\documentclass[a4paper,12pt, total={21cm, 29cm}]{article}

% Required for inserting images

\usepackage{graphicx}

%For adding clicable references

\usepackage{hyperref}

%Math

\usepackage[thinc]{esdiff}

\usepackage{amsmath}

\usepackage{amsthm}

\usepackage{amssymb}

\usepackage{geometry}

\geometry{left=2.5cm, right=2.5cm}

%Package to display line numbers

%\usepackage{lineno}

%\linenumbers

\usepackage[T1]{fontenc}

\usepackage[scaled]{helvet}

\renewcommand\*{\familydefault}{\sfdefault}

\begin{document}

%Commands

\renewcommand{\qed}{\hfill$\blacksquare$}

\newcommand{\R}{\mathbb{R}}

\newcommand{\tx}[1]{\quad \text{#1} \quad}

\newcommand{\lb}[1]{\if\relax\detokenize{#1}\relax{\nonumber}\else{\label{#1}}\fi}

%Equation number depends on secton

\numberwithin{equation}{section}

%Definitions

%\theoremstyle{definition}

\newtheorem{df}{Definition}[section]

%Theorems and Lemmas

\newtheorem{thm}{Theorem}[section]

\newtheorem{corollary}{Corollary}[thm]

\newtheorem{lemma}[thm]{Lemma}

\newtheorem{prop}[thm]{Proposition}

\null\thispagestyle{empty}\newpage

\null\thispagestyle{empty}\newpage

\textbf{Dedication or acknowledgements (if not included, a blank page)}\thispagestyle{empty}\newpage

\null\thispagestyle{empty}\newpage

%Resetting the page numbering

\clearpage

\pagenumbering{arabic}

\tableofcontents

\newpage

%---------------------------Introduction----------------------------------

\section{Introduction}

The main focus of this paper is to provide an introductory dive into some important aspect of Numerical Analysis, with a particular emphasis of analyzing methods to solve integrals and differential equations. In section \ref{section\_integrals} we will see that these two concepts often go hand in hand, which leads to many similarities between the numerical methods. Before starting with the numerical methods I will state some important theorems and I will devote some time to polynomial interpolation as this is the basis of many methods we will see.

%---------------------------Useful theorems----------------------------------

\subsection{Some useful theorems}

The following theorems are give without proof, but are extremely important in some proofs I will be discussing in this paper.

\begin{thm}[Generalized Integral Mean Value Theorem]\label{Gen\_IMVT}

Suppose $f,g:\R \longrightarrow \R$ with $f,g \in C([a,b])$. Furthermore, assume $g$ does not change sign on $[a,b]$. Then there exists a points $\xi \in [a,b]$ such that

\begin{equation}\nonumber

\int\_{a}^{b}g(x)f(x)\,dx = f(\xi)\int\_{a}^{b}g(x)\,dx

\end{equation}

\end{thm}

\begin{proof}

The proof can be found in JE book pp10

\end{proof}

%---------------------------Polynomial Interpolation------------------------------

\section{Polynomial Interpolation}

%https://services.math.duke.edu/~jtwong/math563-2020/lectures/Lec1-polyinterp.pdf

%For proof of lagrange theorem:

%https://math.okstate.edu/people/binegar/4513-F98/4513-l16.pdf

Throughout numerical analysis we encounter problems that have to deal with very intricate and complicated functions which often cause troubles when we try applying our analysis tools on them. To try and overcome these issues we must find clever ways to approximate the complicated functions with simpler ones with which we can apply our mathematical methods too. Polynomials seems to check all the boxes in this case. They are very easy to manipulate and to perform operations on. Furthermore, they are extremely versatile as you can increase their degree and play around with their coefficients to approximate essentially any function. Actually the Weierstrass approximation theorem states that any continuous function on a closed interval $[a,b]$ can be uniformly approximated to an arbitrary level of precision by a polynomial.

This theorem,however, does not tell us how to find such a polynomial. A classic why of finding a possible polynomial is using a Taylor polynomial. While this is very powerful in itself in this chapter we will explore another why of find a polynomial approximation of a function: Lagrange polynomial interpolation.\\

Given a function f and n+1 points $\{x\_0, ... , x\_n\}$(data points for example) define

\begin{equation}

y\_i= f(x\_i) \quad \text{for} \quad i \in \{0,...,n\}

\end{equation}

We say a function $p(x)$ is an interpolant for $f(x)$ if $p(x)$ matches $f(x)$ at the n+1 points,

\begin{equation}\label{poly\_interpolation\_conditions}

p(x\_i) = y\_i \quad \text{for} \quad i \in \{0,...,n\}

\end{equation}

We will focus on polynomial interpolations throughout this paper. On a side note, I often use interpolation and approximation interchangeably, however they are not the same thing. Interpolation is a method that may or may not be a useful approximation. All we say is that it must match the function at a finite amount of points. A priori there is no reason to believe that this may work as a good approximation. Luckily in numerical analysis interpolation works very well! Its main usefulness is when we have a finite number of data points and values of a function. In this case methods like Taylor series will fail as we have no real knowledge about derivative. When we have complete knowledge about the function say $f(x) = cos(x)$ it may be a lot better to use a Taylor approximation compared to a polynomial interpolation.\\

\\

Our objective is to construct a polynomial that satisfies \ref{poly\_interpolation\_conditions}. While I will not present the derivation, the construction of $p(x)$ comes down to understanding the Lagrange basis.

\begin{equation}

l\_i(x)=\prod\_{j: j \neq i} \frac{x-x\_j}{x\_i-x\_j}

\end{equation}

By simple inspection we notice that

\begin{equation}

l\_i(x\_i) = \prod\_{j: j \neq i} \frac{x\_i-x\_j}{x\_i-x\_j} = \prod\_{j: j \neq i} 1 = 1 \quad \text{and} \quad l\_i(x\_k) = \prod\_{j: j \neq i} \frac{x\_k-x\_j}{x\_i-x\_j} = 0

\end{equation}

for all $k \neq i$. This is basically the Kronecker delta constructed with polynomials. Using these n-th degree polynomials we can easily construct our interpolant polynomial as

\begin{equation}\label{poly\_interpolation}

p(x) = \sum\_{i=0}^{n}y\_il\_i(x)

\end{equation}

\begin{prop}[Uniqueness of degree at most n]\label{uniqueness\_of\_interp}

Assume $f \in C^{n+1}([a,b])$ with n+1 points

\begin{equation}

x\_0<x\_1<x\_2<...<x\_{n-1}<x\_n \quad \text{with} \quad x\_i \in [a,b]\quad \forall i \in \{0,...,n\} \nonumber

\end{equation}

Let $p\_n(x)$ be the interpolating polynomial such that $deg(p\_n) \leq n$. Then $p\_n(x)$ is the unique polynomial of degree at most n that satisfies \ref{poly\_interpolation\_conditions}.

\end{prop}

\begin{proof}

Assume, by contradiction, that there exist two polynomials $p\_n$ and $q\_n$ such that

\begin{equation}

deg(p\_n),deg(q\_n) \leq n \quad \text{and} \quad \exists x \in [a,b]: p\_n(x) \neq q\_n(x) \nonumber

\end{equation}

Define $r(x) = p\_n(x) - q\_n(x)$ which implies that $deg(R) \leq n$. Moreover, notice that

\begin{equation}

r(x\_i) = p\_n(x\_i) - q\_n(x\_i) = 0 \quad \forall i \in \{0,...,n\} \nonumber

\end{equation}

By the fundamental theorem of algebra a polynomial of degree at most n, must have at most n roots, however $r(x)$ has n+1 roots and $deg(r) \leq n$ which implies that $r(x) = 0 \quad \forall x \in [a,b]$. This mean that $p\_n(x) = q\_n(x)\quad \forall x \in [a,b]$ which gives a contradiction.

