MA385/MA530 – Numerical Analysis I ("NA1")

Niall Madden

September 29, 2019

Chapter 0

MA385: Preliminaries

0.1 Introduction

0.1.1 Welcome to MA385/MA530 ("NA1")

This is a Semester 1, upper level module on *numerical analysis*. It is often taken in conjunction with MA378 ("NA2").

Lecturer: Dr Niall Madden (he/him), School of Maths.
My office is in room ADB-1013, Arás de Brún.
Email: Niall.Madden@NUIGalway.ie

Lectures: Monday at 9 and Thursday at 3, in AC201.

Tutorial/Lab: TBA (to begin during Week 3)

Assessment:
- Two written assignments (20%)
- Three computer labs (10%)
- One in-class test in Week 6 [tbc] (10%)
- Two-hour exam at the end of semester (60%)

You may be taking this module (MA385/MA530) as part of your degree in Mathematics, Applied Mathematics, Mathematical Science, Computer Science, Financial Mathematics, or as a visiting student or graduate student.

The main text-book is **Süli and Mayers, An Introduction to Numerical Analysis**, Cambridge University Press [1]. The library has eight hard copies (519.4 MAY), *and* the electronic edition. It is very well suited to this course: it has a rigorous, but not over-complicated approach, and has a good selection of interesting problems. The scope of the book is almost perfect for the course, especially for those students taking both semesters. *Please consider buying this book.*

Other useful books include

- G.W. Stewart, *Afternotes on Numerical Analysis*, SIAM [3]. The library has hard and electronic copies. This book is very readable, and suited to students who would enjoy a bit more discussion.
- Cleve Moler, *Numerical Computing with MATLAB* [2]. The emphasis is on the implementation of algorithms in MATLAB, but the techniques are well explained and there are some nice exercises. Also, it is freely available online.
- Anne Greenbaum and Timothy Chartier, *Numerical methods: design, analysis, and computer implementation of algorithms*. Well motivated, and makes great use of the wonderful Chebfun package for MATLAB.
- James F Epperson, *An Introduction to Numerical Methods and Analysis*,[5]. There are five copies in the library at 519.4. It is particularly good a motivating *why* we study particular numerical methods.
- Michelle Schatzman, *Numerical Analysis*,[7].
- Stoer and Bulirsch,*Introduction to Numerical Analysis* [6]. A very complete reference for this course.

Web site The on-line content for the course will be hosted at http://www.maths.nuigalway.ie/MA385. There you'll find various pieces of these notes, copies of slides, problem sets, and lab sheets,

We will also use the MA385 module on BlackBoard for announcements, emails, and the Grade Book. If you are registered for MA385, you should be automatically enrolled onto the blackboard site. If you are enrolled in MA530, please send an email to me.

These notes are a synopsis of the course material. My aim is to provide these in three main sections, and always in advance of the class. They contain most of the main remarks, statements of theorems, results and **exercises**. However, they do not contain proofs of theorems, examples, solutions to exercises, etc.

You should bring these notes to class. It will make following the lecture easier, and you'll know what notes to take down.

Please help! *These notes contain numerous typos and errors. None are deliberate. Please help me, and the rest of the class, by pointing them out so I can fix them.*

0.1.2 What is Numerical Analysis?

Numerical analysis is the design, analysis and implementation of numerical methods that yield *exact* or *approximate* solutions to mathematical problems.

It does not involve long, tedious calculations. We won't (usually) implement Newton's Method by hand, or manually do the arithmetic of Gaussian Elimination, etc.

The *Design* of a numerical method is perhaps the most interesting; it's often about finding a clever way of swapping the problem for one that is easier to solve, but has the same or similar solution. If the two problems have the same solution, then the method is *exact*. If they are similar (but not the same), then it is *approximate*.

The *Analysis* is the mathematical part; it usually culminates in proving a theorem that tells us (at least) one of the following

- The method will work: that our algorithm will yield the solution we are looking for;
- how much effort is required;
- if the method is approximate, determine how close the approximate solution is to the real one. A description of this aspect of the course, to quote Epperson [5], is being "*rigorously imprecise* or *approximately precise*".

