle x^2}$*tridiag*$(+1, -2, +1)$ *are*

$$\lambda_j = -\frac{4}{\triangle x^2}\sin^2\left(\frac{j\pi}{2N}\right), j = 1, \cdot\cdot\cdot, N.$$

Specifically[2]

$$\begin{aligned} j &= 1 : \textit{smallest eigenvalue } \simeq -2\pi\ , \\ j &= N : \textit{largest eigenvalue } \simeq -4\left(\triangle x\right)^{-2} \end{aligned}$$

EXAMPLE 26. *Suppose* $N = 1000$ *(for example) so* $\triangle x = 10^{-6}$ *then the stiffness ratio is*

$$\textit{stiffness ratio} = \frac{4\triangle x^{-2}}{2\pi} = \frac{1}{2\pi}10^{-6}.$$

RK2 is stable if and only if (using $0 > -2\pi \geq \lambda \geq -4\triangle x^{-2}$*)*

$$\begin{aligned} -2 &< \triangle t\lambda_j < 0 \Leftrightarrow \\ \triangle t\frac{4}{10^{-6}} &< 2 \Leftrightarrow \\ \triangle t &< \frac{1}{2}\times 10^{-6}, \end{aligned}$$

which is an extraordinary small size for a problem whose solution is not doing anything dramatic.

This last example is a critical one as it represents all processes that are dominated by diffusion. For these it is typical that the eigenvalues are large (in absolute value), negative and real. A-stability requires more than is needed for this application where the eigenvalues are real and negative (so stability for complex eigenvalues is not necessary). Further, the solution behavior is also very specific: the solution decays monotonically to zero and, when the data is positive, preserves positivity. This is one motivation for studying stability beyond A-stability such as A_0 stability, stiffly stable and L-stability.

DEFINITION 15 (Stability addressing diffusion dominated problems). *Let a method's stability region be* $\mathcal{R}$.

- *A method is* A_0***-stable*** *if* $\mathcal{R} \supset \{z : \mathrm{Re}(z) < 0 \text{ and } \mathrm{Im}(z) = 0\}$.
- *A method is **stiffly stable** if* $\mathcal{R} \supset \mathcal{R}_1 \cup \mathcal{R}_2$ *where*

$$\begin{aligned} \mathcal{R}_1 &= \{z : \mathrm{Re}(z) < -a < 0 \textit{ for some } a > 0\} \textit{ and} \\ \mathcal{R}_2 &= \{z : -a \leq \mathrm{Re}(z) < 0, -c \leq \mathrm{Im}(z) \leq +c \textit{ for some } a > 0, c > 0\}. \end{aligned}$$

- *A method is **L-stable** if it is A−stable and its approximation* y_n *satisfies* $y_n \to 0$ *for* n *fixed but* $\lambda \to -\infty$.

For a nonlinear system,

$$\frac{d}{dt}\overrightarrow{y} = \overrightarrow{f}(t, \overrightarrow{y})$$

[2] This estimate uses

$$\left(\frac{\pi}{2N}\right)^2 \leq \sin^2\left(\frac{n\pi}{2N}\right) \leq 1 = \sin^2(\frac{N\pi}{2N}).$$

stiffness refers to the rates (time scales) at which trajectories squeeze together. If $\overrightarrow{x}(t)$, $\overrightarrow{y}(t)$ are solutions with different initial data then their difference satisfies

$$\begin{aligned}\frac{d}{dt}(\overrightarrow{x}-\overrightarrow{y}) &= \overrightarrow{f}(t,\overrightarrow{x})-\overrightarrow{f}(t,\overrightarrow{y}) \\ &\textit{or} \\ \frac{d}{dt}(\overrightarrow{x}-\overrightarrow{y}) &= \left[\frac{\partial f}{\partial y}(t,\overrightarrow{\xi})\right](\overrightarrow{x}-\overrightarrow{y})\end{aligned}$$

The matrix $\frac{\partial f}{\partial y}$ is the Jacobi matrix or derivative matrix of the system. For a system of N equations, it is an $N \times N$ matrix with entries

$$\left(\frac{\partial f}{\partial y}\right)_{ij} = \frac{\partial f_i}{\partial y_j}.$$

Definition 16 (Another definition of stiffness). *A stiff system* $\frac{d}{dt}\overrightarrow{y} = \overrightarrow{f}(t,\overrightarrow{y})$ *is one for which the eigenvalues of the Jacobi matrix* $\frac{\partial f_i}{\partial y_j}$ *are negative and so large in absolute value as to preclude its solution by explicit methods.*

Stiffness can be measured various ways. One common method is in terms of the Stiffness Ratio.

Definition 17 (Stiffness ratio). *The stiffness ratio is*

$$S := \frac{\max_i |\operatorname{Re}(\lambda_i)|}{\min_j |\operatorname{Re}(\lambda_j)|}.$$

The obvious solution for stiff systems is to use implicit methods and the obvious choices are the trapezoid rule and BDF2, considered in the next section. The most general result for 2 step methods is the following.

Theorem 13 (A-stable 2 step methods). *Concerning 2 step methods*

$$\frac{\alpha_2 y_{n+1}+\alpha_1 y_n+\alpha_0 y_{n-1}}{\triangle t} = \beta_2 f(t_{n+1},y_{n+1})+\beta_1 f(t_n,y_n)+\beta_0 f(t_{n-1},y_{n-1})$$

These have order ≥ 2 *(i.e., LTE=*$O(\triangle t^3)$*) if*

$$\begin{aligned}\alpha_0 = -1+\alpha_2, &\qquad \alpha_1 = 1-2\alpha_2, \\ \beta_0 = \tfrac{1}{2}-\alpha_2+\beta_2, &\quad \beta_1 = \tfrac{1}{2}+\alpha_2-2\beta_2.\end{aligned}$$

They are A stable if

$$\alpha_2 \geq \frac{1}{2}, \beta_2 \geq \alpha_2/2$$

and are L stable if

$$\alpha_2 > \frac{1}{2}, \beta_2 > \alpha_2/2.$$

Alternately, a consistent 2 step method

$$\frac{(1+\xi)y_{n+1}-(1+2\xi)y_n+\xi y_{n-1}}{\triangle t} = \theta f(t_{n+1},y_{n+1})+(1-\theta+\phi)f(t_n,y_n)-\phi f(t_{n-1},y_{n-1}).$$

is A stable if and only if

$$\begin{aligned}\theta &\geq \phi+1/2, \\ \xi &\geq -1/2, \\ \xi &\leq \theta+\phi-1/2.\end{aligned}$$

Exercise 49. *Estimate the stiffness ratio for the discretized heat equation equation. Hint: first show that*

$$0 > -\frac{4}{h^2}\left(\frac{\pi}{2}h\right)^2 \geq \lambda_j \geq -\frac{4}{h^2}.$$

1. The Trapezoid rule

The trapezoid rule

$$\text{(TR)} \qquad \frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_n, y_n).$$

is the optimal A-stable method according to the Dahlquist theory. Thus, there is a lot of experience working with it. This leads to an understanding of its limitations and a number of fixes for its known weak points.

1.1. The Trapezoid rule is not L-stable.

One drawback (we now explore) of the trapezoid rule is that it is not $L-$stable. To explore L-stability return to the TR for the standard test problem $y' = \lambda y$ where λ is real and negative. It is

$$\frac{y_{n+1} - y_n}{\triangle t} = \lambda \frac{y_{n+1} + y_n}{2} \quad \text{or} \quad y_{n+1} = \frac{1 + \frac{1}{2}\triangle t\lambda}{1 - \frac{1}{2}\triangle t\lambda} y_n.$$

This means that

$$y_n = \left(\frac{1 + \frac{1}{2}\triangle t\lambda}{1 - \frac{1}{2}\triangle t\lambda}\right)^n y_0 \text{ and thus}$$

$$y_n \to 0 \text{ as } \lambda \to -\infty \text{ if and only if}$$

$$\left|\frac{1 + \frac{1}{2}\triangle t\lambda}{1 - \frac{1}{2}\triangle t\lambda}\right| \to 0 \text{ as } \lambda \to -\infty.$$

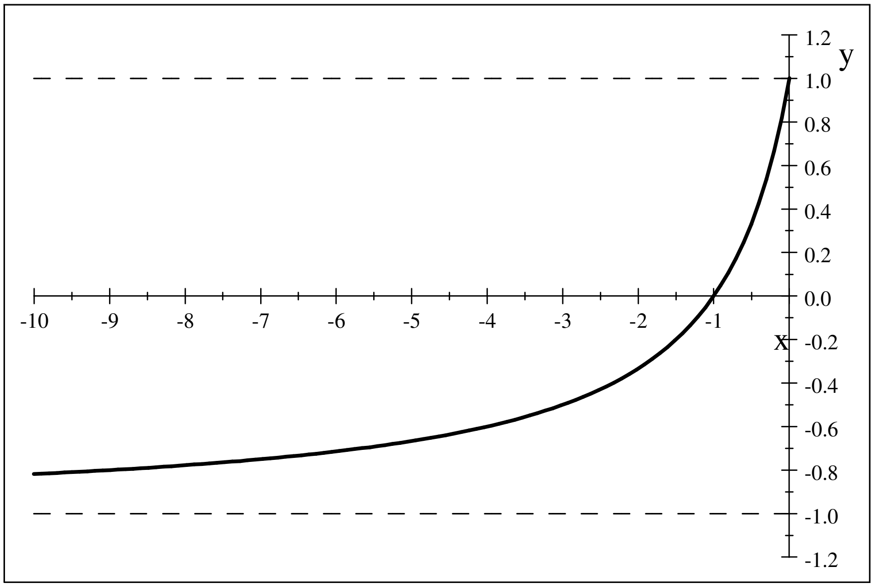
Define

$$a(x) := \frac{1+x}{1-x} \text{ (where } x = \frac{1}{2}\triangle t\lambda).$$

Note that

$$a(\frac{1}{2}\triangle t\lambda) \to -1 \text{ as } \lambda \to -\infty$$

and clearly the trapezoid rule is not $L-$stable. This is also very clear from the plot of $a(x)$, below.



