MATH 609 LECTURE NOTES

ANDREA BONITO AND JOE PASCIAK

Contents

1. Lecture 1: Preliminaries.

Definition 1.1 (Limits)**.** *Let $c \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$. We say that the limit of f when x tends to c exists and is equal to L if*

1. *f is well defined in a neighborhood of c (but not necessarily at c);*
2. *given $\varepsilon > 0$, there exists $\delta > 0$ such that*

$$|f(x) - L| < \varepsilon \qquad \textit{when} \qquad x \in (c - \delta, c + \delta) \setminus \{c\}.$$

In this case, we write

$$\lim_{x \to c} f(x) = L.$$

Definition 1.2 (Continuity)**.** *A function $f : \mathbb{R} \to \mathbb{R}$ is continuous at $c \in \mathbb{R}$ if the limit of f at c exists and*

$$\lim_{x \to c} f(x) = f(c).$$

Definition 1.3 ($C(a, b)$)**.** *We say that $f \in C(a, b)$ if $f : \mathbb{R} \to \mathbb{R}$ is continuous for each $x \in (a, b)$.*

Definition 1.4 ($C[a, b]$)**.** *We say that $f \in C[a, b]$ if $f \in C(a, b)$ and*

$$\lim_{x \to a^+} f(x) = f(a) \qquad \textit{and} \qquad \lim_{x \to b^-} f(x) = f(b).$$

Definition 1.5 (Differentiability)**.** *$f : \mathbb{R} \to \mathbb{R}$ is said to be differentiable at $c \in \mathbb{R}$ if f is defined in a neighborhood of c and*

$$\lim_{x \to c} \frac{f(x) - f(c)}{x - c}$$

exists. In that case, we write

$$f'(c) = \lim_{x \to c} \frac{f(x) - f(c)}{x - c}.$$

Result 1.1 (Differentiability and Continuity)**.** *If f' exists at c, then f is continuous at c.*

Definition 1.6 ($C^1(a, b)$)**.** *We say that $f \in C^1(a, b)$ if both f and f' are in $C(a, b)$.*

Definition 1.7 ($C^1[a, b]$)**.** *We say that $f \in C^1[a, b]$ if $f \in C^1(a, b)$, $f \in C[a, b]$ and $f' \in C[a, b]$.*

Example 1.1 (Continuity and Differentiability)**.** *Two examples:*

1. *$f(x) = 1/x$ is in $C(0, 1)$ but not in $C[0, 1]$;*
2. *$f(x) = x^{1/2}$ is in $C[0, 1]$ not in $C^1[0, 1]$.*

Theorem 1.1 (Intermediate Values)**.** *If $f \in C[a, b]$, then f takes all the calues between $f(a)$ and $f(b)$.*

Theorem 1.2 (Mean Value). *If $f \in C^1[a,b]$, then there exists $c \in (a,b)$ such that*

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

Theorem 1.3 (Rolle's). *If $f \in C^1[a,b]$ and $f(a) = f(b)$, then there exists $c \in (a,b)$ with $f'(c) = 0$.*

The Rolles theorem is illustrated in Figure 1.

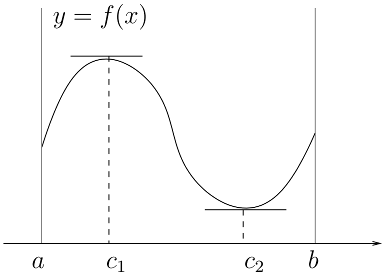

FIGURE 1. Illustration of Rolles theorem. In this case, there are two possible c in the interval (a,b).

Definition 1.8 ($C^m[a,b]$). *We say that $f \in C^m[a,b]$ if $f, f', ..., f^{(m)} \in C[a,b]$.*

Theorem 1.4 (Taylor). *Assume $f \in C^n[a,b]$ and $f^{(n+1)}$ exists and is in $C(a,b)$. Then for $c \in (a,b)$ and $x \in [a,b]$*

$$f(x) = \sum_{j=0}^{n} \frac{f^{(j)}(c)}{j!}(x-c)^j + E_{n+1}(x) := T_n(x) + E_{n+1}(x).$$

The error term $E_{n+1}(x)$ is given by

$$E_{n+1}(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x-c)^{n+1} \tag{1}$$

for some ξ between x and c; or

$$E_{n+1}(x) = \frac{1}{n!} \int_c^x f^{(n+1)}(t)(x-t)^n dt. \tag{2}$$

Taylors Theorem provides a *numerical approximation*

$$T_n(x) = \sum_{j=0}^{n} \frac{f^{(j)}(c)}{j!}(x-c)^j$$

of the function f together with an error bound $E_{n+1}(x)$.

Example 1.2 (Approximation using Taylors Theorem)**.** *We use the* 3 *term Maclaurin series (Taylor series with* $c = 0$*) to approximate* $\cosh(x)$ *for* $x \in [-1, 1]$ *and bound the error. To do this, we compute* $(\cosh(x)' = \sinh(x)$*,* $(\cosh(x)'' = \cosh(x)$ *and* $(\cosh(x)''' = \sinh(x)$*. This leads to the approximation*

$$\cosh(x) \approx \cosh(0) + \sinh(0)x + \frac{1}{2}\cosh(0)x^2 = 1 + \frac{x^2}{2}.$$

To bound the error $|cosh(x) - (1 + \frac{x^2}{2})| = |E_3(x)|$*, we resort to the expression of* E_3 *given by* (1)*, which reads in this case*

$$E_3(x) = \frac{1}{3!}\sinh(\xi)x^3$$

for some $\xi \in (-1, 1)$*. Since* $|\sinh(\xi)| = \sinh(|\xi|)$ *and* $\sinh(t)$ *is increasing for positive* t*, we deduce that*

$$|\sinh(\xi)| \leq \sinh(1) = \frac{e - e^{-1}}{2}.$$

As a consequence, we obtain the error bound

$$|E_3(x)| \leq \frac{e - e^{-1}}{12}.$$

Example 1.3 (Effect of the Length of the Approximation Interval)**.** *Let us consider the quadratic (*3 *term) Maclaurin series approximating* $\cos(x)$ *on* $[-\pi, \pi]$*:*

$$\cos(x) \approx 1 - \frac{1}{2}x^2.$$

The corresponding error term reads

$$E_3(x) = \frac{1}{6}\sin(\xi)x^3$$

for some $\xi \in (-\pi, \pi)$*. As a consequence, we get*

$$|E_3(x)| \leq \frac{\pi^3}{6} \approx 5.16.$$

This indicates that Polynomial approximations may not be very good over large intervals*.*

Exercise 1.1 (Effect of the Interval)**.** *Consider the same problem as in Example 1.3 but on (i)* $[-1/2, 1/2]$ *and (ii)* $[-1/10, 1/10]$*.*

2. Lecture 2.

2.1. **Review of Some Linear Algebra.**

A system of m linear equations with n unknowns $\{x_1, ..., x_n\}$ is of the form

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m, \end{cases}$$

can be rewritten in a matrix-vector form

$$Ax = b,$$

where

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

and

$$A = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix}.$$

We will often deal with square systems, i.e. $m = n$. For a square system:

Theorem 2.1 (Invertibility)**.** *Let A be a $n \times n$ matrix. The following are equivalent*

- $\det(A) \neq 0$*;*
- *A is invertible, i.e. there exists an $n \times n$ matrix B satisfying*

  $$BA = AB = I,$$

  (I denotes the identity matrix);
- $\mathrm{Range}(A) = \mathbb{R}^n$*;*
- $\mathrm{Ker}(A) = \{x \in \mathbb{R}^n \;:\; Ax = 0\} = \{0\}$*;*
- $\mathrm{Rank}(A) = n$*;*
- *For any $b \in \mathbb{R}^n$, the system $Ax = b$ has a unique solution $x \in \mathbb{R}^n$.*

2.2. **Polynomial Interpolation (see Chapt. 6).**

Definition 2.1 (Polynomial Space)**.** *We define $\mathbb{P}^k$ to be the collection of all polynomials of degree at most k, i.e.*

$$\mathbb{P}^k = \mathrm{span}\{1, x, ..., x^k\}.$$

In particular $\dim(\mathbb{P}^k) = k + 1$*.*

Interpolation problem: Given a function f and interpolation points $x_1, ..., x_m$, find $p \in \mathbb{P}^k$ such that

$$p(x_i) = f(x_i), \qquad i = 1, ..., m.$$

Some remarks are in order.

Remark 2.1 (Distinct Interpolation Points). The $x_i's$ should be distinct otherwise the equations are redundant.

Remark 2.2 (Number of Interpolation Points). $\mathbb{P}^k$ has dimension $k+1$ so we need at least $m = k+1$ snapshots.

Remark 2.3 (Approximation). The interpolating polynomial $p(x)$ provides an approximation to f.

Theorem 2.2 (Interpolation)**.** *Let $x_0, ..., x_n$ be distinct real numbers. Then for arbitrary snapshots $y_0, ..., y_n$ there exists a unique $p \in \mathbb{P}^n$ satisfying*

$$p(x_i) = y_i, \qquad i = 0, ..., n.$$

Proof. Let $p(x) = a_0 + a_1 x + ... + a_n x_n$ such that $p(x_j) = y_j$ for $j = 0, ..., n$. The latter conditions can be rewritten

$$a_0 + a_1 x_j + a_2 x_j^2 + ... + a_n x_j^n = y_j, \qquad j = 0, ..., n.$$

This is a linear system of $n+1$ equations and $n+1$ unknowns $\{a_0, ..., a_n\}$. This system corresponds to

$$Ba = y,$$

where

$$y = \begin{pmatrix} y_0 \\ \vdots \\ y_n \end{pmatrix}, \qquad a = \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ \vdots & & \ddots & \vdots & \\ 1 & x_n & x_n^2 & ... & x_n^n \end{pmatrix}.$$

According to Theorem 2.1, we shall show that B is invertible by checking that $\mathrm{Ker}(B) = \{0\}$. Suppose that $Bc = 0$ for some

$$c = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix}.$$

Consider $q(x) = c_0 + c_1 x + ... + c_n x^n$. Then $(Bc)_j = 0$ is equivalent to

$$c_0 + c_1 x_j + c_2 x_j^2 + ... + c_n x_j^n = 0,$$

i.e.

$$q(x_j) = 0.$$

The only polynomial of degree n with $n+1$ distinct roots is the zero polynomial. Thus $c_0 = c_1 = ... = c_n = 0$, i.e. $c = 0$ and $\mathrm{Ker}(B) = \{0\}$. □

Remark 2.4 (Function Interpolation). This implies that there exists a unique polynomial $p \in \mathbb{P}^n$ interpolating f at $x_0, ..., x_n$.

The following method constructs recursively the polynomial $p \in \mathbb{P}^n$ satisfying $p(x_i) = y_i$ for $i = 0, ..., n$.

Newton Form: Given $x_0, ..., x_n$ distincts and $y_0, ..., y_n$.

(1) if $n = 0$ then $\mathbb{P}^0 = \mathrm{span}\{1\}$ and $p_0 \in \mathbb{P}^0$ satisfying $p_0(x_0) = y_0$ is the constant polynomial $p_0(x) = y_0$.

(2) Suppose that $p_k \in \mathbb{P}^k$ has been constructed satisfying $p_k(x_i) = y_i$, $i = 0, ..., k$. We look for $p_{k+1} \in \mathbb{P}^{k+1}$ of the form

$$p_{k+1}(x) = p_k(x) + \underbrace{c_{k+1}(x - x_0)(x - x_1) \cdot ... \cdot (x - x_k)}_{\in \mathbb{P}^{k+1}},$$

where c_{k+1} is to be determined. Note that

$$p_{k+1}(x_i) = p_k(x_i) = y_i, \qquad i = 0, ..., k.$$

Therefore, we determine c_{k+1} imposing the last condition

$$y_{k+1} = p_{k+1}(x_{k+1}) = p_k(x_{k+1}) + c_{k+1}(x_{k+1} - x_1) \cdot ... \cdot (x_{k+1} - x_k)$$

i.e.

$$c_{k+1} = \frac{y_{k+1} - p_k(x_{k+1})}{(x_{k+1} - x_1) \cdot ... \cdot (x_{k+1} - x_k)}.$$

Example 2.1 (Newton form). *Find $p \in \mathbb{P}^3$ interpolating*

$$p(i) = 1/i, \qquad i = 1, 2, 3, 4.$$

The initialization of the algorithm consists in setting $p_0(x) = 1$. We then look for $p_1(x) = 1 + c_1(x - 1)$ using the second interpolation point 2:

$$1/2 = p_1(2) = 1 + c_1(2 - 1), \quad c_1 = -1/2 \quad \Longrightarrow \quad p_1(x) = 1 + \frac{1}{2}(x - 1).$$

Similarly for the third interpolation point: $p_2(x) = p_1(x)+c_2(x-1)(x-2)$ *such that*

$$1/3 = p_2(3) = 1-\frac{1}{2}(2)+c_2(2)(1), \quad c_2 = \frac{1}{6} \quad \Longrightarrow \quad p_2(x) = 1-\frac{1}{2}(x-1)+\frac{1}{6}(x-1)(x-2).$$

Finally, using the last interpolation point: $p_3(x) = p_2(x)+c_3(x-1)(x-2)(x-3)$

$$1/4 = p_3(4) = 1 - \frac{3}{2} + \frac{1}{6}(3\cdot 2) + c_3(3\cdot 2\cdot 1), \quad c_3 = -\frac{1}{24},$$

which leads to the desired polynomial

$$p_3(x) = 1 - \frac{1}{2}(x-1) + \frac{1}{6}(x-1)(x-2) - \frac{1}{24}(x-1)(x-2)(x-3).$$

You can plot this polynomial in matlab

```
x=.5:0.05:4;
y=1-(x-1)/2+times(x-1,x-2)/6 - times(x-1,times(x-2,x-3));
plot(x,y,x,rdivide(1,x));
```

3. Lecture 3.

3.1. Newton form in *matlab*. Recall that the Newton's form algorithm proceeds recursively.

(1) $p_0(x) = y_0$;
(2) $p_{k+1}(x) = p_k(x) + c_{k+1}(x - x_0) \cdot \ldots \cdot (x - x_k)$, where

$$c_{k+1} = \frac{y_{k+1} - p_k(x_{k+1})}{(x_{k+1} - x_0) \cdot \ldots \cdot (x_{k+1} - x_k)}.$$

Note that *matlab* array indices start at 1! Therefore, we rewrite the Newton form algorithm with index starting at 1.

Newton Form with Shifted index: Given $x_1, \ldots, x_{n+1}$ distincts and $y_1, \ldots, y_{n+1}$ find $p_{n+1} \in \mathbb{P}^n$ satisfying

$$p_{n+1}(x_j) = y_j, \qquad j = 1, \ldots, n+1.$$

(1) $p_1(x) = y_1$;
(2) $p_{k+1}(x) = p_k(x) + c_k(x - x_1) \cdot \ldots \cdot (x - x_k)$, where

$$c_k = \frac{y_{k+1} - p_k(x_{k+1})}{(x_{k+1} - x_1) \cdot \ldots \cdot (x_{k+1} - x_k)}.$$

We will need three *matlab* functions.

```
function c=CGEN(n,x,y)
```

which generates $c_1, \ldots, c_n$ from $x_1, \ldots, x_{n+1}$ and $y_1, \ldots, y_{n+1}$,

```
function Px=EVAL(n,c,x,y1,t)
```

which computes $p_{n+1}(t)$ given $c = (c_1, \ldots, c_n), x = (x_1, \ldots, x_{n+1})$, and $y1 = y_1$ and

```
function PLOTINTP(x0,xf,NP,c,x,n,y1,FN),
```

which plots $p_{n+1}(x)$ and the interpolated function, where $x0, xf$ are the plot limits, NP is the number of points used for plotting, c, x, n are as above and FN is the interpolated analytic. All three functions are provided on the class homepage.

We consider the *matlab* code for the $CGEN$ routine

```
function c=CGEN(n,x,y)
          for j=1:n    % compute c(j)
          % compute the denominator of c(j)
          PROD=1.0;
          for I=1:j
                    PROD = PROD*(x(j+1)-x(I));
          end
```

```
    % compute pj(x(j+1))
    if (j==1)
            pj=y(1);
    else
            pj=EVAL(j-1,c,x,y(1),x(j+1));
    end
    c(j) = (y(j+1)-pj)/PROD;
    end
```

You should go over the routine $EVAL$ and see if you can understand it. To generate the result $P_{n+1}(t)$, it has to compute $P_1(t), P_2(t), \ldots$.

The matlab codes are *matlab* m-file function codes and need to be in the subdirectory where you will run *matlab*. There is one file per function.

Problem 3.1. *For $n = 2, 3, 4, 5, 6$ define $\{x_1 = 1, x_2, x_3, ..., x_{n+1} = 4\}$ with $x_1, \ldots, x_{n+1}$ uniformly spaced on $[1, 4]$. Using the above routines, for each n, compute vector c for the polynomial in $\mathbb{P}^n$ interpolating e^x and plot on $[0, 5]$ using $NP = 200$. Report the $L-$infinity error computed by $PLOTINTP$ using the call*

```
PLOTINTP(0,5,200,c,x,y0,n,inline('exp(x)'));
```

Hand in plots for $n = 2$ and $n = 6$.

Problem 3.2. *Repeat the above problem using the interpolation interval $[0, \pi]$ but instead interpolating $\sin(x)$. Plot on $[0, \pi]$ with 200 points.*

3.2. Lagrange Form of the Interpolating Polynomial.

We consider again the problem of finding $p \in \mathbb{P}^n$ satisfying

$$p(x_i) = y_i, \qquad i = 0, ..., n.$$

We saw that there is a unique $l_j \in \mathbb{P}^n$ satisfying (fix $j \in \{0, 1, ..., n\}$)

$$l_j(x_i) = \delta_{ij},$$

where

$$\delta_{ij} := \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

is the Kronecker Delta. In fact,

$$l_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}.$$

These *Lagrange* polynomials allows us to solve the interpolation problem easily. Indeed, the interpolant is given by

$$p(x) = \sum_{i=0}^{n} y_i l_i(x)$$

(check it!).

Example 3.1 (Lagrange Polynomials)**.** *Let* $x_0 = 1$*,* $x_1 = 3/2$ *and* $x_2 = 2$*. Compute* $l_i(x)$ *for* $i = 0, 1, 2$*. They are given by*

$$\begin{aligned} l_0(x) &= \frac{(x-3/2)(x-2)}{(1-3/2)(1-2)} = 2(x-3/2)(x-2) \\ l_1(x) &= \frac{(x-1)(x-2)}{(3/2-1)(3/2-2)} = -4(x-1)(x-2) \\ l_2(x) &= \frac{(x-1)(x-3/2)}{(2-1)(2-3/2)} = 2(x-1)(x-3/2). \end{aligned}$$

Example 3.2 (Lagrange Interpolation)**.** *Use* l_0*,* l_1 *and* l_2 *from the previous example to find* $p \in \mathbb{P}^2$ *satisfying*

$$p(1) = 1, \quad p(2) = -1, \quad p(3/2) = 2.$$

The desired polynomial is directly given by

$$p(x) = 1l_0(x)+2l_1(x)-1l_2(x) = 2(x-3/2)(x-2)-8(x-1)(x-2)-2(x-1)(x-3/2).$$

4. Lecture 4: The Lagrange form: Consequences.

The Lagrange form can be used to deduce properties of polynomial interpolation.

We recall that $C[a, b]$ denote the set of continuous functions defined on $[a, b]$. The space $C[a, b]$ is a linear (vector) space, with vector operations given by

$$(f + g)(x) = f(x) + g(x), \qquad x \in [a, b]$$

and

$$(\alpha f)(x) = \alpha f(x), \qquad x \in [a, b]$$

(for all $f, g \in C[a, b]$ and $\alpha \in \mathbb{R}$). The set $\mathbb{P}^k$ is a subspace of $C[a, b]$ with the above operations. In addition, for $\{x_0, ..., x_k\} \subset [a, b]$ define

$$L : C[a, b] \to \mathbb{P}^k$$

by setting Lf to be the polynomial in $\mathbb{P}^k$ interpolating f at x_i, $i = 0, ..., k$.

Lemma 4.1 (Property of L)**.** *The transformation L is a linear transformation, i.e.*

$$L(\alpha f + \beta g) = \alpha L(f) + \beta L(g)$$

for all $\alpha, \beta \in R$ and $f, g \in C[a, b]$.

Proof. Let $\{l_i\}_{i=0}^k$ be the Lagrange polynomials associated with $x_0, ..., x_k$. Then

$$(Lf)(x) = \sum_{i=0}^k f(x_i) l_i(x) \quad \text{and} \quad (Lg)(x) = \sum_{i=0}^k g(x_i) l_i(x).$$

This implies

$$\begin{aligned} L(\alpha f + \beta g) &= \sum_{i=0}^k (\alpha f(x_i) + \beta g(x_i)) l_i(x) = \alpha \sum_{i=0}^k f(x_i) l_i(x) + \beta \sum_{i=0}^k g(x_i) l_i(x) \\ &= \alpha (Lf)(x) + \beta (Lg)(x). \end{aligned}$$

□

Lemma 4.2. *Let $\{l_i\}_{i=0}^n$ be the Lagrange polynomials corresponding to the nodes $\{x_0, x_1, ..., x_n\}$. Then every $p \in \mathbb{P}^n$ reads*

$$p(x) = \sum_{i=0}^n p(x_i) l_i(x).$$

Proof. The polynomial $q \in \mathbb{P}^k$ given by

$$q(x) = \sum_{i=0}^{n} p(x_i) l_i(x)$$

interpolates $p(x)$. Since p trivially interpolates itself, the uniqueness of the interpolant implies that $q = p$. □

Remark 4.1 (Polynomial Integration). Using the above representation lemma, we directly deduce an integration formula working simultaneously for all polynomials of degree n:

$$\int_a^b p(x)dx = \int_a^b \sum_{i=0}^{n} p(x_i) l_i(x) = \sum_{i=0}^{n} p(x_i) A_i,$$

where $A_i := \int_a^b l_i(x)dx$.

The next theorem is central to derive approximation properties of the interpolant.

Theorem 4.1 (Interpolation Estimates). *Assume that $f \in C^{n+1}[a,b]$ and $p \in \mathbb{P}^n$ interpolates f at $\{x_0, ..., x_n\} \subset [a,b]$ (distinct). To each $x \in [a,b]$, there is a $\xi_x \in (a,b)$ such that*

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{i=0}^{n} (x - x_i). \tag{3}$$

Proof. If $x = x_i$ for some $i = 0, ..., n$ then the assertion is trivial. Therefore, we assume that $x \neq x_i$. Set

$$w(t) = (t - x_0)(t - x_1) \cdot ... \cdot (t - x_n)$$

and

$$\phi(t) = f(t) - p(t) - \lambda w(t), \tag{4}$$

with

$$\lambda = \frac{f(x) - p(x)}{w(x)}.$$

Note that $\phi \in C^{n+1}[a,b]$, $\phi(x_i) = 0$, $i = 0, ..., n$ and $\phi(x) = 0$. Hence, ϕ has $n+2$ distinct roots. Rolles theorem implies that ϕ' has at least $n+1$ distinct roots. Repeating the argument: ϕ'' has at least n roots,..., $\phi^{(n+1)}$ has at least 1 root, denoted $\xi_x \in (a,b)$. Differentiating (4) $n+1$ times, we get

$$0 = \phi^{(n+1)}(\xi_x) = f^{(n+1)}(\xi_x) - \underbrace{\frac{d^{n+1}}{dt^{n+1}} p(t)}_{=0} |_{t=\xi_x} - \lambda \underbrace{\frac{d^{n+1}}{dt^{n+1}} w(t)}_{=(n+1)!} |_{t=\xi_x}.$$

In view of the definition of λ, this simplifies to

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{i=0}^{n} (x - x_i),$$

which is the desired result. □

Let us now see if we can optimize $(x_0, ..., x_n)$ distinct such that

$$\max_{x \in [a,b]} |(x - x_0)...(x - x_n)| \leq \max_{x \in [a,b]} |(x - y_0)...(x - y_n)|$$

for any $y_0, ..., y_n$ distinct. Such points would make the product on the right hand side of (3) the smallest possible (in magnitude). We need to choose $x_0, ..., x_n$ so that

$$q(x) = (x - x_0)...(x - x_n)$$

has minimal absolute value on $[a, b]$.

Suppose that $[a, b] = [-1, 1]$. It is not hard to argue that one should be able to get a smaller maximum by moving the quadrature nodes inside of $(-1, 1)$ so we assume that $\{x_i\} \subset (-1, 1)$. Then $q(x)$ is monotone (increasing or decreasing) for $x > b$ and $x < a$.