The *implementation* is generating solutions with the algorithms, that is, it is the programming part. We'll study this in labs.

If you would like a more thorough discussion on the nature and history of numerical analysis, read Nick Trefethen's essay on Numerical Analysis from the Princeton Companion to Mathematics, 2008.

Topics

0. We'll preface the course with a review of Taylor's theorem. It is central to the algorithms of the following sections.
1. Root-finding and solving non-linear equations.
2. Initial value ordinary differential equations.
3. Matrix Algorithms: solving systems of linear equations and estimating eigenvalues.

We also see how these methods can be applied to so-called "real world" problems, including Financial Mathematics.

Learning outcomes

When you have successfully completed this course, you will be able to demonstrate your factual knowledge of the core topics (root-finding, solving ODEs, solving linear systems, estimating eigenvalues), using appropriate mathematical syntax and terminology.

Moreover, you will be able to describe the fundamental principles of the concepts (e.g., Taylor's Theorem) underpinning Numerical Analysis. Then, you will apply these principles to design algorithms for solving mathematical problems, and discover the properties of these algorithms. course to solve problems.

You will learn how to use MATLAB to implement these algorithms, and adapt the codes for more general problems, and for different techniques.

Mathematical Preliminaries

Anyone who can remember their first and second years of analysis and algebra should be well prepared for this module. Students who know a little about initial value differential equations will find Section 2 somewhat easier than those who haven't.

If it has been a while since you studied basic calculus, you will find it very helpful to revise the following: the Intermediate Value Theorem; Rolle's Theorem; The Mean Value Theorem; **Taylor's Theorem**, and the triangle inequality: $|a+b| \leqslant |a| + |b|$. You'll find them in any good text book, e.g., Appendix 1 of Süli and Mayers.

You'll also find it helpful to recall some basic linear algebra, particularly relating to eigenvalues and eigenvectors. Consider the statement: "all the eigenvalues of a real symmetric matrix are real". If are unsure what the meaning of any of the terms used, or if you didn't know that it's true, you should have a look at a book on Linear Algebra.

0.1.3 Why take this course?

Many industry and academic environments require graduates who can solve real-world problems using a mathematical model, but these models can often only be resolved using numerical methods. To quote one Financial Engineer: "We prefer approximate (numerical) solutions to exact models rather than exact solutions to simplified models".

Another expert, who leads a group in fund management with DB London, when asked "what sort of graduates would you hire", the list of specific skills included

- A programming language and a 4th-generation language such as MATLAB (or R).
- Numerical Analysis

Graduates of our Financial Mathematics, Computing and Mathematics degrees often report to us that they were hired because that had some numerical analysis background, or were required to go and learn some before they could do some proper work. This is particularly true in the financial sector, games development, and mathematics civil services (e.g., MET office, CSO).

Bibliography

[1] E Süli and D Mayers, *An Introduction to Numerical Analysis*, 2003. 519.4 MAY.

[2] Cleve Moler, *Numerical Computing with MATLAB*, Cambridge University Press. Also available free from http://www.mathworks.com/moler

[3] G.W. Stewart, *Afternotes on Numerical Analysis*, SIAM, 1996. 519.4 STE.

[4] G.W. Stewart, *Afternotes goes to Graduate School*, SIAM, 1998. 519.4 STE.

[5] James F Epperson, *An introduction to numerical methods and analysis*. 519.4EPP

[6] Stoer and Bulirsch, *An Introduction to Numerical Analysis*, Springer.

[7] Michelle Schatzman, *Numerical Analysis: a mathematical introduction*, 515 SCH.

[8] Anne Greenbaum and Timothy P. Chartier, *Numerical Methods : Design, Analysis, and Computer Implementation of Algorithms*. Princeton University Press, 2012.

0.2 Taylor's Theorem

Taylor's theorem [1] is perhaps the most important mathematical tool in Numerical Analysis. Providing we can evaluate the derivatives of a given function at some point, it gives us a way of approximating the function by a polynomial.