$a(x) = \frac{1+x}{1-x}$ and $y = \pm 1$.

O. Axelsson[3] has proposed a simple correction. This does not restore L-stability but does introduce some damping for large λ. Recall that the θ-method is

$$(\theta\text{-method}) \qquad \frac{y_{n+1} - y_n}{\triangle t} = \theta f(t_{n+1}, y_{n+1}) + (1-\theta) f(t_n, y_n).$$

and reduces to the TR when $\theta = 1/2$. He suggested simply taking

$$\theta = \frac{1}{2} + \triangle t$$

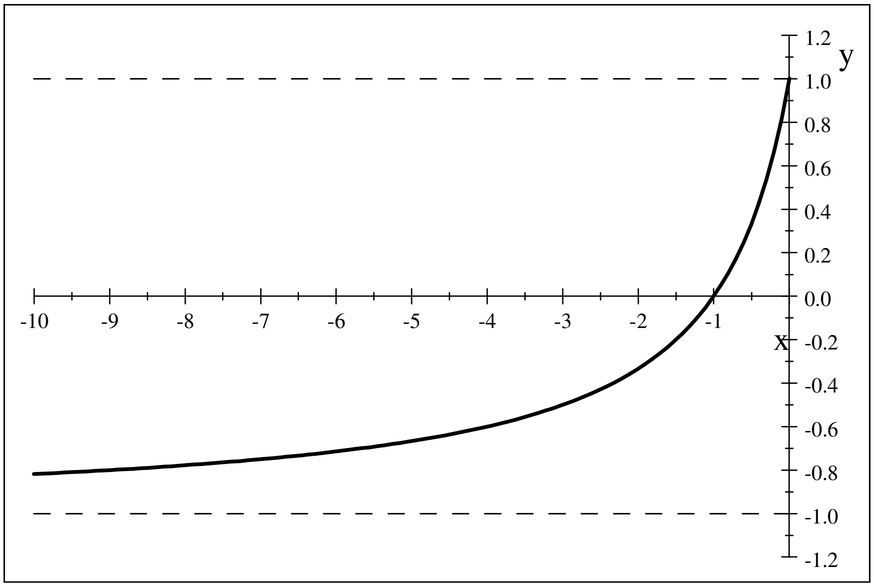
which is a slight bias to the fully implicit method. The resulting method is

$$\begin{aligned} \frac{y_{n+1} - y_n}{\triangle t} &= (\frac{1}{2} + \triangle t) f(t_{n+1}, y_{n+1}) + (\frac{1}{2} - \triangle t) f(t_n, y_n), \\ & \quad \mathit{or} \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(Axelssons correction)} \\ \frac{y_{n+1} - y_n}{\triangle t} &= \left[\frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_n, y_n)\right] + \triangle t \left[f(t_{n+1}, y_{n+1}) - f(t_n, y_n) \right] \end{aligned}$$

Exercise 50. *Show that the correction of Axelsson is second order accurate.*

[3] Professor Owe Axelsson has been a leader in the develpment and analysis of numerical methods for solving large, sparse linear systems, for finite element methods and for timestepping methods. He is the founder of the journal Numerical Linear Algebra with Applications and is on several other. From 1964-1971 he was chairman of the CS department at Chalmers University and the University of Gothenburg, Sweden. He was professor in Numerical Analysis at the University of Nijmegen, The Netherlands from 1979 - 2004. He is currently guest professor at Uppsala University and senior researcher at the Institute of Geonics, Academy of Sciences, Ostrava, Czech Republic. He is listed in the ISI Highly Cited List of Mathematicians.

1.2. The Trapezoid Rule Oscillation. The plot of $a(x)$



$a(x) = \frac{1+x}{1-x}$ and $y = \pm 1$.

also makes the qualitative behavior of the TR solution clear. Note that

$$\begin{aligned} a(x) &= a(\frac{1}{2}\triangle t\lambda) > 0 \text{ only for} \\ -1 &< x \leq 0 \Leftrightarrow \triangle t|\lambda| < 2. \end{aligned}$$

For larger timesteps, i.e., for

$$\triangle t > \frac{2}{|\lambda|},$$

$a(x) < 0$ and thus the approximate solution y_n will alternate sign / *oscillate if the same timestep condition needed for most explicit methods is violated.* The true solution $y(t)$ is positive and decreases to zero monotonically. This oscillation for larger timesteps is a bad feature of the TR and a number of fixes have been developed for it.

1.3. Fixes for the TR Oscillation. Since the TR is used in many practical simulations in which time scales are not well resolved, oscillations occur. Many fixes for these oscillations have been developed. We review some fixes here. The development of a complete and sound mathematical theory for many of there is an open problem.

Axelsson's tilt to the implicit method

As described earlier, it is the following

$$\begin{aligned} \frac{y_{n+1} - y_n}{\triangle t} &= (\frac{1}{2} + \triangle t) f(t_{n+1}, y_{n+1}) + (\frac{1}{2} - \triangle t) f(t_n, y_n), \\ & \quad \textit{or} \qquad \qquad (\textsc{Axelssons correction}) \\ \frac{y_{n+1} - y_n}{\triangle t} &= \left[\frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_n, y_n)\right] + \triangle t \left[f(t_{n+1}, y_{n+1}) - f(t_n, y_n)\right] \end{aligned}$$

Moving averages

Lindberg [**L71**] proposed in 1971 that one average the TR approximation as follows

$$\begin{aligned}\frac{y_{n+1}^{temp}-y_n^{temp}}{\triangle t} &= \frac{1}{2}f(t_{n+1},y_{n+1}^{temp})+\frac{1}{2}f(t_n,y_n^{temp})\\ y_n^{temp} &= \frac{y_{n+1}^{temp}+2y_n^{temp}+y_{n-1}}{4}.\end{aligned}$$

The moving average can be rewritten as

$$\begin{aligned}\frac{y_{n+1}^{temp}-y_n^{temp}}{\triangle t} &= \frac{1}{2}f(t_{n+1},y_{n+1}^{temp})+\frac{1}{2}f(t_n,y_n^{temp})\\ y_n &= y_n^{temp}+\frac{1}{4}\left(y_{n+1}^{temp}-2y_n^{temp}+y_{n-1}\right)\end{aligned}$$

which can easily be shown to reduce the curvature in time $\left(y_{n+1}^{temp}-2y_n^{temp}+y_{n-1}\right)$ of the computed approximation.

The Robert-Asselin time filter

Robert and Asselin proposed something similar[4] with a tunable parameter ν for the CNLF approximation

$$y_n = y_n^{temp}+\frac{\nu}{2}\left(y_{n+1}^{temp}-2y_n^{temp}+y_{n-1}\right), 0<\nu<1.$$

To quantify the effect of the filter step, define a discrete curvature in time, before and after the filter, by

$$\begin{aligned}\kappa_n^{old} &= y_{n+1}^{temp}-2y_n^{temp}+y_{n-1},\\ \kappa_n^{new} &= y_{n+1}^{temp}-2y_n^{temp}+y_{n-1}\end{aligned}$$

We then have the following on reduction in the TR oscillation.

THEOREM 14. *Let* $0<\nu<1$. *The method*

$$\begin{aligned}\frac{y_{n+1}^{temp}-y_n^{temp}}{\triangle t} &= \frac{1}{2}f(t_{n+1},y_{n+1}^{temp})+\frac{1}{2}f(t_n,y_n^{temp})\\ y_n &= y_n^{temp}+\frac{\nu}{2}\left(y_{n+1}^{temp}-2y_n^{temp}+y_{n-1}\right).\end{aligned}$$

reduces the discrete curvature. It satisfies the curvature evolution equation

$$\kappa_n^{new} = (1-\nu)\kappa_n^{old}$$

The proof is a simple rearrangement of the filter step and left as an exercise.

EXERCISE 51. *Let* $\kappa_{old} = y_{n+1}^{temp}-2y_n^{temp}+y_{n-1}$, $\kappa_{new} = y_{n+1}^{temp}-2y_n+y_{n-1}$. *Show that* $\kappa_n^{new} = (1-\nu)\kappa_n^{old}$.

EXERCISE 52. *Consider TR plus the above filter step. Eliminate the temporary variables and write the combination as a LMM. Show it is first order accurate for* $\nu>0$ *fixed and analyze its stability.*

EXERCISE 53. *Prove that the usual TR with Lindberg's moving averages is stable. The proof should be a development of the following strategy: The averaging as formulated does not alter the TR computed solution hence step 1 is stable and Step 2 is a weighted average with positive weights and hence preserves stability.*

[4] They specifically proposed it for the IMEX method CNLF described below. Various improvemments have been developed. The one currently considered best is the RAW or Roberts-Asselin-Williams filter.