To gain more insight, consider $n = 0$. In that case,

$$\max_{x \in [-1,1]} |x - x_0| = \begin{cases} 1 + x_0 & \text{if} x_0 \geq 0, \\ 1 - x_0 & \text{if} x_0 \leq 0 \end{cases}$$

and the minimum occurs at $x_0 = 0$. Note that $\{-1, 1\}$ are “extrema” of the function $q(x) := x - x_0$ on the interval $[-1, 1]$, i.e., $|q(-1)| = |q(1)| = 1$ and $|q(x)| < 1$ for $x \in (-1, 1)$.

Next consider the quadratic case. Intuitively, the two points should be symmetric, i.e. $x_1 = -x_0$ and so

$$q(x) = (x - x_0)(x + x_0) = x^2 - x_0^2.$$

Note that $q(x)$ is a quadratic which is symmetric about $x = 0$. It is easy to see that $\max_{x \in [-1,1]} |q(x)|$ is minimzed when $|q(0)| = |q(1)| = |q(-1)|$, i.e.,

$$x_0^2 = 1 - x_0^2 \quad \text{or} \quad x_0 = \frac{1}{\sqrt{2}}.$$

Note that in this case, q has three extrema at $\{-1, 0, 1\}$.

In the two examples, q has $n + 1$ extreme points with equal magnitude. We need a polynomial $T_n(x)$ with n zeros in $[-1, 1]$ and $n + 1$ extrema’s (with oscillating signs). Such a polynomial is developed in the next lecture.

5. Lecture 5: Chebyschev Polynomials.

The cosine function has a lots of zeros and alternating extrema but unfortunately, it is not a polynomial.

Define for $x \in [-1,1]$ and integer $n \geq 0$

$$T_n(x) := \cos(n \cos^{-1}(x)).$$

Recall that

$$\cos^{-1} : [-1,1] \to [0,\pi]$$

so that

$$n\cos^{-1} : [-1,1] \to [0,n\pi].$$

As a consequence, $T_n(x)$ has $n+1$ extrema oscillating between ± 1. Moreover, $T_n(x)$ has n zeros. Less obvious, $T_n(x)$ is actually a polynomial. Indeed,

$$T_0(x) = \cos(0) = 1$$
$$T_1(x) = \cos(\cos^{-1}(x)) = x.$$

Also,

$$T_{n+1}(x) = \cos((n+1)\theta), \quad \theta = \cos^{-1}(x)$$

and so

$$T_{n+1}(x) = \cos(\theta)\cos(n\theta) - \sin(\theta)\sin(n\theta).$$

Similarly

$$T_{n-1}(x) = \cos(\theta)\cos(n\theta) + \sin(\theta)\sin(n\theta)$$

and therefore

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos(\theta)\cos(n\theta) = 2xT_n(x).$$

We just proved a recurrence formula

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

which implies

1. $T_n(x)$ is a polynomial of degree n;
2. The leading coefficient for $T_n(x)$ is $2^{n-1}x^n$ for $n \geq 1$;
3. For odd n, $T_n(x)$ is an odd function (contain only x^j with j odd) while for even n, $T_n(x)$ is even.

The roots of T_n are related to the zeros of $\cos(n\theta)$, $\theta \in [0,\pi]$ ($\theta = \cos^{-1}(x)$). Since $\cos(y) = 0$ for $y = \pi(j+\frac{1}{2})$, the roots x_j of $T_n(x)$ are such that

$$n\cos^{-1}(x_j) = \pi(j+\frac{1}{2})$$

or

$$x_j = \cos\left(\frac{\pi}{n}(j+\frac{1}{2})\right), \qquad j = 0,1,...,n-1.$$

Example 5.1 ($n = 1$)**.** *When* $n = 1$*,* $T_1(x) = x$ *and* $x_0 = 0$*.*

Example 5.2 ($n = 2$)**.** *When* $n = 2$*,* $T_2(x) = 2x^2 - 1$ *and* $x_0 = -\frac{1}{\sqrt{2}}$*,* $x_1 = \frac{1}{\sqrt{2}}$*.*

Example 5.3 ($n = 3$)**.** *When* $n = 3$*,* $T_3(x) = 4x^3 - 3x$ *and* $x_0 = -\sqrt{\frac{3}{4}}$*,* $x_1 = 0$*,* $x_2 = \sqrt{\frac{3}{4}}$*.*

The following theorem provides a sharp estimate on the interpolation error (see Theorem 4.1).

Theorem 5.1 (Interpolation Error)**.** *Given* n*, let* $x_0, ..., x_n$ *be the roots of* $T_{n+1}(x)$*, i.e.*

$$x_j = \cos\left(\frac{\pi}{n+1}(j + \frac{1}{2})\right), \qquad j = 0, .., n.$$

Let $\{y_0, ..., y_n\} \subset \mathbb{R}^{n+1}$*, then*

$$m_x := 2^{-n} = \max_{x\in[-1,1]} \left|\prod_{i=0}^{n}(x - x_i)\right| \leq \max_{x\in[-1,1]} \left|\prod_{i=0}^{n}(x - y_i)\right| =: m_y.$$

Proof. We proceed by contradiction. Suppose the theorem does not hold. Then, there exists $\{y_0, ..., y_n\}$ with $m_y < m_x$. Set

$$P(x) := \prod_{i=0}^{n}(x - x_i) = 2^{-n}T_{n+1}(x),$$

where we used the fact that x_i are the roots of $T_{n+1}(x)$. Also, we set

$$Q(x) := \prod_{i=0}^{n}(x - y_i) \qquad \text{and} \qquad R(x) := P(x) - Q(x).$$

Now, $m_x = 2^{-n}$ since $T_{n+1}(x)$ oscillate between -1 and 1. Moreover, $m_y < m_x$ implies that $R(x)$ has the same sign as $P(x)$ at each extrema of $P(x)$ (which coincide with those of $T_{n+1}(x)$).

There are $n + 2$ extrema with oscillating signs. Therefore $R(x)$ has oscillating signs at the extrema of $T_{n+1}(x)$. The intermediate value theorem implies that there is a root of $R(x)$ between each pair (of oscillating signs). In turn, $n + 2$ extrema implies that $R(x)$ has $n + 1$ roots. Note however, that both $P(x)$ and $Q(x)$ are monic so their difference, i.e., $R(x)$ is a polynomial of degree n. The only polynomial of degree n with $n + 1$ distinct roots is the zero polynomial, which implies that $P(x) = Q(x)$. This contradicts $m_y < m_x$. □

Example 5.4 (Error Theorem)**.** *Let $x_0, ..., x_n$ be distinct in $[0, \pi]$ and $p \in \mathbb{P}^n$ interpolate $\sin(x)$ at $x_0, ..., x_n$. Derive a bound for the error $|\sin(x) - p(x)|$ for $x \in [0, \pi]$. According to Theorem 4.1, we have*

$$|\sin(x) - p(x)| = \frac{|f^{(n+1)}(\xi_x)|}{(n+1)!} \left| \prod_{i=0}^{n} (x - x_i) \right| .$$

Without information on the distribution of $\{x_i\}$

$$\left| \prod_{i=0}^{n} (x - x_i) \right| = \prod_{i=0}^{n} |(x - x_i)| \leq \pi^{n+1}$$

and $|f^{(n+1)}(\xi_x)| \leq 1$ since

$$|f^{(n+1)}(\xi_x)| = \begin{cases} |\cos(\xi_x)| & \text{when } n+1 \text{ is odd,} \\ |\sin(\xi_x)| & \text{when } n+1 \text{ is even.} \end{cases}$$

As a consequence, we obtain

$$|\sin(x) - p(x)| \leq \frac{\pi^{n+1}}{(n+1)!} \to 0 \qquad \text{when } n \to \infty.$$

We see that the error in the above example actually converges to 0 as n becomes large. This is misleading.

Example 5.5. *We consider approximating the function $f(x) = (2 - x)^{-1}$ on the interval $[-1, 1]$. In this case,*

$$f^{(n+1)}(\zeta) = \frac{(n+1)!}{(2 - \zeta)^{n+2}}$$

so we can only conclude that for $\zeta \in (-1, 1)$,

$$\left| \frac{f^{(n+1)}(\zeta)}{(n+1)!} \right| \leq 1.$$

Now, if we let $x_0, x_1, \ldots x_n$ be arbitrary and distinct in $(-1, 1)$, we can only obtain

$$| \prod_{i=0}^{n} (x - x_i) | \leq 2^{n+1}$$

so that if p interpolates f at $x_0, x_1, \ldots, x_n$, Theorem 4.1 only yields

$$|f(x) - p(x)| \leq 2^{n+1}$$

(hardly useful). In contrast, if we select the nodes using Theorem 5.1, we obtain

$$| \prod_{i=0}^{n} (x - x_i) | \leq 2^{-n}$$

so that

$$|f(x) - p(x)| \leq 2^{-n}$$

in this case. This, at least, shows convergence as n increases.

The above example is somewhat rigged. If we took

$$f(x) = (1 + \gamma - x)^{-1}$$

with $\gamma < 1/2$, even the Chebyshev nodes would not guarantee convergence as n becomes large. **In general, polynomial interpolation with high order polynomials is not a good idea.**

6. Lecture 6: Piecewise Polynomial Interpolation.

The following example, shows the limitation of higher order interpolation methods.

Example 6.1 (No Convergence)**.** *Let* $p \in \mathbb{P}^n$ *interpolate* $f(x) = x^{-\alpha}$ *(*α *defined below) on the interval* $[1/2, 2]$ *using* $n+1$ *distinct nodes* $\{x_i\}$ *in* $[1/2, 2]$*. By the error Theorem 4.1*

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

so

$$|f(x) - p(x)| = \frac{\max_{\xi \in [1/2,2]} |f^{(n+1)}(\xi)|}{(n+1)!} \prod_{i=0}^{n} |x - x_i|.$$

Without further information on the $\{x_i\}$*, we can only conclude*

$$|x - x_i| \le 3/2.$$

Also

$$\begin{aligned} f'(x) &= -\alpha x^{-(\alpha+1)}, \\ f''(x) &= \alpha(\alpha+1) x^{-(\alpha+2)}, \\ &\vdots \\ f^{(n+1)}(x) &= (-1)^{n+1} \alpha(\alpha+1) \cdot \ldots \cdot (\alpha+n) x^{-(\alpha+n+1)}. \end{aligned}$$

Suppose $\alpha \le 1$*, then*

$$\frac{|f^{(n+1)}(\xi)|}{(n+1)!} \le \frac{\alpha(\alpha+1) \cdot \ldots . \cdot (\alpha+n)}{1 \cdot 2 \cdot \ldots \cdot (n+1)} |\xi|^{-(\alpha+n+1)} \le |\xi|^{-(\alpha+n+1)}.$$

In addition, $|\xi|^{-(\alpha+n+1)}$ *is a decreasing function in* ξ *so that*

$$\max_{\xi \in [1/2,2]} |\xi|^{-(\alpha+n+1)} = 2^{\alpha+n+1}.$$

Gathering the above estimates, we obtain the bound

$$|f(x) - p(x)| \le 2^{\alpha+n+1} (3/2)^{n+1} = 2^{\alpha} \cdot 3^{n+1}.$$

Therefore, no convergence can be guaranteed (unless the interpolation points are chosen strategically).

Piecewise polynomial interpolation is a way to circumvent this issue.

Let $I = [a, b]$ be the interval where we want to construct an interpolant and $N > 0$ an integer. Set $z_i := a + (b-a)\frac{i}{N}$ with constitutes a uniform partition of $[a, b]$ with spacing $h = \frac{b-a}{N}$, i.e.

$$a = z_0 < z_1 < \ldots < z_N = b$$

and $z_i - z_{i-1} = h$. On each subinterval $I_i = (z_{i-1}, z_i)$, we use low order approximation.

6.1. **Case 1: $\mathbb{P}^0$ interpolation.** Choose $\tilde{z}_i \in I_i$ and set $f_h(x)$ on each subinterval by

$$f_h(x)|_{I_i} = f(\tilde{z}_i).$$

This is a polynomial interpolation in $\mathbb{P}^0$ on each subinterval, i.e. f_0 is piecewise constant. Using the error Theorem 4.1 on each subinterval I_i:

$$|f(x) - f_h(x)| = |f'(\xi_x)||(x - \tilde{z}_i)| \leq \sup_{\xi \in I_i} |f'(\xi)|h$$

for every $x \in I_i$. As a consequence, the piecewise interpolation converges as the $h \to 0$, i.e. the number of intervals increase to infinity ($N \to \infty$).

Example 6.2 ($x^{-\alpha}$)**.** *We return to the example above. In this case,* $a = 1/2$*,* $b = 2$ *and* $z_i = \frac{1}{2} + \frac{3}{2}\frac{i}{N}$*,* $i = 0, ..., N$ *with* $h = \frac{3}{2N}$*. For* $x \in I_i$*, we have*

$$|f(x) - f_h(x)| \leq \alpha h \sup_{\xi \in I_i} \xi^{-\alpha-1} \leq \alpha h 2^{\alpha+1}.$$

This implies that

$$\max_{x \in [1/2,2]} |f(x) - f_0(x)| \leq \alpha h 2^{\alpha+1}.$$

Notice that the interpolant $f_h(x)$ of $f(x)$ is *not continuous.* We sometimes write $f_h(x) \in C^{-1}(1/2, 2)$.

6.2. **Case 2: Continuous $\mathbb{P}^1$ interpolation.** We construct f_h, continuous on the entire interval and in $\mathbb{P}^1$ on each subinterval. To do this, we use the endpoints of I_i as the interpolation nodes leading to

$$f_h(x) = f(z_{i-1})\frac{(z_i - x)}{h} + f(z_i)\frac{(x - z_{i-1})}{h}$$

for $x \in I_i$. Notice that in particular, $f_h(z_{i-1}) = f(z_{i-1})$ and $f_h(z_i) = f(z_i)$, which implies the resulting piecewise polynomial f_h is *continuous* because $f_h(z_i) = f(z_i)$ from I_i and I_{i+1} for $i = 1, ..., N-1$ (so we do not need to worry that I_i and I_{i+1} defines $f_h(z_i)$).

Remark 6.1 (Discontinuous piecewise linears)*.* Continuity was forced by choosing the endpoints as interpolation nodes. If instead you used interior nodes, you will end up with a discontinuous approximation (in general).

On each I_i we use again error representation provided by Theorem 4.1:

$$|f(x) - f_h(x)| \le \frac{|f^{(2)}(\xi_x)|}{2}|(x - z_{i-1})(x - z_i)|, \qquad x \in I_i.$$

Now $|(x-z_{i-1})(x-z_i)| = (z_i-x)(x-z_{i-1})$ which achieves its maximum at $x = \frac{1}{2}(z_i + z_{i-1})$ with a value of $\frac{h^2}{4}$. Therefore, for $x \in I_i$

$$|f(x) - f_h(x)| \le \sup_{\xi\in I_i} \frac{|f^{(2)}(\xi)|}{8}h^2$$

and for $x \in I_i$

$$|f(x) - f_h(x)| \le \sup_{\xi\in I} \frac{|f^{(2)}(\xi)|}{8}h^2.$$

Notice that again the error is converging but this time the rate is quadratic (instead of linear).

Example 6.3. $x^{-\alpha}$ *For the above example, this is*

$$|f(x) - f_h(x)| \le \alpha(\alpha+1)\frac{h^2}{8} \max_{\xi\in[1/2,2]} \xi^{-\alpha-2} \le \alpha(\alpha+1)2^{\alpha-1}h^2.$$

The following theorem gathers both cases discussed above.

Theorem 6.1. *Let f be defined on $[a,b]$ with $f \in C^1[a,b]$. Set $z_i := a + ih$, $i = 0, ..., N$ with $h = (b-a)/N$. If f_h is the piecewise constant approximation of f then*

$$|f_h(x) - f(x)| \le \|f'\|_{L^\infty(a,b)}h.$$

Moreover, if $f \in C^2[a,b]$ and f_h is the continuous, piecewise linear approximation of f then

$$|f_h(x) - f(x)| \le \|f''\|_{L^\infty(a,b)}\frac{h^2}{8}.$$

Here $\|v\|_{L^\infty(a,b)} := \max_{\xi\in[a,b]} |v(\xi)|$.

7. Lecture 7.

In the previous lecture, we discussed piecewise interpolation with $\mathbb{P}^k$, $k \geq 1$. These approximations are globally continuous if the endpoints of each interval are interpolation points.

We are now contemplating the possibility of constructing globally C^1 interpolants.

Example 7.1 (Globally $\mathbb{P}^2$ and C^1 interpolant)**.** *Find $p \in \mathbb{P}^2$ satisfying*

$$p(-1) = a, \quad p'(0) = b, \quad p(1) = c \tag{5}$$

for given $a, b, c \in \mathbb{R}$.

Let $p(x) = \alpha + \beta x + \gamma x^2$ for α , β and $\gamma \in \mathbb{R}$. The above 3 equations lead to the linear system

$$\underbrace{\begin{pmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} a \\ b \\ c \end{pmatrix}}_{B}.$$

Note that $\det(A) = 1 \det \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = 0$ and so A is singular. This means that depending on B, there are either infinitely many solutions or no solutions. To give an example, if $p(x)$ solves (5), so does $q(x) = p(x) + \eta(x^2 - 1)$ for any $\eta \in \mathbb{R}$. Also there are no solutions to (5) when $a = 0$, $b = 1$ and $c = 0$ because otherwise both $(x-1)$ and $(x+1)$ would have to divide $p(x)$, i.e. $p(x)$ would have to be a multiple of $(x^2 - 1)$ and so $p'(0) = 0$.

Exercise 7.1. *Consider the interpolation problem, find $p \in \mathbb{P}^3$ satisfying*

$$p(-1) = y_0, \quad p(0) = y_1, \quad p'(0) = y_2, \qquad p(1) = y_3.$$

Show that the above problem always as a unique solution.
Hint: Look for a solution $p(x) = \alpha + \beta x + \gamma x^2 + \delta x^3$, derive the matrix system for the unknowns α, β, γ and δ and show that its determinant is non-zero.

7.1. **A general uniquely solvable interpolation problem.**

Assume $\{x_0, x_1, ..., x_n\}$ are distinct. Find $p \in \mathbb{P}^N$ satisfying for $j = 0, 1, ..., m_i$ and $i = 0, 1, ..., n$

$$p^{(j)}(x_i) = y_{i,j},$$

where $y_{i,j}$ are given.

The number of equations is

$$\sum_{i=0}^{n}(m_i+1)$$

so we must take $N=\sum_{i=0}^{n}(m_i+1)-1$.

Theorem 7.1 (Globally Smooth Interpolation). *The above interpolation problem has a unique solution given any set* $\{y_{i,j}\}_{i=0;j=0}^{n;m_j}$

Notice that if you include a derivative of order j at x_i, you must also include $p^{(l)}(x_i)$, $l=0,1,...,j-1$.

Proof. We leave the proof as an exercise. □

7.2. **The general Hermite interpolation problem.** Given $\{x_0,...,x_n\}$ distinct and $f\in C^1[x_0,x_n]$. Find $p\in\mathbb{P}^{2n+1}$ satisfying

$$p(x_i)=f(x_i), \qquad p'(x_i)=f'(x_i).$$

According to the previous theorem, this interpolation problem has a unique solution.

Theorem 7.2 (Error with Hermite Interpolation). *Suppose that* $f\in C^{2n+2}[a,b]$ *and that* $\{x_i\}\subset[a,b]$. *Let* p *be the Hermite interpolant. Then for* $x\in[a,b]$, *there is a* $\xi_x\in(a,b)$ *satisfying*

$$f(x)-p(x)=\frac{f^{(2n+2)}(\xi_x)}{(2n+2)!}\prod_{i=0}^{n}(x-x_i)^2.$$

Proof. The formula is valid when $x=x_i$, $i=0,...,n$. Therefore, we assume that $x\notin\{x_0,...,x_n\}$.

Set $w(t)=\prod_{i=0}^{n}(t-x_i)^2$ and

$$\phi(t)=f(t)-p(t)-\lambda w(t), \tag{6}$$

where

$$\lambda=\frac{f(x)-p(x)}{w(x)}.$$

Note that $\phi'(x_i)=0$ since $f'(x_i)=p'(x_i)$ and $w'(x_i)=0$. Now $\phi(x_i)=0$ for $i=0,1,...,n$ and $\phi(x)=0$ from the definition of λ. Rolle's theorem implies that ϕ' has at least $n+1$ additional zeros which are not in $\{x_0,...,x_n\}$. Therefore, ϕ' has at least $2n+2$ distinct zeros. Applying Rolle's theorem again but to these zeros implies that ϕ'' has at least $2n+1$ zero. Repeating this process we find that $\phi^{(2n+2)}$ has at least 1 zero, i.e. $\phi^{(2n+2)}(\xi_x)=0$ for some $\xi_x\in(a,b)$.

Differentiating (6) $2n+1$ times and evaluating at ξ_x

$$0=\phi^{(2n+2)}(\xi_x)=f^{(2n+2)}(\xi_x)-\lambda(2n+2)!$$

and the claim follows by simple algebra. □

Example 7.2 (Hermite $n = 0$)**.** *Find $p \in \mathbb{P}^1$ satisfying*

$$p(x_0) = f(x_0), \qquad p'(x_0) = f'(x_0).$$

The solution is

$$p(x) = f(x_0) + f'(x_0)(x - x_0),$$

i.e. the 2 term Taylor polynomial.

Example 7.3 (Cubic Hermite $n = 1$)**.** *Find $p \in \mathbb{P}^3$ satisfying*

$$p(x_i) = f(x_i), \qquad p'(x_i) = f'(x_i), \qquad i = 0, 1. \tag{7}$$

We use the Newton form solution. From the first example

$$p_1(x) = f(x_0) + f'(x_0)(x - x_0),$$

solves $p_1(x_0) = f(x_0)$ and $p_1'(x_0) = f'(x_0)$. Look for

$$p_2(x) = p_1(x) + c_2(x - x_0)^2$$

satisfying $p_2(x_1) = f(x_1)$, i.e.

$$c_2 = \frac{f(x_1) - p_1(x_1)}{(x_1 - x_0)^2} = \frac{f(x_1) - f(x_0) - f'(x_0)(x_1 - x_0)}{(x_1 - x_0)^2}.$$

Then we look for

$$p_3(x) = p_2(x) + c_3(x - x_0)^2(x - x_1)$$

satisfying $p_3'(x_1) = f'(x_1)$. As both problems above have a unique solution, so does (7)*.*

Exercise 7.2. *Derive an expression for $p_3(x)$ in term of*

$$x_0, x_1, f(x_0), f'(x_0), f(x_1), f'(x_1).$$

7.3. **Piecewise Hermite interpolation.**

Let $a = x_0 < x_1 < ... < x_m = b$ and consider piecewise Hermite cubic approximation f_h, i.e.

$$f_h(x) = p_i(x) \qquad \text{on } [x_{i-1}, x_i]$$

with $p_i \in \mathbb{P}^3$ solving

$$\begin{aligned} p_i(x_{i-1}) = f(x_{i-1}), &\qquad p_i'(x_{i-1}) = f'(x_{i-1}) \\ p_i(x_i) = f(x_i), &\qquad p_i'(x_i) = f'(x_i), \end{aligned}$$

where $h = \max_{i=1,..,N}(x_i - x_{i-1})$. Note that $f_h \in C^1[a, b]$.

Exercise 7.3. *Assume that $f \in C^4[a, b]$. Use the previous theorem to prove a 4th order (in h) error bound for $|f_h(x) - f(x)|$.*

8. Lecture 8.

8.1. **Fourier Series.** We now recall some results on Fourier series.

We set $\Omega = [0, 2\pi]$, $\psi_j(x) = e^{ijx}$ for $x \in \Omega$, $i = \sqrt{-1}$ and $j \in \mathbb{Z}$. We recall the Euler's formula for $\theta \in \mathbb{R}$

$$e^{i\theta} = \cos(\theta) + i\sin(\theta),$$

see Figure 2 for an illustration.

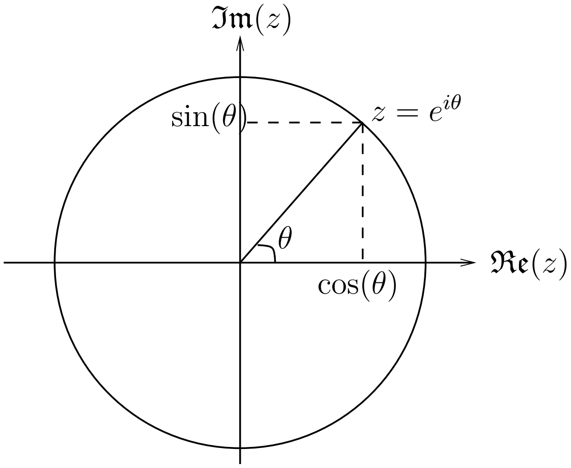

Figure 2. Illustration of the Euler's formula on the Complex plane.