\end{proof}

%---------------------------Error analysis of interpolation---------------------

\subsection{Error analysis of Lagrange polynomial interpolation}

Many of the numerical method I will cover start from a polynomial interpolation of the function in question. To provide a detailed error analysis of the methods to come we must first understand the error that comes from polynomial interpolation. To this end we have the following theorem.

\begin{thm}[Lagrange error formula]\label{Err\_interpolation\_thm}

Assume $f \in C^{n+1}([a,b])$ with n+1 points

\begin{equation}

x\_0<x\_1<x\_2<...<x\_{n-1}<x\_n \quad \text{with} \quad x\_i \in [a,b]\quad \forall i \in \{0,...,n\} \nonumber

\end{equation}

Let $p\_n(x)$ be the interpolating polynomial of degree n. Then for all $x \in [a,b]$

\begin{equation}

f(x) = p\_n(x) + E\_n(x)\nonumber

\end{equation}

with $E\_n(x)$ the interpolation error which takes value

\begin{equation}\label{Err\_interpolation}

E\_n(x) = \frac{f^{(n+1)}(\eta\_x)}{(n+1)!}\prod\_{j=0}^{n}(x-x\_j)

\end{equation}

for some $\eta\_x \in [a,b].$

\end{thm}

\begin{proof}

\end{proof}

Let's take a better look at equation \ref{Err\_interpolation}. The first term we see is the (n+1)-th derivative. As we are working with a compact interval we can easily bound this by taking the $\max\_{x \in [a,b]} |f^{(n+1)}(x)|$. As we decrease the subinterval size this value may decrease or stay the same depending on the function.\\

The second term is $\frac{1}{(n+1)!}$ which, lucky for us, goes to 0 very quickly as $n \to +\infty$.\\

The last term is what we call $\omega(x)$:

\begin{equation}

\omega(x) := \prod\_{j=0}^{n}(x-x\_j)

\end{equation}

We can see that $\omega(x)$ will be small if $x$ is close to one of the points and will be large otherwise.\\

We will see that error analysis essentially boils down which factor will have a bigger effect on the upper bound as we increase the number of points at our disposal. The upper bound on an interval $I$ is of the form

\begin{equation}

|E\_n(x)| \leq \frac{\max\_{x \in I} |f^{(n+1)}(x)|}{(n+1)!}\max\_{x \in I} |\omega(x)|

\end{equation}

%---------------------------Runge phenomenon----------------------------------

\subsection{Convergence of Lagrangian interpolation}

Now this is all well and good, however are we sure that this error goes to 0 as we increase the degree of the polynomial. The answer to this depends on our choice of the interpolation points. All of the methods I will present will assume that the data points are evenly spaced with size $h$. In doing so, however, we risk running into a problem of the interpolation polynomial diverging from the function towards the end points. The next example illustrates an instance in which this happens.

\subsubsection{Runge's phenomenon}

Consider the interval $[-5,5]$ and the following $C^{\infty}([-5,5])$ function over that interval,

\begin{equation}\label{Runge\_function}

G(x) := \frac{1}{1 + x^2}

\end{equation}

If we try and interpolate this function using a uniform grid we see that towards the edges the polynomial behaving badly as it starts to rapidly oscillate\footnote{All the code for the graphs can be found \href{https://github.com/MattiaBarbiere/Numerical-Method-for-ODE-s}{here}}.

\begin{center}

\includegraphics[width=\textwidth]{runge phenomenon.png}

\end{center}

For larger values of n this behaviour only gets worse. The problem of this function is that the n-th derivatives blow up in the interval $[-5,5]$. The tables below provide some empirical insight of this occurring. \\

\begin{minipage}[b]{0.5\textwidth}

\begin{flushright}

\includegraphics[width=0.3\textwidth]{max table.png}

\hspace{10mm}

\end{flushright}

\end{minipage}%

\begin{minipage}[b]{0.5\textwidth}

\includegraphics[width=0.3\textwidth]{min table.png}

\end{minipage}%

\\

More precisely if we plot $\max\_{x \in [-5,5]} |G^{(n)}(x)|$ and $\min\_{x \in [-5,5]} |G^{(n)}(x)|$ as a function of $n$ on a logarithmic scale we can see that this blow up actually grows exponentially in n.\\

\begin{minipage}[b]{0.5\textwidth}

\includegraphics[width=\textwidth]{max over log scale.png}

\end{minipage}%

\begin{minipage}[b]{0.5\textwidth}

\includegraphics[width=\textwidth]{min over log scale.png}

\end{minipage}%

\\

It might seem that we have little hope of working with the polynomial interpolation with evenly spaced points, luckily for us there are a few workaround that we can exploit. The main solution that all the methods I will tackle use is \textit{spline interpolation}. essentially what this does is to break up the interval into many subintervals. With this in mind we can use our thoery of interpolation to interpolate over the function with a lower degree polynomial. If the subintervals are small and the function is smooth enough, our polynomial interpolation over the subinterval will be a very good approximation of the function. Below we can see how spline interpolation works very well even with the problematic Runge function $G(x)$.

\begin{center}

\includegraphics[width=\textwidth]{Spline interpolation.png}

\end{center}

With this solution in mind, numerical methods boil down to understanding them over a single interval and then summing over all the subintevals.

%---------------------------Integrals----------------------------------

\section{Numerical Methods for Solving Integrals}\label{section\_integrals}

Before directly analyzing methods for solving differential equations, we first explore some methods for numerically solving integrals. This is motivated by the fact that often differential equations and integrals are essentially the same thing.

Take a differential equation as defined in \ref{IVP}. Often we need to assume that $f$ is continuous on an interval $(a, b)$ containing $t\_{0}$ for a solution to exist. This would also make $f$ an integrable function. Taking the integral of the ODE.

\begin{equation}

\int\_{t\_{0}}^{t}f(s,y(s))ds = \int\_{t\_{0}}^{t}y'(s) ds = y(t) - y(t\_{0}) = y(t) - y\_{0}

\end{equation}

Giving us the equation

\begin{equation}\label{Integral\_ODE}

y(t) = y\_{0} + \int\_{t\_{0}}^{t}f(s,y(s))ds

\end{equation}

%---------------------------Trap Rule----------------------------------

\subsection{Trapezoidal Rule for Integration}\label{trap\_method}

Given any function $f: \R \longrightarrow \R$ our objective is to compute

\begin{equation}\label{integral}

I(f) = \int\_{a}^{b}f(x)\,dx

\end{equation}

Starting from a first degree polynomial interpolation of $f$, from \ref{poly\_interpolation} we take

\begin{equation}

p(x) = \frac{x-a}{b-a} f(b) + \frac{b-x}{b-a} f(a)

\end{equation}

This polynomial provides a linear approximation of the function $f$ in the interval $[a,b]$. By integrating the polynomial we could approximate the integral we are after. Geometrically, by integrating $p(x)$ we are calculating the area of a (rotated) trapezoid with vertices $(a,0),$ $ (a,f(a)), (b, f(b))$ and $(b,0)$. Using the formula for the area of a trapezoid with height equal to $(b-a)$ we have

\begin{equation}\label{def\_trap\_single\_interval}

T\_{1}(f) := \int\_{a}^{b}p(x)\,dx = \frac{1}{2}(b-a) (f(a) + f(b))

\end{equation}

If is very evident that if $b-a$ is very large this approximation could be very far from the correct value of the integral. The error denoted as $E\_T(f)$ in fact is given by the following theorem.

\begin{thm}[Trapezoid method error with a single interval]\label{thm\_trap\_error\_1\_interval}

Let $f \in C^2([a,b])$ and let $T\_1(f)$ be the trapezoid approximation to I(f) as defined in \ref{def\_trap\_single\_interval}. Then there exists $\xi \in [a,b]$ such that

\begin{equation}\nonumber

E\_T(f) := I(f) - T\_n(f) = -\frac{1}{12}f''(\xi)(b-a)^3

\end{equation}

\end{thm}

\begin{proof}

\begin{equation}

\begin{split}

E\_T(f) & = I(f) - T\_{1}(f) =

\int\_{a}^{b}f(x)\,dx - \int\_{a}^{b}p(x)\,dx = \\

& =\int\_{a}^{b}(f(x) - p(x))\,dx = \frac{1}{2} \int\_{a}^{b}((x-a)(x-b)f''(\eta\_{x}))\,dx \nonumber

\end{split}

\end{equation}

Where the last step comes from theorem \ref{Err\_interpolation\_thm}.

As $(x-a)(x-b) \leq 0$ on $[a,b]$ we use theorem \ref{Gen\_IMVT} we get that for some $\xi \in [a,b]$:

\begin{equation}\nonumber

\begin{split}

E\_T(f) &= \frac{1}{2} \int\_{a}^{b}(x-a)(x-b)f''(\eta\_{x})\,dx=\\

&= \frac{1}{2} f''(\xi) \int\_{a}^{b}(x-a)(x-b)\,dx= \\

&= \frac{-1}{12} f''(\xi) (b-a)^3

\end{split}

\end{equation}

As requested.

\end{proof}

As we expected the error depends $b - a$ this means that if we have a large interval we may be very far away from the true value of the integral. The trick is to divide the interval into many smaller intervals and then summing up the area of all the smaller trapezoids. Luckily for us differentiable function look almost linear at each point if we "zoom" in close enough so by taking small enough subintervals we will get a good approximation.\\

Take n subintervals $[x\_{i-1}, x\_{i}]$ for $ i \in \{1,..,n\}$ with $x\_0 = a$ and $x\_n = b$ then our integral becomes.

\begin{equation}

I(f) = \sum\_{i=1}^{n} \int\_{x\_{i-1}}^{x\_{i}} f(x)\,dx

\end{equation}

For each subinterval we follow the approximation of the previous part and we get

\begin{equation}\label{def\_T\_n}

\sum\_{i=1}^{n} \int\_{x\_{i-1}}^{x\_{i}} f(x)\,dx \approx

\sum\_{i=1}^{n} \frac{1}{2}(x\_{i} - x\_{i-1}) (f(x\_{i}) + f(x\_{i-1})) =: T\_n(f)

\end{equation}

How we divide the interval $[a,b]$ will have an effect on overall error of the approximation as we will see next.