Occasionally adding a BE step

One method sometimes advocated is every unit time add 2 steps of backward Euler. These strongly damp the oscillations generated during the previous steps.

Adapt the time step

This is the method strongly advocated by Phil Gresho[5] who notes that, while there is no proof, there is extensive evidence that oscillations do not arise when time adaptivity is added to the TR. He proposes the following as an efficient implementation of an adaptive estimator.

$$\text{Predict\_with\_AB2}$$

$$y_{n+1}^{P} = y_n + \frac{\triangle t_n}{2}\left[(2+\frac{\triangle t_n}{\triangle t_{n-1}})f(t_n,y_n) - \frac{\triangle t_n}{\triangle t_{n-1}}f(t_{n-1},y_{n-1})\right]$$

$$\text{Reuse\_function\_Evaluations\_for\_TR:}$$

$$\frac{y_{n+1}-y_n}{\triangle t} = \frac{1}{2}f(t_{n+1},y_{n+1}) + \frac{1}{2}f(t_n,y_n)$$

$$EST = \frac{|y_{n+1}^{P} - y_{n+1}|}{3(1+\frac{\triangle t_n}{\triangle t_{n-1}})},$$

$$\text{Invert\_TR to get } f_{n+1}$$

$$f(t_{n+1},y_{n+1}) = \frac{2}{\triangle t_n}(y_{n+1}-y_n) - f(t_n,y_n)$$

$$\text{and reuse for next predictor step.}$$

1.4. The one-leg vs. two-leg Trapezoid rule. The issue of "leggedness" has been considered in great detail for the trapezoid rule. The two-leg (usually considered the normal TR) and the one-leg (also called the implicit midpoint rule) versions are, respectively,

$$\begin{aligned}\frac{\overrightarrow{y}_{n+1} - \overrightarrow{y}_n}{\triangle t} &= \frac{1}{2}\overrightarrow{f}(t_{n+1},\overrightarrow{y}_{n+1}) + \frac{1}{2}\overrightarrow{f}(t_n,\overrightarrow{y}_n) \text{ or}\\ \frac{\overrightarrow{y}_{n+1} - \overrightarrow{y}_n}{\triangle t} &= \overrightarrow{f}(t_{n+\frac{1}{2}},\frac{\overrightarrow{y}_{n+1} + \overrightarrow{y}_n}{2}).\end{aligned}$$

To explain the difference, consider the nonautonomous test problem

$$y' = \lambda(t)y, \text{ where } \lambda(t) < 0.$$

Here there is a difference between the 1 leg and 2 leg methods. They are respectively

$$\begin{array}{cc} \text{1 leg\_ TR:} & \text{2 leg\_ TR:} \\ y_{n+1} = \left(\frac{2+\triangle t\lambda(t_{n+1/2})}{2-\triangle t\lambda(t_{n+1/2})}\right)y_n & y_{n+1} = \left(\frac{2+\triangle t\lambda(t_{n+1})}{2-\triangle t\lambda(t_{n+1})}\right)y_n \end{array}.$$

Clearly, the 1 leg method has the property, not shared by the 2 leg version that

$$|y_{n+1}| \leq |y_n| \text{ for all } n \text{ since } \left|\frac{2+\triangle t\lambda(t_{n+1/2})}{2-\triangle t\lambda(t_{n+1/2})}\right| < 1.$$

This impacts stability. Suppose $f = f(y)$ satisfies a monotonicity condition.

[5] Dr Philip Gresho is one of the founders of the International Journal for Numerical Methods in Fluids. He made many major technical contributions to the field. His book 'Incompressible Flow and the Finite Element Method' is a source for "what works" in Computational Fluid Dynamics. It is filled with important insights into numerical methods.

CONDITION 1. *Suppose there is an $\alpha > 0$ such that for all x, y*

(MONOTONICITY) $$(f(x) - f(y)) \cdot (x - y) \leq -\alpha |x - y|^2.$$

In this case the following is easy to show (and its proof is an exercise).

THEOREM 15. *Suppose $f(y)$ satisfies (*MONOTONICITY*). Then any two solutions $x(t), y(t)$ from different initial conditions must squeeze together exponentially fast:*

$$|x(t) - y(t)|^2 \leq e^{-\alpha t} |x(0) - y(0)|^2$$

The natural question is whether discretizations preserve this property. It is very easy to show that the one leg TR has this property for averages of the solution over 2 time levels..

THEOREM 16 (Squeezing of averages). *Suppose $f(y)$ satisfies (*MONOTONICITY*). Then averages of any two solutions x_n, y_n from different initial conditions of the one leg TR must squeeze together: let $w_n = x_n - y_n$ then*

$$\frac{w_{n+1} + w_n}{2} \to 0 \text{ as } n \to \infty.$$

PROOF. Begin with

$$\begin{aligned} \frac{x_{n+1} - x_n}{\triangle t} &= f(\frac{x_{n+1} + x_n}{2}) \\ \frac{y_{n+1} - y_n}{\triangle t} &= f(\frac{y_{n+1} + y_n}{2}). \end{aligned}$$

Set $w_n = x_n - y_n$. Subtract and take the dot product with $(w_{n+1} + w_n)/2$. This gives

$$\frac{w_{n+1} - w_n}{\triangle t} \cdot \frac{w_{n+1} + w_n}{2} =$$
$$= \left(f(\frac{x_{n+1} + x_n}{2}) - f(\frac{y_{n+1} + y_n}{2}) \right) \cdot \left(\frac{x_{n+1} + x_n}{2} - \frac{y_{n+1} + y_n}{2} \right).$$

The LHS is

$$LHS = \frac{w_{n+1} - w_n}{\triangle t} \cdot \frac{w_{n+1} + w_n}{2} = \frac{1}{2\triangle t} \left(|w_{n+1}|^2 - |w_n|^2 \right).$$

The RHS fits the monotonicity assumption perfectly. It satisfies

$$\begin{aligned} RHS &= \left(f(\frac{x_{n+1} + x_n}{2}) - f(\frac{y_{n+1} + y_n}{2}) \right) \cdot \left(\frac{x_{n+1} + x_n}{2} - \frac{y_{n+1} + y_n}{2} \right) \\ &\leq -\alpha \left| \frac{x_{n+1} + x_n}{2} - \frac{y_{n+1} + y_n}{2} \right|^2 = -\frac{\alpha}{4} \left| w_{n+1} + w_n \right|^2. \end{aligned}$$

Putting these together we have

$$\frac{1}{2\triangle t} \left(|w_{n+1}|^2 - |w_n|^2 \right) + \frac{\alpha}{4} \left| w_{n+1} + w_n \right|^2 \leq 0.$$

Summing $n = 1, ..., N-1$ gives

$$\frac{1}{2\triangle t} |w_N|^2 + \frac{\alpha}{4} \sum_{n=0}^{N-1} \left| w_{n+1} + w_n \right|^2 \leq \frac{1}{2\triangle t} |w_0|^2.$$

Since the RHS is independent of N we conclude that, uniformly in N,

$$\begin{aligned}\sup_N |w_N|^2 &< \infty \\ & \textit{and} \\ \sum_{n=0}^{\infty} |w_{n+1}+w_n|^2 &< \infty.\end{aligned}$$

Since the infinite series converges it must follow that the nth term → 0

$$|w_{n+1}+w_n| \to 0, \text{ as } n \to \infty$$

□

Averages squeeze together but what about trajectories?

Now, the previous proof simply does not work with the two leg version of the TR. Thus it was believed that the one leg version had preferable stability properties, until the following was noticed.

THEOREM 17 (Equivalence of 1 leg and 2 leg methods). *If (t_n, y_n) satisfies the 1 leg TR. Then*

$$\begin{aligned}\widehat{t_n} &= \frac{t_{n+1}+t_n}{2} \\ \widehat{y_n} &= \frac{y_{n+1}+y_n}{2}\end{aligned}$$

satisfies the 2 leg TR. Conversely, if $(\widehat{t_n}, \widehat{y_n})$ satisfies the 2 leg TR then (t_n, y_n) satisfies the 1 leg TR where

$$\begin{aligned}y_n &= \widehat{y_n} - (h/2)f(\widehat{t_n}, \widehat{y_n}), \\ t_n &= \widehat{t_n} - h/2.\end{aligned}$$

To my knowledge, this relationship has not [yet] been used to post process the TR.

EXERCISE 54. *Prove the equivalence theorem about the 1 and 2 leg TR.*

EXERCISE 55. *If $f(x) = Ax$ where A is negative definite show that $w_n = x_n - y_n \to 0$ as $n \to \infty$.*

1.5. Error Estimation. Error be combined efficiently with methods for generating initial guesses for solving the nonlinear system. This is the method[6] strongly advocated by Phil Gresho who notes that while there is no proof, oscillations do not arise when time adaptivity is added to the TR. He proposes the following as

[6] This is a repetition of an earlier note.

an efficient implementation of an adaptive estimator.

$$\text{Predict\_with\_AB2}$$

$$y_{n+1}^P = y_n + \frac{\triangle t_n}{2}\left[(2+\frac{\triangle t_n}{\triangle t_{n-1}})f(t_n, y_n) - \frac{\triangle t_n}{\triangle t_{n-1}} f(t_{n-1}, y_{n-1})\right]$$

$$\text{Reuse\_function\_Evaluations\_for\_TR:}$$

$$\frac{y_{n+1}-y_n}{\triangle t} = \frac{1}{2}f(t_{n+1}, y_{n+1}) + \frac{1}{2}f(t_n, y_n)$$

$$EST = \frac{|y_{n+1}^P - y_{n+1}|}{3(1+\frac{\triangle t_n}{\triangle t_{n-1}})},$$

$$\text{Invert\_TR to get } f_{n+1}$$

$$f(t_{n+1}, y_{n+1}) = \frac{2}{\triangle t_n}(y_{n+1}-y_n) - f(t_n, y_n)$$

and reuse for next predictor step.