Using the Euler's formula we find that

$$\psi_j(x) = \cos(jx) + i\sin(jx), \qquad j \in \mathbb{Z}.$$

Define the space

$$L^2(\Omega) := \left\{ f : [0, 2\pi] \to \mathbb{C} \ : \ \int_0^{2\pi} |f(x)|^2 dx < \infty \right\}$$

with norm

$$\|f\|_{L^2(\Omega)} := \left(\int_0^{2\pi} |f(x)|^2 dx \right)^{1/2}.$$

Remark 8.1 ($L^2(\Omega)$).
(1) $L^2(\Omega)$ is a vector space (infinite dimensional) of functions on $[0, 2\pi]$ with scalar field $\mathbb{C}$.
(2) A norm $\|.\|$ on a vector space $\mathbb{V}$ over $\mathbb{C}$ satisfies
   (a) $\|v\| \geq 0$ for all $v \in \mathbb{V}$ and $\|v\| = 0$ only if $v = 0$.
   (b) $\|\alpha v\| = |\alpha|\|v\|$ for all $\alpha \in \mathbb{C}$ and $v \in \mathbb{V}$.
   (c) $\|v + w\| \leq \|v\| + \|w\|$ for $v, w \in \mathbb{V}$ (triangle inequality).

(3) Norms on a vector space $\mathbb{V}$ give a notion of distances between elements of $\mathbb{V}$, i.e. the distance between two elements $v, w \in \mathbb{V}$ is $\|v - w\|$.

Definition 8.1 (Convergence of a Series in a Normed Vector Space)**.** *Let $\mathbb{V}$ be a vector space and $\|.\|$ a norm on $\mathbb{V}$. Assume $\{v_j\} \subset \mathbb{V}$ and $v \in \mathbb{V}$. Then $\sum_{j=1}^{\infty} v_j$ converges to v in $\|.\|$ if the sequence of partial sums $S_l := \sum_{j=1}^{l} v_j$ converges to v. This means that given $\varepsilon > 0$, there exists $N = N(\varepsilon)$ satisfying*

$$\|v - S_l\| \leq \varepsilon \qquad \text{when } l > N.$$

Theorem 8.1 (Fourier Series)**.** *For $f \in L^2(\Omega)$, the series*

$$\sum_{j=-\infty}^{+\infty} c_j \psi_j(x),$$

with

$$c_j = \frac{1}{2\pi} \int_0^{2\pi} f(x) \overline{\psi_j(x)}\, dx \in \mathbb{C}$$

converges to f in $\| \cdot \|_{L^2(\Omega)}$. (In this case we set $S_l(x) := \sum_{j=-l}^{l} c_j \psi_j(x)$ and the bar denotes the complex conjugate.) In addition,

$$\|f\|^2_{L^2(\Omega)} = 2\pi \sum_{j \in \mathbb{Z}} |c_j|^2.$$

The $\{c_j\}$ are called the Fourier coefficients of f.

Remark 8.2. Some remarks are in order.

(1) For $f \in L^2(\Omega)$, the series $\sum_{j=-\infty}^{\infty} |c_j|^2$ converges.

(2) Note that the functions $\psi_j(x)$ are periodic, i.e.

$$\lim_{x \to 0^+} \psi_j(x) = \lim_{x \to 2\pi^-} \psi_j(x)$$

and

$$\lim_{x \to 0^+} \psi_j'(x) = \lim_{x \to 2\pi^-} \psi_j'(x)$$

and so on for all derivatives.

(3) Depending on the smoothness of f, i.e. $f \in C^n[0, 2\pi]$ and $f, f', f^{(2)}, ..., f^{(n-1)}$ are periodic, then the series

$$\sum_{j \in \mathbb{Z}} |c_j|^2 (1 + j^2)^n$$

converges. The set of functions in $L^2(\Omega)$ for which the above series converges is denoted by $\dot{H}^n$ and defines a Hilbert space.

We set

$$\|f\|_{\dot{H}^n} := \left(\sum_{j\in\mathbb{Z}} |c_j|^2 (1+j^2)^n \right)^{1/2}, \quad \text{for } f \in \dot{H}^n.$$

Note that just like functions in $L^2(\Omega)$ can be discontinuous, functions in $\dot{H}^n$ may have non-periodic n'th order derivatives.

Theorem 8.2 (Spectral approximation)**.** *Suppose that $f \in \dot{H}^n$. Then the truncated series*

$$S_l := \sum_{j=-l}^{l} c_j \psi_j(x)$$

satisfies

$$\|f - S_N\|^2_{L^2(\Omega)} = 2\pi \sum_{|j|>N} |c_j|^2 \le \frac{2\pi}{(N+1)^{2n}} \|f\|^2_{\dot{H}^n}.$$

Proof. We have

$$\|f - S_N\|^2_{L^2(\Omega)} = 2\pi \sum_{|j|>N} |c_j|^2 \le \frac{2\pi}{1+(1+N)^2} \sum_{|j|>N} |c_j|^2 (1+j^2)^n$$
$$\le \frac{2\pi}{(1+N)^2} \|f\|^2_{\dot{H}^n}.$$

□

Some remarks are in order.

Remark 8.3.

(1) The rate of convergence for the truncated series is better for smoother periodic f.
(2) To compute the coefficients c_j, you need to compute integrals with complex integrands. We next provide an alternative approximation which avoids these integrals.

8.2. **Trigonometric Interpolation.** Given an integer $N \ge 0$, set

$$h = \frac{2\pi}{2N+1} \qquad \text{and} \qquad x_j = jh, \quad j = 0, ..., 2N.$$

Also define

$$\mathbb{V}_{2N+1} := \text{span}\{\psi_j \ , \ j = -N, ..., N\} = \left\{ \sum_{|j|\le N} d_j \psi_j \ , \ \{d_j\} \subset \mathbb{C} \right\}.$$

The *trigonometric interpolation problem* reads: Find $f_{2N+1} \in \mathbb{V}_{2N+1}$ satisfying

$$f_{2N+1}(x_j) = f(x_j), \qquad j = 0, 1, ..., 2N.$$

Note that $\dim(\mathbb{V}_{2N+1}) = 2N+1$ so f_{2N+1} involves $2N+1$ coefficients in any basis and there are $2N+1$ equations. We shall see that there is a unique solution to this interpolation problem.

8.3. **The DFT - Discrete Fourier Transform.** Define $E(j) := e^{\frac{2\pi i j}{2N+1}}$ for $j \in \mathbb{Z}$ and note that

$$E(j + (2N+1)) = e^{\frac{2\pi i j}{2N+1}} e^{\frac{2\pi i(2N+1)}{2N+1}} = E(j)\underbrace{e^{2\pi i}}_{=1} = E(j).$$

This shows that $E(j)$ is periodic with period $2N+1$. Also

$$E(j)^{(2N+1)} = e^{\frac{2\pi i j(2N+1)}{2N+1}} = e^{2\pi i j} = 1.$$

In fact, $E(j)$, $j = 0, 1, ..., 2N$ are the $2N+1$ roots of $x^{2N+1} - 1 = 0$, see Figure 3.

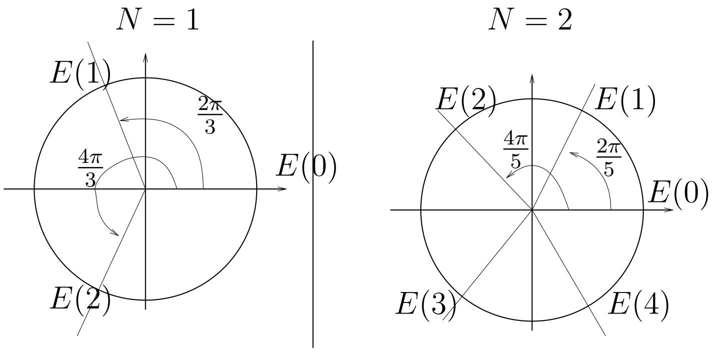

FIGURE 3. Illustration of DFT: (left) $N = 1$ where $E(0) = 1$, $E(1) = e^{\frac{2\pi i}{3}}$, $E(2) = e^{\frac{4\pi i}{3}}$, $E(3) = E(0) = 1$ and (right) $N = 2$ where $E(0) = 1$, $E(1) = e^{\frac{2\pi i}{5}}$, $E(2) = e^{\frac{4\pi i}{5}}$, $E(3) = e^{\frac{6\pi i}{5}}$, $E(4) = e^{\frac{8\pi i}{5}}$, $E(5) = E(0) = 1$.

For $d \in \mathbb{C}^{2N+1}$, we define $DFT_{\pm}(d) \in \mathbb{C}^{2N+1}$ by

$$DFT_{\pm}(d)(j) = \sum_{m=0}^{2N} d_m E(\pm jm).$$

We now return to the interpolation problem: Find

$$f_{2N+1} = \sum_{|j| \le N} c_j \psi_j \in \mathbb{V}_{2N+1}$$

satisfying

$$f_{2N+1}(x_l) = f(x_l), \qquad l = 0, 1, ..., 2N. \tag{8}$$

Note that

$$\begin{aligned} f(x_l) = f_{2N+1}(x_l) &= \sum_{j=-N}^{N} c_j e^{ijx_l} = \sum_{j=-N}^{N} c_j e^{\frac{ijl2\pi}{2N+1}} \\ &= \sum_{j=-N}^{N} c_j E(jl) = \sum_{j=0}^{2N} d_j E(jl) = DFT_+(d)(l), \end{aligned}$$

where

$$d_j = \begin{cases} c_j & \text{if } 0 \le j \le N, \text{ and} \\ c_{j-(2N+1)} & \text{if } N < j \le 2N. \end{cases}$$

This means that

$$F := \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{2N}) \end{pmatrix} = DFT_+(d).$$

The next theorem guarantees that the interpolation problem, i.e., finding the coefficients c_j or d_j so that f_{2N+1} satisfies (8), can be solved for any data and as this is a square linear system, unique solvability follows.

Theorem 8.3 (Inverse DFT)**.**

$$(DFT_+)^{-1} = \frac{1}{2N+1} DFT_-$$

so that

$$d = \frac{1}{2N+1} DFT_-(F).$$

The trigonometric interpolant also satisfies spectral approximation as demonstrated by the following theorem.

Theorem 8.4. *Suppose that f is in $\dot{H}^n$, $n \ge 1$ and $f_{2N+1} \in V_{2N+1}$ is its trigonometric interpolant, i.e., satisfies* (8)*. Then*

$$\|f - f_{2N+1}\|_{L^2(\Omega)} \le \frac{C}{(N+1)^n} \|f\|_{\dot{H}^n}$$

with C not depending on N or f.

Proof. The proof is beyond the scope of this class. See, Spectral and pseudospectral methods for advection equations by Joseph E. Pasciak, Math. Comp. 35 (1980), no. 152, 1081-1092 for details. □

9. Lecture 9: Proof of Theorem 8.3.

Proof of Theorem 8.3. We begin by proving the identity

$$(DFT)_+^{-1} = \frac{1}{2N+1} DFT_- \qquad .$$

For $c \in \mathbb{C}^{2N+1}$, we have

$$DFT_+(c)(j) = \sum_{l=0}^{2N} c_l E(jl),$$

where $E(jl) = e^{\frac{2\pi i jl}{2N+1}}$. Then,

$$\begin{aligned} DFT_-(DFT_+(c))(m) &= \sum_{j=0}^{2N} DFT_+(c)(j)E(-jm) = \sum_{j=0}^{2N}\left(\sum_{l=0}^{2N} c_l E(lj)\right)E(-jm) \\ &= \sum_{j=0}^{2N}\sum_{l=0}^{2N} c_l E((l-m)j) = \sum_{l=0}^{2N} c_l \left(\sum_{j=0}^{2N} E((l-m)j)\right). \end{aligned}$$

Now, if $l = m$,

$$\sum_{j=0}^{2N} E((l-m)j) = \sum_{j=0}^{2N} 1 = 2N+1.$$

If $l \neq m$,

$$E((l-m)j) = e^{\frac{2\pi i (l-m)j}{2N+1}} = \xi^j,$$

with

$$\xi := e^{\frac{2\pi i (l-m)}{2N+1}}.$$

Therefore, we have

$$\begin{aligned} \sum_{j=0}^{2N} E((l-m)j) &= 1 + \xi + ... + \xi^{2N} \\ &= \frac{1-\xi^{2N+1}}{1-\xi} = \frac{1 - e^{\frac{2\pi i (l-m)j\,\cancel{(2N+1)}}{\cancel{2N+1}}}}{1-\xi} = \frac{1-1}{1-\xi} = 0 \end{aligned}$$

so that

$$DFT_-(DFT_+(c))(m) = (2N+1)c_m. \tag{9}$$

Note that $DFT_\pm$ applied to a vector c corresponds to multiplication by the square matrix $M^\pm$ defined by

$$M^\pm_{jl} = E(\pm jl), \quad j, l = 0, 1 \ldots, , 2N$$

and hence (9) is the same as

$$((2N+1)^{-1}M^-)M^+ = I.$$

□

Remark 9.1 (Computational Cost using the DFT)*.* We recall that

$$DFT_{\pm}(c)(j) = \sum_{l=0}^{2N} c_l E(lj).$$

Each value j requires $2N+1$ multiplications (complex) and $2N$ additions or a total of $O(N)$ flops (floating point operations). This entails an overall cost of $O(N^2)$ to compute all the output values.

Remark 9.2 (The Fast "Discrete" Fourier Transform - FFT)*.* If the number of points, k, is highly factorable, the k-point FFT algorithm will compute the DFTs in $O(k\log(k))$ operations.

9.1. **Trigonometric interpolation using $2N$ points.** To take advantage of the FFT, we shall consider problems with, for example, $2N := 2^k$ points, so consider the interpolation problem with $2N$ points: Find

$$f_{2N}(x) = \sum_{j=-N+1}^{N} c_j \psi_j(x) \tag{10}$$

such that

$$f_{2N}(x_j) = f(x_j), \qquad j = 0, 1, ..., 2N-1, \tag{11}$$

where $\psi_j(x) = e^{ijx}$ and $x_j = jh$ with $h = \pi/N$.

The discrete Fourier transforms using $2N$ points read

$$DFT_{\pm} : (c_0, ..., c_{2N-1}) \subset \mathbb{C}^{2N} \to \mathbb{C}^{2N},$$

where

$$DFT_{\pm}(c)(j) = \sum_{l=0}^{2N-1} c_l E(\pm lj), \qquad j = 0, ..., 2N-1,$$

and

$$E(m) = e^{\frac{2\pi i m}{2N}}.$$

The analogue of Theorem 8.3 holds, namely,

$$DFT_{+}^{-1} = \frac{1}{2N} DFT_{-}.$$

Hence, the coefficients appearing in (10) solving the trigonometric interpolation problem (11) are given by

$$c_j = \left\{ \begin{array}{ll} d_j & \text{for } j = 0, ..., N \\ d_{j+2N} & \text{for } j = -N+1, ..., -1, \end{array} \right.$$

where

$$d = \frac{1}{2N} DFT_{-}(F), \qquad \text{with } F_j = f(x_j), \quad j = 0, 1, ..., 2N-1.$$

The above formulas will be used in the next homework assignment.

10. Lecture 10.

10.1. **Extension of Trigonometric interpolation.** On $[0, \pi]$ use

$$\mathbb{V}_{N-1} = \text{span}\{\sin(jx) \ : \ j = 1, 2, ..., N-1\}$$

so that the interpolation problem reads: find $S_N \in \mathbb{V}_{N-1}$ satisfying

$$S_N(x_j) = f(x_j), \qquad \text{for } j = 1, ..., N-1,$$

with $x_j = \frac{\pi}{N} j = hj$ where $h := \frac{\pi}{N}$.

Remark 10.1 (Existence and Uniqueness). The above interpolation problem has a unique solution. Moreover, computing the coefficients c_j in the interpolating polynomial

$$S_N(x) = \sum_{j=1}^{N-1} c_j \sin(jx)$$

can be done using $FFT's$ of size $2N$.

Remark 10.2 (Spectral Convergence). This sine approximation also exhibits spectral convergence as in Theorem 8.2.

On $[0, \pi]$ one can also use

$$\mathbb{V}_{N+1} = \text{span}\{\cos(jx) \ : \ j = 0, 1, ..., N\}.$$

In this case, $x_j = hj$ for $j = 0, ..., N$ and the interpolation problem consists in finding $C_{N+1} \in \mathbb{V}_{N+1}$ such that

$$C_{N+1}(x_j) = f(x_j), \qquad j = 0, ..., N.$$

As in the sine case, similar remarks holds.

10.2. **Numerical Differentiation.** We start with the simplest example: use the difference quotient from the definition of the derivative, i.e.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}, \qquad h > 0. \tag{12}$$

This is called a forward difference. Similarly,

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}, \qquad h > 0. \tag{13}$$

is called a backward difference.

10.3. **Error estimates.** The Taylor series around x reads

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$$

for some $\xi \in (x, x+h)$. Whence,

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \underbrace{\frac{h}{2} f''(\xi)}_{O(h)}.$$

This implies that error due to the approximation (12) is $O(h)$ or first order!

The same property holds for (13) since

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(\xi)$$

for some $\xi \in (x-h, x)$ and so

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \underbrace{\frac{h}{2} f''(\xi)}_{O(h)}.$$

Example 10.1. *[Method of Undetermined Coefficients] We propose to find an approximation of the derivative of highest possible order of the form*

$$f'(x) \approx af(x) + bf(x-h) + cf(x-2h).$$

To do this, we use the method of undetermined coefficients. *We write the Taylor's series around x and expand $f(x-h)$ and $f(x-2h)$*

$$\begin{aligned} f(x) &= f(x) \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2} f''(x) + ... \\ f(x-2h) &= f(x) - 2hf'(x) + 2h^2 f''(x) + ... \end{aligned}$$

Now, use these expression to expand $af(x) + bf(x-h) + cf(x-2h)$ and regroups terms with the same power of h

$$\begin{aligned} af(x) + bf(x-h) + cf(x-2h) &= (a+b+c)f(x) - (b+2c)hf'(x) \\ &\quad + \frac{h^2}{2}(b+4c)f''(x) - \frac{h^3}{6}(b+8c)f'''(x) + ... \end{aligned}$$

We want this to equal $f'(x) + O(h^r)$ for r as large as possible. To do this, we constraint the coefficients to satisfy

$$\begin{aligned} a + b + c &= 0 \\ -(b + 2c)h &= 1 \\ b + 4c &= 0. \end{aligned}$$

This has a unique solution. Indeed, it is equivalent to (upon multiplying the second constraint by $-1/h$)

$$\underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 4 \end{pmatrix}}_{=:A} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ -1/h \\ 0 \end{pmatrix}.$$

We compute $\det(A) = 1.\det\begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix} = 2 \neq 0$. *This implies that the system has a unique solution and we cannot get rid of higher order terms. The solution is given by $a = \frac{3}{2h}$, $b = -\frac{2}{h}$ and $c = \frac{1}{2h}$ so that the desired approximation is*

$$f'(x) \approx \frac{\frac{3}{2}f(x) - 2f(x-h) + \frac{1}{2}f(x-2h)}{h}.$$

To derive an expression of the error, we use Taylor series with remainder (at the lowest remaining term), i.e.

$$\begin{aligned} f(x) &= f(x) \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\xi_1), \qquad \xi_1 \in (x-h, x) \\ f(x-2h) &= f(x) - 2hf'(x) + 2h^2 f''(x) - \frac{8h^3}{6}f'''(\xi_2), \qquad \xi_2 \in (x-2h, x). \end{aligned}$$

Then, we compute

$$\begin{aligned} \frac{\frac{3}{2}f(x) - 2f(x-h) + \frac{1}{2}f(x-2h)}{h} &= f'(x) - \frac{h^3}{6}\left(-\frac{2}{h}f'''(\xi_1) + \frac{8}{2h}f'''(\xi_2)\right) \\ &= f'(x) + O(h^2). \end{aligned}$$

This is the highest order method of this form and its order is 2.

Example 10.2 (Second Derivatives). *We use the method of undetermined coefficients to get a difference approximation to*

$$-f''(x) \approx af(x-h) + bf(x) + cf(x+h)$$

of highest possible order. We have

$$\begin{aligned} f(x) &= f(x) \\ f(x \pm h) &= f(x) \pm hf'(x) + \frac{h^2}{2}f''(x) \pm \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(x) \pm ... \end{aligned}$$

Hence,

$$\begin{aligned} af(x-h) + bf(x) + cf(x+h) =& (a+b+c)f(x) + h(c-a)f'(x) + \frac{h^2}{2}(c+a)f''(x) \\ &+ \frac{h^3}{6}(c-a)f'''(x) + \frac{h^4}{24}(c+a)f^{(4)}(x) + ... \end{aligned}$$

To have this expression agree with $-f''(x)$ *to highest order we set*

$$\begin{aligned} a+b+c &= 0 \\ c-a &= 0 \\ \frac{h^2}{2}(c+a) &= -1. \end{aligned}$$

This is probably as far as we can go and leads to the system

$$\underbrace{\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}}_{=:A} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -2/h^2 \end{pmatrix}.$$

We compute $\det(A) = -1 \det \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} = 2 \neq 0$*. The unique solution is given by* $a = -\frac{1}{h^2}$*,* $b = \frac{2}{h^2}$ *and* $c = -\frac{1}{h^2}$ *so that the desired approximation is*

$$f''(x) \approx \frac{-f(x-h) + 2f(x) - f(x+h)}{h^2}.$$

Note that $c = a$ *implies that the leading error term is the one involving* $f^{(4)}$*. The Taylor series with remainder (at that term) are*

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{h^2}{2}f''(x) \pm \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\xi_\pm),$$

where $\xi_+ \in (x, x+h)$ *and* $\xi_- \in (x-h, x)$*. Thus,*

$$\frac{-f(x-h) + 2f(x) - f(x+h)}{h^2} = -f''(x) - \frac{h^2}{24}\left(f^{(4)}(\xi_+) + f^{(4)}(\xi_-)\right).$$

The approximation is second order but requires $f \in C^4[x-h, x+h]$*.*

11. Lecture 11: Differentiation via Polynomial Interpolation.

Suppose we want a differentiation formula of the form

$$f'(x) = \sum_{i=0}^{n} \alpha_i f(x_i),$$

with $\{x_0, ..., x_n\}$ distinct in $[a, b]$.

Let $p \in \mathbb{P}^n$ be the polynomial interpolating f at $x_0, ..., x_n$, i.e.

$$p(x) = \sum_{i=0}^{n} l_i(x) f(x_i),$$

where $l_i(x)$ are the Lagrange polynomials (see Section 3.2). Then $p'(x) = \sum_{i=0}^{n} l_i'(x) f(x_i)$ should approximate $f'(x)$. This is indeed the case at $x = x_i$, $i = 0, ..., n$ but it is less clear what happen when x is not an interpolation point.

Theorem 11.1. *Assume that $f \in C^{n+1}[a, b]$, $\{x_0, ..., x_n\}$ distinct in $[a, b]$ and $p \in \mathbb{P}^n$ interpolates f at $x_0, ..., x_n$. Then, there exists $\xi_j \in [a, b]$ such that*

$$f'(x_j) - p'(x_j) = \frac{f^{(n+1)}(\xi_j)}{(n+1)!} \prod_{l \neq j} (x_j - x_l).$$

Proof. For $x \notin \{x_0, ..., x_n\}$, define

$$\Theta(x) := \frac{f(x) - p(x)}{\prod_{i=0}^{n} (x - x_i)} (n+1)! \qquad .$$

Note that Theorem 4.1 guarantees that

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

so

$$\Theta(x) = f^{(n+1)}(\xi_x)$$

Since we do not know what happens to ξ_x when $x \to x_j$, we cannot use this representation further but it shows that

$$\Theta(x) \in \text{Range}(f^{(n+1)}), \qquad \text{for } x \in [a, b] \setminus \{x_0, x_1, \dots, x_n\}.$$

Instead, we evaluate

$$\lim_{x \to x_j} \Theta(x) = \lim_{x \to x_j} \frac{(f(x) - p(x))(n+1)!}{\prod_{i=0}^{n} (x - x_i)}.$$

This is a 0/0 type of indetermination so we can use L'Hospital's rule:

$$\lim_{x\to x_j}\Theta(x)=\lim_{x\to x_j}\frac{(f'(x)-p'(x))(n+1)!}{\left(\prod_{i=0}^n(x-x_i)\right)'}.$$

We now compute the denominator

$$\left(\prod_{i=0}^n(x-x_i)\right)'=\sum_{i=0}^n\prod_{l\neq i}(x-x_l).$$

Each product of the above sum has a factor $(x-x_j)$ except when $i=j$ and so the only one that is not vanishing when $x\to x_j$ is the j'th. This implies that

$$\lim_{x\to x_j}\left(\prod_{i=0}^n(x-x_i)\right)'=\prod_{l\neq j}(x_j-x_l)$$

and thus

$$\lim_{x\to x_j}\Theta(x)=\frac{(f'(x_j)-p'(x_j))(n+1)!}{\prod_{l\neq j}(x_j-x_l)}.$$

Now, $f^{(n+1)}$ is continuous on $[a,b]$ by assumption and we have already seen that

$$\Theta(x)\in\text{Range}(f^{(n+1)}),\qquad\text{for } x\in[a,b].$$

As the range of a continuous function on $[a,b]$ is a closed set,

$$\lim_{x\to x_j}\Theta(x)\in\text{Range}(f^{(n+1)})$$

or, there exists $\xi_j\in[a,b]$ with

$$\lim_{x\to x_j}\Theta(x)=f^{(n+1)}(\xi_j).$$

Thus

$$f^{(n+1)}(\xi_j)=\frac{(f'(x_j)-p'(x_j))(n+1)!}{\prod_{l\neq j}(x_j-x_l)},$$

which is the desired estimate after simple algebraic manipulations. □

Remark 11.1 (Continuity of $f^{(n+1)}(\xi_x)$). Within the proof of the above theorem, we actually showed that the function

$$x\mapsto f^{(n+1)}(\xi_x)$$

is continuous on $[a,b]$ provided $f\in C^{(n+1)}[a,b]$. This fact will be used later.