Working with polynomials, particularly ones of degree 3 or less, is much easier than working with arbitrary functions. For example, polynomials are easy to differentiate and integrate. Most importantly for the next section of this course, their zeros are easy to find.

Our study of Taylor's theorem starts with one of the first theoretical results you learned in university mathematics: *the mean value theorem.*

. . . .

Theorem 0.2.1 (Mean Value Theorem)**.** *If* f *is function that is continuous and differentiable for all* $a \leqslant x \leqslant b$*, then there is a point* $c \in [a, b]$ *such that*

$$\frac{f(b) - f(a)}{b - a} = f'(c).$$

This is just a consequence of Rolle's Theorem, and has few different interpretations. One is that the slope of the line that intersects f at the points a and b is equal to the slope of the tangent to f at some point between a and b.

<u>Take notes</u>:

There are many important consequences of the MVT, some of which we'll return to later. Right now, we are interested in the fact that the MVT tells us that we can approximate the value of a function by a near-by value, with accuracy that depends on f':

<u>Take notes</u>:

---

[1] Brook Taylor, 1865 – 1731, England. He (re)discovered this polynomial approximation in 1712, though its full importance was not realised for another 50 years.

Or we can think of it as approximating f by a line:
`Take notes`:

....

But what if we want a better approximation? We could replace our function with, say, a quadratic polynomial. Let $p_2(x) = b_0 + b_1(x-a) + b_2(x-a)^2$ and solve for the coefficients b_0, b_1 and b_2 so that

$$p_2(a) = f(a), \quad p_2'(a) = f'(a), \quad p_2''(a) = f''(a).$$

`Take notes`:

This gives that

$$p_2(x) = f(a) + f'(a)(x-a) + (x-a)^2\frac{f''(a)}{2}.$$

Next, if we try to construct an approximating cubic of the form

$$p_3(x) = b_0 + b_1(x-a) + b_2(x-a)^2 + b_3(x-a)^3, \\ = \sum_{k=0}^{3} b_k(x-a)^k,$$

with the property that

$$\begin{aligned} p_3(a) &= f(a), & p_3'(a) &= f'(a), \\ p_3''(a) &= f''(a), & p_3'''(a) &= f'''(a). \end{aligned} \tag{0.2.1}$$

Note: we can write (0.2.1) in a more succinct way, using the mathematical short-hand:

$$p_3^{(k)}(a) = f^{(k)}(a) \text{ for } k = 0, 1, 2, 3.$$

Again we find that

$$b_k = \frac{f^{(k)}(a)}{k!} \text{ for } k = 0, 1, 2, 3.$$

As you can probably guess, this formula can be easily extended for arbitrary k, giving us the *Taylor Polynomial*.

Definition 0.2.2 (Taylor Polynomial). The *Taylor Polynomial* of degree k (also called the *Truncated Taylor Series*) that approximates the function f about the point $x = a$ is

$$p_k(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''(a) + \\ \frac{(x-a)^3}{3!}f'''(a) + \cdots + \frac{(x-a)^k}{k!}f^{(k)}(a).$$

We'll return to this topic later, with a particular emphasis on quantifying the "error" in the Taylor Polynomial.

Example 0.2.3. Write down the Taylor polynomial of degree k that approximates $f(x) = e^x$ about the point $x = 0$.

`Take notes`:

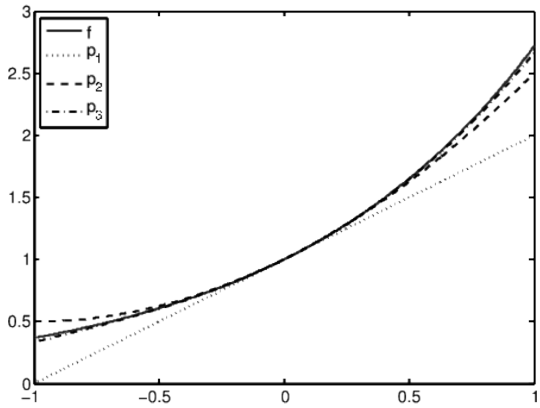

Fig. 0.1: Taylor polynomials for $f(x) = e^x$ about $x = 0$

As Figure 0.1 suggests, in this case $p_3(x)$ is a more accurate estimation of e^x than $p_2(x)$, which is more accurate than $p_1(x)$. This is demonstrated in Figure 0.2 were it is shown the difference between f and p_k.