2. Solving the nonlinear system for stiff problems

For a nonlinear system,

$$\frac{d}{dt}\overrightarrow{y} = \overrightarrow{f}(t, \overrightarrow{y})$$

the trapezoid rule can be interpreted in two ways:

$$\frac{\overrightarrow{y}_{n+1} - \overrightarrow{y}_n}{\triangle t} = \frac{1}{2}\overrightarrow{f}(t_{n+1}, \overrightarrow{y}_{n+1}) + \frac{1}{2}\overrightarrow{f}(t_n, \overrightarrow{y}_n) \text{ or}$$

$$\frac{\overrightarrow{y}_{n+1} - \overrightarrow{y}_n}{\triangle t} = \overrightarrow{f}(t_{n+\frac{1}{2}}, \frac{\overrightarrow{y}_{n+1} + \overrightarrow{y}_n}{2}).$$

These are sometimes called the "two leg" and the "one leg" trapezoid rule and the one leg version is also called the implicit midpoint rule. (They are the same for linear systems.) For either selection, one must solve a nonlinear system for $\overrightarrow{y}_{n+1}$ and the issues are much the same for one as for the other. For the two legged version, at every step we must perform:

$$\begin{aligned} \text{given } \overrightarrow{y}_n &: \\ \text{assemble the vector :} \quad & \overrightarrow{b} = \overrightarrow{y}_n + \frac{\triangle t}{2}\overrightarrow{f}(t, \overrightarrow{y}_n) \\ \text{solve the nonlinear system} &: \quad \overrightarrow{y}_{n+1} - \frac{\triangle t}{2}\overrightarrow{f}(t, \overrightarrow{y}_{n+1}) = \overrightarrow{b} \end{aligned}$$

Essentially the same challenge occurs for any implicit method including the one leg trapezoid rule and BDF methods. So that we can be specific, we shall thus study the above system. Thus the problem is:

$$\text{solve for } \overrightarrow{y} :$$

$$F(\overrightarrow{y}) := \overrightarrow{y} - \frac{\triangle t}{2}\overrightarrow{f}(\overrightarrow{y}) = \overrightarrow{b}.$$

The general method for solving nonlinear systems $F(y) = b$ consists of three ingredients:

- Select initial guess to the solution y^{old};
- Rewrite $F(y) = b$ as $y = G(y)$ and iterate $y^{new} = G(y^{old})$;

- Stop when appropriate stopping criteria are satisfied.

We will examine these three main steps for the above nonlinear system.

The initial guess. Finding good initial guesses is usually the hardest part of solving nonlinear systems. When, as here, they arise in time dependent problems, it becomes the easiest part of the problem. Good options for initial guesses $y^{old} =$ approximation of y_{n+1} include:

$$\begin{aligned} \text{the last time step:} \quad & y^{old} := y_n \\ \text{extrapolation from previous values:} \quad & y^{old} := 2y_n - y_{n-1} \\ \text{one step of an explicit method} \quad : \quad & y^{old} := y_n + \triangle t f(t_n, y_n) \end{aligned}$$

Stopping Criteria. Every iterative method must contain a user selected, preset tolerance, a user selected, preset maximum number of iterations and three tests:

STOP and signal non-convergence if **too many iterations**,

STOP if both below hold:

Small Residual: test if $|F(y^{new}) - b| < Tolerance$

Small Update: test if $|y^{new} - y^{old}| < Tolerance$

The choice of iterative method. If the time step $\triangle t$ is small it is very natural to try to solve the nonlinear system by a simple iteration of the form:

$$\text{Given: } y^{old},$$

$$y^{new} - \frac{\triangle t}{2} f(y^{old}) = b$$

$$\text{Replace: } y^{old} \Leftarrow y^{new}.$$

Implementing this is particularly easy since the iteration uses only the function $f(y)$ already programmed and noting further. The convergence of this iteration is governed by the contraction mapping theorem. There are various versions. The relevant one for our purposes here is as follows.

Theorem 18 (Contraction Mapping Theorem). *Suppose $G(y)$ is continuously differentiable and has a fixed point*

$$y^* = G(y^*).$$

Consider the fixed point iteration

$$Guess : y^{old}$$

$$\textit{Until Convergence:}$$

$$y^{new} = G(y^{old})$$

$$y^{new} \Leftarrow y^{old}.$$

This iteration converges locally (i.e., for good enough initial guess) provided

$$|\frac{d}{dy} G(y^*)| < 1.$$

Applied to $y^{new} - \frac{\triangle t}{2} f(y^{old}) = b$, we take

$$G(y) := b + \frac{\triangle t}{2} f(y).$$

The contraction mapping theorem assures convergence provided the timestep $\triangle t$ is small enough to satisfy

$$\frac{\triangle t}{2}|f_y(t_{n+1}, y_{n+1})| < 1.$$

Since getting good initial guesses is no problem, this means the nonlinear system can be solved easily provided the system is not stiff.

The problem with simple iteration for stiff problems is that it requires the above timestep restriction for convergence

$$\frac{\triangle t}{2}|f_y(t_{n+1}, y_{n+1})| < 1.$$

When $f(y) = \lambda y$, this is

$$|\triangle t\lambda| < 2,$$

exactly the time step restriction required for stability of forward Euler! Thus,

Simple iteration is useful if and only if the system is not stiff.

Fortunately, **Newton's**[7] **method works well** for solving the nonlinear system arising in stiff systems. Newton's method for $F(y) = b$ reads

$$\text{Until Satisfied : Given } y^{old}, r = b - F(y^{old})$$

$$\text{Assemble the } N \times N \text{ matrix : } A = \frac{\partial F_i}{\partial y_j}(y^{old}) \text{ and solve } Ax = r$$

$$\text{Update : } y^{new} = y^{old} + x \text{ and } y^{old} \Leftarrow y^{new}$$

$$\text{Compute : } r = b - F(y^{old}) \text{ and test for convergence.}$$

With a good initial guess, the second step is normally where Newton's method would break down. For stiff systems, $F(y) = y - \frac{\triangle t}{2} f(y)$ so the eigenvalues of the matrix $\frac{\partial f_i}{\partial y_j}$ are large and negative. Thus the eigenvalues of $\frac{\partial F_i}{\partial y_j}$ are positive and bounded away from zero:

$$\begin{aligned}
\lambda\left(\frac{\partial F_i}{\partial y_j}\right) &= \lambda\left(I - \frac{\triangle t}{2}\frac{\partial f_i}{\partial y_j}\right) \\
&= 1 - \frac{\triangle t}{2}\lambda\left(\frac{\partial f_i}{\partial y_j}\right) \\
&= 1 - \frac{\triangle t}{2} \times \{\text{something large and negative}\} \geq 1.
\end{aligned}$$

As a consequence, no breakdown is possible for Newton's method.

3. Energy proofs of stability

There are problems for which proof of stability by direct assault (by energy methods) is essential. In particular, energy proofs, beyond scalar , autonomous problems, yield stability for nonlinear, non-autonomous systems. they also give estimates of dependence of stability on the size of the system (thus are relevant for PDEs) and other parameters. Further, IMEX methods are the current standard for multi-physics systems and the scalar test problem gives necessary but not sufficient conditions for IMEX methods. Energy proofs, when known, are the gold standard for stability and, when compatible for the component methods, can yield stability of IMEX methods.

[7] "The" Isaac Newton of course!

On the other hard, energy proofs are not systematic[8] and are generally more intricate than the beautiful, complete and systematic theory of A-stability by root conditions.

There are a few methods where the "tricks" needed for energy stability analysis are known. In this section we review 3: the backward Euler method, the Trapezoid rule and BDF2. The goal of an energy proof of stability is to develop an exact energy equality where the equivalent of the kinetic energy is revealed and where the numerical dissipation of the method is exactly quantified. To give this much detail, often the analysis is restricted to a linear system of evolution equations such as

$$\begin{aligned} \overrightarrow{y}' + A\overrightarrow{y} &= f(t), t > 0, \\ \overrightarrow{y}(0) &= \overrightarrow{y}_0, \\ & \quad where \\ A &= \text{ a square SPD matrix.} \end{aligned}$$

In the analysis the A-norm is important for a precise result.

Definition 18 (A-norm and inner product). *For an SPD matrix A the associated A−norm and inner product are*

$$\langle x, y\rangle_A := x^T A y \text{ and } |x|_A := \sqrt{\langle x, x\rangle_A}.$$

Taking the dot product of $\overrightarrow{y}' + A\overrightarrow{y} = f(t)$ with $\overrightarrow{y}$ and integrating in time gives, for any T >0,

$$\frac{1}{2}|y(T)|^2 + \int_0^T |y(T)|_A^2 dt = \frac{1}{2}|y(0)|^2 + \int_0^T f(t)\cdot y(t)dt.$$

This is the basic energy equality that a discrete energy analysis seeks to mimic. The terms have the interpretation below.