Example 11.1 (A 3 point differentiation scheme)**.** *Use polynomial interpolation to derive an approximation to the derivative of the form*

$$f'(x) \approx af(x) + bf(x-h) + cf(x-2h).$$

We first compute the Lagrange basis for the interpolation points $\{x, x-h, x-2h\}$ *(we use* t *for the variable for a fixed* x*)*

$$\begin{aligned} l_0(t) &= \frac{(t-(x-h))(t-(x-2h))}{(x-(x-h))(x-(x-2h))} = \frac{t^2-(2x-3h)t+(x-h)(x-2h)}{2h^2};\\ l_1(t) &= \frac{(t-x)(t-(x-2h))}{((x-h)-x)((x-h)-(x-2h))} = \frac{t^2-(2x-2h)t+x(x-2h)}{-h^2};\\ l_2(t) &= \frac{(t-x)(t-(x-h))}{(x-2h-x)(x-2h-(x-h))} = \frac{t^2-(2x-h)t+x(x-h)}{2h^2}. \end{aligned}$$

The associated interpolant is

$$p(t) = l_0(t)f(x) + l_1(t)f(x-h) + l_2(t)f(x-2h)$$

so that

$$p'(x) \approx l_0'(x)f(x) + l_1'(x)f(x-h) + l_2'(x)f(x-2h).$$

This means

$$\begin{aligned} f'(x) \approx& \frac{2x-(2x-3h)}{2h^2}f(x) + \frac{2x-(2x-2h)}{-h^2}f(x-h) + \frac{2x-(2x-h)}{2h^2}f(x-2h)\\ &= \frac{3}{2h}f(x) - \frac{2}{h}f(x-h) + \frac{1}{2h}f(x-2h). \end{aligned}$$

The error term is

$$\begin{aligned} f'(x) - \left(\frac{3}{2h}f(x) - \frac{2}{h}f(x-h) + \frac{1}{2h}f(x-2h)\right) &= \frac{f'''(\xi_0)}{6}(x-(x-h))(x-(x-2h))\\ &= \frac{f'''(\xi_0)}{3}h^2 \end{aligned}$$

and hence the scheme is second order in h*.*

Remark 11.2. This formula was already derived using the method of undetermined coefficients, see Example 10.1. Note, however, that the error terms are slightly different.

Remark 11.3. Suppose that $[a,b]$ is of order h, i.e., $b-a = O(h)$ and $p(t)$ interpolates $f(t) \in C^{n+1}[a,b]$ at $n+1$ distinct nodes $\{x_0, x_1, \dots, x_n\} \subset [a,b]$. Then, by Theorem 4.1,

$$|f(t) - p(t)| = O(h^{n+1}), \quad \text{for } t \in [a,b]$$

and by Theorem 11.1,

$$|f'(x_j) - p'(x_j)| \le O(h^n), \quad \text{for } j = 0, 1, \dots, n.$$

12. Lecture 12: Numerical Integration.

We use polynomial interpolation techniques to derive numerical integration schemes to approximate

$$I(f) = \int_\alpha^\beta f(x)\ dx,$$

for $\alpha < \beta$. Let $\{x_0, ..., x_n\} \subset [a, b]$ be distinct, where $a < b$ are such that $[\alpha, \beta] \subseteq [a, b]$. Let $p \in \mathbb{P}^n$ interpolates f at $\{x_0, ..., x_n\}$. We propose to approximate $I(f)$ by

$$Q(f) = \int_\alpha^\beta p(x)\ dx.$$

Using the Lagrange polynomials $\{l_i(x)\}_{i=0}^n$ associated with the interpolations points $\{x_i\}_{i=0}^n$, we write

$$p(x) = \sum_{i=0}^n f(x_i) l_i(x)$$

so that

$$\begin{aligned} Q(f) &= \int_\alpha^\beta \left(\sum_{i=0}^n f(x_i) l_i(x) \right)\ dx = \sum_{i=0}^n f(x_i) \int_\alpha^\beta l_i(x)\ dx \\ &= \sum_{i=0}^n w_i f(x_i), \end{aligned}$$

where we defined

$$w_i := \int_\alpha^\beta l_i(x)\ dx. \tag{14}$$

This leads to the following definition of quadrature approximation.

Definition 12.1 (Quadrature)**.** *An integral approximation of the form*

$$I(f) \approx Q(f) = \sum_{i=0}^n w_i f(x_i)$$

is called a quadrature*. The real numbers $\{w_i\}$ are the weights and $\{x_i\}$ are the nodes.*

Remark 12.1. This is not the most general form of a quadrature approximations. It is possible to introduce formulas having derivatives at the nodes as well however such formulas will not be be studied in this course.

Example 12.1 (Rectangle quadrature)**.** *Let $x_0 \in [a, b]$. Find the quadrature approximating*

$$I(f) = \int_a^b f(x)\ dx$$

based on polynomial interpolation using x_0 and $\mathbb{P}^0$. In this case, $l_0 = 1$ so $p(x) = f(x_0)l_0(x) = f(x_0)$ and

$$Q(f) = \int_a^b f(x_0)\ dx = (b-a)f(x_0),$$

which is the area of the shaded region in Figure 4.

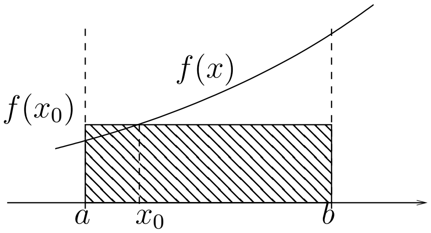

FIGURE 4. Rectangle quadrature. The approximation $Q(f)$ corresponds to the area of the shaded region.

Example 12.2 (Trapezoidal quadrature)**.** *Consider $p \in \mathbb{P}^1$ interpolating f at $x_0 = a$ and $x_1 = b$ to approximate*

$$I(f) = \int_a^b f(x)\ dx.$$

In that case,

$$p(x) = l_0(x)f(a) + l_1(x)f(b) = \frac{b-x}{b-a}f(a) + \frac{x-a}{b-a}f(b)$$

so that

$$Q(f) = \int_a^b p(x)\ dx = \frac{f(a)}{b-a}\int_a^b (b-x)\ dx + \frac{f(b)}{b-a}\int_a^b (x-a)\ dx.$$

Both integrals equal $\frac{1}{2}(b-a)^2$ so

$$Q(f) = \frac{b-a}{2}\left(f(a) + f(b)\right).$$

See Figure 5 for an illustration.

Example 12.3 (A 3 Point Quadrature)**.** *Consider the interpolation nodes $\{x, x-h, x-2h\}$ for some $h > 0$ and $x \in \mathbb{R}$ and use a quadratic polynomial interpolant to derive a quadrature scheme to approximate*

$$I(f) = \int_{x-h}^x f(t)\ dt.$$

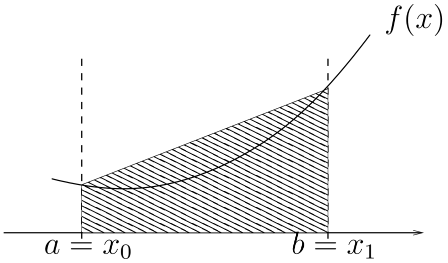

FIGURE 5. Trapezoidal quadrature. The approximation $Q(f)$ corresponds to the area of the shaded region.

Here x and h are parmeters. Note that this will use interpolation points outside the integration region. In Example 11.1, we have already computed the corresponding Lagrange polynomials l_0, l_1 and l_2. By (14)*, we need to compute*

$$w_i = \int_{x-h}^{x} l_i(t)\ dt.$$

We leave this as an exercise.

Remark 12.2. The weights in the above quadrature end up being independent of x but not h. This is an important example in that the quadrature will be used to develop a multistep ODE approximation later in this course. We will develop better ways to derive its weights.

The next theorem addresses how well $Q(f)$ approximate $I(f)$. One part of it depends on what we will call the "mean value theorem for integrals" (following the textbook by Kincaid and Cheney mentioned in the syllabus). This states that if $g(x)$ is continuous on $[a, b]$ and $f(x)$ does not change sign on $[a, b]$, then there is a $\zeta \in (a, b)$ such that

$$\int_a^b f(x)g(x)\,dx = g(\zeta)\int_a^b f(x)\,dx.$$

A warning: there are other definitions of the MVT for integrals in the literature.

Theorem 12.1 (Interpolation error)**.** *Let $f \in C^{(n+1)}[a, b]$, $\{x_0, ..., x_n\}$ distinct in $[a, b]$ and $a \le \alpha < \beta \le b$. If $p \in \mathbb{P}^n$ inteprolates f at x_i, $i = 0, ..., n$, and $Q(f) = \sum_{i=0}^n w_i f(x_i)$, with $w_i = \int_\alpha^\beta l_i(x)\ dx$, then we have*

$$I(f) - Q(f) = \frac{1}{(n+1)!}\int_\alpha^\beta f^{(n+1)}(\xi_x)\prod_{i=0}^{n}(x - x_i)\ dx,$$

where for every $x \in [\alpha,\beta]$, $\xi_x \in [a,b]$. Moreover, if $\prod_{i=0}^n (x-x_i)$ does not change sign on $[a,b]$, then there exists $\xi \in [a,b]$ with

$$I(f) - Q(f) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_\alpha^\beta \prod_{i=0}^n (x-x_i)\ dx.$$

Proof. In view of the interpolation error provided by Theorem 4.1, for $x \in (\alpha,\beta)$ there exists $\xi_x \in (a,b)$ such that

$$I(f)-Q(f) = \int_\alpha^\beta (f(x)-p(x))\ dx = \frac{1}{(n+1)!}\int_\alpha^\beta f^{(n+1)}(\xi_x)\prod_{i=0}^n (x-x_i)\ dx.$$

This is the first claim. To continue further, it suffices to note that $f^{(n+1)}(\xi_x)$ is continuous (see Remark 11.1) and invoke the mean value theorem for integrals to write

$$\frac{1}{(n+1)!}\int_\alpha^\beta f^{(n+1)}(\xi_x)\prod_{i=0}^n (x-x_i)\ dx = \frac{f^{(n+1)}(\xi)}{(n+1)!}\int_\alpha^\beta \prod_{i=0}^n (x-x_i)\ dx,$$

for some $\xi \in (a,b)$. This implies the second claim. □

Example 12.4 (Error for the Trapezoidal quadrature)**.** *For some $\xi \in (a,b)$ the above theorem guarantees that*

$$\int_a^b f(x)\ dx - \frac{b-a}{2}\left(f(a)+f(b)\right) = \frac{f''(\xi)}{2}\int_a^b (x-a)(x-b)\ dx$$

using the fact that $(x-a)(x-b)$ does not change sign. Hence, computing the integral leads to

$$\begin{aligned}\int_a^b f(x)\ dx - \frac{b-a}{2}\left(f(a)+f(b)\right) &\overset{y=x-a}{=} \frac{f''(\xi)}{2}\int_0^{b-a} y(y-(b-a))\ dy\\ &= -\frac{f''(\xi)}{12}(b-a)^3.\end{aligned}$$

Example 12.5 (Simpson rule)**.** *We want to approximate*

$$I(f) = \int_{-1}^1 f(x)\ dx$$

using a polynomial of degree 2 interpolating f at $x_0 = -1$, $x_1 = 0$ and $x_2 = 1$. We first compute the lagrange polynomials

$$l_0(x) = \frac{(x-0)(x-1)}{(-1-0)(-1-1)} = \frac{x^2 - x}{2}$$
$$l_1(x) = \frac{(x+1)(x-1)}{(0+1)(0-1)} = 1 - x^2$$
$$l_2(x) = \frac{(x-0)(x+1)}{(1-0)(1+1)} = \frac{x^2 + x}{2}$$

Therefore,

$$w_0 = \int_{-1}^{1} l_0(x)\ dx = \int_{-1}^{1} \frac{1}{2}(x^2 - x)\ dx = \frac{1}{3}$$
$$w_1 = \int_{-1}^{1} l_1(x)\ dx = \int_{-1}^{1} (1 - x^2)\ dx = \frac{4}{3}$$
$$w_2 = \int_{-1}^{1} l_3(x)\ dx = \int_{-1}^{1} \frac{1}{2}(x^2 + x)\ dx = \frac{1}{3}.$$

and the quadrature rule reads

$$I(f) \approx Q(f) = \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1).$$

Remark 12.3. Note that for Simpson's rule,

$$\prod_{i=0}^{2}(x - x_i) = (x+1)x(x-1) = (x^2 - 1)x$$

changes sign on $[-1, 1]$ so it is not possible to derive a formula of the form

$$I(f) - Q(f) = cf^{(3)}(\zeta)$$

using the MVT for integrals. However, it is possible to derive a formula of the form

$$I(f) - Q(f) = -\frac{1}{90}f^{(4)}(\zeta) \tag{15}$$

but requires using more powerful tools, This can be shown using Sard's theory of approximating functions and the Peano Kernal Theorem, (Kincaid and Cheney, Section 7.6).

We now discuss a possibly simpler way to compute the weights w_i. This relies on the observation that the interpolant of $p \in \mathbb{P}^n$ is p itself. This means that

$$I(p) = \int_a^b p(x)\ dx = \sum_{i=0}^{n} w_i p(x_i) = Q(p).$$

In other words, the quadrature is *exact* for $p \in \mathbb{P}^n$. In particular, this means that

$$\begin{aligned} J_0 &:= \int_a^b 1 \ dx = \sum_{i=0}^n w_i 1 = Q(1) \\ J_1 &:= \int_a^b x \ dx = \sum_{i=0}^n w_i x_i = Q(x) \\ &\vdots \\ J_n &:= \int_a^b x^n \ dx = \sum_{i=0}^n w_i x_i^n = Q(x^n). \end{aligned}$$

Hence, the weights $\{w_i\}_{i=0}^n$ satisfy the linear system

$$A^t w := \begin{pmatrix} 1 & 1 & 1 & ... & 1 \\ x_0 & x_1 & x_2 & ... & x_n \\ x_0^2 & x_1^2 & x_2^2 & ... & x_n^2 \\ & & \vdots & & \\ x_0^n & x_1^n & x_2^n & ... & x_n^n \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} J_0 \\ J_1 \\ \vdots \\ J_n \end{pmatrix}.$$

Recall that you get the linear system

$$Ac := \begin{pmatrix} 1 & x_0 & x_0^2 & ... & x_0^n \\ 1 & x_1 & x_1^2 & ... & x_1^n \\ 1 & x_2 & x_2^2 & ... & x_2^n \\ & & \vdots & & \\ 1 & x_n & x_n^2 & ... & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix},$$

when solving the interpolation problem $p(x_i) = y_i$, where

$$p(x) = c_0 + c_1 x + ... c_n x^n.$$

We already know that A is non-singular and it follows that A^t is also non-singular. As a consequence, we realize that

Remark 12.4 (Uniqueness of Weights). The weights w_i making the quadrature Q exact on $\mathbb{P}^n$ are uniquely determined from the exactness conditions

$$Q(x^j) = \sum_{i=0}^n w_i x_i^j = \int_a^b x^j \ dx, \qquad j = 0, ..., n.$$

Example 12.6 (Simpson's quadrature)**.**

$$Q(f) = w_0 f(-1) + w_1 f(0) + w_2 f(1) \approx \int_{-1}^1 f(x) \ dx.$$

The exactness conditions (for $\mathbb{P}^2$) are

$$\begin{aligned} 2 &= \int_{-1}^{1} 1 \; dx = w_0(1) + w_1(1) + w_2(1) \\ 0 &= \int_{-1}^{1} x \; dx = w_0(-1) + w_1(0) + w_2(1) \\ \frac{2}{3} &= \int_{-1}^{1} x^2 \; dx = w_0(-1)^2 + w_1(0)^2 + w_2(1)^2, \end{aligned}$$

i.e.

$$w_0 = w_2 = \frac{1}{3} \qquad \textit{and } w_1 = \frac{4}{3}$$

as previously obtained.

Remark 12.5 (Higher order exactness for Simpson). Note that in addition of being exact for any polynomial of degree 2, the Simpson's quadrature rule also satisfies

$$0 = \int_{-1}^{1} x^3 \; dx = \frac{1}{3}(-1)^3 + 0 + \frac{1}{3}(1)^3$$

while

$$\frac{2}{5} = \int_{-1}^{1} x^4 \; dx \neq \frac{1}{3}(-1)^4 + 0 + \frac{1}{3}(1)^4 = \frac{2}{3}.$$

Hence, the Simpson's quadrature rule is exact for $\mathbb{P}^3$ but not $\mathbb{P}^4$. This is the reason that $f^{(4)}$ appears in (15).

13. Lecture 13.

The last lecture introduced Simpson's rule:

$$I(f) := \int_{-1}^{1} f(x)\, dx \approx \frac{1}{3} f(-1) + \frac{4}{3} f(0) + \frac{1}{3} f(1) =: Q(f).$$

We found that Q was exact for cubics.

Consider instead the quadrature scheme based on the nodes $\{x_0, x_1, x_2, x_3\} := \{-1, 0, 1/2, 1\}$, i.e.

$$I(f) \approx Q(f) = \sum_{i=0}^{3} w_i f(x_i).$$

We also saw during the last lecture that there exists a unique set of weights w_i, $i = 0, 1, 2, 3$ making Q exact for cubics (see Remark 12.4). Note that the Simpson's scheme can be interpreted as

$$Q(f) = \frac{1}{3} f(-1) + \frac{4}{3} f(0) + 0 f(1/2) + \frac{1}{3} f(1),$$

is exact on cubics and hence is the unique scheme based on these nodes. Applying the error formula provided by Theorem 12.1 gives

$$I(f) - Q(f) = \frac{1}{4!} \int_{-1}^{1} f^{(4)}(\xi_x) \prod_{i=0}^{3} (x - x_i)\, dx.$$

The quantity on the right side of the above relation is usually quite large. To get quadrature to approximate integrals, we need *composite schemes.*

13.1. **Composite Schemes.**

Suppose you have a scheme

$$Q(f) = \sum_{i=0}^{n} w_i f(x_i) \approx \int_a^b f(x)\, dx = I(f),$$

exact on $\mathbb{P}^n$, where $\{x_0, ..., x_n\}$ are distinct in $[a, b]$.

We want to deduce a scheme on $[\alpha, \beta]$ from that on $[a, b]$. Let λ be the linear mapping taking $[a, b]$ onto $[\alpha, \beta]$, i.e.

$$\lambda(x) = \alpha + \frac{x - a}{b - a} (\beta - \alpha)$$

($\lambda(a) = \alpha$ and $\lambda(b) = \beta$). Also,

$$\lambda^{-1}(t) = a + \frac{t - \alpha}{\beta - \alpha} (b - a)$$

is a linear mapping from $[\alpha,\beta]$ to $[a,b]$. Now, for $q\in\mathbb{P}^n$

$$\tilde{I}(q) := \int_\alpha^\beta q(t)\ dt = \frac{\beta-\alpha}{b-a}\int_a^b q(\lambda(x))\ dx,$$

where we have used the change of variable $x=\lambda^{-1}(t)$ or $t=\lambda(x)$ so that $\frac{dx}{dt}=\frac{b-a}{\beta-\alpha}$ and $dt=\frac{\beta-\alpha}{b-a}dx$. Note that if $q\in\mathbb{P}^n$ is defined on $[\alpha,\beta]$, then $q\circ\lambda$ is a polynomial in $\mathbb{P}^n$ defined on $[a,b]$ (we leave this as an exercise). Since the quadrature scheme is exact on $\mathbb{P}^n$,

$$\tilde{I}(q) := \frac{\beta-\alpha}{b-a}\sum_{i=0}^n w_i q(\lambda(x_i)).$$

We set

$$\tilde{w_i} = \frac{\beta-\alpha}{b-a}w_i \qquad \text{and} \qquad \tilde{x}_i=\lambda(x_i) \tag{16}$$

to deduce that

$$\int_\alpha^\beta q(t)dt = \sum_{i=0}^n \tilde{w}_i q(\tilde{x}_i) =: \tilde{Q}(q).$$

In conclusion, given a scheme

$$I(f)=\int_a^b f(x)\ dx \approx Q(f)=\sum_{i=0}^n w_i f(x_i)$$

which is exact on $\mathbb{P}^n$, we get a *translated* scheme

$$\tilde{I}(f)=\int_\alpha^\beta f(t)\ dt \approx \tilde{Q}(f)=\sum_{i=0}^n \tilde{w}_i f(\tilde{x}_i)$$

which is also exact on $\mathbb{P}^n$ using the notation (16).

Remark 13.1. Note that reversing the argument shows that if the translated quadrature is exact on $\mathbb{P}^n$, so is the original. This means that if n is the maximum order of exactness for the original scheme then it is the maximum order of exactness for the translated scheme and visa versa.

Remark 13.2 (Property of the translated scheme). The map λ is a linear map of $[a,b]$ onto $[\alpha,\beta]$ so it maps points in a proportional way: $a\to\alpha$ and $b\to\beta$ implies $(a+b)/2\to(\alpha+\beta)/2$. More generally, for any $t\in[0,1]$

$$[a,b]\ni ta+(1-t)b\to t\alpha+(1-t)\beta\in[\alpha,\beta].$$

13.2. **Composite Quadrature.** We want to approximate

$$I(f) = \int_a^b f(x)\ dx$$

and introduce $N+1$ distinct points

$$a = x_0 < x_1 < ... < x_N = b$$

and set $h = \max_{i=1,...,N}(x_i - x_{i-1})$. Hence, we split the integral over $[a,b]$ onto N pieces

$$I(f) = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} f(x)\ dx$$

and use a fixed quadrature scheme translated to each sub-interval in the partitioning.

13.3. **Simpson's Composite Quadrature Rule.** If we use the Simpson's rule

$$\int_{-1}^{1} g(t)\ dt \approx \frac{1}{3}g(-1) + \frac{4}{3}g(0) + \frac{1}{3}g(1)$$

to approximate

$$\int_{x_{i-1}}^{x_i} f(x)\ dx,$$

we have

$$\int_{x_{i-1}}^{x_i} f(x)\ dx \approx \sum_{i=0}^{2} \tilde{w}_i f(\tilde{x}_i),$$

where

$$\tilde{w}_2 = \tilde{w}_0 = \frac{x_i - x_{i-1}}{2}\frac{1}{3} \qquad \text{and} \qquad \tilde{w}_1 = \frac{x_i - x_{i-1}}{2}\frac{4}{3}$$

and the nodes are moved proportionally

$$-1 \to x_{i-1}, \qquad 1 \to x_i \qquad \text{and} \qquad 0 \to \frac{x_{i-1}+x_i}{2}.$$

This implies

$$\int_{x_{i-1}}^{x_i} f(x)\ dx \approx \frac{x_i - x_{i-1}}{6}\left(f(x_{i-1}) + 4f(x_{i-1/2}) + f(x_i)\right),$$

where

$$x_{i-1/2} := \frac{x_{i-1}+x_i}{2}.$$

Gathering all the approximations in all subinterval, we arrive at the *composite Simpson's rule* approximation

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^N \frac{x_i - x_{i-1}}{6}\big(f(x_{i-1}) + 4f(x_{i-1/2}) + f(x_i)\big) =: \sum_{i=1}^N \tilde{Q}_i(f).$$

The integration error will still be based on the error formula of Theorem 12.1 but applied on the subintervals. As discussed above, Simpson's rule is the unique scheme which is exact on $\mathbb{P}^3$ based on the nodes $\{-1, 0, 1/2, 1\}$. This means that the translated scheme on the interval $[x_{i-1}, x_i]$ is the unique scheme based on the nodes

$$\{x_{i-1}, \frac{1}{2}(x_{i-1}+x_i), \frac{3}{4}x_i + \frac{1}{4}x_{i-1}, x_i\} := \{\tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3\},$$

which is exact on cubics. Theorem 12.1 gives

$$\int_{x_{i-1}}^{x_i} f(x)\ dx - \tilde{Q}_i(f) = \frac{1}{24}\int_{x_{i-1}}^{x_i} f^{(4)}(\xi_x)\prod_{j=0}^3 (x - \tilde{x}_j)\ dx,$$

provided that $f \in C^4[a,b]$. Let $\|f^{(4)}\|_\infty = \max_{t\in[a,b]} |f^{(4)}(t)|$, then

$$\left|\int_{x_{i-1}}^{x_i} f(x)\ dx - \tilde{Q}(f)\right| \le \frac{1}{24}\|f^{(4)}\|_\infty h^4 \int_{x_{i-1}}^{x_i} dx$$

(since $|x - \tilde{x}_i| \le h$ as $x, \tilde{x}_j \in [x_{i-1}, x_i]$). Therefore, we obtain that

$$\left|\int_{x_{i-1}}^{x_i} f(x)\ dx - \tilde{Q}(f)\right| \le \frac{1}{24}\|f^{(4)}\|_\infty h^4 (x_i - x_{i-1}). \tag{17}$$

Summing over all subintervals gives an estimate for the error

$$\begin{aligned}\left|\int_a^b f(x)\ dx - \sum_{i=1}^N \tilde{Q}_i(f)\right| &\le \sum_{i=1}^N \left|\int_{x_{i-1}}^{x_i} f(x)\ dx - \tilde{Q}_i(f)\right| \\ &\le \frac{b-a}{24}\|f^{(4)}\|_\infty h^4.\end{aligned} \tag{18}$$

Remark 13.3. In general, a quadrature rule which is exact on $\mathbb{P}^n$ translated to an interval of size h has a "local accuracy" of h^{n+2} as illustrated by (17). The "global accuracy" is the error for the integral over the full interval. For smooth problems, the global accuracy is of order h^{n+1} as illustrated by (18). This should be compared with Remark 11.3.