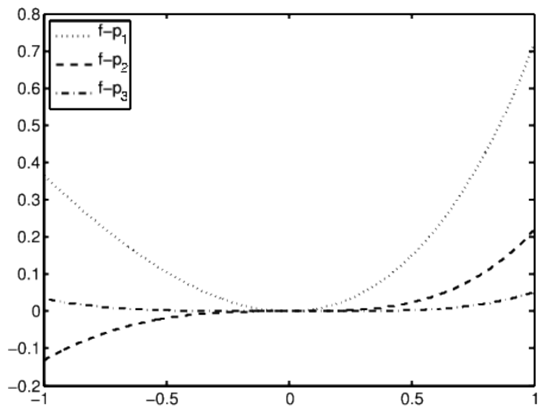

Fig. 0.2: Error in Taylor polys for $f(x) = e^x$ about $x = 0$

0.2.1 The Remainder

We now want to examine the *accuracy* of the Taylor polynomial as an approximation. In particular, we would like to find a formula for the *remainder* or *error*:

$$R_k(x) := f(x) - p_k(x).$$

With a little bit of effort one can prove that:

$$R_k(x) := \frac{(x-a)^{k+1}}{(k+1)!} f^{(k+1)}(\sigma), \text{ for some } \sigma \in [x, a].$$

We won't prove this in class, since it is quite standard and features in other courses you have taken. But for the sake of completeness, a proof is included below in Section 0.2.4 below.

Example 0.2.4. With $f(x) = e^x$ and $a = 0$, we get that

$$R_k(x) = \frac{x^{k+1}}{(k+1)!} e^{\sigma}, \text{ some } \sigma \in [0, x].$$

Example 0.2.5. How many terms are required in the Taylor Polynomial for e^x about $a = 0$ to ensure that the error at $x = 1$ is

- no more than 10^{-1}?
- no more than 10^{-2}?
- no more than 10^{-6}?
- no more than 10^{-10}?

Take notes:

0.2.2 An application of Taylor's Theorem

The reasons for emphasising Taylor's theorem so early in this course are that

- It introduces us to the concept of approximation, and error estimation, but in a very simple setting;
- It is the basis for deriving methods for solving both nonlinear equations, and initial value ordinary differential equations.

With the last point in mind, we'll now outline how to derive Newton's method for nonlinear equations. (This is just a *taster*: we'll return to this topic in the next section).

Take notes:

0.2.3 Exercises

Exercise 0.1. Write down the formula for the Taylor Polynomial for

(i) $f(x) = 3x^2 + 3x - 12$

(ii) $f(x) = \sqrt{1+x}$ about the point $a = 0$,

(iii) $f(x) = \log(x)$ about the point $a = 1$.

Exercise 0.2. Write out the Taylor polynomial at x, about $a = 0$, of degree 7 for $f(x) = \sin(x)$. How does its derivative compare to the corresponding Taylor polynomial for $f(x) = \cos(x)$?

The purpose of the next exercise is to demonstrate that, usually, the closer x is to a, the better the Taylor polynomial approximates that function's value.

Exercise 0.3. Write out the Taylor Polynomial about $a = 1$ of degree 4 and corresponding remainder for $f(x) = \ln(x)$. Give an upper bound for this remainder when $x = 2$, $x = 1.1$ and $x = 1.01$.

The purpose of the next exercise is to demonstrate that some functions do not have sensible Taylor polynomials.

Exercise 0.4. Write out the Taylor polynomial about $a = 0$, of degree 4, for $f(x) = e^{-1/x^2}$.
Hint: $\lim_{x\to 0} e^{-1/x^2}x^{-p} = 0$ for any positive, finite p.