Interpretation		Corresponding term
energy at time T	:	$\frac{1}{2}\lvert y(T)\rvert^2$
energy dissipated at time t	:	$\lvert y(t)\rvert_A^2$
total dissipated over $0 \le t \le T$	:	$\int_0^T \lvert y(T)\rvert_A^2 dt$
Initial energy	:	$\frac{1}{2}\lvert y(0)\rvert^2$
Energy input over $0 \le t \le T$	:	$\int_0^T f(t)\cdot y(t)dt$

The goal is to produce a discrete equivalent for the solution of the methods difference approximation.

3.1. Implicit Euler. Consider the implicit Euler approximation to

$$\begin{aligned} \overrightarrow{y}' + A\overrightarrow{y} &= f(t), t > 0, \\ \overrightarrow{y}(0) &= \overrightarrow{y}_0, \\ & \quad \text{where} \\ A &= \text{ a square SPD matrix.} \end{aligned}$$

[8] There is a systematic approach to energy estimates of stability known as G-stability theory. It is intricate and quite technical.

It is given by

$$\text{(Implicit Euler)} \qquad \frac{\overrightarrow{y}_{n+1}-\overrightarrow{y}_n}{\triangle t}+A\overrightarrow{y}_{n+1}=f(t_{n+1}).$$

The proof will need a vector identity known as the polarization identity.

Lemma 2 (Polarization Identity). *For any vectors x, y we have*

$$x\cdot y=\frac{1}{2}|x|^2+\frac{1}{2}|y|^2-\frac{1}{2}|x-y|^2$$

Proof. This is an identity. To prove it first expand each side and cancel terms until it reduces to 0=0. The proof is then simply writing the sequence of steps in the opposite order starting with "0=0". The realization of this proof is left as an exercise. □

We then have energy stability.

Theorem 19 (Energy stability of the implicit method). *We have, for any N>0*

$$\frac{1}{2}|y_N|^2+\triangle t\sum_{n=0}^{N-1}\left[|y_{n+1}|_A^2+\frac{\triangle t}{2}|\frac{y_{n+1}-y_n}{\triangle t}|^2\right]=\frac{1}{2}|y_0|^2+\triangle t\sum_{n=0}^{N-1}f(t_{n+1})\cdot y_{n+1}.$$

Proof of energy stability. We parallel the steps in deriving the energy equality

$$\frac{1}{2}|y(T)|^2+\int_0^T|y(T)|_A^2dt=\frac{1}{2}|y(0)|_A^2+\int_0^T f(t)\cdot y(t)dt.$$

for the IVP. Take the inner product with $\overrightarrow{y}_{n+1}$ and multiply through by$\triangle t$. This gives

$$\left(|y_{n+1}|^2-y_n\cdot y_{n+1}\right)+\triangle t\,|y_{n+1}|_A^2=\triangle t f(t_{n+1})\cdot y_{n+1}$$

At this point we need a vector identity known as the polarization identity. The polarization identity is now used on the term $y_n\cdot y_{n+1}$. This gives

$$\left(|y_{n+1}|^2-\left[\frac{1}{2}|y_n|^2+\frac{1}{2}|y_{n+1}|^2-\frac{1}{2}|y_{n+1}-y_n|^2\right]\right)+\triangle t\,|y_{n+1}|_A^2=\triangle t f(t_{n+1})\cdot y_{n+1},$$

or

$$\left(\frac{1}{2}|y_{n+1}|^2-\frac{1}{2}|y_n|^2\right)+\triangle t\,|y_{n+1}|_A^2+\frac{1}{2}|y_{n+1}-y_n|^2=\triangle t f(t_{n+1})\cdot y_{n+1}.$$

Note that

$$\frac{1}{2}|y_{n+1}-y_n|^2=\triangle t\frac{\triangle t}{2}|\frac{y_{n+1}-y_n}{\triangle t}|^2.$$

Summing this over $n=0,\cdots,N-1$ gives

$$\frac{1}{2}|y_N|^2+\triangle t\sum_{n=0}^{N-1}\left[|y_{n+1}|_A^2+\frac{\triangle t}{2}|\frac{y_{n+1}-y_n}{\triangle t}|^2\right]=\frac{1}{2}|y_0|^2+\triangle t\sum_{n=0}^{N-1}f(t_{n+1})\cdot y_{n+1}.$$

□

The discrete energy stability resembles term by term the continuous case:

$$\begin{array}{rcl} & Continuous & : \\ \frac{1}{2}|y(T)|^2 + \int_0^T |y(T)|_A^2 dt & = & \frac{1}{2}|y(0)|_A^2 + \int_0^T f(t)\cdot y(t)dt \\ & Discrete & : \\ \frac{1}{2}|y_N|^2 + \triangle t \sum_{n=0}^{N-1}\left[|y_{n+1}|_A^2 + \frac{\triangle t}{2}|\frac{y_{n+1}-y_n}{\triangle t}|^2\right] & = & \frac{1}{2}|y_0|^2 + \triangle t \sum_{n=0}^{N-1} f(t_{n+1})\cdot y_{n+1} \end{array}$$

This means we have the following interpretations of each term.

Interpretation		Corresponding term
energy at time t_N	:	$\frac{1}{2}\lvert y_N\rvert^2$
energy dissipated at time t_n	:	$\lvert y_{n+1}\rvert_A^2 + \frac{\triangle t}{2}\lvert\frac{y_{n+1}-y_n}{\triangle t}\rvert^2$
Energy dissipated over $0 \leq t \leq T$	:	$\triangle t \sum_{n=0}^{N-1}\left[\lvert y_{n+1}\rvert_A^2 + \frac{\triangle t}{2}\lvert\frac{y_{n+1}-y_n}{\triangle t}\rvert^2\right]$
Initial energy	:	$\frac{1}{2}\lvert y_0\rvert^2$
Energy input over $0 \leq t \leq T$	:	$\triangle t \sum_{n=0}^{N-1} f(t_{n+1})\cdot y_{n+1}$

The dissipation has two components:

Interpretation	*and*	*Term*
IVP dissipation	:	$\lvert y_{n+1}\rvert_A^2$
Extra numerical dissipation	:	$\frac{\triangle t}{2}\lvert\frac{y_{n+1}-y_n}{\triangle t}\rvert^2$

3.2. The Trapezoid Rule. Consider the implicit Euler approximation to

$$\begin{array}{rcl} \overrightarrow{y}' + A\overrightarrow{y} & = & f(t), t > 0, \\ \overrightarrow{y}(0) & = & \overrightarrow{y}_0, \\ & & \text{where} \\ A & = & \text{a square SPD matrix.} \end{array}$$

It is given by

(Implicit Euler) $$\frac{\overrightarrow{y}_{n+1} - \overrightarrow{y}_n}{\triangle t} + A\frac{\overrightarrow{y}_{n+1} + \overrightarrow{y}_n}{2} = f(t_{n+1/2}).$$

The proof will need use the vector identity.

Lemma 3. *We have*

$$(x-y)\cdot(x+y) = |x|^2 - |y|^2$$

Proof. Expand both sides. □

With that identity energy stability follows easily.

Theorem 20 (Energy stability of the TR). *We have, for any $N > 0$*

$$\frac{1}{2}|y_N|^2 + \triangle t \sum_{n=0}^{N-1}\left|\frac{y_{n+1}+y_n}{2}\right|_A^2 = \frac{1}{2}|y_0|^2 + + \triangle t \sum_{n=0}^{N-1} f(t_{n+1/2})\cdot\frac{y_{n+1}+y_n}{2}.$$

Proof of energy stability. We parallel the steps in deriving the energy equality

$$\frac{1}{2}|y(T)|^2 + \int_0^T |y(T)|_A^2 dt = \frac{1}{2}|y(0)|_A^2 + \int_0^T f(t)\cdot y(t)dt.$$

for the IVP. Taking the dot product of the TR with $\frac{\overrightarrow{y}_{n+1}+\overrightarrow{y}_n}{2}$ and using the above identity gives

$$\frac{1}{2\triangle t}\left(|y_{n+1}|^2-|y_n|^2\right)=\left|\frac{y_{n+1}+y_n}{2}\right|_A^2=f(t_{n+1/2})\cdot\frac{y_{n+1}+y_n}{2}$$

Summing $n=0,...,N-1$ gives

$$\frac{1}{2}|y_N|^2+\triangle t\sum_{n=0}^{N-1}\left|\frac{y_{n+1}+y_n}{2}\right|_A^2=\frac{1}{2}|y_0|^2++\triangle t\sum_{n=0}^{N-1}f(t_{n+1/2})\cdot\frac{y_{n+1}+y_n}{2}. \tag{3.1}$$

□

The discrete energy stability resembles term by term the continuous case:

$$\begin{array}{rcl} & \mathit{Continuous} & : \\ \frac{1}{2}|y(T)|^2+\int_0^T|y(T)|_A^2dt & = & \frac{1}{2}|y(0)|_A^2+\int_0^T f(t)\cdot y(t)dt \\ & \mathit{discrete} & : \\ \frac{1}{2}|y_N|^2+\triangle t\sum_{n=0}^{N-1}\left|\frac{y_{n+1}+y_n}{2}\right|_A^2 & = & \frac{1}{2}|y_0|^2++\triangle t\sum_{n=0}^{N-1}f(t_{n+1/2})\cdot\frac{y_{n+1}+y_n}{2}. \end{array}$$

This means we have the following interpretations of each term.