14. Lecture 14.

14.1. **Gaussian Quadrature.** We noted last lecture that the order of a quadrature is determined by exactness on $\mathbb{P}^n$. It is natural to optimize the order by allowing the nodes to move.

We start with examples.

Example 14.1 (One point)**.**

$$I(f) = \int_a^b f(x)\ dx \approx (b-a)f(x_i).$$

Note that the weight $(b-a)$ *makes the quadrature exact on constants. For the quadrature to be exact on linears, we need that*

$$\frac{b^2-a^2}{2} = \int_a^b x\ dx = (b-a)x_i,$$

or

$$x_i = \frac{b+a}{2},$$

which implies that

$$Q(f) = (b-a)f\left(\frac{b+a}{2}\right).$$

This is called the mid-point rule*. It is exact for linears but not quadratics since*

$$\frac{b^3-a^3}{3} = \int_a^b x^2 \neq (b-a)\left(\frac{a+b}{2}\right)^2.$$

Example 14.2 (A two point formula)**.** *The two-point formula has 4 unknowns (2 weights and 2 interpolation points). We show that we can determine these unknowns for the quadrature to be exact on cubics. For this, the following* 4 *exactness conditions must hold:*

$$\begin{aligned}
2 &= \int_{-1}^1 \ dx = w_1 + w_2 \\
0 &= \int_{-1}^1 x\ dx = w_1 x_1 + w_2 x_2 \\
\frac{2}{3} &= \int_{-1}^1 x^2\ dx = w_1 x_1^2 + w_2 x_2^2 \\
0 &= \int_{-1}^1 x^3\ dx = w_1 x_1^3 + w_2 x_2^3.
\end{aligned}$$

From the second condition, we deduce that $w_1x_1 = -w_2x_2$. *Putting this into the* 4*th condition yields*

$$0 = w_2x_2(x_1^2 - x_2^2).$$

Note that $w_2 \neq 0$ *and* $x_2 \neq 0$*, for otherwise it would be a one-point rule which, as we saw earlier, cannot be exact even for quadratics. Therefore, we must have* $x_1^2 = x_2^2$*, i.e.*

$$x_1 = -x_2$$

Using the second constraint again implies that

$$w_1 = w_2.$$

Now for the first constraint to hold:

$$w_1 + w_2 = 2 \qquad \Longrightarrow \qquad w_1 = w_2 = 1.$$

Finally, the third condition implies

$$\frac{2}{3} = 2x_1^2 \qquad or \qquad x_1 = \pm\sqrt{\frac{1}{3}}.$$

Finally, the scheme is

$$\int_{-1}^{1} f(x)\ dx \approx f\Big(-\frac{1}{\sqrt{3}}\Big) + f\Big(\frac{1}{\sqrt{3}}\Big) := Q(f)$$

and is exact on cubics. It is not exact on quartics since

$$\frac{2}{5} = \int_{-1}^{1} x^4\ dx \neq Q(x^4) = \frac{1}{9} + \frac{1}{9}.$$

Example 14.3. *[A three point formula] Can we make a three point quadrature rule exact on* $\mathbb{P}^5$*? Here the unknowns are* $\{w_i, x_i\}_{i=0}^2$. *We assume that the scheme is* symmetric *about the origin, i.e.*

$$Q(f) = w_1f(-x_1) + w_0f(0) + w_1f(x_1).$$

Notice that the symmetry implies that for all odd degree conditions:

$$0 = \int_{-1}^{1} x^{2j+1}\ dx = -w_1x_1^{2j+1} + w_0 0 + w_1x_1^{2j+1} = Q(x^{2j+1}), \qquad j \geq 0.$$

We now check the even degree conditions:

$$\begin{aligned} 2 &= \int_{-1}^{1} 1\ dx \overset{?}{=} 2w_1 + w_0 \\ \frac{2}{3} &= \int_{-1}^{1} x^2\ dx \overset{?}{=} 2w_1 x_1^2 \\ \frac{2}{5} &= \int_{-1}^{1} x^4\ dx \overset{?}{=} 2w_1 x_1^4. \end{aligned}$$

Divide the third relation by the second to get

$$\frac{3}{5} = x_1^2 \qquad \Longrightarrow \qquad x_1 = \sqrt{\frac{3}{5}}.$$

From the second relation we compute w_1:

$$\frac{1}{3} = w_1 \frac{3}{5} \qquad \Longrightarrow \qquad w_1 = \frac{5}{9}.$$

This in the first constraint implies that

$$\frac{10}{9} + w_0 = 2 \qquad \Longrightarrow \qquad w_0 = \frac{8}{9}$$

and the scheme reads

$$Q(f) = \frac{5}{9} f\Big(-\sqrt{3/5}\Big) + \frac{8}{9} f(0) + \frac{5}{9} f\Big(\sqrt{3/5}\Big).$$

This scheme is exact on $\mathbb{P}^5$ but not on $\mathbb{P}^6$.

Definition 14.1 (Gaussian Quadrature)**.** *A quadrature involving $n+1$ points, which is exact on $\mathbb{P}^{2n+1}$ is called a* Gaussian quadrature*.*

14.2. **Generalization: Weighted Gaussian Quadrature.** Given a non-negative weight functions $w(x)$ defined on $[a, b]$ and only vanishing at a discrete set of points, we want to derive Gaussian quadrature schemes such that

$$\int_a^b w(x) f(x)\ dx \approx \sum_{i=0}^{n} w_i f(x_i).$$

Notice that the assumption on the weight function implies that when $[\alpha, \beta] \subseteq [a, b]$ with $\alpha < \beta$ then

$$\int_\alpha^\beta w(x)\ dx > 0.$$

We define

$$\langle f, g \rangle_w := \int_a^b w(x) f(x) g(x)\ dx.$$

The mapping $\langle ., . \rangle_w$ provides an inner product on $C[a, b]$, i.e.

(1) $\langle ., . \rangle_w$ is bilinear, i.e.

$$\langle \alpha f + \beta g, h \rangle_w = \alpha \langle f, h \rangle_w + \beta \langle g, h \rangle_w,$$

and

$$\langle h, \alpha f + \beta g \rangle_w = \alpha \langle h, f \rangle_w + \beta \langle h, g \rangle_w,$$

for $f, g, h \in C[a, b]$ and $\alpha, \beta \in \mathbb{R}$.

(2) $\langle ., . \rangle_w$ is symmetric, i.e.

$$\langle f, g \rangle_w = \langle g, f \rangle_w, \qquad f, g \in C[a, b].$$

(3) $\langle ., . \rangle_w$ is positive definite, i.e.

$$\langle f, f \rangle_w \geq 0, \qquad f \in C[a, b]$$

and equals 0 only if f is the zero function, i.e. $f(x) = 0$.

The above three properties implie that

$$\|f\|_w := (\langle f, f \rangle_w)^{1/2}$$

is a norm on $C[a, b]$ and

$$|\langle f, g \rangle_w| \leq \|f\|_w \|g\|_w$$

(Cauchy-Schwartz inequality).

Definition 14.2 (w-orthogonality). *We say that f is $w-$*orthogonal *to $\mathbb{P}^k$ if*

$$\langle f, p \rangle_w = 0 \qquad \textit{for all} \quad p \in \mathbb{P}^k.$$

We introduce the following theorem whose proof will be given in the next lecture.

Theorem 14.1 (Gaussian Quadrature). *Suppose there is a nonzero $q_{k+1} \in \mathbb{P}^{k+1}$ which is $w-$orthogonal to $\mathbb{P}^k$. If q_{k+1} has $k+1$ distinct roots $\{x_0, ..., x_k\}$, then quadrature based on the nodes $\{x_0, ..., x_k\}$ approximating*

$$I(f) = \int_a^b w(x) f(x) \, dx$$

which is exact on $\mathbb{P}^k$ is in fact exact on $\mathbb{P}^{2k+1}$, i.e. it is a Gaussian quadrature.

In the next section, we shall prove that there is a unique monic polynomial q_{k+1} that is w-orthogonal to $\mathbb{P}^k$ and it has exactly $k+1$ distinct roots, $\{x_0, x_1, \ldots, x_k\}$, in the interval $[a, b]$. Thus, by the previous theorem, the quadrature using its roots as nodes is a Gaussian quadrature. The following theorem gives a formula for the quadrature error.

Theorem 14.2. *Suppose that f is in $C^{2k+2}[a,b]$, q_{k+1} is the monic polynomial described above and*

$$Qf = \sum_{i=0}^{k} w_i f(x_i)$$

is the corresponding Gaussian quadrature. Then there exists $\zeta \in (a,b)$ satisfying

$$\int_a^b w(x)f(x)\ dx - Qf = \frac{f^{(2k+2)}(\zeta)}{(2k+2)!}\int_a^b q_{k+1}^2(x)w(x)\,dx.$$

Proof. Let $p(x) \in \mathbb{P}^{2k+1}$ solve the Hermite interpolation problem:

$$p(x_i) = f(x_i) \text{ and } p'(x_i) = f'(x_i), \qquad \text{for } i = 0,1,\dots,k.$$

By Theorem 7.2, for each $x \in [a,b]$, there is a ζ_x in $[a,b]$ satisfying

$$f(x) - p(x) = \frac{f^{(2k+2)}(\zeta_x)}{(2k+2)!} q_{k+1}^2(x).$$

Now, since $p(x) \in \mathbb{P}^{2k+1}$ and the quadrature is exact on P^{2k+1},

$$I(p) = Qp = Qf$$

and hence

$$\begin{aligned} I(f) - Q(f) = I(f-p) &= \int_a^b w(x)(f(x) - p(x))\,dx \\ &= \frac{1}{(2k+2)!}\int_a^b f^{(2k+2)}(\zeta_x)w(x)q^2(x)\,dx. \end{aligned}$$

Applying an argument like that used in the proof of Theorem 11.1 shows that $f^{(2k+2)}(\zeta_x)$ is a continuous function of x and hence applying the mean value theorem for integrals gives the desired result. □

Note that the quadrature Q is exact on $\mathbb{P}^k$ (or $\mathbb{P}^{2k+1}$) means that

$$I(p) = \int_a^b w(x)p(x)\ dx = Q(p)$$

for every $p \in \mathbb{P}^k$ (or $\mathbb{P}^{2k+1}$). Also, by Remark 12.4, the weights of the quadrature are uniquely defined from exactness on $\mathbb{P}^k$.

Remark 14.1. If $q_{k+1} \in \mathbb{P}^{k+1}$ is $w-$orthogonal to $\mathbb{P}^k$ and is nonzero then

$$q_{k+1}(x) = a_{k+1}x^{k+1} + a_k x^k + ... + a_0$$

and $a_{k+1} \neq 0$. Indeed, if $a_{k+1} = 0$ then $q_{k+1} \in \mathbb{P}^k$ and

$$0 = \langle q_{k+1}, q_{k+1}\rangle_w = \int_a^b w(x)q_{k+1}^2(x)\ dx,$$

which implies that $q_{k+1} = 0$ and contradicts our assumption. Moreover, since we are only interested in the roots of q_{k+1}, we may assume that q_{k+1} is monic, i.e. $a_{k+1} = 1$ and

$$q_{k+1}(x) = x^{k+1} + a_k x^k + ... + a_0.$$

Example 14.4 ($k = 1$ and $w(x) = 1$)**.** *Find a monic $q \in \mathbb{P}^2$ with*

$$\langle f, g\rangle_w = \int_{-1}^{1} f(x)g(x)\ dx \quad \text{for all } g \in \mathbb{P}^1.$$

We are looking for α and β such that

$$0 = \langle q, 1\rangle_w = \int_{-1}^{1} (x^2 + \alpha x + \beta)\ dx = \frac{2}{3} + \alpha 0 + 2\beta,$$

i.e. $2\beta = -\frac{2}{3}$ or $\beta = -\frac{1}{3}$. In addition, we want

$$0 = \langle q, x\rangle_w = \int_{-1}^{1} (x^3 + \alpha x^2 + \beta x)\ dx = \frac{2}{3}\alpha,$$

and so $\alpha = 0$. This implies that the desired polynomial is

$$q(x) = x^3 - \frac{1}{3},$$

which has two roots, namely

$$\pm\frac{1}{\sqrt{3}}.$$

There are the quadrature nodes derived in Example 14.2.

Example 14.5 ($k = 2$ and $w(x) = 1$)**.** *Find $q \in \mathbb{P}^3$,*

$$q(x) = x^3 + \alpha x^2 + \beta x + \gamma,$$

which is $w-$orthogonal to $\mathbb{P}^2$ with $w(x) = 1$. The desired polynomial must satisfy the following 3 constraints

$$\begin{aligned} \alpha\frac{2}{3} + 2\gamma &= \langle q, 1\rangle_w = 0 \\ \frac{2}{5} + \frac{2}{3}\beta &= \langle q, x\rangle_w = 0 \\ \alpha\frac{2}{5} + \frac{2}{3}\gamma &= \langle q, x^2\rangle_w = 0. \end{aligned}$$

The first and last constraints hold only if

$$A := \begin{pmatrix} \frac{2}{3} & 2 \\ \frac{2}{5} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} \alpha \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Note that $\det(A) = \frac{4}{9} - \frac{4}{5} \neq 0$, *so the only solution is* $\alpha = \gamma = 0$. *From the second constraint, we find that*

$$\frac{1}{5} + \frac{\beta}{3} = 0 \qquad \Longrightarrow \qquad \beta = -\frac{3}{5}.$$

The desired polynomial reads

$$q(x) = x^3 - \frac{3}{5}x = \left(x^2 - \frac{3}{5}\right)x$$

and has roots $-\sqrt{3/5}$, 0, $\sqrt{3/5}$. *These are the quadrature nodes of Example 14.3.*

15. Lecture 15.

We start with the proof of the quadrature theorem (Theorem 14.1).

Proof of Theorem 14.1. Let $\{x_0, .., x_k\}$ be distinct roots of q_{k+1} and let Q be the associated exact quadrature scheme on $\mathbb{P}^k$. Let $p \in \mathbb{P}^{2k+1}$ and factor (using polynomial division with reminder)

$$p = q_{k+1}s + r,$$

with $r, s \in \mathbb{P}^k$. Then,

$$I(p) = \int_a^b w(x)p(x)\ dx = \underbrace{\int_a^b w(x)q_{k+1}(x)s(x)\ dx}_{=0} + \int_a^b w(x)r(x)\ dx$$

using the fact that q_{k+1} is $w-$orthogonal to $\mathbb{P}^k$ and $s \in \mathbb{P}^k$. Now, since Q is exact on $\mathbb{P}^k$, then

$$I(r) = \sum_{j=0}^{k} w_j r(x_j).$$

Moreover, the nodes $\{x_0, ..., x_k\}$ are the roots of q_{k+1}, so that computing further

$$I(p) = \sum_{j=0}^{k} w_j r(x_j) = \sum_{j=0}^{k} w_j \left(\underbrace{q_{k+1}(x_j)}_{=0} s(x_j) + r(x_j) \right) = Q(p),$$

which proves the quadrature is exact on $\mathbb{P}^{2k+1}$. □

Lemma 15.1 (Roots of $w-$orthogonal polynomial)**.** *If $q_{k+1} \in \mathbb{P}^{k+1}$ is non zero and is $w-$orthogonal to $\mathbb{P}^k$, then all of the roots of q are distinct and in (a, b).*

Proof. We will see in the next lemma (Lemma 15.2) that q as real coefficients. If q does not have any root in (a, b) then $q_{k+1} > 0$ or $q_{k+1} < 0$ in (a, b) and so

$$\int_a^b w(x)q_{k+1}(x)\ dx > 0 \qquad \text{or} \qquad \int_a^b w(x)q(x)\ dx < 0,$$

either contradicting the $w-$orthogonality of q_{k+1} in $\mathbb{P}^0 \subset \mathbb{P}^k$.

Suppose now that q_{k+1} has $1 \leq l < k+1$ roots in (a, b), denoted y_1, y_2, ..., y_l (repeated according to their multiplicity), and set

$$r(x) = \prod_{y_j \text{ root of odd multiplicity}} (x - y_j).$$

As the polynomial q changes sign across a root of odd multiplicity (as does $r(x)$), the product $q_{k+1}(x)r(x)$ has the same sign except for $x = y_j$, where y_j is a root of odd multiplicity. This implies that

$$\int_a^b w(x)q_{k+1}(x)r(x)\ dx \neq 0.$$

As $r \in \mathbb{P}^k$, this is a contradiction with the $w-$orthogonality in $\mathbb{P}^k$. As a consequence, there must be $k+1$ roots of q_{k+1} in (a, b) and so they must be distinct for the resulting r to belongs to $\mathbb{P}^{k+1}$. □

Lemma 15.2 (Real coefficients)**.** *There is a unique monic real polynomial $q \in \mathbb{P}^{k+1}$, which is $w-$orthogonal to $\mathbb{P}^k$.*

Proof. Let $q_{k+1} = x^{k+1} + \alpha_k x^k + ... + \alpha_0$. Then q_{k+1} is a $w-$orthogonal to $\mathbb{P}^k$ if and only if

$$\langle q_{k+1}, x^j\rangle_w = 0, \qquad j = 0, ..., k,$$

i.e.

$$\langle x^{k+1}, x^j\rangle_w + \sum_{l=0}^{k} \alpha_l \langle x^l, x^j\rangle_w = 0, \qquad j = 0, ..., k.$$

This is equivalent to

$$A \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_k \end{pmatrix} = F,$$

where the coefficients of the matrix A are given by

$$A_{j,l} = \langle x^l, x^j\rangle_w \qquad j, l = 0, ..., k$$

and

$$F_j = -\langle x^{k+1}, x^j\rangle_w, \qquad j = 0, ..., k.$$

Suppose that $A\beta = 0$ for some $\beta \in \mathbb{R}^{k+1}$. Set

$$r(x) = \beta_k x^k + \beta_{k-1}x^{k-1} + ... + \beta_0.$$

The jth equation of $A\beta = 0$ is

$$0 = \sum_{l=0}^{k} A_{j,l}\beta_l = \sum_{l=0}^{k} \langle x^l, x^j\rangle_w \beta_l = \sum_{l=0}^{k} \langle \beta_l x^l, x^j\rangle_w = \langle r(x), x^j\rangle_w, \quad j = 0, 1, ..., k,$$

i.e. $r(x)$ is $w-$orthogonal to $\mathbb{P}^k$. As $r \in \mathbb{P}^k$

$$0 = \langle r(x), r(x)\rangle_w \qquad \Longrightarrow \qquad r(x) = 0, \quad \text{i.e.} \quad \beta = 0.$$

This proves that A is nonsingular. As A is real valued, so is A^{-1}. (For instance, the inverse can be computed by row reducing $(A : I) \to (I : A^{-1})$.) □

Every weighted quadrature problem gives rise to a sequence of orthogonal polynomials. The sequence follows a 3 term recurrence.

Start with $\tilde{p}_0 = 1 \in \mathbb{P}^0$ (nothing to be orthogonal to). Then $\tilde{p}_1 \in \mathbb{P}^1$ must be orthogonal to 1. If $\tilde{p}_1(x) = x + \alpha$ then α must satisfy

$$0 = \langle x + \alpha, 1\rangle_w, \qquad \text{or} \qquad \alpha = -\langle x, 1\rangle_w.$$

Suppose we have computed $\tilde{p}_{j-1}$ and $\tilde{p}_j$. Write

$$\tilde{p}_{j+1} = (x + \alpha)\tilde{p}_j + \beta\tilde{p}_{j-1}.$$

Then for $\theta \in \mathbb{P}^{j-2}$

$$\langle \tilde{p}_{j+1}, \theta\rangle_w = \langle (x+\alpha)\tilde{p}_j, \theta\rangle_w + \beta \underbrace{\langle \tilde{p}_{j-1}, \theta\rangle_w}_{=0} = \underbrace{\langle \tilde{p}_j, (x+\alpha)\theta\rangle_w}_{=0} = 0.$$

We also need

$$0 = \langle \tilde{p}_{j+1}, \tilde{p}_{j-1}\rangle_w = \langle (x+\alpha)\tilde{p}_j, \tilde{p}_{j-1}\rangle_w + \beta\langle \tilde{p}_{j-1}, \tilde{p}_{j-1}\rangle_w.$$

The α term goes away so

$$\beta = -\frac{\langle x\tilde{p}_j, \tilde{p}_{j-1}\rangle_w}{\langle \tilde{p}_{j-1}, \tilde{p}_{j-1}\rangle_w}.$$

Also

$$0 = \langle \tilde{p}_{j+1}, \tilde{p}_j\rangle_w = \langle (x+\alpha)\tilde{p}_j, \tilde{p}_j\rangle_w + \beta \underbrace{\langle \tilde{p}_{j-1}, \tilde{p}_j\rangle_w}_{=0},$$

and so we find

$$\alpha = -\frac{\langle x\tilde{p}_j, \tilde{p}_j\rangle_w}{\langle \tilde{p}_j, \tilde{p}_j\rangle_w}.$$

The values of α and β determines $\tilde{p}_{j+1}$. Note that the orthogonal polynomials always satisfy 3 term recurrence relations!

15.1. **Rodrigues Formula for Legendre Polynomials.**

Example 15.1. $w(x) = 1$, $a = -1$, $b = 1$] *Consider the approximation* $I(f) = \int_{-1}^{1} f(x)\, dx$.. *Then*

$$\tilde{p}_n(x) = \frac{1}{(2n)(2n-1)\cdot \ldots . \cdot (n+1)} \frac{d^n}{dx^n}[(x^2-1)^n].$$

To see this, we check that it is a monic polynomial of degree n *and* $w-$*orthogonal to* $\mathbb{P}^{n-1}$. *We leave the first part as an exercise (Exercise 15.1). Now, if* $p \in \mathbb{P}^{n-1}$

$$\int_{-1}^{1} \frac{d^n}{dx^n}[(x^2-1)^n]p(x)\, dx = \underbrace{\frac{d^{n-1}}{dx^{n-1}}[(x^2-1)^n]p(x)\Big|_{-1}^{1}}_{=0} - \int_{-1}^{1} \frac{d^{n-1}}{dx^{n-1}}[(x^2-1)^n]p'(x)\, dx.$$

Repeating this by moving all derivatives over to p, we arrive at

$$\int_{-1}^{1} \frac{d^n}{dx^n}[(x^2-1)^n]p(x)\ dx = (-1)^n \int_{-1}^{1} (x^2-1)^n \underbrace{p^{(n)}(x)}_{=0}\ dx = 0$$

because $p \in \mathbb{P}^{n-1}$.