%---------------------------Trap uniform----------------------------------

\subsubsection{Trapezoid Method with uniform grid}

Under the assumption that the grid is uniform we can find an exact value of the error directly from the calculation of $E\_T(f)$ in the previous section. This lead us to the following theorem.

\begin{thm}[Trapezoid method error with a uniform grid] \label{Err\_uni}

Let $f \in C^2([a,b])$ and let $T\_n(f)$ be the n-subinterval trapezoid approximation to $I(f)$, using a uniform grid with subinterval size $h = \frac{b-a}{n}$. Then there exists $\eta\_{h} \in [a,b]$ such that

\begin{equation}\nonumber

E\_T(f) := I(f) - T\_n(f) = -\frac{b-a}{12}h^2f''(\eta\_{h})

\end{equation}

\end{thm}

\begin{proof}

The proof is basically just summing of $E\_{T}(f)$ n times(JE 2013 book).

\end{proof}

This leads us to the following important corollary.

\begin{corollary}\label{cor\_err\_uni}

With the assumptions of the previous theorem, by letting the size $h$ of the uniform grid go to zero and so letting the number of subintervals go to infinity we get that

\begin{equation}\nonumber

\lim\_{n\to\infty} T\_n(f) = I(f)

\end{equation}

\begin{proof}

Taking the absolute value of Theorem \ref{Err\_uni} and taking the limit

\begin{equation}

\lim\_{n\to\infty} |E\_T(f)| =

\lim\_{n\to\infty} \frac{b-a}{12}h^2|f''(\eta\_{h})|

\leq \left(\lim\_{n\to\infty} \frac{b-a}{12}h^2\right)\max\_{x \in [a,b]}|f''(x)| = 0\nonumber

\end{equation}

\end{proof}

\end{corollary}

So we are able to get the error sufficiently small by choosing $h$ small enough which in turn depends on the amount of $x\_i$'s. So to lower the error all we have to do is increase the number of subintervals. Furthermore, from theorem \ref{Err\_uni} we can clearly see that the trapezoidal rule with uniform grid has an error of second order $O(h^2)$.

%---------------------------Trap non-uni----------------------------------

\subsubsection{Trapezoid Method with non-uniform grid}

If we have an arbitrary grid we are able to find an upper bound for the error depending on the size of the largest subinterval we have.

\begin{thm}[Trapezoid method error with a non-uniform grid] \label{Err\_non\_uni}

Let $f \in C^2([a,b])$ and let $T\_n(f)$ be the n-subinterval trapezoid approximation to $I(f)$, using a non-uniform grid defined by

\begin{equation}\nonumber

a=x\_0<x\_1<x\_2<...<x\_{n-1}<x\_n = b

\end{equation}

with $h\_i = x\_{i+1} - x\_i$ and $h = \max\_{i} h\_i$. Then

\begin{equation}\nonumber

|E\_n(f)| := |I(f) - T\_n(f)| \leq \frac{b-a}{12}h^2\max\_{x \in [a,b]}|f''(x)|

\end{equation}

\end{thm}

\begin{proof}

Similar to before but working with the new definition of h

\end{proof}

We can construct a corollary similar to corollary \ref{cor\_err\_uni} however in this case increasing the number of subintervals will not guarantee that the error converges to 0. To see why take a subinterval $\left[a,\frac{a+b}{2}\right]$. Now we can divide $\left[\frac{a+b}{2}, b\right]$ infinitely many times, however this will not lower the upper bound of our error as the upper bound depends on $h = \max\_{i} h\_i$. So to make sure that our error goes to 0 we must make sure that all grid sizes go to 0 which is equivalent to imposing that the maximum grid size goes to 0. Similarly to the uniform case we also see that in this case the error has second order $O(h^2)$.

%---------------------------Stability of trap----------------------------------

\subsubsection{Stability of the trapezoid rule}

Other than the convergence to the correct value, another important aspect of approximations is stability. That is, if we slightly perturb the input function how does the value of our integral approximation change? For numerical integration often the methods are stable however we will see that when we do a similar analysis for numerical ODE's this is not always the case.\\

Define $g(x) := f(x) + \epsilon(x)$ with $\epsilon(x)$ the slight perturbation of $f(x)$. What can we say about $|T\_n(f) - T\_n(g)|$ given that $|f(x) - g(x)| = |\epsilon(x)|$? Recall from equation \ref{def\_T\_n} that

\begin{equation}\nonumber

\begin{split}

&T\_n(f) = \sum\_{i=1}^{n} \frac{1}{2}(x\_{i} - x\_{i-1}) (f(x\_{i}) + f(x\_{i-1})) \\

&T\_n(g) = \sum\_{i=1}^{n} \frac{1}{2}(x\_{i} - x\_{i-1}) (g(x\_{i}) + g(x\_{i-1}))

\end{split}

\end{equation}

Taking the difference gives

\begin{equation}\nonumber

\begin{split}

T\_n(f) - T\_n(g) & = \sum\_{i=1}^{n} \frac{1}{2}(x\_{i} - x\_{i-1}) (f(x\_{i}) + f(x\_{i-1}) - g(x\_{i}) - g(x\_{i-1}))\\

& = \sum\_{i=1}^{n} \frac{1}{2}(x\_{i} - x\_{i-1}) (\epsilon(x\_{i}) + \epsilon(x\_{i-1}))

\end{split}

\end{equation}

Taking the absolute value and bounding it from above gives

\begin{equation}

|T\_n(f) - T\_n(g)| \leq 2\max\_{x \in [a,b]}|\epsilon(x)|\sum\_{i=1}^{n} \frac{1}{2}(x\_{i} - x\_{i-1}) = (b-a)\max\_{x \in [a,b]}|\epsilon(x)|

\end{equation}

Where the last step comes from the fact that we have a telescoping sum and that $x\_0 = a$ and $x\_n = b$.\\

This tells us that if the maximum perturbation is sufficiently small also $|T\_n(f) - T\_n(g)|$ will be small. Actually if we let $\max\_{x \in [a,b]}|\epsilon(x)| \to 0$ we get that $|T\_n(f) - T\_n(g)| \to 0$ which means $T\_n(f) \to T\_n(g)$ proving that the trapezoid method is numerically stable.

%---------------------------intro Simpson's rule----------------------------------

\subsection{Simpson's Rule for Integration}\label{Simp\_section}

%https://www.uio.no/studier/emner/matnat/math/MAT-INF1100/h11/kompendiet/chap12.pdf

In wake of the derivation of the trapezoid method we can take the polynomial interpolation one step further. That is, for each (sub)interval $[a,b]$ we interpolate the function $f$ at $a, b, \frac{a+b}{2}$ with a second order polynomial. Recall that in the trapezoid method we only interpolated the end points of the intervals with a first degree polynomial. By increasing the degree of the polynomial we get Simpson's rule for integration.\\

Definite $m:=\frac{a+b}{2}$ then our polynomial interpolation of f at $a,b,$ and $m$ is

\begin{equation}\label{second\_deg\_interpolation}

q\_{a,b}(x) = f(a)\frac{(x-m)(x-b)}{(a-m)(a-b)} + f(m)\frac{(x-a)(x-b)}{(m-a)(m-b)} +

f(b)\frac{(x-a)(x-m)}{(b-a)(b-m)}

\end{equation}

We could follow the steps as we did with the trapezoid method, however there is a more elegant derivation that I will present.