Interpretation		Corresponding term
energy at time t_N	:	$\frac{1}{2}\|y_N\|^2$
energy dissipated at time t_n	:	$\left\|\frac{y_{n+1}+y_n}{2}\right\|_A^2$
total dissipated over $0\leq t\leq T$	:	$\triangle t\sum_{n=0}^{N-1}\left\|\frac{y_{n+1}+y_n}{2}\right\|_A^2$
Initial energy	:	$\frac{1}{2}\|y_0\|^2$
Energy input over $0\leq t\leq T$	:	$\triangle t\sum_{n=0}^{N-1}f(t_{n+1/2})\cdot\frac{y_{n+1}+y_n}{2}$

The dissipation components for the TR are:

Interpretation	*and*	*Term*
IVP dissipation	:	$\left\|\frac{y_{n+1}+y_n}{2}\right\|_A^2$
Extra numerical dissipation	:	*none*

3.3. BDF2. Analogous but more complex manipulations are known for BDF2.

3.4. The discrete energy evolution of leapfrog. Analogous but more complex manipulations are known for leapfrog.

4. Implicit RK Methods

The first Runge Kutta methods were explicit and developed quite early (Runge 1895, Kutta 1901, Heun 1900). The subsequent work of Dahlquist revealed that implicit methods are necessary. It took substantially longer to develop good implicit RK methods. This section presents some good implicit RK methods.

First, RKs methods are one step methods. Thus they take the general form

$$y_{n+1}-y_n=\triangle t\Phi(\cdot)$$

where Φ takes the form:

$$\Phi = \sum_{i=1}^{s} b_i k_i.$$

The stages are of the form

$$k_i = \triangle t f(t_n + c_i \triangle t, y_n + \sum_{j=1}^{s} a_{ij} k_j) \text{ for } i = 1, \cdots, s.$$

Thus, the method is determined by specifying the parameters b_i, c_i, a_{ij}. These parameters are determined by the twin constraints of high consistency and desired stability. RK methods are thus codified by presenting these parameters are an array called the "*Butcher array*" or "*Butcher tableau*" due to the work of Butcher[9] in 1964:

$$\begin{bmatrix} \overrightarrow{c} & | & A \\ - & | & \\ & | & \overrightarrow{\bar{b}}^{T} \end{bmatrix} = \begin{bmatrix} c_i & | & A_{ij} \\ - & | & \\ & | & \bar{b}_j^{T} \end{bmatrix}.$$

EXAMPLE 27 (Butcher array for RK2). *RK2 is*

$$\begin{aligned} k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \triangle t, y_n + k_1) \\ y_{n+1} &= y_n + \frac{1}{2} k_1 + \frac{1}{2} k_2. \end{aligned}$$

Thus,

$$\begin{aligned} b_1 &= b_2 = \frac{1}{2}, \\ c_1 &= 0, c_2 = 1 \\ a_{11} &= 0, a_{12} = 0 \\ a_{21} &= 1, a_{22} = 0. \end{aligned}$$

This corresponds to the Butcher array

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} .$$

Indeed, rewrite RK2 as follows (and use subscripts to indicate where in the array each number goes:

$$\begin{aligned} k_1 &= \triangle t f(t_n + 0_{(c_1)} \triangle t, y_n + \big(0_{(A_{1,1})} k_1 + 0_{(A_{1,2})} k_2\big)) \\ k_2 &= \triangle t f(t_n + 1_{(c_2)} \triangle t, y_n + \big(1_{(A_{2,1})} k_1 + 0_{(A_{2,2})} k_2\big)) \\ y_{n+1} &= y_n + \left(\frac{1}{2}\right)_{(b_1)} k_1 + \left(\frac{1}{2}\right)_{(b_2)} k_2. \end{aligned}$$

[9] Adapted from Wikipedia:

John Charles Butcher ONZM (born 1933) is a New Zealand mathematician who is a leader in the development of numerical methods for the solution of ordinary differential equations. Butcher works Runge-Kutta and general linear methods. The Butcher group and the Butcher tableau are named after him. Butcher was awarded the Jones Medal from the Royal Society of New Zealand in 2010, for his "exceptional lifetime work on numerical methods for the solution of differential equations and leadership in the development of New Zealand mathematical sciences."

Written this way, it is clear that in step 1, if $A_{1,1} \neq 0$ *one must solve a nonlinear equation or system of equations for* k_1 *and similarly for solving for* k_2 *in step 2. If* $A_{1,2} \neq 0$ *then the nonlinear system is twice as large as the nonlinear equations for* k_1 *and* k_2 *are coupled.*

EXAMPLE 28 (Butcher array for the Ralston rule RK2 method). *The Ralston rule is:*

$$\begin{array}{rcl} given & : & y_n \qquad\qquad\qquad\qquad \textit{(Ralston Rule again)} \\ k_1 & = & \triangle t f(t_n, y_n) \\ k_2 & = & \triangle t f(t_n + \frac{2}{3}\triangle t, y_n + \frac{2}{3}k_1) \\ y_{n+1} & = & y_n + \frac{1}{4}k_1 + \frac{3}{4}k_2. \end{array}$$

This corresponds to the Butcher array

$$\begin{array}{r|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}.$$

We observe that a RK method is

- explicit if $A_{ij} = 0\ for\ j \geq i$;
- implicit if A has a nonzero entry on or above its diagonal;
- diagonally implicit if $A_{ij} = 0\ for\ j > i$ and $A_{ii} \neq 0$ for some i.

EXERCISE 56. *Write the Butcher array for RK4.*

The Calahan (1968) DIRK. The following is a diagonally implicit RK (DIRK)[10] method proposed in 1968 by Calahan. It is given by the Butcher array:

$$\begin{array}{r|cc} \alpha & \alpha & 0 \\ 1-\alpha & 1-2\alpha & \alpha \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

A second order DIRK Method

The following is known.

THEOREM 21. *The* ***Calahan method*** *is* A_0*-stable and second order accurate (local truncation error* $\mathcal{O}(\triangle t^3)$*) for all* α , $0 \leq \alpha \leq 1$, *and third order accurate (local truncation error* $\mathcal{O}(\triangle t^4)$*) if*

$$\alpha = \frac{3+\sqrt{3}}{6}.$$

For that value of α*, it is A−stable (but not L−stable).*

Implementation of the method is as follows.

$$\begin{array}{rcl} \text{Given } \alpha, y_n & : & \\ \text{Solve the nonlinear equation for } k_1 & : & k_1 = \triangle t f(t_n + \alpha\triangle t, y_n + \alpha k_1) \\ \text{Solve the nonlinear equation for } k_2 & : & k_2 = \triangle t f(t_n + (1-\alpha)\triangle t, y_n + (1-2\alpha)k_1 + \alpha k_2) \\ y_{n+1} & = & y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2. \end{array}$$

[10] K. Dekker and J.G. Verwer, Stability of RK methods for stiff nonlinear equations, North Holland, Amsterdam, 1984.

A fourth order implicit RK method. The 2 stage RK method given by the following Butcher array is known to be $A-$stable and fourth order accurate.

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

An $A-$stable, 2 stage, implicit RK Method

4.1. The discoverers. D.A. Calahan was a professor at the University of Michigan. His method originated in a 1968 paper of his.

Adapted from Wikipedia:

John Charles Butcher ONZM (born 1933) is a New Zealand mathematician who is a leader in the development of numerical methods for the solution of ordinary differential equations. Butcher works Runge-Kutta and general linear methods. The Butcher group and the Butcher tableau are named after him. Butcher was awarded the Jones Medal from the Royal Society of New Zealand in 2010, for his "*exceptional lifetime work on numerical methods for the solution of differential equations and leadership in the development of New Zealand mathematical sciences.*"

5. Stability of RK Methods

Stability of RK methods is analyzed in the usual manner. We apply the RK method to the test problem

$$y' = \lambda y, y(0) = 1$$

Since these are all one step methods, this yields

$$y_{n+1} = a(\triangle t\lambda)y_n.$$

We then ask under what conditions does $y_n \to 0$ as $n \to \infty$, i.e., when is $|a(\triangle t\lambda)| < 1$.

Example 29 (RK2). *RK2 is*

$$\begin{aligned} k_1 &= \triangle t f(t_n, y_n) \\ k_2 &= \triangle t f(t_n + \triangle t, y_n + k_1) \\ y_{n+1} &= y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2. \end{aligned}$$

which becomes for $f(t, y) = \lambda y$

$$k_1 = \triangle t\lambda y_n \textit{ then } k_2 = \triangle t\lambda(y_n + k_1)$$

$$y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2.$$

Eliminating intermediate steps gives

$$y_{n+1} = y_n + \frac{1}{2}(\triangle t\lambda y_n) + \frac{1}{2}(\triangle t\lambda(y_n + \triangle t\lambda y_n)).$$

Collecting terms yields $y_{n+1} = a(\triangle t\lambda)y_n$ where

$$a(\triangle t\lambda) = 1 + \triangle t\lambda + \frac{1}{2}(\triangle t\lambda)^2.$$

Like RK2, if the RK method is explicit, $a(\triangle t\lambda)$ will be a polynomial in $\triangle t\lambda$. Since every polynomial $a(z) \to \infty$ as $|z| \to \infty$, we can immediately conclude one special case of the Dahlquist theory for RK methods.