Definition 15.1 (Legendre Polynomials)**.** *The polynomial*

$$\frac{1}{2^n n!}\frac{d^n}{dx^n}[(x^2-1)^n].$$

is called the Legendre polynomial (different normalization) and satisfies

$$(n+1)p_{n+1}(x) = (2n+1)xp_n(x) - np_n(x).$$

Exercise 15.1. *Show that*

$$\tilde{p}_n(x) = \frac{1}{(2n)(2n-1)\cdot....\cdot(n+1)}\frac{d^n}{dx^n}[(x^2-1)^n]$$

is a polynomial of degree n and is monic (i.e. the leading coefficient is 1*).*

15.2. Chebyshev Polynomials.

Example 15.2 (Chebyshev polynomials)**.** *We recall that the Chebyshev polynomials are given by*

$$T_n(x) = \cos(n\cos^{-1}(x)).$$

Note that for $n \neq j$

$$\int_0^{\pi} \cos(n\theta)\cos(j\theta)\ d\theta = 0.$$

We leave the above claim as exercise.

Set $\theta = \cos^{-1}(x)$, then $x = \cos(\theta)$ and

$$dx = -\sin(\theta)d\theta = -\sqrt{1-\cos^2(\theta)}\ d\theta = -\sqrt{1-x^2}\ d\theta.$$

Using the orthogonality above

$$\begin{aligned} 0 = \int_0^{\pi} \cos(n\theta)\cos(j\theta)\ d\theta &= \int_{-1}^{1} \cos(n\cos^{-1}(x))\cos(j\cos^{-1}(x))\frac{dx}{\sqrt{1-x^2}} \\ &= \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}}T_n(x)T_j(x)\ dx, \end{aligned}$$

i.e. the Chebyshev polynomial T_n are orthogonal polynomials on $-1, 1$ with weights $w(x) = \frac{1}{\sqrt{1-x^2}}$ and satisfy the recurrence

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

as we have seen already in Section 5.

16. Lecture 16: Rodrigues Formula for Chebyshev Polynomials.

We saw last lecture that the Chebyshev polynomials are orthogonal with respect to the scalar product

$$\langle f, g\rangle_w = \int_{-1}^{1} \underbrace{\frac{1}{\sqrt{1-x^2}}}_{=:w(x)} f(x)g(x)\ dx,$$

i.e. T_{n+1} satisfies $T_{n+1} \in \mathbb{P}^{n+1}$ and

$$\langle T_{n+1}, T_j\rangle_w = 0, \qquad 0 \le j \le n.$$

As $\{T_j\}_{j=0}^n$ is a basis for $\mathbb{P}^n$, T_{n+1} is $w-$orthogonal to $\mathbb{P}^n$.

The Rodrigues formula for the Chebyshev polynomials reads

$$\tilde{T}_n = w(x)^{-1}\frac{d^n}{dx^n}\left(w(x)(1-x^2)^n\right) = (1-x^2)^{1/2}\frac{d^n}{dx^n}\left((1-x^2)^{n-1/2}\right)$$

using the definition of the weight $w(x) = (1-x^2)^{-1/2}$. Note that

$$\frac{d^n}{dx^n}(fg) = \sum_{j=0}^{n} \binom{n}{j} f^{(j)} g^{(n-j)}$$

with the parenthesis term involving n and j denoting the binomial coefficient (you can prove this formula by induction). Therefore,

$$\begin{aligned}
\tilde{T}_n(x) &= (1-x^2)^{1/2}\frac{d^n}{dx^n}\left((1-x)^{n-1/2}(1+x)^{n-1/2}\right)\\
&= (1-x^2)^{1/2}\sum_{j=0}^{n}\binom{n}{j} c_{j,n}(1-x)^{n-j-1/2}(1+x)^{j-1/2} \qquad (19)\\
&= \sum_{j=0}^{n}\binom{n}{j} c_{j,n}(1-x)^{n-j}(1+x)^{j}.
\end{aligned}$$

Here $c_{j,n}$ is a constant depending only on j and n. This proves that $\tilde{T}_n \in \mathbb{P}^n$.

We now check that $\tilde{T}_n(x)$ is $w-$orthogonal to $\mathbb{P}^{n-1}$. We use integration by parts again: for $p \in \mathbb{P}^{n-1}$,

$$\begin{aligned} I &:= \int_{-1}^{1} (1-x^2)^{-1/2}\tilde{T}_n(x)p(x)\ dx \\ &= \int_{-1}^{1} \frac{d^n}{dx^n}\left((1-x^2)^{-1/2}(1-x^2)^n\right)p(x)\ dx \\ &= p(x)\frac{d^{n-1}}{dx^{n-1}}\left((1-x)^{n-1/2}(1+x)^{n-1/2}\right)\Big|_{x=-1}^{x=1} \\ &\qquad - \int_{-1}^{1} \frac{d^{n-1}}{dx^{n-1}}\left((1-x^2)^{-1/2}(1-x^2)^n\right)p'(x)\ dx \end{aligned}$$

The end point contribution vanishes since

$$\frac{d^{n-1}}{dx^{n-1}}\left((1-x)^{n-1/2}(1+x)^{n-1/2}\right) = \\ \sum_{j=0}^{n-1}\binom{n-1}{j} c_{n-1,j}(1-x)^{n-j-1/2}(1+x)^{j+1-1/2}$$

and all terms evaluated at the end points are zero because they have positive powers of $(1-x)$ and $(1+x)$. Repeating the argument gives

$$I = (-1)^n\int_{-1}^{1}(1-x^2)^{-1/2}(1-x^2)^n p^{(n)}(x)\ dx = 0$$

for $p \in \mathbb{P}^{n-1}$. Since, T_n and $\tilde{T}_n$ differ at most by a normalization constant, T_n is also a polynomial of degree n, $w-$orthogonal to $\mathbb{P}^{n-1}$.

Finally, we need to show that the polynomial $\tilde{T}_n(x)$ given by (19) is nonzero (w-orthogonality will imply that the coefficient of its highest order term is non-zero). This is not immediately obvious as the highest order coefficient of each term in the right hand sum in (19) is non-zero and the terms are of oscillating signs. We shall proceed by contradiction, i.e., suppose that $\tilde{T}_n(x) = 0$. Set

$$\psi_j(x) = \frac{d^j}{dx^j}(1-x^2)^{n-1/2}, \qquad \text{for } j = 0, 1, \ldots, n.$$

In the integration by parts argument above, we showed that

$$\psi_j(-1) = \frac{d^j}{dx^j}(1-x^2)^{n-1/2}\Big|_{-1} = 0, \qquad \text{for } j = 0, 1, \ldots, n-1.$$

Thus, by the fundamental theorem of calculus,

$$\psi_{n-1}(x) = \psi_{n-1}(-1) + \int_{-1}^{x}\psi_n(t)\,dt = 0$$

where we used $\tilde{T}_n(x) = 0$ to conclude that the integral term vanished. Repeating this argument leads eventually to

$$\psi_0(x) = 0,$$

which is a contradiction.

17. Lecture 17.

17.1. Nonlinear Equations.

Essentially, the only way that one can solve nonlinear equations is by iteration.

The quadratic formula enables one to compute the roots of $p(x) = 0$ when $p \in \mathbb{P}^2$. Formulas were derived for finding the roots of $p \in \mathbb{P}^3$ and $p \in \mathbb{P}^4$ by expressions involving radicals (~ 1545). The case of quintics was studied unsuccessfully for almost 300 years until Abel (1824) proved *no such formula existed.* As we shall see, the roots of quintics (and other nonlinear equations) can be solved by iteration!

Example 17.1 (Eigenvalues)**.** *Let A be a $n \times n$ matrix with $n \geq 5$. We propose to compute the eigenvalues of A. The eigenvalues of A are the roots of the characteristic polynomial, i.e.*

$$p(\lambda) = \det(A - \lambda I).$$

The characteristic polynomial p has degree n and leading term is $(-1)^n \lambda^n$. In general, this problem can only be solved by iteration when $n \geq 5$.

17.2. Bisection method.

Suppose f is continuous on $[a, b]$ and satisfies $f(a)f(b) < 0$ (not the same sign). The intermediate value theorem tells us there is at least 1 root x^* of $f(x^*) = 0$ ($x^* \in (a, b)$). The bisection method can be used to iteratively find such a root.

```
Bisection Algorithm (mathematical)
    Set $a_0 = a$ and $b_0 = b$
    For $i = 0, 1, 2, ...$
        Set $a_{i+\frac{1}{2}} = \frac{a_i+b_i}{2}$
        If $f(a_{i+\frac{1}{2}}) = 0$
            STOP, $a_{i+\frac{1}{2}}$ is the desired root
        Else If $f(a_{i+\frac{1}{2}})f(a_i) > 0$
            Set $a_{i+1} = a_{i+\frac{1}{2}}$ and $b_{i+1} = b_i$
        Else
            Set $a_{i+1} = a_i$ and $b_{i+1} = a_{i+\frac{1}{2}}$
    End For
```

It is obvious that after j steps, either we have found a root or there is a root in (a_j, b_j). Note that in that case the root x^* is at most half of the interval length away from either a_j or b_j, i.e.

$$x^* = \frac{a_j + b_j}{2} + \varepsilon,$$

where

$$\varepsilon < b_j - \frac{a_j + b_j}{2} = \frac{b_j - a_j}{2} = \frac{b - a}{2^j}.$$

We now describe the *matlab* version of the mathematical bisection algorithm using minimal memory usage.

```
%%% R is the root approximation after N steps
%%% A,B are real numbers
%%% F is a continuous function
%%% F(A)F(B)<=0
Function R=BISECTION(A,B,N,F)

%% Preliminaries: A or B are a root, F(A)F(B)>0
    SA = sign(F(A));
    if (SA == 0)
        R=A;
        return;
    end

    SB = sign(F(B));
    if (SB == 0)
        R=B;
        return;
    end

    if (SA == SB)
      fprintf ('Input error to bisection\n');
      R=NaN(1);  %return not a number
      return;
    end

%% all the preliminaries are done
   for I=1:N
      AV = (A-B)/2;
      FAV = F(AV);
      S=sign(FAV);
      if (S==0)
        R=AV;
        return;
      end
      if (S==SA)
        A=AV;
      else
```

```
            B=AV;
         end
      end
   end
```

This algorithm will only get the real roots of a real valued function $f : \mathbb{R} \to \mathbb{R}$.

Example 17.2. *[Cubic] Let $f(x) = x^3 + x = x(x^2 + 1)$. The roots of f are $(0, i, -i)$ and bisection will only get the real roots, i.e. $x = 0$ To find the complex roots, we need to treat f as a complex valued functions. We define $F : \mathbb{R}^2 \simeq \mathbb{C} \to \mathbb{C} \simeq \mathbb{R}^2$ by*

$$\begin{aligned} F\begin{pmatrix} R \\ I \end{pmatrix} &\sim f(R + iI) \\ &= (R + iI)^3 + (R + iI) = R^3 + 3iR^2I - 3RI^2 - iI^3 + (R + iI) \\ &= (R^3 - 3RI^2 + R) + (3R^2I - I^3 + I)i \sim \begin{pmatrix} R^3 - 3RI^2 + R \\ 3R^2I - I^3 + I \end{pmatrix}. \end{aligned}$$

for real number R, I. Now, we try to find the real roots corresponding to the system

$$F\begin{pmatrix} R \\ I \end{pmatrix} := \begin{pmatrix} R^3 - 3RI^2 + R \\ 3R^2I - I^3 + I \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{20}$$

Remark 17.1. Note that the bisection algorithm cannot be applied to (20). This illustrates the need for *iterative techniques for systems* (but is far from the only reason).

Remark 17.2. It is not necessary to reformulate the problem of Example 17.2 into a problem on $\mathbb{R}^2$. Instead, it can be formulated as a nonlinear problem involving one complex variable, i.e., find $x^* \in \mathbb{C}$ satisfying

$$f(x^*) = 0.$$

Even though this is a single variable equation, the bisection method still cannot be applied. (FUNDAMENTAL DIFFICULTIES IN MATHEMATICS ARE INVARIENT UNDER REFORMULATION!)

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a vector valued function on $\mathbb{R}^n$. We want to find roots $x^* \in \mathbb{R}^n$ satisfying

$$F(x^*) = 0 \in \mathbb{R}^n.$$

This gives n equations and $x_1^*, ..., x_n^*$ are the n unknown.

Example 17.3 ($n = 2$)**.**

$$F\begin{pmatrix}x_1\\x_2\end{pmatrix} := \begin{pmatrix}x_1^2 + 4x_2^2 - 1\\4x_1^2 + x_2^2 - 1\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}.$$

Refer to Figure 6 for an illustration of the situation.

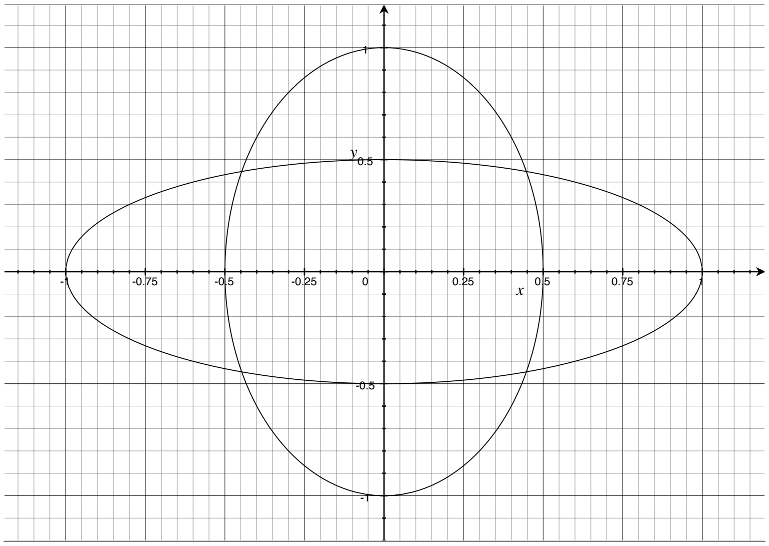

FIGURE 6. There are 4 roots of the system $x_1^2+4x_2^2-1 = 0$ and $4x_1^2 + x_2^2 - 1 = 0$.

Example 17.4 ($n = 1$)**.** *Consider*

$$f(x) = \sin(x) = 0.$$

The roots are $x = j\pi$*,* $j \in \mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$*.*

Definition 17.1 (Fixed Point Equation)**.** *A* fixed point equation *is one of the form*

$$x = G(x),$$

with $x \in \mathbb{R}^n$ *and* $G : \mathbb{R}^n \to \mathbb{R}^n$*.*

A solution to a fixed point equations is $x^* \in \mathbb{R}^n$ satisfies

$$x^* = G(x^*).$$

We can turn $F(x) = 0$ into a fixed point problem, i.e.

$$x = x - F(x), \qquad \text{i.e. } G(x) := x - F(x).$$

This means that x^* solves $F(x^*) = 0$ if and only if $x^* = G(x^*)$. Obviously fixed point problems can be turned into $F(x) = 0$ by setting $F(x) = x - G(x)$. We could have also used

$$x = x - BF(x) =: G(x)$$

for any non singular $n \times n$ matrix B. We will see the importance of the matrix B soon.

18. Lecture 18: Fixed Point Iteration or Picard Iteration.

We considered in the last lecture the fixed point formulation: Find $x^* \in \mathbb{R}^n$ satisfying

$$x^* = G(x^*),$$

where $G : \mathbb{R}^n \to \mathbb{R}^n$. We now discuss an algorithm approximating such x^*.

The fixed point or Picard iteration: Start with an initial iterate (guess) $x^0 \in \mathbb{R}^n$. Then, for $i = 0, 1, 2, ...$ set

$$x^{i+1} = G(x^i).$$

We now want to understand when $x^i \to x^*$.

Definition 18.1 (Lipschitz Continuous)**.** *Let $\Omega \subset \mathbb{R}^n$. The vector-valued function $G : \Omega \to \mathbb{R}^n$ is called* Lipschitz continuous *if there is an $M \geq 0$ with*

$$\|G(x) - G(y)\|_\infty \leq M\|x - y\|_\infty$$

for all $x, y \in \Omega$. Here for $w \in \mathbb{R}^n$

$$\|w\|_\infty := \max_{i=1,\ldots,n} |w_i|.$$

Definition 18.2. *[Contraction Mapping] Let $\Omega \subset \mathbb{R}^n$. The vector-valued function $G : \Omega \to \Omega$ is a* contraction mapping *if G is Lipschitz continuous with constant $M = \rho < 1$.*

Remark 18.1. It is possible to use any norm on $\mathbb{R}^n$ in the definition of Lipschitz continuity and, since all norms on $\mathbb{R}^n$ are equivalent, G will be Lipschitz continuous with respect to any such norm if it is Lipschitz continuous with repect to one. Note, however, that G may be a contraction with respect to one norm without being a contraction with respect to another.

Theorem 18.1 (Contraction Mapping Theorem)**.** *Let Ω be a closed subset of $\mathbb{R}^n$ and G be a contraction mapping of Ω into Ω with constant ρ. Then, there is a unique fixed point $x^* \in \Omega$ (satisfying $x^* = G(x^*)$) and the Picard iteration $\{x^j\}_{j=0}^\infty$, starting with any $x^0 \in \Omega$, converges to x^* and satisfies*

$$\|x^j - x^*\|_\infty \leq \frac{\rho^j}{1-\rho}\|x^0 - x^*\|_\infty.$$

This is often called linear convergence. We postpone the proof for later.

To understand the contraction mapping hypothesis, we consider the equation $F(x) = 0$, where $F : \mathbb{R} \to \mathbb{R}$, and assume that it has as solution $x^* \in \mathbb{R}$, i.e. $F(x^*) = 0$. Furthermore, we assume that $F \in C^2$ in a neighborhood $B_{\delta_1}(x^*) := [x^* - \delta_1, x^* + \delta_1]$ and that $F'(x^*) \neq 0$. Notice that the latter condition guarantees that there is a neighborhood $B_{\delta_2}(x^*)$ ($\delta_2 \leq \delta_1$) such that

$$|F'(\xi)| \geq \frac{1}{2}|F'(x^*)| \tag{21}$$

for every $\xi \in B_{\delta_2}(x^*)$. This implies that

$$|F'(\xi)|^{-1} \leq 2|F'(x^*)|^{-1} \qquad \xi \in B_{\delta_2}(x^*).$$

Now for $\delta \leq \delta_2$ (to be determined), we pick $w \in B_\delta(x^*)$ and set

$$G(x) = x - (F'(w))^{-1}F(x). \tag{22}$$

Clearly x^* is a fixed point of G and for $x, y \in B_\delta(x^*)$

$$G(x) - G(y) = G'(y)(x - y) + \frac{G''(\xi)}{2}(x - y)^2,$$

for some ξ between x and y. Therefore,

$$|G(x) - G(y)| \leq |G'(y)|\,|x - y| + \frac{|G''(\xi)|}{2}(x - y)^2. \tag{23}$$

Now by the C^2 assumption, there exists a constant M such that $|F''(\theta)| \leq M$ for every $\theta \in B_{\delta_1}(x^*)$ hence by (21),

$$\frac{|G''(\xi)|}{2}(x - y)^2 = \frac{1}{2}|F'(w)|^{-1}|F''(\xi)|\,|x - y|^2 \leq M|F'(X^*)|^{-1}|x - y|^2.$$

For the other term in (23), we note that

$$\begin{aligned} G'(y) &= 1 - (F'(w))^{-1}F'(y) = (F'(w) - F'(y))(F'(w))^{-1} \\ &= F''(\xi_1)(w - y)(F'(w))^{-1} \end{aligned}$$

for ξ_1 between w and y. Thus,

$$|G'(y)| \leq 2M|F'(x^*)|^{-1}|w - y|$$

and hence

$$|G(x) - G(y)| \leq 6M|F'(x^*)|^{-1}\delta|x - y| \tag{24}$$

where we used $|w - y| < 2\delta$ and $|x - y| < 2\delta$.

Given $0 < \rho < 1$, we chose δ so that

$$6M|F'(x^*)|^{-1}\delta < \rho.$$

This implies that G is a contraction mapping and so the Picard iterates

$$x^{i+1} = x^i - (F'(w))^{-1}F(x^i)$$

converges to x^* provided $|x^0 - x^*| \leq \delta$ and $|w - x^*| \leq \delta$.

The next theorem generalize this argumentation to $\mathbb{R}^n$.

Theorem 18.2 (Secant Algorithm)**.** *Assume $F : \mathbb{R}^n \to \mathbb{R}^n$ and $F(x^*) = 0$ for some $x^* \in \mathbb{R}^n$ with*

(1) *$F \in C^2$ in a neighborhood of x^*;*
(2) *$DF(x^*) =$ the derivative matrix at x^* given by*

$$(DF(x^*))_{ij} = \frac{\partial}{\partial x_j} F_i(x^*)$$

is nonsingular.

Then there is a $\delta > 0$ such that if $w \in B_\delta(x^*)$ and $x^0 \in B_\delta(x^*)$, the iteration

$$x^{i+1} = x^i - (DF(w))^{-1} F(x^i)$$

converges to x^*.

The proof is as in the case of scalar functions discussed before the theorem but more complicated due to matrix notations. The major obstacle to apply this algorithm is getting close enough to the root to come up with w (then we can always take $x^0 = w$).

19. Lecture 19.

19.1. **The proof of the contraction mapping theorem.**

We start with the proof of the contraction mapping theorem (Theorem 18.1).

Proof of Theorem 18.1. Let $j \leq k$ and $l \geq 0$, $x^{j+1} = G(x^j)$ with $x^0 \in \Omega \subset \mathbb{R}^n$ (closed) and G a contraction (Lipschitz constant $\rho < 1$) on Ω. Note that

$$x^{k+l+1} - x^k = x^{k+l+1} - x^{k+l} + x^{k+l} - x^{k+l-1} + ... + x^{k+1} - x^k.$$

Now,

$$x^{m+1} - x^m = G(x^m) - G(x^{m-1})$$

and so

$$\|x^{m+1} - x^m\|_\infty = \|G(x^m) - G(x^{m-1})\|_\infty \leq \rho\|x^m - x^{m-1}\|_\infty.$$

Repeating

$$\|x^{m+k+1} - x^{m+k}\|_\infty \leq \rho^k\|x^{m+1} - x^m\|_\infty$$

and thus

$$\begin{aligned}\|x^{k+l+1} - x^k\|_\infty &\leq \|x^{k+l+1} - x^{k+l}\|_\infty + \|x^{k+l} - x^{k+l-1}\|_\infty + ... + \|x^{k+1} - x^k\|_\infty \\ &\leq (\rho^l + \rho^{l-1} + ... + 1)\|x^{k+1} - x^k\|_\infty \\ &\leq (\rho^l + \rho^{l-1} + ... + 1)\rho^k\|x^1 - x^0\|_\infty \\ &\leq \underbrace{\rho^{k-j}}_{\leq 1}\frac{\rho^j}{1-\rho}\|x^1 - x^0\|_\infty \leq \frac{\rho^j}{1-\rho}\|x^1 - x^0\|_\infty.\end{aligned}$$

This means that if $m, t > j$

$$\|x^m - x^t\|_\infty \leq \frac{\rho^j}{1-\rho}\|x^1 - x^0\|_\infty.$$

The quantity on the right side can be made as small as we want by taking j large. This implies that the sequence $\{x^j\}$ is a Cauchy sequence and so converges to some $x^* \in \Omega$. (Recall that $\mathbb{R}^n$ is complete and Ω closed implies that Ω is complete). Moreover,

$$\begin{aligned}\|x^* - G(x^*)\|_\infty &\leq \|x^* - x^j + G(x^{j-1}) - G(x^*)\|_\infty \\ &\leq \|x^* - x^j\|_\infty + \rho\|x^* - x^{j-1}\|_\infty.\end{aligned}$$

As x^j converges to x^*, the quantity on the right can be made as small as desired buy taking j large, i.e.

$$x^* = G(x^*).$$

This shows that every Picard iteration converges to a fixed point. These fixed points are unique. Indeed, if $x_1^* = G(x_1^*)$ is another fixed point, then

$$\|x_1^* - x^*\|_\infty = \|G(x_1^*) - G(x^*)\|_\infty \leq \rho \|x_1^* - x^*\|_\infty,$$

i.e.

$$(1-\rho)\|x_1^* - x^*\|_\infty \leq 0$$

and therefore $\|x_1^* - x^*\|_\infty = 0$ or $x_1^* = x^*$. □

19.2. **Newton's Method.** We start with a motivation. We look for a root of $F(x) = 0$, where $F(x) := (f_1(x), f_2(x), \ldots, f_n(x))^t : \mathbb{R}^n \to \mathbb{R}^n$. Here the t denotes the matrix transpose. Assume $F \in C^2$ and that there is a root x^* of F such that $DF(x^*)$ is nonsingular.

We will need to use the second order multivariate Taylor series with remainder, namely, for $x, y \in \mathbb{R}^n$,

$$F(x) = F(y) + DF(y)(x-y) + \varepsilon_2(x,y). \tag{25}$$

When $F \in C^2$ in a neighborhood of y, $\varepsilon_2(x,y) = O(|x-y|^2)$ for x in that neighborhood. Ignoring the error term and recalling that we want $F(x^*) = 0$, we chose x^{j+1} by

$$0 = F(x^j) + DF(x^j)(x^{j+1} - x^j), \quad \text{i.e.,}$$
$$x^{j+1} = x^j - DF(x^j)^{-1}F(x^j).$$

This is the *Newton's* iterative method.