%---------------------------Simp on [-1.1]----------------------------------

\subsubsection{Particular case of Simpson's method}

Let's solve for the case when $a = -1, m=0$ and $b=1$:

\begin{equation}

q\_{-1,1}(x) = f(-1)\frac{x(x-1)}{2} - f(0)(x+1)(x-1) +

f(1)\frac{x(x+1)}{2}\nonumber

\end{equation}

Integrating $q\_{-1,1}(x)$ on $[-1,1]$ gives

\begin{equation}

\int\_{-1}^{1}q\_{-1,1}(x)\,dx = f(-1)\int\_{-1}^{1}\frac{x(x-1)}{2}\,dx - f(0)\int\_{-1}^{1}(x+1)(x-1)\,dx +

f(1)\int\_{-1}^{1}\frac{x(x+1)}{2}\,dx =\nonumber

\end{equation}

\begin{equation}\label{Simp\_-1\_1}

= \frac{1}{3}(f(-1) + 4f(0) + f(1))

\end{equation}

%---------------------------Simpson's rule----------------------------------

\subsubsection{Simpson's rule}\label{Simp\_single\_interval}

Now notice the following function

\begin{equation}\label{map\_a\_b}

y=\frac{2(x-a)}{b-a}-1

\end{equation}

This function maps $x \in [a,b]$ to $y \in [-1,1]$.\\

We can now move to the general case. Using the inverse of \ref{map\_a\_b} we can perform a substitution and we get

\begin{equation}

x=\frac{b-a}{2}(y+1)+a \implies \,dx = \frac{b-a}{2}\,\,dy \nonumber

\end{equation}

\begin{equation}

I(f) = \int\_{a}^{b}f(x)\,\,dx = \frac{b-a}{2}\int\_{-1}^{1}f(\frac{b-a}{2}(y+1)+a)\,dy

\end{equation}\\

For clarity let's define $F(y):=f(\frac{b-a}{2}(y+1)+a)$, and using \ref{Simp\_-1\_1} with $F(y)$ we get\\

\begin{equation}

\frac{b-a}{2}\int\_{-1}^{1}F(y)\,dy \approx \frac{b-a}{2}\int\_{-1}^{1}q\_{-1,1}(x)\,dx = \frac{b-a}{6}(F(-1) + 4F(0) + F(1))

\end{equation}\\

Recalling that $F(-1) = f(a), F(0) = f(m), F(1) = f(b)$ and letting $h=\frac{b-a}{2}$ we have the Simpson's method for integration defined as

\begin{equation}

S\_2(f) := \frac{h}{3}(f(a) + 4f(m) + f(b)) \approx I(f)

\end{equation}

%---------------------------Composite simp rule----------------------------------

\subsubsection{Composite Simpson's rule}

To give a better approximation of the integral, we divide the interval $[a,b]$ in many subintervals and perform what we did in section \ref{Simp\_single\_interval} on each subinterval. Often in practice if we are given data points $x\_i$ and $x\_{i+1}$ we are not always sure what the value of $f(\frac{x\_i + x\_{i+1}}{2})$ is. So we can only evaluate the function in the data points we know.\\

Given an uniformly distributed even number of points

\begin{equation}

a=x\_0<x\_1<x\_2<...<x\_{n-1}<x\_n = b \tx{and} h = \frac{x\_{2k}-x\_{2k-2}}{2} = x\_{2k}-x\_{2k-1}\nonumber

\end{equation}

we apply the Simpson's method on each subinterval of the form $[x\_{2k-2}, x\_{2k}]$ for \\$k \in \{1,...,\frac{n}{2}\}$ using $x\_{2k-1}$ as the midpoint of each subinterval.\\

For a single interval section \ref{Simp\_single\_interval} tell us

\begin{equation}

\int\_{x\_{2k-2}}^{x\_{2k}}f(x)\,dx \approx \frac{h}{3}(f(x\_{2k-2}) + 4f(x\_{2k-1}) + f(x\_{2k}))

\end{equation}

By summing over all the subintervals we get the complete Simpson's rule as

\begin{equation}\label{Simpson's estimator}

\int\_{a}^{b}f(x)\,dx = \sum\_{k=1}^{n/2}\int\_{x\_{2k-2}}^{x\_{2k}}f(x)\,dx \approx \frac{h}{3}\sum\_{k=1}^{n/2}(f(x\_{2k-2}) + 4f(x\_{2k-1}) + f(x\_{2k})) =: S\_n(f)

\end{equation}

%---------------------------power of simp rule----------------------------------

\subsubsection{The power of Simpson's rule}

Recall from section \ref{Simp\_section} that Simpson's rule starts from a second degree polynomial interpolation of the function $f$. Consider the special case when $f$ is itself polynomial of at most second degree. Then by interpolating $f$ as we did in \ref{second\_deg\_interpolation} we would get that $q\_{a,b}$ is exactly equal to $f$. If this were not the case we could view $f$ as a polynomial interpolation of itself and we would contradict proposition \ref{uniqueness\_of\_interp}. In this case Simpson's rule exactly calculates the integral. Now this seems like a reasonable thing to happen, however what special about this rule is that the same is true for third degree polynomials! This gives rise to the following lemma.

\begin{lemma}\label{power\_of\_simpson}

Let $f(x)$ be a polynomial with $deg(f) \leq 3$. Then Simpson's rule comes out to be an equality

\begin{equation}

S\_2(f) = \int\_{a}^{b}f(x)\,dx \nonumber

\end{equation}

\end{lemma}

\begin{proof}

I have already talked about the case when $deg(f) \leq 2$. Let's focus on $deg(f) = 3$.

\begin{equation}

f(x) = Cx^3 + g(x) \tx{with} deg(g) \leq 2 \nonumber

\end{equation}

Notice that both $I$ and $S\_2$ are linear so the error becomes

\begin{equation}\nonumber

I(f) - S\_2(f) = C(I(x^3) - S\_2(x^3)) + (I(g) - S\_2(g))

\end{equation}

By our discussion above $I(g) - S\_2(g) = 0$. Let's evaluate $S\_2(x^3)$

\begin{equation}\nonumber

\begin{split}

S\_2(x^3) & = \frac{b-a}{6}\left(a^3 + 4 \left(\frac{a+b}{2}\right)^3+b^3\right)=\\

& = \frac{b-a}{6}\left(a^3 + \frac{1}{2} a^3 + \frac{3}{2}a^2b + \frac{3}{2}ab^2 + \frac{1}{2}b^3+b^3\right) = \\

& = \frac{b-a}{4}(a^3 + a^2b + ab^2 + + b^3) = \\

& = \frac{(b-a)^4}{4}=\\

& = I(x^3)

\end{split}

\end{equation}

Which gives us that $I(f) - S\_2(f) = 0$ concluding the proof.

\end{proof}

%---------------------------Error of simp rule----------------------------------

\subsubsection{Error analysis of Simpson's rule}

One might be tempted to follow a similar reasoning as we did with the trapezoid method. We have three points $a,m,$ and $b$, and starting from \ref{Err\_interpolation} we integrate and that give us the error. Due to lemma \ref{power\_of\_simpson}, this gives us an error of 0,

\begin{equation}\nonumber

I(f) - S\_2(f) = \int\_{a}^{b}E\_2(x)\,dx = \frac{f^{(n+1)}(\xi)}{(n+1)!}\int\_{a}^{b}((x-a)(x-m)(x-b))\,dx = 0

\end{equation}

Since

\begin{equation}\nonumber

\int\_{a}^{b}\left((x-a)\left(x-\frac{a+b}{2}\right)(x-b)\right)\,dx = 0

\end{equation}

After having seem lemma \ref{power\_of\_simpson} this comes as no surprise, however this means that we have to come up with a better method for deriving the error of Simpson's rule. I will next prove the following theorem.

\begin{thm}[Error of Simpson's rule]

Suppose $f \in C^4([a,b])$, then there exists a point $\xi \in [a,b]$ such that

\begin{equation}

I(f) - S\_2(f) = -\frac{(b-a)^5}{2880}f^{(4)}(\xi)

\end{equation}

\end{thm}

In Simpson's rule we are only given 3 points which means that we can interpolate with a second degree polynomial. As we have already discussed directly computing the error from $E\_2(x)$ is of little help. We would like to go up by one degree, that is calculate the error using $E\_3(x)$. We only have 3 points at our disposal and ultimately to derive Simpson's rule we only need these. To overcome this we start with 4 points and by taking a limit we end up with only 3.

\begin{proof}

Let $p\_3$ interpolate $f$ at the points $x\_0=a, x\_1=m-\epsilon, x\_2=m+\epsilon,$ and $x\_3=b$ with $\epsilon > 0$ a small deviation. By \ref{Err\_interpolation}, for $x \in [a,b]$

\begin{equation}

f(x) - p\_3(x) = E\_3(x) = \frac{1}{4!}(x-x\_0)(x-x\_1)(x-x\_2)(x-x\_3)f^{(4)}(\eta\_x)

\end{equation}

for some $\eta\_x \in [a,b]$. By lemma \ref{power\_of\_simpson} we have that

\begin{equation}

I(f) - S\_2(f) = I(E\_3) - S\_2(E\_3) + I(p\_3) - S\_2(p\_3)=I(E\_3) - S\_2(E\_3)

\end{equation}

As $p\_3$ is an interpolating polynomial of $f$ we know that at $x=a $ and $x=b$ the error $E\_3(x)$ will vanish and so we get

\begin{equation}

\begin{split}

S\_2(E\_3) & = \frac{b-a}{6}(E\_3(a) + 4E\_3(m) + E\_3(b)) =\\

& = \frac{b-a}{6}(4E\_3(m)) = \\

& = \frac{4(b-a)}{6 \* 4!}(m-x\_0)(m-x\_1)(m-x\_2)(m-x\_3)f^{(4)}(\eta\_m)