Proposition 6. *Explicit RK methods must have bounded stability regions and thus cannot be A-stable.*

If the RK method is implicit, $a(\triangle t\lambda)$ will be a rational function $R(\triangle t\lambda)$ (a quotient of two polynomials) in $\triangle t\lambda$. In terms of the methods Butcher array, letting $z = \triangle t\lambda$, we obtain

$$R(z) = 1 + z\,\overrightarrow{b}^T[I - zA]^{-1}\overrightarrow{\mathbf{1}} \tag{5.1}$$
$$\text{where } \overrightarrow{\mathbf{1}} = (1, 1, \cdots, 1)^T.$$

With $R(z)$ known, stability properties of the RK method can then be read off.

Theorem 22 (Stability of RK methods). *The RK method with $R(z)$ given above is*

$A-$stable if $|R(z)| < 1$ for all z with $\mathrm{Re}(z) < 0$;

A_0-stable if $|R(z)| < 1$ for all z with $Im(z) = 0, \mathrm{Re}(z) < 0$;

$A_\alpha-$stable if $|R(z)| < 1$ for all z in the infinite wedge: $\mathrm{Re}(z) < 0, \pi - \alpha < \arg(z) < \pi + \alpha$;

$L-$stable if it is $A-$stable and $|R(z)| \to 0$ when z is real and $z \to -\infty$.

Exercise 57. *For RK4 find $a(\triangle t\lambda)$. If is λ real (and negative), plot it and verify RK4 is stable for the interval predicted by its stability region.*

Exercise 58. *Consider an implicit RK method with*

$$R(z) = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2}.$$

a. Find the RK method (hint: (5.1)). b. Given this $R(z)$, show that $R(z)$ factors in the form

$$R(z) = \frac{(z+p)(z+q)}{(z-p)(z-q)}$$

c. use the factorization to analyze its stability.

CHAPTER 9

IMEX Methods

This chapter collects a list of some particular combinations of schemes (known as an IMEX = Implicit-Explicit method) that have been useful in some applications. The analysis of these schemes for stability and convergence is largely an open question so use them at your own risk.

1. CNLF

CNLF is a method combining the Trapezoid rule[1] with doubled timestep with Leapfrog. This combination is commonly used in some applications when the system takes the form

$$\overrightarrow{y}' = \overrightarrow{f}(t, \overrightarrow{y}(t)) + \Lambda \overrightarrow{y}(t),$$

where Λ is skew symmetric, i.e., $\Lambda^T = -\Lambda$.

The combination CNLF is then

$$\text{(CNLF)} \qquad \frac{y_{n+1} - y_{n-1}}{2\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_{n-1}, y_{n-1}) + \Lambda y_n.$$

CNLF is usually combined with time filters. Robert and Asselin proposed the first filter with a tunable parameter ν

$$y_n = y_n^{temp} + \frac{\nu}{2}\left(y_{n+1}^{temp} - 2y_n^{temp} + y_{n-1}\right), 0 < \nu < 1.$$

This filter does reduce the oscillation that sometimes occurs but it also reduces the accuracy to first order unless $\nu = \mathcal{O}(\triangle t)$. Subsequent developments include an important modification by Williams that restores accuracy.

To quantify the effect of the filter step, define a discrete curvature in time by

$$\begin{aligned} \kappa_n^{old} &= y_{n+1}^{temp} - 2y_n^{temp} + y_{n-1}, \\ \kappa_n^{new} &= y_{n+1}^{temp} - 2y_n^{temp} + y_{n-1} \end{aligned}$$

We then have the following on reduction in the TR oscillation.

Theorem 23. *Let $0 < \nu < 1$. The method*

$$\frac{y_{n+1}^{temp} - y_{n-1}}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}^{temp}) + + \frac{1}{2} f(t_{n-1}, y_{n-1}) + \Lambda y_n^{temp}$$

$$y_n = y_n^{temp} + \frac{\nu}{2}\left(y_{n+1}^{temp} - 2y_n^{temp} + y_{n-1}\right).$$

reduces the discrete curvature. It satisfies the curvature evolution equation

$$\kappa_n^{new} = (1 - \nu)\kappa_n^{old}$$

[1] It is called CNLF and not TRLF because in the applications where it is commonly used the trapezoid rule is called the CN = Crank-Nicolson method.

2. CN-AB2

AB2 is commonly used with the trapezoid rule. This combination (another IMEX = Implicit-Explicit method) is used in some applications when the system takes the form

$$y' = f(t, y(t)) + g(t, y(t)).$$

The combination CN-AB2[2] is then

$$\frac{y_{n+1} - y_n}{\triangle t} = \frac{1}{2} f(t_{n+1}, y_{n+1}) + \frac{1}{2} f(t_n, y_n) + \frac{3}{2} g(t_n, y_n) - \frac{1}{2} g(t_{n-1}, y_{n-1}).$$

3. Semi-Implicit Predictor Corrector IMEX schemes

We consider next some linearly implicit schemes for the system

$$y' = f(t, y(t)).$$

Here y is a vector. We split $f(t, y)$ into a linear part and a nonlinear part. The linear part is generally described as "fast" and the nonlinear part as "slow".

$$y' = Ly(t) + N(y(t)).$$

3.1. CNLF again. The combination CNLF is in this case

$$\frac{y_{n+1} - y_{n-1}}{2\triangle t} = \frac{1}{2} L y_{n+1} + \frac{1}{2} L y_{n-1} + N(y_n). \tag{CNLFagain}$$

Unfortunately, CNLF can be unstable so special time filters are always used with CNLF when the LF is applied to a nonlinear term.

3.2. The LFT predictor corrector scheme. Kurihara in 1965 proposed

$$\begin{aligned} \frac{y^P_{n+1} - y_{n-1}}{2\triangle t} &= f(t_n, y_n) \qquad \text{(LFT)} \\ \frac{y_{n+1} - y_n}{\triangle t} &= \frac{1}{2} f(t_{n+1}, y^P_{n+1}) + \frac{1}{2} f(t_n, y_n). \end{aligned}$$

3.3. The ABT scheme. Kar proposed in 2012:

$$\begin{aligned} \frac{y^P_{n+1} - y_n}{\triangle t} &= \frac{3}{2} f(t_n, y_n) - \frac{1}{2} f(t_{n-1}, y_{n-1}) \qquad \text{(ABT)} \\ \frac{y_{n+1} - y_n}{\triangle t} &= \frac{1}{2} f(t_{n+1}, y^P_{n+1}) + \frac{1}{2} f(t_n, y_n). \end{aligned}$$

3.4. The ABM scheme. Durran proposed in 1999 the Adams-Bashforth-Moulton scheme:

$$\begin{aligned} \frac{y^P_{n+1} - y_n}{\triangle t} &= \frac{3}{2} f(t_n, y_n) - \frac{1}{2} f(t_{n-1}, y_{n-1}) \qquad \text{(ABM)} \\ \frac{y_{n+1} - y_n}{\triangle t} &= \frac{5}{12} f(t_{n+1}, y^P_{n+1}) + \frac{8}{12} f(t_n, y_n) - \frac{1}{12} f(t_{n-1}, y_{n-1}). \end{aligned}$$

[2] It is called CN-AB2 and not TR-AB2 because in the applications where it is commonly used the trapezoid rule is called the CN = Crank-Nicolson method.

3.5. The T-ABT scheme.

$$\begin{aligned}\frac{y^P_{n+1}-y_n}{\triangle t} &= \frac{1}{2}Ly^P_{n+1}+\frac{1}{2}Ly_n+\frac{3}{2}N(y_n)-\frac{1}{2}N(y_{n-1}) \qquad \text{(T-ABT)}\\ \frac{y_{n+1}-y_n}{\triangle t} &= \frac{1}{2}Ly_{n+1}+\frac{1}{2}Ly_n+\frac{1}{2}N(y^P_{n+1})+\frac{1}{2}N(y_n)\end{aligned}$$

3.6. The AM2-ABM scheme.

$$\begin{aligned}\frac{y^P_{n+1}-y_n}{\triangle t} &= \frac{3}{4}Ly^P_{n+1}+\frac{1}{4}Ly_{n-1}+\frac{3}{2}N(y_n)-\frac{1}{2}N(y_{n-1}) \qquad \text{(AM2-ABM)}\\ \frac{y_{n+1}-y_n}{\triangle t} &= \frac{3}{4}Ly_{n+1}+\frac{1}{4}Ly_{n-1}+\frac{5}{12}N(y^P_{n+1})+\frac{8}{12}N(y_n)-\frac{1}{12}N(y_{n-1})\end{aligned}$$

EXERCISE 59. *Choose a test problem and compare the performances of the above IMEX schemes. The goal is to differentiate among the methods and draw a conclusion. If they perform more or less the same, try a different test problem. Use an adaptive RKF45 simulation as a "truth" solution.*

CHAPTER 10

Modularity: Splitting Methods

insert

CHAPTER 11

Modelling Errors

> "In the future, proponents of numerical fluid dynamics should explain the limitations (as well as statistical uncertainties)..." Garrett Birkhoff, p. 29 in: Numerical Fluid Dynamics, SIAM Review, 25(1983), 1-34.

Problems associated with missing initial data or unrepresented processes occur widely in atmospheric science and the methods used to handle the issues have been highly developed there. We will only look at the first step in their development: "**nudging**"[1] or "**Newtonian damping**". There are two cases: forward nudging and backward nudging.