Exercise 0.5. Prove the *Integral Mean Value Theorem*: there exists a point $c \in [a, b]$ such that

$$f(c) = \frac{1}{b-a}\int_a^b f(x)dx.$$

0.2.4 A proof of Taylor's Theorem

Here is a proof of Taylor's theorem. It wasn't covered in class. One of the ingredients need is *Generalised Mean Value Theorem*: if the functions F and G are continuous and differentiable, etc, then, for some point $c \in [a, b]$,

$$\frac{F(b) - F(a)}{G(b) - G(a)} = \frac{F'(c)}{G'(c)}. \qquad (0.2.2)$$

Theorem 0.2.6 (Taylor's Theorem)**.** *Suppose we have a function* f *that is sufficiently differentiable on the interval* $[a, x]$*, and a Taylor polynomial for* f *about the point* $x = a$

$$p_n(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2}f''(a) + \frac{(x-a)^3}{3!}f''(a) + \dots \frac{(x-a)^k}{k!}f^{(n)}(a). \qquad (0.2.3)$$

If the remainder *is written as* $R_n(x) := f(x) - p_n(x)$*, then*

$$R_n(x) := \frac{(x-a)^{n+1}}{(n+1)!}f^{(n+1)}(\sigma), \qquad (0.2.4)$$

for some point $\sigma \in [x, a]$.

Proof. We want to prove that, for any $n = 0, 1, 2, \dots$, there is a point $\sigma \in [a, x]$ such that

$$f(x) = p_n(x) + R_n(x).$$

If $x = a$ then this is clearly the case because $f(a) = p_n(a)$ and $R_n(a) = 0$.

For the case $x \neq a$, we will use a *proof by induction*. The Mean Value Theorem tells us that there is some point $\sigma \in [a, x]$ such that

$$\frac{f(x) - f(a)}{x - a} = f'(\sigma).$$

Using that $p_0(a) = f(a)$ and that $R_0(x) = (x-a)f'(\sigma)$ we can rearrange to get

$$f(x) = p_0(a) + R_0(x),$$

as required.

Now we will *assume* that (0.2.3)–(0.2.4) are true for the case $n = k - 1$; and use this to show that they are true for $n = k$. From the Generalised Mean Value Theorem (0.2.2), there is some point c such that

$$\frac{R_k(x)}{(x-a)^{k+1}} = \frac{R_k(x) - R_k(a)}{(x-a)^{k+1} - (a-a)^{k+1}} = \frac{R_k'(c)}{(k+1)(c-a)^k},$$

where here we have used that $R_k(a) = 0$. Rearranging we see that we can write R_k in terms of its own derivative:

$$R_k(x) = R_k'(c)\frac{(x-a)^{k+1}}{(k+1)(c-a)^k}. \qquad (0.2.5)$$

So now we need an expression for R_k'. This is done by noting that it also happens to be the remainder for the Taylor polynomial of degree $k - 1$ for the function f'.

$$R_k'(x) = f'(x) - \frac{d}{dx}\Big(f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \dots + \frac{(x-a)^k}{k!}f^{(k)}(a)\Big).$$

$$R_k'(x) = f'(x) - \Big(f'(a) + (x-a)f''(a) + \dots + \frac{(x-a)^{k-1}}{(k-1)!}f^{(k)}(a)\Big).$$

But the expression on the last line of the above equation is the formula for the Taylor Polynomial of degree -1 for f'. By our inductive hypothesis:

$$R_k'(c) := \frac{(c-a)^k}{k!}f^{(k+1)}(\sigma),$$

for some σ. Substitute into (0.2.5) above and we are done. ☐

Chapter 1

Solving nonlinear equations

1.1 Bisection

1.1.1 Introduction

Linear equations are of the form:

$$\textit{find } x \textit{ such that } ax + b = 0$$

and are easy to solve. Some nonlinear problems are also easy to solve, e.g.,

$$\textit{find } x \textit{ such that } ax^2 + bx + c = 0.$$

Similarly, there are formulae for all cubic and quartic polynomial equations. But most equations to not have simple formulae for this soltuions, so numerical methods are needed.