\end{split}

\end{equation}

Now by taking the limit as $\epsilon \to 0$ we get that $x\_1 \to m$ and $x\_2 \to m$. In turn this gives $E\_3(m) \to 0$ and so also $S\_2(E\_3)$ goes to 0. Lastly, as $\epsilon \to 0$ we get

\begin{equation}

I(E\_3) \to \frac{1}{4!}\int\_{a}^{b}(x-a)(x-m)^2(x-b)f^{(4)}(\eta\_x)\,dx

\end{equation}

So in the limit we are left with

\begin{equation}

I(f) - S\_2(f) = \frac{1}{4!}\int\_{a}^{b}(x-a)(x-m)^2(x-b)f^{(4)}(\eta\_x)\,dx

\end{equation}

As $(x-a)(x-m)^2(x-b) \leq 0 \quad \forall x \in [a,b]$ we can use theorem \ref{Gen\_IMVT} and for some $\xi \in [a,b]$ we get

\begin{equation}

I(f) - S\_2(f) = \frac{f^{(4)}(\xi)}{4!}\int\_{a}^{b}(x-a)(x-m)^2(x-b)\,dx

\end{equation}

Computing the integral gives

\begin{equation}

\int\_{a}^{b}(x-a)(x-m)^2(x-b)\,dx = -\frac{(b-a)^5}{120}

\end{equation}

So

\begin{equation}\nonumber

I(f) - S\_2(f) = -\frac{(b-a)^5}{2880}f^{(4)}(\xi)

\end{equation}

\end{proof}

By applying this theorem to each subinterval one can prove the following theorem.

\begin{thm}[Error of composite Simpson's rule]\label{composite\_simp\_error}

Assume we have a uniform grid of n points with grid size $h=\frac{b-a}{n}$ and that $f \in C^4([a,b])$, then for some $\xi \in [a,b]$

\begin{equation}\nonumber

I(f) - S\_n(f) = -\frac{(b-a)h^4}{180}f^{(4)}(\xi)

\end{equation}

\end{thm}

Here we clearly see that in fact Simpson's method has error order 4, $O(h^4)$. The remarkable thing about this rule is that we only increase the interpolation degree by one, compared to the trapezoid method, but the error increase by 2 orders!

%---------------------------Example simp rule--------------------------------

\subsubsection{Simpson's example}

Does this theoretical order really hold? With the following simple example we will see that this does in fact occur. We would like to calculate the following integral:

\begin{equation}

\int\_{0}^{1}e^x\,dx

\end{equation}

Which we how has value $e-1$. By implementing \ref{Simpson's estimator} in a computer program we get the following table.

\begin{center}

\includegraphics[width=0.65\textwidth]{Simpson'r rule example.png}

\end{center}

In the table above we compute the integral with Simpson's method doubling $n$ every time. Consequently this means that h gets halved each time. The third column is the approximate value given by Simpson's method. We see that it gets remarkably close to the correct value. In fact the error becomes very small for large $n$. Now to compute the order we calculate the ratio using theorem \ref{composite\_simp\_error}

\begin{equation}

\frac{I(f) - S\_{n}(f)}{I(f) - S\_{2n}(f)} = \frac{(\frac{b-a}{n})^4}{(\frac{b-a}{2n})^4}=16

\end{equation}

By looking at the last column of the table we see that this theoretical convergence does hold. The example was chosen because the ratio converges to 16 rather quickly, for other examples $n$ might have to be larger to see the ratio approach 16.

%---------------------------Num for oDE----------------------------------

\section{Numerical Methods for Solving Differential Equations}

\subsection{Mathematical preliminaries}

%---------------------------ODE----------------------------------

\subsubsection{Ordinary Differential Equations}

Differential equation have as unknown a function, I will denote it $y$. For the rest of this paper we will be solving for function of one variable, commonly denoted $t$. Given information about the first derivative of the function in question we would like to solve for the function $y$.

\begin{equation}

\diff{y}{t}(t) = y'(t) = f(t,y(t))

\end{equation}

As we know that the function $y$ only depends only on one variable we often omit writing the dependence on $t$.

In solving this differential equation we would find a whole family of functions. When we differentiate we lose information about the translation up or down of the function. To solve for this we need some other information. The next to section show how to get this information.

%---------------------------IVP----------------------------------

\subsubsection{Initial Value Problems}

As the name suggests this tell us a known specific value $y\_{0}$ our function $y(t)$ takes at some initial time which we denote $t\_{0}$. Often for examples we will take $t\_{0} = 0$ for simplicity. A complete initial value problems (also called a Cauchy problem) is as follows.

\begin{df}[Initial Value Problem]\label{IVP}

Let $t\_{0}, t\_{1}, y\_{0}, y\_{1} \in \R$ such that $t\_{0} < t\_{1}$. Let $f:\R^{3} \longrightarrow \R$ and $y:\R \longrightarrow \R$. A Initial Value Problem is the following:

\begin{equation}

\begin{cases}

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

y(t\_{0}) = y\_{0}

\end{cases}

\end{equation}

Where $t\_{0}, y\_{0}$ and $f$ are all known and $y(t)$ in unknown.

\end{df}

%---------------------------BVP----------------------------------

% \subsection{Boundary Value Problems}

% These problems add a layer of difficulty as we are now talking about a second order differential equation. These problems however constrain the function in a close internal $[t\_{0}, t\_{1}]$ and provide the values of the function at the end points. Since the differential equation is of second order we need two conditions to fully define the function.

% \begin{df}[Boundary Value Problem]\label{BVP}

% Let $t\_{0}, y\_{0} \in \R$. Let $f:\R^{2} \longrightarrow \R$ and $y:\R \longrightarrow \R$. A Boundary Value Problem is the following:

% \begin{equation}

% \begin{cases}

% y''(t) = f(t,y(t),y'(t)) \quad for \quad t \in [t\_{0}, t\_{1}]\\

% y(t\_{0}) = y\_{0}\\

% y(t\_{1}) = y\_{1} \nonumber

% \end{cases}

% \end{equation}

% Where $t\_{0}, t\_{1}, y\_{0}, y\_{1}$ and $f$ are all known and $y(t)$ in unknown.

% \end{df}

% While this definition is general sometimes we will have to impose further assumption on the domain of $y$, for example, so we are guaranteed that a unique solution exists.

We are finally ready to to talk about differential equations. These become more complex numerical method compared to integration since an error in the beginning may lead to a much bigger error further down the line, causing a lot of instability with our methods. We start of with the trapezoid method which essentially comes down to applying section \ref{trap\_method} to equation \ref{Integral\_ODE}.

\subsection{Trapezoidal Method}

Recalling equation \ref{Integral\_ODE}, by discretizing time we have that at time $t\_{n+1}$

\begin{equation}

y(t\_{n+1}) = y(t\_n) + \int\_{t\_n}^{t\_{n+1}}f(s,y(s))\,ds

\end{equation}

For clarity when I write numerical methods for ODEs I will set $y(t\_n)=y\_n$. Now applying $T\_1(f)$ from equation \ref{def\_trap\_single\_interval} to the integral

\begin{equation}\label{trap\_method\_for\_ode}

y\_{n+1} = y\_n + \frac{1}{2}h(f(t\_n, y\_n) + f(t\_{n+1}, y\_{n+1})) \tx{and} y(t\_0) = y\_0

\end{equation}

Lucky for us we have already done all the calculation needed to calculate the truncation error in theorem \ref{thm\_trap\_error\_1\_interval} and we get that for all $n>0$ there exists some $\xi\_n \in [a,b]$

\begin{equation}

E\_{t\_n} = -\frac{1}{12}h^3f''(\xi\_n,y(\xi\_n)) = -\frac{1}{12}h^3y'''

(\xi\_n)

\end{equation}

As we can see this is an implicit method, that is, to find the value of $y\_{n+1}$ we must solve an equation. Depending on $f$ this is easier said than done.

%---------------------------Adams-bashforth----------------------------------

\subsection{Adams-Bashforth Method}

It has become common practice in this paper to start numerical methods from a polynomial interpolation. This method is no exception. The derivation of the Adams-Bathforth method is reminiscent of what we did for the trapezoid method and in fact there is a derivation starting from the trapezoid method. Nevertheless, I will again start from polynomial interpolation.