Note that this iteration is of the form

$$x^{j+1} = G_j(x^j), \qquad G_j(x) = x - DF(x^j)^{-1}F(x)$$

so it is not quite a fixed point iteration because G changes at each step!

We first show that Newton's method is well defined if we are sufficiently close to x^* under our assumptions on F, i.e, $DF(\xi)$ is nonsingular in a neighborhood of x^*. We will only sketch the argument showing this.

To do this, we use the following norm on $n \times n$ matrices:

$$\|A\|_\infty := \sum_{x \in \mathbb{R}^n, x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty}.$$

Here

$$\|x\|_\infty = \max_{i=1,2,\ldots,n} |x_i|, \quad x \in \mathbb{R}^n,$$

is the "max" vector norm in $\mathbb{R}^n$. We leave it as an exercise to show that $\|A\|_\infty$ provides a norm on the set of $n \times n$ matrices. Furthermore, its definition immediately implies

$$\|Ax\|_\infty \leq \|A\|_\infty \|x\|_\infty$$

and

$$\|AB\|_\infty \le \|A\|_\infty \|B\|_\infty.$$

Assume F is in $C^2(\overline{B_\delta(x^*)})$ for some $\delta > 0$ and that $\xi \in B_{\delta_1}(x^*)$ with $\delta_1 \le \delta$ to be determined. Using the first order Taylor series approximation applied to $(DF)_{ij}$, for $i, j =, 1 \ldots, n$:

$$(DF)_{ij}(\xi) = (DF)_{ij}(x^*) + \varepsilon_1(\xi, x^*) \tag{26}$$

with $\varepsilon_1(\xi, x^*) = O(|\xi - x^*|)$. This in turn implies that

$$\|DF(\xi) - DF(x^*)\|_\infty \le M\|\xi - x^*\|_\infty.$$

We next manipulate the equation, for $b, v \in \mathbb{R}^n$

$$DF(\xi)v = b \tag{27}$$

to obtain

$$[I - DF(x^*)^{-1}(DF(x^*) - DF(\xi))]v = DF(x^*)^{-1}DF(\xi)v = DF(x^*)^{-1}b.$$

Setting $B = DF(x^*)^{-1}(DF(x^*) - DF(\xi))$, we rewrite this as

$$(I - B)v = DF(x^*)^{-1}b. \tag{28}$$

This we solve by a "Neuman series" argument writing

$$v = (I - B)^{-1}DF(x^*)^{-1}b = \left[\sum_{j=0}^{\infty} B^j\right] DF(x^*)^{-1}b.$$

The sequence of matrices in the brackets above converges if $\|B\|_\infty < 1$ and the resulting vector v satisfies (27). This shows that $DF(\xi)$ is nonsingular if

$$\begin{aligned}\|B\|_\infty &= \|DF(x^*)^{-1}(DF(x^*) - DF(\xi))\|_\infty \\ &\le \|DF(x^*)^{-1}\|_\infty \|(DF(x^*) - DF(\xi))\|_\infty \\ &\le M\|DF(x^*)^{-1}\|_\infty \|\xi - x^*\|_\infty < 1.\end{aligned}$$

This holds if $\delta_1 < \min\{M^{-1}(\|DF(x^*)^{-1}\|_\infty)^{-1}/2, \delta\}$ in which case $\|(I-B)^{-1}\|_\infty < 2$ so $DF(\xi)$ is nonsingular and

$$\|DF(\xi)^{-1}\|_\infty \le 2\|DF(x^*)^{-1}\|_\infty.$$

For simplicity, we now continue in the single variable case. Applying the argumentation given before the proof of Theorem 18.2 shows that the Newton iterates satisfy

$$|x^* - x^{j+1}| = |G_j(x^*) - G_j(x^j)| \le 6M|f'(x^*)|^{-1}|x^* - x^j|^2.$$

This shows that once the iteration starts converging, the error at the $j + 1$'st step is bounded by a constant times the SQUARE of the error at the j'th step. This is called QUADRATIC CONVERGENCE.

This result holds in the multivariate case BUT the argmuent, although similar, is more complicated due to the vector notation.

For a geometric interpretation of the Newton's method, we refer to Figure 7.

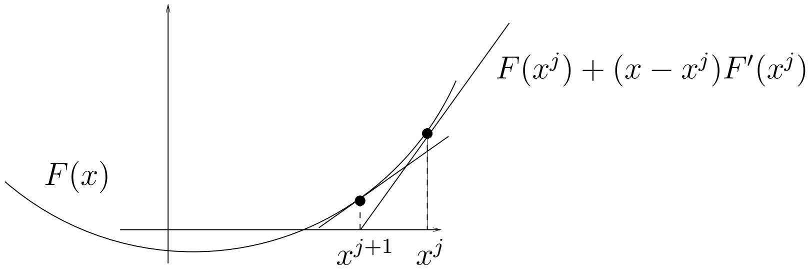

FIGURE 7. Geometric Interpretation of the Newton's method.

Example 19.1 (Newton)**.** *Consider the function $f(x) = x^3 e^{-x^2}$, which has only one root (at $x = 0$), see Figure 8. We compute*

$$f'(x) = (3x^2 - 2x^4)e^{-x^2}$$

so

$$f'(x) > 0 \qquad \textit{for} \qquad |x| < \sqrt{3/2}$$

and

$$f'(x) < 0 \qquad \textit{for} \qquad |x| > \sqrt{3/2}$$

The geometric interpretation of the Newton method implies that

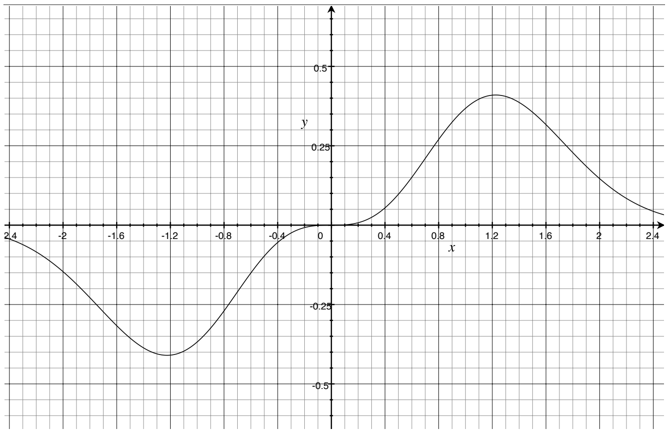

FIGURE 8. $f(x) = x^3 e^{-x^2}$.

(1) *if $x^0 > \sqrt{3/2}$, Newton's method diverge to ∞*
(2) *if $x^0 < -\sqrt{3/2}$, Newton's method diverge to $-\infty$*

(3) *it diverge in a neighborhood of both* $-\sqrt{3/2}$ *and* $\sqrt{3/2}$
(4) *it converges to* 0 *otherwise.*

20. Lecture 20: Ordinary Differential Equations (ODEs).

Look for $x(t) \in C^1(a,b)$ such that

$$\left\{ \; x'(t) = f(t, x(t)), \qquad t \in (a,b); x(t_0) = x_0 \right.$$

for some $t_0 \in (a,b)$, $x_0 \in \mathbb{R}$ and a function $f : (a,b) \times \mathbb{R} \to \mathbb{R}$.

We ask the following questions:

(1) Is there a function $x(t)$ satisfying the ODE and is it unique?
(2) How to approximate solutions?

Example 20.1 (Blowup). *Consider the following ODE*

$$x' = x\tan(t), \qquad x(0) = 1.$$

We check that $x(t) = \sec(t) = \frac{1}{\cos(t)}$ *is a solution. Indeed, we check*

$$x' = \frac{\sin(t)}{\cos^2(t)} = \frac{1}{\cos(t)}\tan(t) = x\tan(t)$$

and $x(0) = \frac{1}{\cos(0)} = 1$. *The function* $x(t) = \frac{1}{\cos(t)}$ *is pictured in Figure 9. This solution is continuously defined on* $(-\pi/2, \pi/2)$ *and is unique in that interval. The blow up of the solution at* $t = \pm\pi/2$ *is caused by the blow up of* $\tan(t)$ *at* $t = \pm\pi/2$ *(see later).*

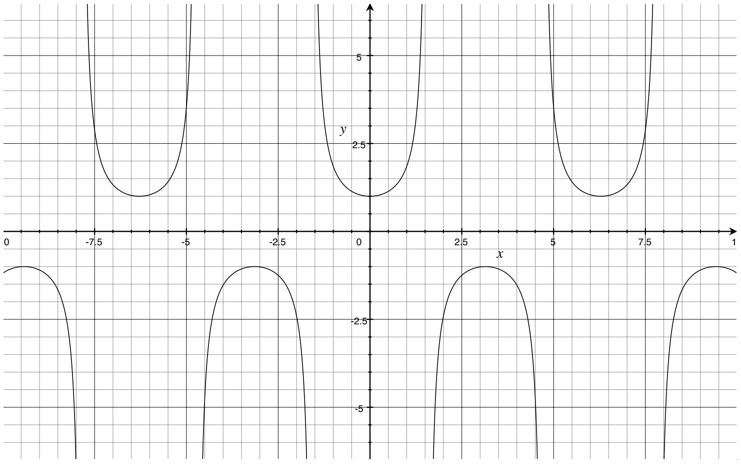

Figure 9. Function $x(t) = 1/\cos(t)$.

Example 20.2 (Blowup 2). *Consider now the ODE*

$$x' = 1 + x^2, \qquad x(0) = 0.$$

In this case, we find the solution by separating variables and taking anti-derivatives

$$\int \frac{dx}{1+x^2} = \int 1 \; dt + C.$$

This implies that

$$\arctan(x) = t + C$$

or

$$x = tan(t + C).$$

Using the initial condition, we find that $C = 0$, *i.e.*

$$x(t) = \tan(t).$$

Figure 10 depicts this function. Although the ODE looks harmless, the solution $x(t) = \tan(t)$ *is continuously defined (and unique) on* $(-\pi/2, \pi/2)$ *and blows up at the endpoints.*

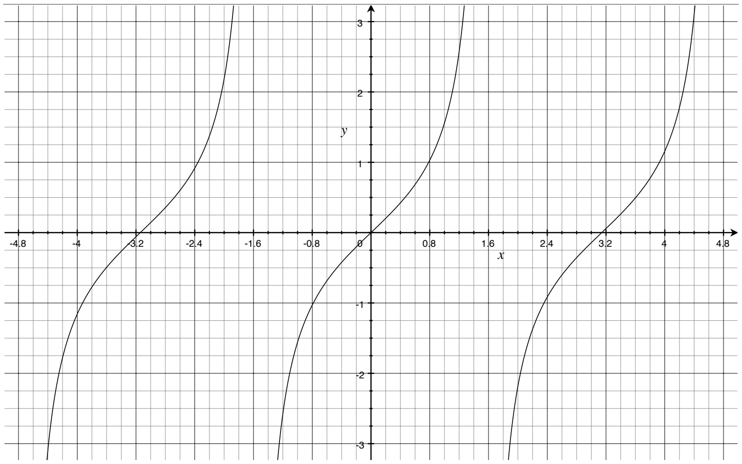

FIGURE 10. Function $x(t) = \tan(t)$.

Example 20.3 (Non-Uniqueness)**.** *Consider now the ODE*

$$x' = x^{2/3}, \qquad x(0) = 0.$$

Clearly $x(t) = 0$ *is a solution. Alternatively, by variable separation,*

$$\int \frac{dx}{x^{2/3}} = \int 1 \; dt + C.$$

This implies that

$$3x^{1/3} = t + C$$

or

$$x(t) = \left(\frac{t + C}{3}\right)^3.$$

Using the initial condition, we find that $C = 0$, *i.e.*

$$x(t) = \left(\frac{t}{3}\right)^3.$$

We check that

$$x'(t) = \frac{3}{3}(t/3)^2 = \left((t/3)^6\right)^{1/3} = (x^2)^{1/3}.$$

Hence, we found 2 solutions (non-unique). There are, in fact, infinitly many solutions.

Systems of ODEs are treated similarly. We are given $F(t,x) : (a,b)\times \mathbb{R}^n \to \mathbb{R}^n$,

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \qquad x'(t) = \begin{pmatrix} x_1'(t) \\ x_2'(t) \\ \vdots \\ x_n'(t) \end{pmatrix}.$$

We look for solutions $x_i(t) \in C^1(a,b)$ for $i = 1, ..., n$ satisfying

$$\begin{cases} x'(t) = F(t,x), & t \in (a,b) \\ x(0) = x_0, \end{cases} \tag{29}$$

where $x_0 \in \mathbb{R}^n$ and $t_0 \in (a,b)$ are given.

Definition 20.1 (Uniform Lipschitz)**.** *Assume that F is continuous on $[a,b]\times\mathbb{R}^n$. Then F satisfies a uniform Lipschitz condition if*

$$|F(t,x_1) - F(t,x_2)| \le L|x_1 - x_1|$$

for all $t \in [a,b]$ and $x_1, x_2 \in \mathbb{R}^n$.

Theorem 20.1 (Existence and Uniqueness)**.** *If F satisfies a uniform Lipschitz condition, the system of ODES* (29) *has a unique solution on $[a,b]$.*

Proof. (Sketch using Picard Iteration). We start by noting that integrating the differential equation gives

$$x(t) = x_0 + \int_{t_0}^{t} x'(s)\, ds$$

or

$$x(t) = x_0 + \int_{t_0}^{t} F(s, x(s))\, ds \tag{30}$$

This is a fixed point equation for the vector valued function $x(t)$ on $[a,b]$, or more precisely, $x \in (C([a,b]))^n$.

We shall show that this is, in fact, a contraction on a neighborhood $B_\delta(t_0) = [t_0 - \delta, t_0 + \delta]$ for some $\delta > 0$ to be determined. For $y : B_\delta(t_0) \to \mathbb{R}^n$ with $y \in (C(B_\delta(t_0)))^n$, we define

$$|||y||| = \max_{t\in B_\delta(t_0)} \|y(t)\|_\infty$$

and set

$$G(y)(t) = x_0 + \int_{t_0}^{t} F(s, y(s))\ ds.$$

Note that when y is in $(C(B_\delta(t_0)))^n$ so is $G(y)$. Moreover, $y, z \in (C(B_\delta(t_0)))^n$,

$$\begin{aligned} \|G(y)(t) - G(z)(t)\|_\infty &= \left\| \int_{t_0}^{t} [F(s, y(s)) - F(s, z(s))]\, ds \right\|_\infty \\ &\leq L|t - t_0|\ |||y - z||| \end{aligned}$$

and hence

$$|||G(y) - G(z)||| \leq \delta L |||y - z|||.$$

This implies that $G : (C(B_\delta(t_0)))^n \to (C(B_\delta(t_0)))^n$ satisfies

$$|||G(y) - G(z||| \leq \rho |||y - z|||, \qquad \text{for all } t \in B_\delta(t_0)$$

with $\rho < 1$ when $\delta < L^{-1}$. This is a generalization of the contraction mapping definition given in Definition 18.2 and shows that G is a contraction mapping of $(C(B_\delta(t_0)))^n$ into itself. The corresponding contraction mapping theorem implies that there is a unique fixed point $x^* \in (C(B_\delta(t_0)))^n$ satisfying (30). In fact, the Picard iteration

$$x^{j+1} = G(x^j)$$

converges to x^* for any starting iterate, e.g. $x^0(t) = x_0$ for all t.

Th above argument guarantees the existence and uniqueness of the solution $x(t)$ in the δ neighborhood of t_0. Repeating this argument with t_0 replaced by $t_0 - \delta$ and t_0 replaced by $t_0 + \delta$ guarantees existence and uniqueness on the neighborhood of t_0 of size 2δ. It is clear that repeating this argument a sufficient number of times continues the solution on all of $[a, b]$.

Finally, differentiating the fixed point equation (30) shows that

$$(x^*)' = F(t, x^*)$$

and completes the proof of the theorem. □

21. Lecture 21.

21.1. Numerical Ordinary Differential Equations (ODEs).

Without loss of generality, we shall develop numerical ODE schemes for the initial value problem, i.e.,

$$\begin{aligned} x'(t) &= f(t, x(t)), \qquad \text{for } t > 0, \\ x(0) &= x_0. \end{aligned} \tag{31}$$

In this case, the numerical ODE approximations are based on a sequence $0 = t_0 < t_1 < \cdots$ and provide an approximation only at these points.

We discuss two techniques to develop numerical ODE schemes for (31) in these lectures:

(1) Replace $\frac{d}{dt}x(t)$ by a finite difference approximation.
(2) Integrate:

$$x(t_{k+1}) - x(t_k) = \int_{t_k}^{t_{k+1}} \frac{d}{dt}x(t)\ dt = \int_{t_k}^{t_{k+1}} f(t, x(t))\ dt$$

or

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} f(t, x(t))\ dt$$

and replace the integral by a quadrature.

In this lecture, we only consider the first approach.

21.2. Finite Difference Approximations.

We start with the finite difference approximation to the derivative, namely,

$$\frac{d}{dt}x(t) \approx \frac{x(t) - x(t_k)}{t - t_k}.$$

The quantity on the right is a forward difference and we have the error identity:

$$\frac{x(t) - x(t_k)}{t - t_k} = f(t_k, x(t_k)) + O(|t - t_k|).$$

To derive the numerical ODE scheme, we throw away the error term, replace t by t_{k+1} and replace $x(t_j)$ by x_j to obtain

$$\frac{x_{k+1} - x_k}{h_k} = f(t_k, x_k),$$

where $h_k = t_{k+1} - t_k$ and $x_k \approx x(t_k)$. This scheme is called the *Forward Euler* time stepping method.

Similarly, but starting from the backward difference approximation to the derivative yields

$$\frac{x(t_{k+1}) - x(t)}{t_{k+1} - t} = f(t_{k+1}, x(t_{k+1})) + O(|t_{k+1} - t|).$$

leads to (replacing t by t_k in this case) the *Backward Euler* scheme:

$$\frac{x_{k+1} - x_k}{h_k} = f(t_{k+1}, x_{k+1}),$$

where again $x_k \approx x(t_k)$.

Remark 21.1 (Explicit/Implicit Schemes)*.* Given x_0, the Forward Euler scheme determines x_k recursively according to the relation

$$x_{k+1} = x_k + h_k f(t_k, x_k).$$

Given x_k, the computation of the right side of the above relation only involves addition, multiplication and the evaluation of $f(t_k, x_k)$. This is called an explicit method. In contrast, the Backward Euler determines x_k recursively according to the relation

$$x_{k+1} - h_k f(t_{k+1}, x_{k+1}) = x_k.$$

Given x_k, the problem of determining x_{k+1} is generally nonlinear. It can, however, be written as a fixed point iteration

$$x = x_k + h_k f(t_{k+1}, x)$$

and one could use the Picard iteration or Newton's method to compute its solution (if it exists). ODE schemes that require the solution of (nonlinear) equations are called implicit. For certain types of problems, implicit methods have better stability properties.

Remark 21.2 (Systems of ODEs)*.* We always derive ODE methods for a single ODE. As we shall see, schemes for systems of ODEs follow immediately from schemes for the single variable ODE.

21.3. **Taylor series schemes.** ODE schemes can also be derived using a "Taylor series technique" as illustrated in the following examples.

Example 21.1 (Forward Euler)**.** *Set $h_j = t_{j+1} - t_j$ so that*

$$x(t_{j+1}) = x(t_j) + h_j x'(t_j) + O(h_j^2).$$

Replace $x(t_j)$ by an approximation x_j and throw away the $O(h_j^2)$ term:

$$x_{j+1} = x_j + h_j x'(t_j) = x_j + h_j f(t_j, x_j).$$

This is again the forward Euler scheme.

Example 21.2 (Second Order)**.**

$$x(t_{j+1}) = x(t_j) + h_j x'(t_j) + \frac{h_j^2}{2} x''(t_j) + O(h_j^3).$$

As before,

$$x'(t_j) = f(t_j, x_j)$$

but now

$$\begin{aligned} x''(t) &= \frac{d}{dt} x'(t) = \frac{d}{dt} f(t, x(t)) \\ &= \frac{\partial}{\partial t} f(t, x(t)) + \frac{\partial}{\partial x} f(t, x(t)) x'(t) := f_t + f_x f. \end{aligned}$$

We throw away the $O(h_j^3)$ term and replace $x(t_j)$ by x_j to arrive at

$$x_{j+1} = x_j + h_j f(t_j, x_j) + \frac{h_j^2}{2} \left(f_t(t_j, x_j) + f_x(t_j, x_j) f(t_j, x_j) \right).$$

Example 21.3 (Third Order)**.**

$$x(t_{j+1}) = x(t_j) + h_j x'(t_j) + \frac{h_j^2}{2} x''(t_j) + \frac{h_j^3}{6} x'''(t_j) + O(h_j^4).$$

Here, we also use

$$x'''(t) = \frac{d}{dt} x''(t) = \frac{d}{dt}(f_t + f_x f) = f_{tt} + f_{tx} f + f(f_{xt} + f_{xx} f) + f_x(f_t + f_x f)$$

to deduce the scheme

$$\begin{aligned} x_{j+1} = x_j &+ h_j f(t_j, x_j) + \frac{h_j^2}{2} \left(f_t + f_x f \right) \\ &+ \frac{h_j^3}{6} \left(f_{tt} + 2 f_{tx} f + f^2 f_{xx} + f_x f_t + f_x^2 f \right). \end{aligned}$$

All the f's and derivatives are evaluated at (t_j, x_j).

The methods of Examples 21.2 and 21.3 are not used in practice. The reason is that you can derive higher order methods which only require the user to provide a routine for $f(t, x)$ (not its derivatives).

However, we shall use these methods to derive higher order methods only involving $f(t, x)$ for various values of t and x.

22. Lecture 22: Runge-Kutta Methods.

The methods below are based on the Taylor series method. As in the previous lecture, the order is obtained by scaling the equation so that there is a difference which limits to a derivative. The order is then the power of the error appearing in the scaled equation,

Example 22.1 (Second Order)**.** *From Example 21.2:*

$$\begin{aligned} x(t_{k+1}) &= x(t_k) + h_k f(t_k, x(t_k)) \\ &\quad + \frac{h_k^2}{2}\left(f_t(t_k, x(t_k)) + f_x(t_k, x(t_k)) f(t_k, x(t_k))\right) + O(h_k^3) \\ &= x(t_k) + h_k \widetilde{f} + \frac{h_k^2}{2}\left(\widetilde{f}_t + \widetilde{f}_x \widetilde{f}\right) + O(h_k^3). \end{aligned} \tag{32}$$

Here and what follows, $\widetilde{f}$, $\widetilde{f}_t$, $\widetilde{f}_x$ and $\widetilde{\nabla} f = (\widetilde{f}_t, \widetilde{f}_x)^t$ denotes the corresponding variable evaluated at $(t_k, x(t_k))$, e.g., $\widetilde{f} := f(t_k, x(t_k))$. We look for a method based on

$$x(t_{k+1}) \overset{?}{=} x(t_k) + \omega_1 h_k \widetilde{f} + \omega_2 h_k f\left(t_k + \alpha h_k, x(t_k) + \alpha h_k \widetilde{f}\right) + O(h_k^3)$$

for some parameters ω_1, ω_2, α chosen so that the error term is still third order.

By Taylor's Theorem,

$$\begin{aligned} f\left(t_k + \alpha h_k, x(t_k) + \alpha h_k \widetilde{f}\right) &= \widetilde{f} + \widetilde{\nabla} \widetilde{f} \cdot (\alpha h_k, \alpha h_k \widetilde{f}) + O(h_k^2) \\ &= \widetilde{f} + \alpha h_k (\widetilde{f}_t + \widetilde{f}_x \widetilde{f}) + O(h_k^2) \end{aligned}$$

where we expanded around $(t_k, x(t_k))$. Putting these together gives

$$x(t_{k+1}) \overset{?}{=} x(t_k) + h_k(\omega_1 + \omega_2)\widetilde{f} + \alpha h_k^2 \omega_2 (\widetilde{f}_t + \widetilde{f}_x \widetilde{f}) + O(h_k^3).$$

Comparing this with (32) *we see that to maintain third order, we need to set*

$$\omega_1 + \omega_2 = 1 \quad \textit{and} \quad \omega_2 \alpha = \frac{1}{2} \tag{33}$$

(3 unknowns, 2 equations, multiple solutions). The corresponding numerical methods read

$$\begin{aligned} x_{k+1} &= x_k + h_k(\omega_1 F_1 + \omega_2 F_2) \quad \textit{where} \\ F_1 &= f(t_k, x_k) \quad \textit{and } F_2 = f(t_k + \alpha h_k, x_k + \alpha h_k F_1) \end{aligned}$$

and they are all second order provided that (33) holds.