References

- Süli and Mayers [1, Chapter 1]. We'll follow this pretty closely in lectures, but we'll do the sections in reverse order!
- Stewart (*Afternotes ...*), [3, Lectures 1–5]. A well-presented introduction, with lots of diagrams to give an intuitive introduction.
- Moler (Numerical Computing with MATLAB) [2, Chap. 4]. Gives a brief introduction to the methods we study, and a description of MATLAB functions for solving these problems.
- The proof of the convergence of Newton's Method is based on the presentation in [5, Thm 3.2].

Our generic problem is:

Let f *be a continuous function on the interval* $[a, b]$*. Find* $\tau \in [a, b]$ *such that* $f(\tau) = 0$*.*

Here f is some specified function, and τ is the *solution* to $f(x) = 0$.

This leads to two natural questions:

(1) How do we know there is a solution?

(2) How do we find it?

The following gives *sufficient* conditions for the existence of a solution:

Theorem 1.1.1. *Let* f *be a real-valued function that is defined and continuous on a bounded closed interval* $[a, b] \subset \mathbb{R}$*. Suppose that* $f(a)f(b) \leqslant 0$*. Then there exists* $\tau \in [a, b]$ *such that* $f(\tau) = 0$*.*

<u>`Take notes`</u>`:`

Now we know there is a solution τ to $f(x) = 0$, but how to we actually solve it? Usually we don't! Instead we construct a sequence of estimates $\{x_0, x_1, x_2, x_3, \dots\}$ that *converge* to the true solution. So now we have to answer these questions:

(1) How can we construct the sequence $x_0, x_1, \dots$?

(2) How do we show that $\lim_{k\to\infty} x_k = \tau$?

There are some subtleties here, particularly with part (2). What we would like to say is that at each step the error is getting smaller. That is

$$|\tau - x_k| < |\tau - x_{k-1}| \qquad \text{for } k = 1, 2, 3, \dots.$$

But we can't. Usually all we can say is that the *bounds* on the error is getting smaller. That is: *let* ε_k *be a bound on the error at step* k

$$|\tau - x_k| < \varepsilon_k,$$

then $\varepsilon_{k+1} < \mu\varepsilon_k$ for some number $\mu \in (0, 1)$. It is easiest to explain this in terms of an example, so we'll study the simplest method: *Bisection*.

1.1.2 Bisection

The most elementary algorithm is the "*Bisection Method*" (also known as "Interval Bisection"). Suppose that we know that f changes sign on the interval $[a, b] = [x_0, x_1]$ and, thus, $f(x) = 0$ has a solution, τ, in $[a, b]$. Proceed as follows

1. Set x_2 to be the midpoint of the interval $[x_0, x_1]$.
2. Choose one of the sub-intervals $[x_0, x_2]$ and $[x_2, x_1]$ where f change sign;
3. Repeat Steps 1–2 on that sub-interval, until f is sufficiently small at the end points of the interval.

This may be expressed more precisely using some *pseudocode*.

Method 1.1.2 (Bisection).
Set eps to be the stopping criterion.
If $|f(a)| \leqslant eps$, return a. Exit.
If $|f(b)| \leqslant eps$, return b. Exit.
Set $x_0 = a$ and $x_1 = b$.
Set $x_L = x_0$ and $x_R = x_1$.
Set $k = 1$
while($|f(x_k)| > eps$)
  $x_{k+1} = (x_L + x_R)/2$;
  if $(f(x_L)f(x_{k+1}) < eps)$
    $x_R = x_{k+1}$;
  else
    $x_L = x_{k+1}$
  end if;
  $k = k + 1$
end while;

Example 1.1.3. *Find an estimate for $\sqrt{2}$ that is correct to 6 decimal places.*
Solution: Try to solve the equation $f(x) := x^2-2 = 0$ on the interval $[0, 2]$. Then proceed as shown in Figure 1.1 and Table 1.1.