Discretize time into many steps of size $h$. Take the interpolating polynomial of an arbitrary function $g$ at the points $t\_{n-1}$ and $t\_n$

\begin{equation}

p(x) = \frac{x-t\_{n-1}}{t\_n-t\_{n-1}} g(t\_n) + \frac{t\_n-x}{t\_n-t\_{n-1}} g(t\_{n-1})

\end{equation}

Where this method differs compared to the trapezoid method is that we now integrate over $[t\_n,t\_{n+1}]$ as oppose to $[t\_{n-1},t\_{n}]$. As $p(x)$ interpolates $g$ we get the following approximation

\begin{equation}

\int\_{t\_n}^{t\_{n+1}}g(s)\,ds \approx \int\_{t\_n}^{t\_{n+1}}p(s)\,ds

\end{equation}

By computations we get that the integral of the interpolating polynomial is

\begin{equation}

AB(g) := \int\_{t\_n}^{t\_{n+1}}p(s)\,ds = \frac{h}{2}(3g(t\_n) - g(t\_{n-1}))

\end{equation}

To find the truncation error we again start from theorem \ref{Err\_interpolation\_thm}.

\begin{equation}

E\_{t\_n}^{AB} := \int\_{t\_n}^{t\_{n+1}}g(s)\,ds - \int\_{t\_n}^{t\_{n+1}}p(s)\,ds = \int\_{t\_n}^{t\_{n+1}}\frac{g''(\eta\_s)}{2}(s-t\_{n-1})(s-t\_n)\,ds

\end{equation}

We using theorem \ref{Gen\_IMVT} and computation, for some $\xi\_n \in [t\_{n},t\_{n+1}] $

\begin{equation}

\begin{split}

\int\_{t\_n}^{t\_{n+1}}\frac{g''(\eta\_s)}{2}(s-t\_{n-1})(s-t\_n)\,ds & = \frac{g''(\xi\_n)}{12}(t\_{n+1} - t\_n)^2(t\_n + 2t\_{n+1} - 3t\_{n-1})=\\ & = \frac{5}{12}h^3g''(\xi\_n)

\end{split}

\end{equation}

This indicates that the Adams-Bathforth has error of order 3, $O(h^3)$.

Starting from \ref{Integral\_ODE} and using $f$ instead of $g$ we get

\begin{equation}\label{AB\_method}

y\_{n+1} = y\_n + \frac{1}{2}h(3f(t\_n, y\_n) - f(t\_{n-1}, y\_{n-1})) \tx{and} y(t\_0) = y\_0

\end{equation}

To start the recursion we also need $y\_1$. The simplest way to do this is to use Euler method to give the first value.

%---------------------------Runge-Kutta----------------------------------

\subsection{Runge-Kutta of fourth order (RK4)}\label{Runge\_kutta\_sec}

%https://lpsa.swarthmore.edu/NumInt/NumIntFourth.html

The Runge-Kutta method is a whole family of methods. I will present the Runge-Kutta method of forth degree. I will not however present the derivation and jump immediately to the analysis.

The method is best written as

\begin{align}\nonumber

k\_1 &= f(t\_n,y\_n)\\ \nonumber

k\_2 &= f(t\_n+\frac{1}{2}h,y\_n+k\_1\frac{h}{2})\\\nonumber

k\_3 &= f(t\_n+\frac{1}{2}h,y\_n+k\_2\frac{h}{2})\\\nonumber

k\_4 &= f(t\_n+h,y\_n+k\_3h)\\\nonumber

y\_{n+1} &= y\_n + \frac{h}{6}(k\_1 + 2k\_2 + 2k\_3 + k\_4)

\end{align}

For a more intuitive understanding let's understand what the various $k$'s mean in the RK4 method.

\begin{itemize}

\item $k\_1$ is the slope of the start of time step $n+1$ (equal to the slope at the end of time step $n$)

\item Following the slope give by $k\_1$ but only taking half a step, we then recalculate the slope at this midpoint and we get $k\_2$

\item Now if we go back to the start of the time step and take half a step following the slope given by $k\_2$ and recalculate agian the slope at this point we get $k\_3$

\item Lastly we take a complete step following the slope given by $k\_3$ and the slope at that final point is exactly $k\_4$

\item Taking inspiration from Simpson's method for approximating integrals we get the last equation.

\end{itemize}

The last point justifies the fact that this method, like Simpson's method in section \ref{Simp\_section}, has an error of fourth order $O(h^4)$.\\

The last step can also be seen as a weighted average of the slopes that we calculated in the RK4 method. With this perspective we notice that the slope at the middle points denoted $k\_2$ and $k\_3$ have a higher weight compared to the other two slopes. If we go back to the derivation of Simpson's method we can see why this would be the case. When we interpolate the function $f$ by construction the interpolant agrees exactly with the function $f$ at the midpoint. Seeing as the both the function and polynomial will have to pass through the extremes of the interval, it wouldn't be so far fetched to say that the slopes of the two function are similar at the midpoint. Furthermore, at the extremes even though the function start at the same point, they may have very different slopes as in the interpolation we do not consider any information we have have outside the interval.

%---------------------------Runge-Kutta example----------------------------

\subsubsection{Example of Runge-Kutta of fourth order}

As an example take the following initial value problem

\begin{equation}\nonumber

\begin{cases}

y'(t) = -2y\\

y(0) = 3

\end{cases}

\end{equation}

We can easily see that the exact solution is $y(t) = 3e^{-2t}$. By varying the values of $h$ we get the following graph

\vspace{-2.5mm}

\begin{center}

\includegraphics[width=0.8\textwidth]{RK4 example.png}

\end{center}

%Adavnateges and disadavntages: https://testbook.com/maths/runge-kutta-4th-order

%---------------------------Stability analysis LMM-----------------------------

\section{Stability Analysis of Linear Multistep methods}

A general r-step linear multistep method is of the form

\begin{equation}\label{LMM}

\sum\_{j=0}^r\alpha\_jy\_{n+j} = h\sum\_{j=0}^r\beta\_jf(t\_{n+j}, y\_{n+j})

\end{equation}

Notice that the trapezoidal method and the second order Adams-Bashforth method are both particular cases of this general equation. To get the trapezoidal method in equation \ref{trap\_method\_for\_ode} we choose

\begin{equation}\label{trap\_parameters}

r=1,\quad \alpha\_0 = -1,\quad \alpha\_1 = 1 \tx{and} \beta\_0 = \beta\_1 = \frac{1}{2}

\end{equation}

To get Adams-Bashforth seen in equation \ref{AB\_method}, with a slight change of indexing, we must set

\begin{equation}\label{AB\_parameters}

r=2,\quad \alpha\_0 = 0,\quad \alpha\_1 = -1, \quad \alpha\_2 = 1, \quad \beta\_0 =-\frac{1}{2}, \quad \beta\_1 = \frac{3}{2} \tx{and} \beta\_2 = 0

\end{equation}

%---------------------------local trunc error------------------------------

\subsection{Local truncation error}

The local truncation error for linear multistep methods is defined as the following

\begin{equation}

\tau(t\_{n+r}) = \frac{1}{h}\left(\sum\_{j=0}^r\alpha\_jy(t\_{n+j}) - h\sum\_{j=0}^r\beta\_jy'(t\_{n+j})\right)

\end{equation}

Assuming that $y$ is smooth enough we can take the Taylor series of $y$ and $y'$

\begin{align}

y(t\_{n+j}) & = y(t\_n) + jky'(t\_n) + \frac{1}{2}(jk)^2y''(t\_n)+...\\

y'(t\_{n+j}) & = y'(t\_n) + jky''(t\_n) + \frac{1}{2}(jk)^2y'''(t\_n)+...

\end{align}

Substituting this back into the local truncation error gives

\begin{align}

\tau(t\_{n+r}) & = \frac{1}{h}\left(\sum\_{j=0}^r\alpha\_j\right)y(t\_n) + \left(\sum\_{j=0}^r(j\alpha\_j-\beta\_j)\right)y'(t\_n) + \\&+ h\left(\sum\_{j=0}^r\left(\frac{1}{2}j^2\alpha\_j - j\beta\_j\right)\right)y''(t\_n) + o(h)

\end{align}

We call a numerical method \textit{consistent} if $\tau \to 0$ as $h \to 0$. For this condition to hold we must have that

\begin{equation}

\sum\_{j=0}^r\alpha\_j=0 \tx{and} \sum\_{j=0}^rj\alpha\_j=\sum\_{j=0}^r\beta\_j

\end{equation}

An important note is that these conditions only depend on the method and not the actual IVP we are solving.

%---------------------------Char polynomial--------------------------------

\subsection{Characteristic polynomials}

For linear multistep methods it is useful to define the two following polynomials called characteristic polynomials.

\begin{equation}

\rho(x):=\sum\_{j=0}^r\alpha\_jx^j \tx{and} \sigma(x):=\sum\_{j=0}^r\beta\_jx^j

\end{equation}

With this in mind we can rewrite the the consistency conditions as

\begin{equation}\label{consistent\_conditions}

\rho(1)=0 \tx{and} \rho'(1)=\sigma(1)

\end{equation}

%---------------------------zero-stability---------------------------------

\subsection{Zero-stability of a linear multistep method}\label{zero\_stab\_section}

The next step is introducing the notion of \textit{zero-stability}. This is one of many different notions of stability for numerical methods. It is very simple but nonetheless very impactful thanks to theorem \ref{convergence\_with\_zero\_stability}.