1. Forward nudging for errors in the model.

> "There are two kinds of prognosticators: those who know nothing and those who don't know they know nothing." -an old proverb about economic forecasting
>
> As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality. — Albert Einstein

Treating these errors requires extra solution measurements. Measurements or observations are averages. Thus, they containing necessarily less information than $y(t)$. (Otherwise, we would just use the observation as a new initial condition.) Thus, we have a matrix C which is not of full rank and observations $y_{data}(t)$ and want to minimize $y_{data}(t) - Cy(t)$. Thus the problem becomes:

$$\begin{array}{ll} \text{minimize :} & |y_{data}(t) - Cy(t)|^2 \\ \text{subject to:} & y' = f(t, y), t > 0, \ \& \ y(0) = y_0. \end{array}$$

The idea of nudging is to penalize the deviation of the solution from its observed averages. Thus, it chooses a small penalty parameter $\chi > 0$ and replaces $y' = f(t, y)$ by the IVP

$$y' = f(t, y) + \chi^{-1} C^T \left(y_{data}(t) - Cy(t) \right), \text{ for } t > 0,$$
$$y(0) = y_0.$$

[1] For recent work see:

1 D.G. Luenberger, IEEE. T. A. Control, 11(1966) and 16(1971).

2 F.E. Thau, Int. J. Control 17(1973), 471–479.

3 H. Nijmeijer, PhysicaD, 154(2001), 219–228.

4 D. Bloemker, K.J.H. Law, A.M. Stuart and K. Zygalalkis, Nonlinearity 26(2013), 2193–2219.

5 K.J.H. Law, A. Shukla and A.M. Stuart, Discrete and Continuous Dynamical Systems A, 34(2014), 1061–1078.

2. Backward nudging for unknown initial data.

It is also quite common for a complete initial condition to be unknown. This can be treated by starting the IVP at a (later) time when some observed value $y(T)$ is known. The nudging term is added and the following IVP is solved *backward* in time down to $t = \mathit{0}$

$$y' = f(t, y) - \chi^{-1} C^T \left(y_{data}(t) - Cy(t) \right), \text{ for } t < T,$$
$$y(T) \text{ given.}$$

Finally, these two steps can be repeated iteratively giving forward and back nudging. These two cases are optimization problems with IVP sitting at their center. Optimization problems are solved by iteration, (in simple form, given a guess of the unknown data, change it a bit and see if the quantity minimized goes up or down and use that information to improve the guess of the unknown data) which requires solving the IVP many times. Forward and backward nudging is only one example is such an iteration.

Bibliography

[ALP04] M. Anitescu, W. Layton and F. Pahlevani, *Implicit for local effects, explicit for nonlocal is unconditionally stable*, ETNA 18 (2004), 174-187.

[ARW95] U Asher, S Ruuth and B. Wetton, *Implicit-Explicit methods for time dependent partial differential equations,* SINUM 32(1995) 797-823.

[AP98] U. Asher and L. Petzold, *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM, Philadelphia, 1998.

[1] A. Aoyagi, *Nonlinear leapfrog instability for Fornberg's pattern*, Journal of Computational Physics, 120(1995) 316–322

[BPS89] Bogacki, Przemyslaw; Shampine, Lawrence F.. (1989), "*A 3(2) pair of Runge–Kutta formulas*", Applied Mathematics Letters 2 (4): 321–325, doi:10.1016/0893-9659(89)90079-7, ISSN 0893-9659

[2] Calvo and Sanz-Serna 1993.

[CN47] J Crank and P. Nicolson. *A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type*, Proc. Cambridge Philos. Soc. 43 (1947). 50-67.

[C80] M Crouzeix, *Une méthode multipas implicite-explicite pour l'approximation des équationes d'évolution paraboliques*, Numer. Math. 35(1980) 257-276.

[D78] G. Dahlquist, *Positive functions and some applications to stability questions for numerical methods*, pp. 1-29 in: Recent Advances in Numerical Analysis, (eds.:C. deBoor and G. Golub) Academic Press, 1978.

[DP80] Dormand, J. R.; Prince, P. J. (1980), "*A family of embedded Runge-Kutta formulae*", Journal of Computational and Applied Mathematics 6 (1): 19–26, doi:10.1016/0771-050X(80)90013-3

[DD] J Douglas and T Dupont, INSERT

[DV84] K. Dekker and J.G. Verwer, *Stability of RK methods for stiff nonlinear equations*, North Holland, Amsterdam, 1984.

[3] Dong Wang, *Variable step-size implicit-explicit linear multistep methods for time dependent partial differential equations*, MS thesis, Simon Fraser University, 2005.

[4] D.R. Durran, *Numerical methods for fluid dynamics with applications to geophysics*, Second Edition, Springer, Berlin, 2010.

[F69] Erwin Fehlberg (1969). *Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems.* NASA Technical Report 315.

[F70] Erwin Fehlberg (1970). "*Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme,*" Computing (Arch. Elektron. Rechnen), vol. 6, pp. 61–71. doi:10.1007/BF02241732

[5] J. Frank, W. Hundsdorfer and J Verwer, *Stability of Implicit-Explicit linear multistep methods*, CWI Report 1996.

[6] B. Fornberg, *On the instability of the Leap-Frog and Crank-Nicolson approximation of a nonlinear partial differential equation*, Math. Comp. 27(1973), 45-57.

[HNW08] Hairer, Ernst; Nørsett, Syvert Paul; Wanner, Gerhard (2008), *Solving ordinary differential equations I: Nonstiff problems,* Berlin, New York: Springer-Verlag, ISBN 978-3-540-56670-0.

[7] W.H. Hundsdorfer and J. Verwer, *Numerical solution of time dependent advection diffusion reaction equations*, Springer, Berlin, 2003.

[8] W Huang and B. Leimkuhler, *The adaptive Verlet method*, SIAM J Sci. Comput. 18, 1997, 239-256.

[9] P. Hut, J. Makino and S. McMillan, *Building a better leapfrog*, Astrophysical J. 443(1995), L93-L96.

[GS] P.M. Gresho and R.L. Sani, Incompressible Flow and the Finite Element Method, Volume 2: Isothermal Laminar Flow, Wiley, NY, 2000.

[H02] W. Hundsdorfer, *Accuracy and stability of splitting with Stabilizing Corrections*, Applied Numerical Mathematics, Volume 42, Issues 1-3, Numerical Solution of Differential and Differential-Algebraic Equations, 4-9 September 2000, Halle, Germany, August 2002, Pages 213-233, ISSN 0168-9274, DOI: 10.1016/S0168-9274(01)00152-0.

[JK63] O. Johansson, H.-O. Kreiss, *Über das Verfahren der zentralen Differenzen zur Lösung des Cauchy problems für partielle Differentialgleichungen,* Nordisk Tidskr. Informations-Behandling 3 (1963) 97–107.

[10] W. Layton and C. Trenchea, *Stability of two IMEX methods, CNLF and BDF2-AB2, for uncoupling systems of evolution equations*, ANM 62(2012), 112-120.

[L71] B. Lindberg, *On smoothing and extrapolation of the trapezoid rule*, BIT 11(1971) 29-52.

[O] O. Osterby, *5 ways to reduce the trapezoid rule oscillation*, BIT

[R62] A. Ralston, *Runge-Kutta methods with minimum error bounds*, Math. Comp., 16 (1962), 431-437

[11] J.M. Sans-Serna, *Studies in Numerical Nonlinear Instability I. Why do Leapfrog Schemes Go Unstable?* , SIAM J. Sci. and Stat. Comput., 6(1985), 923–938.

[12] J.M. Sans-Serna and M.P. Calvo, *Numerical Hamiltonian Problems,* Chapman and Hall, London, 1994.

[SG79] L.F. Shampine and C.W. Gear, A *users view of solving stiff ordinary differential equations,* SIAM Review 21(1979) 1-17.

[13] R.D. Skeel and J.J. Biesiadecki, *Symplectic integrators with variable timesteps,* Annals of Numerical Mathematics, 191-198, 1994.

[14] Skeel and Gear 1992.

[15] Skeel, R. D., *Variable Step Size Destabilizes the Stömer/Leapfrog/Verlet Method,* BIT Numerical Mathematics, Vol. 33, 1993, pp. 172-175

[16] I. Sloan and A.R. Mitchell, *On nonlinear instabilities in leap-frog finite difference schemes*, Journal of Computational Physics, 67(1986) 372–395.

[17] S. J. Thomas, R. D. Loft, *The NCAR spectral element climate dynamical core: semi-implicit Eulerian formulation*, J. Sci. Comput. 25 (1-2) (2005) 307–322.

[V80] J.M. Varah, *Stability restrictions on a second order, three level finite difference schemes for parabolic equations,* SINUM 17(1980) 300-309.

[18] J. Verwer, *Convergence and component splitting for the Crank-Nicolson Leap-Frog integration method*, CWI report MAS-E0902, 2009.

[W11] P.D. Williams, *The RAW Filter: An Improvement to the Robert–Asselin Filter in Semi-Implicit Integrations.* Mon. Wea. Rev., 139 (2011) 1996–2007.

[19] W. Zhou, *An alternative Leap-Frog scheme for surface gravity wave equations*, J. Amer. Meteorological Soc. 19(2002) 1415-1423.