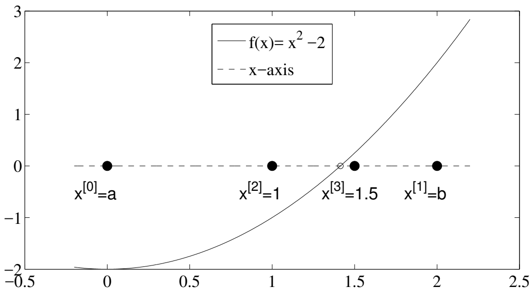

Fig. 1.1: Solving $x^2 - 2 = 0$ with the Bisection Method

Note that at steps 4 and 10 in Table 1.1 the error actually *increases*, although the bound on the error is decreasing.

k	x_k	$\lvert x_k - \tau\rvert$	$\lvert x_k - x_{k-1}\rvert$
0	0.000000	1.41	
1	2.000000	5.86e-01	
2	1.000000	4.14e-01	1.00
3	1.500000	8.58e-02	5.00e-01
4	1.250000	1.64e-01	2.50e-01
5	1.375000	3.92e-02	1.25e-01
6	1.437500	2.33e-02	6.25e-02
7	1.406250	7.96e-03	3.12e-02
8	1.421875	7.66e-03	1.56e-02
9	1.414062	1.51e-04	7.81e-03
10	1.417969	3.76e-03	3.91e-03
⋮	⋮	⋮	⋮
22	1.414214	5.72e-07	9.54e-07

Table 1.1: Solving $x^2 - 2 = 0$ with the Bisection Method

1.1.3 The bisection method works

The main advantages of the Bisection method are

- It will always work.
- After k steps we know that

  Theorem 1.1.4.
  $$|\tau - x_k| \leqslant \left(\frac{1}{2}\right)^{k-1}|b - a|, \quad \textit{for } k = 2, 3, 4, ...$$

Take notes:

A disadvantage of bisection is that it is not as efficient as some other methods we'll investigate later.

1.1.4 Improving upon bisection

The bisection method is not very efficient. Our next goals will be to derive better methods, particularly the *Secant Method* and *Newton's method*. We also have to come up with some way of expressing what we mean by "better"; and we'll have to use Taylor's theorem in our analyses.

1.1.5 Exercises

Exercise 1.1. Does Proposition 1.1.1 mean that, if there is a solution to $f(x) = 0$ in $[a, b]$ then $f(a)f(b) \leqslant 0$? That is, is $f(a)f(b) \leqslant 0$ a *necessary* condition for their being a solution to $f(x) = 0$? Give an example that supports your answer.

Exercise 1.2. Suppose we want to find $\tau \in [a, b]$ such that $f(\tau) = 0$ for some given f, a and b. Write down an estimate for the number of iterations K required by the bisection method to ensure that, for a given ε, we know $|x_k - \tau| \leqslant \varepsilon$ for all $k \geqslant K$. In particular, how does this estimate depend on f, a and b?

Exercise 1.3. How many (decimal) digits of accuracy are gained at each step of the bisection method? (If you prefer, how many steps are needed to gain a single (decimal) digit of accuracy?)

Exercise 1.4. Let $f(x) = e^x - 2x - 2$. Show that there is a solution to the problem: *find* $\tau \in [0, 2]$ *such that* $f(\tau) = 0$.
Taking $x_0 = 0$ and $x_1 = 2$, use 6 steps of the bisection method to estimate τ. You may use a computer program to do this, but please note that in your solution.
Give an upper bound for the error $|\tau - x_6|$.

Exercise 1.5. We wish to estimate $\tau = \sqrt[3]{4}$ numerically by solving $f(x) = 0$ in $[a, b]$ for some suitably chosen f, a and b.

(i) Suggest suitable choices of f, a, and b for this problem.

(ii) Show that f has a zero in $[a, b]$.

(iii) Use 6 steps of the bisection method to estimate $\sqrt[3]{4}$. You may use a computer program to do this, but please note that in your solution.

(iv) Use Theorem 1.1.4 to give an upper bound for the error $|\tau - x_6|$.