To get some intuition of what zero-stability means, let's look at the simplest example of ODE: $y'(t) = 0$ with $y(0)=0$. An obvious solution of this is $y(t) = 0$, but you can be surprised as too how many methods fail to reach this solution given a small initialization error. The next examples illustrates this very well.

Take the following linear multistep method trying to solve $y'(t) = 0$ with $y(0)=0$.

\begin{equation}\label{bad\_multistep}

y\_{n+2} - 3y\_{n+1} + 2y\_n = 0, \quad \quad y\_0 = 0

\end{equation}

While there is so real justification why someone would use this method, it seems like a perfectly reasonable 2-step linear multistep method. To start the recurrence we need to estimate $y\_1$. Assume that in doing so we get that $y\_1 = \epsilon > 0$. With methods I will present soon, one can solve \ref{bad\_multistep} for $y\_n$ starting from $y\_1$ and $y\_0$ giving

\begin{equation}

y\_n = 2y\_0 - y\_1 + 2^n(y\_1 - y\_0)

\end{equation}

Under our assumption of $y\_1 = \epsilon$ we are left with

\begin{equation}

y\_n = (2^n - 1)\epsilon

\end{equation}

Which obviously diverges as $n \to \infty$.\\

Why did this happen? Well in general it comes down to solving the linear equation

\begin{equation}\label{difference\_equation}

\sum\_{j=0}^r\alpha\_jy\_{n+j} = 0

\end{equation}

As $y\_n$ depends on the previous $y\_i$s a reasonable guess for a solution would be

\begin{equation}

y\_n = k^n

\end{equation}

For some value of $k$. For this to happen $k$ must satisfy

\begin{equation}

\sum\_{j=0}^r\alpha\_jk^{n+j} = 0

\end{equation}

which is exactly like saying that $k$ must be a root of $\rho(x)$. Whatsmore, by the fundamental theorem of algebra $\rho(x)$ must have $r$ roots and as \ref{difference\_equation} is linear the general solution is

\begin{equation}

y\_n = \sum\_{j=1}^rc\_jx\_j^{n}

\end{equation}

where $x\_j$ for $j = 1,...,r$ are the roots of $\rho(x)$. To solve for the coefficients we must use the initial conditions, which we must have $r$ of to start the recurrence, to solve a $r x r$ system of equations.

If we want $y\_n$ to not diverge as $n \to \infty$ we must have that $|x\_j| \leq 1$ for all the roots. There is a slight catch if the roots repeat as the system to solve for the initial conditions becomes singular and so we must impose that $|x\_j| < 1$ if $x\_j$ is a repeated root.

With this in mind we can define zero-stability.

\begin{df}[zero-stability]

A r-step linear multistep method is said to be zero-stable if the roots of the characteristic polynomial $\rho(x)$ denoted $x\_1, ... , x\_r$ satisfy the following conditions:

\begin{align}

&|x\_j| \leq 1 \tx{for} j=1,...,r\\

& \text{If} \,x\_j \,\text{is a repeated root, then} \,|x\_j| < 1

\end{align}

\end{df}

Following from my discussion above, we notice that zero-stability is essentially guaranteeing stability when we send $h \to 0$.

This important definition as it leads to the following theorem.

\begin{thm}\label{convergence\_with\_zero\_stability}

Given a linear multistep method used to solve an initial value problem as in \ref{IVP} we have that:

\begin{equation}

\text{consistency} \quad + \quad\text{zero stability} \quad \iff \quad\text{convergence}

\end{equation}

\end{thm}

%---------------------------Euler methods-------------------------------

\subsection{Particular case: Euler Methods}

The explicit and implicit Euler methods are one of the simplest linear multistep methods there is.\\

\textbf{Explicit Euler}: The relation of the explicit Euler method is as follows

\begin{equation}

y\_{n+1} = y\_n + h f(t\_n, y\_n), \quad y\_0 = y(t\_0)

\end{equation}

This formula gives rise to the following characteristic polynomials

\begin{equation}\label{char\_poly\_exp\_ee}

\rho\_{EE}(x) = -1 + x \tx{and} \sigma\_{EE}(x) = 1

\end{equation}

Where $\rho\_{EE}(x)$ has a single root namely $x\_1=1$. As the conditions \ref{consistent\_conditions} are satisfied we conclude that the explicit Euler method is zero-stable and consistent which, by the previous theorem, is equivalent to being convergent.\\

\textbf{Implicit Euler}: The implicit Euler method is similar to the previous with a slight variation

\begin{equation}

y\_{n+1} = y\_n + h f(t\_{n+1}, y\_{n+1}), \quad y\_0 = y(t\_0)

\end{equation}

Which has the following characteristic polynomials

\begin{equation}\label{char\_poly\_exp\_ie}

\rho\_{IE}(x) = -1 + x \tx{and} \sigma\_{IE}(x) = x

\end{equation}

Where $\rho\_{IE}(x)$ has a single root $x\_1=1$. Similarly to the explicit Euler method also the implicit Euler is zero-stable and consistent, which is to say convergent.

%---------------------------trap method case-------------------------------

\subsection{Particular case: Trapezoidal method for IVPs}

From \ref{trap\_parameters} we clearly see that the trapezoidal method is a consistent method. Now to check if the method is zero-stable let's first find it's characteristic polynomials.

\begin{equation}\label{char\_poly\_trap}

\rho\_{T}(x) = -1 + x \tx{and} \sigma\_T(x) = \frac{1}{2} + \frac{1}{2}x

\end{equation}

Where $\rho\_{T}(x) = \rho\_{EE}(x)$ and has root $x\_1=1$. With these characteristic polynomials in mind we can easily state that the trapezoidal method is both zero-stable and consistent, and so convergent.

%---------------------------Ab case----------------------------------

\subsection{Particular case: Adams-Bashforth}

Similarly to the trapezoidal method, also Adams-Bashforth is consistent. Furthermore, it's polynomial is

\begin{equation}

\rho\_{AB}(x) = -x + x^2 \tx{and} \sigma\_{AB}(x) = -\frac{1}{2} + \frac{3}{2}x

\end{equation}

Since $\rho\_{AB}(x)$ has two roots $x\_1 = 0$ and $x\_2 = 1$ we conclude that also this method is convergent.\\

Zero-stability however is not the only kind of stability we care about. In fact often this stability is not useful (example of this ????). These condition basically tell us that "eventually" for $h$ small enough we will reach the exact answer. Say, however, we are looking for an arbitrary precision to the real solution. How small does $h $ have to be to ensure that precision? Actually, it can also happen that for values of $h$ not small enough the error is extremely large. Essentially there are no guarantees that a given $h$ will give a small error. To dive deeper into stability analysis we will introduce absolute stability.

%---------------------------Abs stability-------------------------------

\subsection{Absolute stability for multistep methods}

This notion of stability focuses on a specific linear example of ODE. We will call this the test equation.

\begin{equation} \label{test\_ODE}

y'(t) = \lambda y(t)

\end{equation}

While this equation seems extremely simple, by varying $\lambda$ we can really put to the test many of the methods we have seen. If we take one step further we can even take $\lambda$ to be complex allowing us to visualize regions of absolute stability in the complex plane. Why do we care so much about equation \ref{test\_ODE} who has a very simple solution $Ce^{\lambda t}$? Well the reason for this is that this is the simplest example of whats called a stiff differential equation. A stiff differential equation causes many numerical method to behave unstably if $h$ is not chosen small enough. If, for example, we take $\lambda$ to be very large, then the step the numerical method makes is multiplied by $\lambda$ and might lead to the numerical method overshooting the real value. So the numerical method will have a hard time converging to the correct result. We see however, that this can be mitigated by choosing $h$ small enough so to counteract the impact $\lambda$ has on the method.

For example applying the explicit Euler method to \ref{test\_ODE} gives

\begin{equation}\label{euler\_abs\_stab}

y\_{n+1} = (1+h\lambda)y\_n

\end{equation}

For the explicit Euler method to to absolutely stable we must have that $|1+h\lambda|\leq 1$. Note that while $\lambda$ is complex we still take $h$ to be real. It is evident that the important aspect here is the product of $h$ and $\lambda$. This is perfectly inline with what we discussed above since taking $h$ small can reduce the effect of $\lambda$. So the important aspect to monitor is the product $\lambda h$ and it is precisely this that we are going to plot to find what is called the stability regions of the numerical methods.\\

In the explicit Euler case we get a circle with radius 1 and center $(-1,0)$.

%---------------------------Abs stability for LMM-------------------------------

\subsubsection{Absolute stability for general multistep methods}

Using \ref{LMM} to solve our test problem and rearranging we obtain

\begin{equation}\label{difference\_equation\_abs\_stab}

\sum\_{j=0}^r(\alpha\_j - h \lambda\beta\_j) y\_{n+j} = 0

\end{equation}

Similar to the previous example, the important parameter to consider is the product of $h$ and $\lambda$. Setting $z=h\lambda$ we define the stability polynomial defined as

\begin{equation}

\pi(x;z)=\rho(x)-z\sigma(x)

\end{equation}

Where $\pi(x;z)$ is a polynomial in $x$ whose coefficients depend on $z$. The essential point in this argument is that equation \ref{difference\_equation\_abs\_stab} is identical to equation \ref{difference\_equation} with slightly different coefficients. Using the exact argument we did in section \ref{zero\_stab\_section} with $\pi(x;z)$ instead of $\rho(x)$ we get another kind of stability: \textit{Absolute stability}.

\begin{df}[Absolute stability]

A r-step linear multistep method is said to be absolutely stable if the roots of the stability polynomial $\pi(x;z)$ denoted $x\_1, ... , x\_r$ satisfy the following conditions:

\begin{align}

&|x\_j| \leq 1 \tx{for} j=1,...,r\\

& \text{If} \,x\_j \,\text{is a repeated root, then} \,|x\_j| < 1

\end{align}

\end{df}

I will not be presenting the formal reasoning to understand why the definitions of stability have this. The intuition is that an error in the current time step does not grow in future time steps.

%---------------------------euler method case-------------------------------

As i hinted to before also Euler methods fall under the umbrella of linear multistep method allowing all the theory we have discussed to be applied to them as well. For examples we can plot the stability regions.

\begin{minipage}[b]{.5\textwidth}

\includegraphics[width=\textwidth]{Stab region of Explicit Euler.png}

\end{minipage}%

\begin{minipage}[b]{.5\textwidth}

\includegraphics[width=\textwidth]{Stab region of Implicit Euler.png}

\end{minipage}%

\\

%---------------------------trap case stab-------------------------------

\subsection{Particular case: Trapezoidal method for IVPs}

From the characteristic polynomials found in \ref{char\_poly\_trap} we have that the stability polynomial

\begin{equation}

\pi\_T(x;z) = -\left(1+\frac{1}{2}z\right) + \left(1-\frac{1}{2}z\right)x

\end{equation}

has a single root, which is

\begin{equation}

x\_1 = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}

\end{equation}

and taking the absolute value we get

\begin{equation}

|x\_1| \leq 1 \iff \left|1 + \frac{1}{2}z \right| \leq \left|1-\frac{1}{2}z \right|

\end{equation}

which is like saying all the complex numbers $z$ that are closer to $-1$ than to $1$, giving us the left half plane as below.

\vspace{-2.5mm}

\begin{center}

\includegraphics[width=0.5\textwidth]{Stab region of Trapezoidal Method.png}

\end{center}

%---------------------------AB case stab-------------------------------

\subsection{Particular case: Adams-Bashforth}

The stability polynomial of the Adams-Bashforth method of order 2 is

\begin{equation}

\pi\_{AB}(x;z) = \frac{1}{2}z + \left(-\frac{3}{2}z-1\right)x + x^2

\end{equation}

In this case checking the root condition may become complicated, and more so for higher orders. In this case we can resort to the computer programs which display the region of stability of the various methods. Thanks to the construction of the stability polynomial it is very easy to generalize the program to any linear multistep method. Below we the stability regions for the Adams-Bashforth methods up to order 5.

\\

\begin{minipage}[b]{.5\textwidth}

\includegraphics[width=\textwidth]{Stab region of AB2.png}

\end{minipage}%

\begin{minipage}[b]{.5\textwidth}

\includegraphics[width=\textwidth]{Stab region of AB3.png}

\end{minipage}%

\\

\begin{minipage}[b]{.5\textwidth}

\includegraphics[width=\textwidth]{Stab region of AB4.png}

\end{minipage}%

\begin{minipage}[b]{.5\textwidth}

\includegraphics[width=\textwidth]{Stab region of AB5.png}

\end{minipage}%

\\

\section{Stability Analysis of Runge-Kutta of fourth order}

In the previous section we managed to analysis the stability of a general linear multistep method. With Runge-Kutta methods, unfortunetly, it is much hard to make a general stability analysis. For this reason in this section I will be focusing on the stability of Runge-Kutta of fourth order as defined in section \ref{Runge\_kutta\_sec}.\\

In the footsteps of absolute stability for multistep methods, we will also apply RK4 to the differentail equation defined as

\begin{equation}\label{test\_ODE\_abs\_stab}

y'(t) = \lambda y(t)

\end{equation}

By direct computation we get the following values for $k\_1$, $k\_2$, $k\_3$, and $k\_4$.

\begin{align}

k\_1 &= \lambda y\_n\\ \nonumber

k\_2 &= y\_n \left(\lambda + \frac{\lambda^2 h}{2} \right)\\ \nonumber

k\_3 &= y\_n \left(\lambda + \frac{\lambda^2 h}{2} + \frac{\lambda^3 h^2}{4} \right)\\ \nonumber

k\_4 &= y\_n \left(\lambda + \lambda^2 h + \frac{\lambda^3 h^2}{2} + \frac{\lambda^4 h^3}{4} \right)\\ \nonumber

\end{align}

Putting everything together we get the value of $y\_{n+1}$.

\begin{equation}

y\_{n+1} = y\_n \left[1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24} \right]

\end{equation}

The spectacular part about this is that the multiplicative coefficient of $y\_n$ is exactly the fourth order taylor expansion of $e^{\lambda h}$ which, up to multiplicative constant, is the correct solution to \ref{test\_ODE\_abs\_stab}. \\

Using $z = \lambda h $ as above let us define

\begin{equation}

R(z) := 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!}

\end{equation}

Similarly to how to analysed the absolute stability of the explicit Euler method in equation \ref{euler\_abs\_stab}, for the RK4 method to be absolutely stable we need that the coefficient of $y\_n$ does not blow up to infinity, which is to say $|R(z)| \leq 1$. Plotting this inequality similarly to what we did with the other methods results in the following graph.

\vspace{-2.5mm}

\begin{center}

\includegraphics[width=0.5\textwidth]{Stab region of Trapezoidal Method.png}

\end{center}

\section{Concluding Remarks}

\section{What's left to do}

\begin{itemize}

\item More numerical example for methods like trapezoid and Adams-Bashforth

\item Compare various methods and discuss pros and cons of each one

\item discuss the down side of using methods of higher orders

\item stiff with y'= ay

\item why not higher order?

\item use the complex plane with a

\item examples when assumptions are not met

\item examples when methods break

\item does interpolation error go to 0?

\item https://en.wikipedia.org/wiki/Runge%27s\_phenomenon

\item https://webspace.science.uu.nl/~frank011/Classes/numwisk/ch10.pdf

\end{itemize}

\end{document}

%-----------------------------Extra code------------------------

%Linear interpolation error

% \begin{thm}[Linear Interpolation error]

% Let $f \in C^2([a,b])$ and let $p(x)$ be the linear polynomial that interpolates $f$ at $a$ and $b$. Then, for all $x \in [a,b]$ there exists $\eta\_{x} \in [a,b]$ such that

% \begin{equation}

% f(x) - p(x) = \frac{1}{2}(x-a)(x-b)f''(\eta\_{x})

% \end{equation}

% Furthermore,

% \begin{equation}

% |f(x) - p(x)| \leq \frac{1}{8}(b - a)^2 \max\_{x \in [a,b]}|f''(x)|

% \end{equation}

% \end{thm}

% \begin{proof}

% The proof of this theorem boils down to applying Rolle's Theorem in a clear way. The proof can be found in the book (JE 2013 book).

% \end{proof}

%Example of how to estimate asymptoic behaviour of ometa(x)

% \subsection{Example of asymptotic behaviour of error bound}

% As an example assume we have a uniform grid of n points with grid size $h = \frac{b-a}{n}$. As the (n+1)-th derivative depends on the function we are tackling there is little we can say with certainty. The only real factor we can play with is $\omega(x)$. We notice that $|\omega(x)|$ is largest if $x$ is in one of the end points. This is where the product of the distances is maximized. So if $x \in [x\_0, x\_1]$ then

% \begin{equation}

% |x-x\_j| \leq (j+1)h \tx{for} j \in \{0,...,n\} \nonumber

% \end{equation}

% Taking the product give an upper bound for $\omega(x)$

% \begin{equation}

% |\omega(x)| \leq \prod\_{j=0}^{n}(|x-x\_j|) \leq (n+1)!h^{n+1} \quad \text{for} \quad x \in [a,b]\nonumber

% \end{equation}

% With Stirling's approximation for $n!$

% \begin{equation}

% n! \sim e^{-n}n^n\sqrt{2\pi n } \tx{as} n \to +\infty \nonumber

% \end{equation}h

%for continue: https://services.math.duke.edu/~jtwong/math563-2020/lectures/Lec1-polyinterp.pdf pp:11