Example 22.2 (Heun's Method)**.** *We set $\omega_1 = \omega_2 = \frac{1}{2}$ and $\alpha = 1$ to arrive at*

$$x_{k+1} = x_k + \frac{1}{2} h_k (F_1 + F_2),$$

where

$$
\begin{aligned}
F_1 &= f(t_k, x_k)\\
F_2 &= f(t_k + h_k, x_k + h_k F_1).
\end{aligned}
$$

Example 22.3 (Modified Euler)**.** *We set* $\omega_1 = 0$*,* $\omega_2 = 1$ *and* $\alpha = \frac{1}{2}$ *to arrive at*

$$x_{k+1} = x_k + h_k F_2,$$

where

$$
\begin{aligned}
F_1 &= f(t_k, x_k)\\
F_2 &= f(t_k + \frac{1}{2}h_k, x_k + \frac{1}{2}h_k F_1).
\end{aligned}
$$

Remark 22.1 (Heun's method)*.* Consider the case of Heun's method applied to the ODE

$$x'(t) = f(t). \tag{34}$$

In this case,

$$\int_{t_k}^{t_{k+1}} x'(t)\ dt = x(t_{k+1}) - x(t_k) = \frac{h_k}{2}\left(f(t_k) + f(t_k + h_k)\right) + O(h_k^3).$$

Thus

$$\int_{t_k}^{t_{k+1}} f(t)\ dt = \int_{t_k}^{t_{k+1}} x'\ dt \approx \frac{h_k}{2}\left(f(t_k) + f(t_k + h_k)\right),$$

which is the Trapezoidal rule. The latter is locally 3'rd order and globally 2'nd order which is in agreement with the 2'nd order rate already assigned to Heun's method.

Exercise 22.1 (Backward and Forward Euler)**.** *Use a similar reasoning to show that Backward and Forward Euler methods are first order when applied to* (34)

Remark 22.2 (Modified Euler)*.* Similar to the above remark, in the case of Modifie Euler applied to (34),

$$\int_{t_k}^{t_{k+1}} f(t)\ dt = x(t_{k+1}) - x(t_k) \approx h_k f(t_k + h_k/2).$$

This is the midpoint quadrature which is also globally second order which is agreement the (global) order of the modified Euler being 2.

Heun's method and the modified Euler method are called (explicit) Runge-Kutta methods. High order Runge-Kutta methods are tedious to derive but have been extensively studied. One such method, which is widely used, is the Runge-Kutta 4'th order method:

$$x_{k+1} = x_k + \frac{h_k}{6}\left(F_1 + 2F_2 + 2F_3 + F_4\right),$$

where

$$\begin{aligned}
F_1 &= f(t_k, x_k),\\
F_2 &= f(t_k + \frac{h_k}{2}, x_k + \frac{h_k}{2}F_1),\\
F_3 &= f(t_k + \frac{h_k}{2}, x_k + \frac{h_k}{2}F_2),\\
F_4 &= f(t_k + h_k, x_k + h_kF_3).
\end{aligned}$$

This method is 4'th order.

Remark 22.3 (Simpson). Applying the Runge-Kutta 4'th order method to

$$x'(t) = f(t)$$

gives

$$\begin{aligned}
\int_{t_k}^{t_{k+1}} f(t)dt = \int_{t_k}^{t_{k+1}} x'(t)\ dt &\approx x_{k+1} - x_k\\
&= \frac{h_k}{6}\left(f(t_k) + 4f(t_k + h_k/2) + f(t_k + h_k)\right).
\end{aligned}$$

This is Simpson's Rule, which is locally 5'th order and globally 4'th order.

23. Lecture 23: Multistep Methods.

We start with the integral representation

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} f(t, x(t))\ dt$$

and apply quadrature to the integral. Note that one only has approximation for $x(t)$ at t_j, $j = 0, ..., k$ so we have to use $t_0, ..., t_k$ and possibly t_{k+1} as quadrature nodes.

The *Adams Bashford* schemes use the $m+1$ nodes $t_k, ..., t_{k-m}$:

$$x(t_{k+1}) = x(t_k) + \sum_{j=0}^{m} w_j f(t_{k-j}, x(t_{k-j})) + O(h^{m+2})$$

when the weights w_j are taken so that

$$I(g) := \int_{t_k}^{t_{k+1}} g(t)\ dt = \sum_{j=0}^{m} w_j g(t_{k-j}) = I_m(g),$$

i.e. the quadrature is exact on $\mathbb{P}^m$. Note that Adams Bashford methods

$$x_{k+1} = x_k + \sum_{j=0}^{m} w_j f(t_{k-j}, x_{k-j})$$

are explicit and globally of order $m+1$.

The *Adams Moulton* schemes use instead the $m+1$ nodes $t_{k+1}, ..., t_{k-m+1}$:

$$x(t_{k+1}) = x(t_k) + \sum_{j=-1}^{m-1} w_j f(t_{k-j}, x(t_{k-j})) + O(h^{m+2}).$$

Adams Moulton methods

$$x_{k+1} = x_k + \sum_{j=-1}^{m-1} w_j f(t_{k-j}, x_{k-j})$$

are implicit since $f(t_{k+1}, x_{k+1})$ appears and globally of order $m+1$.

In general, the ODE schemes are

$$x_{k+1} = x_k + \sum_{j=\sigma}^{m+\sigma} w_j f(t_{k-j}, x_{k-j})$$

with $\sigma = 0$ for Adams Bashford and $\sigma = -1$ for Adams Moulton.

Remark 23.1 (Non Uniform Time-steps). If one only has

$$t_0 < t_1 < t_2 < ...$$

(i.e. no structure), then new quadrature weights need to be computed at each time step. Also, the scheme cannot be analyzed.

Remark 23.2 (Starting Values). When $\sigma = 0$, one needs the starting values

$$x_0, x_1, ..., x_m$$

to compute $x_{m+1}, x_{m+2}, ...$ The values for $x_1, ..., x_m$ need to be computed by some other method.

When $\sigma = -1$, one needs the starting values

$$x_0, x_1, ..., x_{m-1}$$

to compute $x_m, x_{m+1}, ...$ The values for $x_1, ..., x_{m-1}$ need to be computed by some other method.

Remark 23.3 (More General). The methods can be made still more general, e.g.

$$\sum_{i=0}^{m} \alpha_i x_{j-i} = \sum_{i=0}^{m} \beta_i f(t_{j-i}, x_{j-i})$$

with $\alpha_0 = 1$. In this case, the method is explicit if $\beta_0 = 0$, otherwise, it is implicit. The method computes x_j from $x_{j-1}, \dots, x_{j-m}$ and requires starting values $x_0, \dots, x_{m-1}$.

From here onward, we assume *a uniform time step*, i.e.

$$t_j = t_0 + jh$$

for some fixed $h > 0$.

We return to Adams Bashford and compute the weights for the quadrature problem

$$\widehat{I}(g) := \int_m^{m+1} g(x)\ dx \approx \sum_{j=0}^{m} \widehat{w}_j g(m-j) =: \widehat{I}_m(g)$$

so that $\widehat{I}_m$ is exact on $\mathbb{P}^m$. Now for $t_j = t_0 + jh$, we get the quadrature

$$I(g) := \int_{t_k}^{t_{k+1}} g(t)\ dt \approx h \sum_{j=0}^{m} \widehat{w}_j g(t_{k-j}) = I_m(g)$$

by translating $\widehat{I}_m(g)$.

Example 23.1 (3rd order). *We set $m = 2$:*

$$\widehat{I}(g) = \int_2^3 g(x)\ dx \approx \sum_{j=0}^{2} \widehat{w}_j g(2-j) =: \widehat{I}_2(g).$$

To make it exact for $\mathbb{P}^2$, we require

$$\begin{aligned}
\widehat{I}(1) &= \int_2^3 1\ dx = 1 = \widehat{w}_0 + \widehat{w}_1 + \widehat{w}_2 = \widehat{I}_2(1), \\
\widehat{I}(x) &= \int_2^3 x\ dx = \left.\frac{x^2}{2}\right|_2^3 = \frac{5}{2} = 2\widehat{w}_0 + \widehat{w}_1 = \widehat{I}_2(x), \\
\widehat{I}(x^2) &= \int_2^3 x^2\ dx = \left.\frac{x^3}{3}\right|_2^3 = \frac{19}{3} = 4\widehat{w}_0 + \widehat{w}_1 = \widehat{I}_2(x^2).
\end{aligned}$$

The last two conditions imply

$$2\widehat{w}_0 = \frac{19}{3} - \frac{5}{2} = \frac{23}{6}$$

and so

$$\widehat{w}_0 = \frac{23}{12}.$$

Using this in the second constraint yields

$$\frac{23}{6} + \widehat{w}_1 = \frac{5}{2}$$

and so

$$\widehat{w}_1 = \frac{5}{2} - \frac{23}{6} = -\frac{8}{6} = -\frac{4}{3}.$$

It remains to use the first constraint to get

$$\widehat{w}_2 = 1 - \frac{23}{12} + \frac{4}{3} = \frac{12 - 23 + 16}{12} = \frac{5}{12}.$$

After translating the quadrature to the equally spaced grid of size h, we obtain

$$x_{k+1} = x_k + \frac{h}{12}\left\{23f(t_k, x_k) - 16f(t_{k-1}, x_{k-1}) + 5f(t_{k-2}, x_{k-2})\right\}.$$

Remark 23.4. We could have translated the scheme on any equally spaced unit size grid and obtained the same result. For example, we could have computed the scheme

$$\widehat{I}(g) = \int_0^1 g(x)\,dx \approx \sum_{i=-2}^{0} \widehat{w}_i g(i) := \widehat{I}_m(g).$$

Making this exact on $\mathbb{P}^2$ leads to the system:

$$\begin{aligned}\widehat{I}(1) &= \int_0^1 1\ dx = 1 = \widehat{w}_{-2} + \widehat{w}_{-1} + \widehat{w}_0 = \widehat{I}_2(1),\\ \widehat{I}(x) &= \int_0^1 x\ dx = \frac{1}{2} = -2\widehat{w}_{-2} - \widehat{w}_{-1} = \widehat{I}_2(x),\\ \widehat{I}(x^2) &= \int_0^1 x^2\ dx = \frac{1}{3} = 4\widehat{w}_{-2} + \widehat{w}_{-1} = \widehat{I}_2(x^2).\end{aligned}$$

Its solution is easily seen to be $(w_{-2}, w_{-1}, w_0) = (5/12, -4/3, 23/12)$ and translates to the same scheme on the grid of size h.

24. Lecture 24.

Similarly as for Adams-Bashford, we now derive the Adams-Moulton (for equally spaced nodes, i.e. $x_i = x_0 + hi$), we solve the interpolation problem

$$\widehat{I}(g) := \int_{m-1}^{m} g(s)\ ds \approx \sum_{j=0}^{m} \widehat{w}_j g(m-j) =: \widehat{I}_m(g), \tag{35}$$

which is exact on $\mathbb{P}^m$. The resulting ODE scheme is

$$x_k = x_{k-1} + h \sum_{j=0}^{m} \widehat{w}_j f(t_{k-j}, x_{k-j}).$$

Exercise 24.1 (Undetermined Coefficients)**.** *Compute the coefficients* $\{\widehat{w}_j\}_{j=0}^2$ *which makes the quadrature* (35) *exact on* $\mathbb{P}^2$ *using undetermined coefficients.*

24.1. Backwards Differences.

Consider approximating the derivative

$$f'(x) \approx \alpha_0 f(x) + \alpha_1 f(x-h) + ... + \alpha_m f(x-hm).$$

Recall that to find such approximation, we first find the polynomial $p_m \in \mathbb{P}^m$ interpolating

$$p_m(t) = f(t) \tag{36}$$

for $t = x, x-h, ..., x-hm$ and we then set

$$f'(x) \approx p_m'(x).$$

Applying Newton form to solve the interpolation problem (36), we get

$$p_m(t) = \sum_{i=0}^{m} c_i q_i(t), \tag{37}$$

where

$$q_0(t) \equiv 1, \qquad q_j(t) = \prod_{l=0}^{j-1}(t - x + lh).$$

As we shall see in the Appendix at the end of this lecture,

$$c_i = \frac{1}{i!} D_h^i f(x). \tag{38}$$

Here

$$\begin{aligned} D_h^0 f(x) &= f(x), \\ D_h^1 f(x) &= \frac{f(x)-f(x-h)}{h}, \\ D_h^{j+1} f(x) &= D_h^1[D_h^j f(x)]. \end{aligned}$$

Example 24.1 (D_h^2)**.**

$$\begin{aligned} D_h^2 f(x) = D_h^1 &\left[\frac{f(x)-f(x-h)}{h}\right] \\ &= \frac{\frac{f(x)-f(x-h)}{h} - \frac{f(x-h)-f(x-2h)}{h}}{h} = \frac{f(x)-2f(x-h)+f(x-2h)}{h^2}. \end{aligned}$$

Example 24.2 (D_h^3)**.**

$$D_h^3 f(x) = \frac{f(x)-3f(x-h)+3f(x-2h)-f(x-3h)}{h^3}.$$

Notice that the coefficients $1, -3, 3, -1$ *in front of the functions are the binomial coefficients.*

Returning to the Newton's form for p_m, we find that

$$\begin{aligned} p_m(t) = f(x_0) + D_h^1 f(x)(t-x) &+ \frac{1}{2!} D_h^2 f(x)(t-x)(t-x+h) \\ &+ \frac{1}{3!} D_h^3 f(x)(t-x)(t-x+h)(t-x+2h) \\ &+ ... + \frac{1}{m!} D_h^m f(x)(t-x)(t-x+h) \cdot ... \cdot (t-x+(m-1)h). \end{aligned}$$

Differentiating the above expression and taking $t = x$, we arrive at

$$\begin{aligned} p_m'(x) = 0 + D_h^1 f(x) &+ \frac{1}{2!} D_h^2 f(x) h \\ &+ \frac{1}{3!} D_h^3 f(x) h \cdot 2h \\ &+ ... + \frac{1}{m!} D_h^m f(x) h(2h) \cdot ... \cdot (m-1)h, \end{aligned}$$

i.e.

$$p_m'(x) = \sum_{i=1}^{m} \frac{h^{i-1}}{i} D_h^i f(x)$$

and so

$$f'(x) \approx \sum_{i=1}^{m} \frac{h^{i-1}}{i} D_h^i f(x). \tag{39}$$

The corresponding numerical ODE schemes approximating (31) of Lecture 21 come from first applying the above finite difference approximation to the derivative $x'(t_m)$, i.e., replacing $f(x)$ by $x(t)$. We obtain

$$\sum_{i=1}^{m} \frac{h^{i-1}}{i} D_h^i x(t_m) = f(t_m, x_m) + O(h^m)$$

with the difference becoming

$$D_h x(t_m) = \frac{x(t_m) - x(t_m - h)}{h}.$$

The corresponding numerical ODE scheme follows as usual and is given by

$$\sum_{i=1}^{m} \frac{h^{i-1}}{i} D_h^i x_m = f(t_m, x_m)$$

with

$$D_h x_m = \frac{x_m - x_{m-1}}{h}.$$

They are all implicit.

Example 24.3 (1st order method)**.**

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}.$$

The corresponding ODE scheme is the Backward Euler scheme

$$\frac{x_j - x_{j-1}}{h} = f(t_j, x_j).$$

Example 24.4 (2nd order method)**.**

$$\begin{aligned} f'(x) \approx & \frac{f(x) - f(x-h)}{h} + \frac{f(x) - 2f(x-h) + f(x=2h)}{2h} \\ = & \frac{3/2 f(x) - 2f(x-h) + 1/2 f(x-2h)}{h}. \end{aligned}$$

The corresponding ODE scheme is

$$\frac{3/2 x_j - 2x_{j-1} + 1/2 x_{j-2}}{h} = f(t_j, x_j).$$

Example 24.5 (3rd order method)**.**

$$\begin{aligned} f'(x) \approx & \frac{3/2 f(x) - 2f(x-h) + 1/2 f(x-2h)}{h} \\ & + \frac{f(x) - 3f(x-h) + 3f(x-2h) - f(x-3h)}{3h} \\ = & \frac{1}{6h} \left(11 f(x) - 18 f(x-h) + 9 f(x-2h) - 2 f(x-3h) \right). \end{aligned}$$

The corresponding ODE scheme is

$$\frac{11x_j - 18x_{j-1} + 9x_{j-2} - 2x_{j-3}}{6h} = f(t_j, x_j).$$

24.2. **Appendix of Lecture 24.** We now return to the justification of (38) and refer to Section 6.2 of the book (Kincaid-Cheney). There they introduce the divided difference notation. In the case of our Newton form problem (37), the divided difference symbol $f[x, x-h, \ldots, x-ih]$ is the coefficient c_i, i.e.,

$$f[x, x-h, ..., x-hi] := c_i,$$

where c_i are the coefficients appearing in the Newton form expansion (37).

Clearly, we have

$$c_0 = f(x) \text{ and hence } f[x] := f(x). \tag{40}$$

On the other hand, Theorem 1 of Section 6.2 of Kincaid-Cheney implies that

$$f[x_0, x_1, ..., x_n] = \frac{f[x_1, ..., x_n] - f[x_0, ..., x_{n-1}]}{x_n - x_0}. \tag{41}$$

We will show, by induction, that

$$f[x, x-h, ..., x-hi] = \frac{1}{i!} D_h^i f(x). \tag{42}$$

That (42) holds for $i = 0$ follows from (40). Suppose that (42) holds for $i = l$. This implies that implies that

$$f[x, x-h, ..., x-hl] = \frac{1}{l!} D_h^l f(x)$$

and

$$f[x-h, x-2h, ..., x-h(l+1)] = \frac{1}{l!} D_h^l f(x-h).$$

Therefore, by (41),

$$\begin{aligned} f[x, x-h, ..., x-(l+1)h] &= \frac{f[x-h, ..., x-(l+1)h] - f[x, ..., x-lh]}{x-(l+1)h-x} \\ &= \frac{D_h^l f(x) - D_h^l f(x-h)}{h(l+1)} \frac{1}{l!} = \frac{1}{(l+1)!} D_h^{l+1} f(x). \end{aligned}$$

This ends the induction proof.

25. Lecture 25: Errors in ODE schemes.

Stability and local truncation error are two central notions when studying numerical methods for ODE schemes. We first provide an intuitive description.

(1) *Local truncation error:* The numerical schemes that we have seen were obtained by dropping error terms. The truncation error characterized this error made at each step.
(2) *Stability:* If you make an error computing x_j when approximating $x(t_j)$, that error propagates through all remaining steps (this is unavoidable). A stable scheme prevents this error from increasing.

We will only consider the Adams-Bashford (AB) and Adams-Moulton (AM) schemes for the analysis.

Regarding the *Stability*, the error at any time propagates but does not grow too much. A necessary condition for stability, which we will take as the definition of stability in this class, is the following:

Definition 25.1 (Stability). *We say that a scheme is* stable *if when applied to*

$$x'(t) = 0, \qquad x(t_0) = \varepsilon,$$

the approximate solutions remains bounded at any fixed time independently of the mesh size h.

In the case $t_0 = 0$ and $h = 1/m$, $x^h_m \approx x(1)$ then stability holds at time $t = 1$ if

$$|x^h_m| \leq C \qquad \text{as} \qquad h = 1/m \to 0.$$

Example 25.1. *AB/AM The AB or AM schemes applied to the above ODE read*

$$x^h_{k+1} = x^h_k, \qquad x^h_0 = \varepsilon \qquad \Longrightarrow \qquad x^h_k = \varepsilon$$

and are therefore stable.

We now discuss the local truncation error and start with a definition.

Definition 25.2 (Local truncation error). *The* local truncation error *is the error made when one substitutes the solution into the numerical ODE scheme.*

For example,

$$\rho_h(k+1) := \frac{x(t_{k+1}) - x(t_k)}{h} - \sum_{j=\sigma}^{m+\sigma} \widehat{w}_j f(t_{k-j}, x(t_j - k))$$

is the local truncation for the numerical scheme

$$\frac{x_{k+1} - x_k}{h} = \sum_{j=\sigma}^{m+\sigma} \widehat{w}_j f(t_{k-j}, x_{j-k}).$$

Note that $\rho_h(k+1) = O(h^{m+1})$ for AB/AM.

The error between $x(t_k)$ and x_k given by the AB scheme is described in the following theorem.

Theorem 25.1 (Error Estimate for Adams Bashford)**.** *Assume that f satisfies a uniform Lipschitz condition, i.e.*

$$|f(t,x) - f(t,y)| \leq M|x-y|, \qquad x, y \in \mathbb{R}, \qquad t \in [0,T].$$

Let $h > 0$, $t_k := kh$ and $x_0, x_1, ..$ be the approximation sequence. Set $\rho_h := \max_{m<k\leq T/h} |\rho_h(k)|$, $\overline{w} := \max_{j=0,\dots,m} |\widehat{w}_j|$ and $\varepsilon_k := \max_{j=0,\dots,k} |x(t_j) - x_j|$. Then for $m < k \leq T/h$, there holds

$$\varepsilon_k \leq e^{\delta t_k}[\varepsilon_m + \rho_h/\delta],$$

where $\delta := (m+1)\overline{w}M$.

Proof. We start by noting that the Lipschitz assumption on f guarantees a unique solution of the ODE for $t \geq t_0 = 0$. The AB approximate solutions are given by

$$x_{k+1} = x_k + h \sum_{j=0}^{m} \widehat{w}_j f(t_{k-j}, x_{k-j}).$$

Subtracting this from the local truncation relation

$$\frac{x(t_{k+1}) - x(t_k)}{h} - \sum_{j=0}^{m} \widehat{w}_j f(t_{k-j}, x(t_{k-j})) =: \rho_h(k+1)$$

and setting $e_k := x(t_k) - x_k$ gives

$$e_{k+1} = e_k + h \sum_{j=0}^{m} \widehat{w}_j [f(t_{k-j}, x(t_{k-j})) - f(t_{k-j}, x_{k-j})] + h\rho_h(k+1).$$

Now apply the Lipschitz condition and the triangle inequality

$$\begin{aligned} |e_{k+1}| &\leq |e_k| + h \sum_{j=0}^{m} |\widehat{w}_j| |f(t_{k-j}, x(t_{k-j})) - f(t_{k-j}, x_{k-j})| + h|\rho_h(k+1)| \\ &\leq |e_k| + hM \sum_{j=0}^{m} |\widehat{w}_j| |x(t_{k-j}) - x_{k-j})| + h|\rho_h(k+1)| \\ &\leq |e_k| + hM\overline{w} \sum_{j=0}^{m} |e_{k-j}| + h|\rho_h(k+1)|. \end{aligned}$$

Using the definition of ρ_h, ε_k and δ, we get

$$|e_{k+1}| \leq |e_k| + \delta h \varepsilon_k + h\rho_h,$$

i.e.

$$\varepsilon_{k+1} \leq (1 + \delta h)\varepsilon_k + h\rho_h.$$

Repeating the application of the above estimate gives

$$\begin{aligned}\varepsilon_k &\leq (1 + \delta h)\varepsilon_{k-1} + h\rho_h \\ &\leq (1 + \delta h)^2 \varepsilon_{k-2} + h\rho_h(1 + (1 + \delta h)) \\ &\vdots \\ &\leq (1 + \delta h)^{k-m}\varepsilon_m + h\rho_h \sum_{j=0}^{k-m-1} (1 + \delta h)^j.\end{aligned}$$

Now $k \leq T/h$, so

$$(1 + \delta h)^{k-m} \leq (e^{\delta h})^{k-m} \leq e^{\delta k h} \leq e^{\delta T}.$$

Also

$$\sum_{j=0}^{k-m-1} (1 + \delta h)^j = \frac{(1 + \delta h)^{k-m} - 1}{(1 + \delta h) - 1} \leq \frac{e^{\delta T}}{\delta h}.$$

Thus,

$$\varepsilon_k \leq e^{\delta T}(\varepsilon_m + \rho_h/\delta),$$

which is the desired result. □

Before providing a similar result for the AM scheme, we first show that the scheme is well defined. Recall the AM is implicit and x_k is the solution (if it exists) to

$$x_{k+1} = x_k + h \sum_{j=-1}^{m-1} \widehat{w}_j f(t_{k-j}, x_{k-j}).$$

Lemma 25.1 (AM - Well defined)**.** *Assume that f satisfies a uniform Lipschitz condition with constant M. Given $x_0, ..., x_k$. Then the $k+1$ iterate of the $m+1$ points AM is well defined provided $h < 1/(|\widehat{w}_{-1}|M)$.*

Proof. We consider the fixed point iteration for x_{k+1}, namely

$$\begin{aligned}y_0 &= x_k \\ y_{l+1} &= x_k + h\widehat{w}_{-1} f(t_{k+1}, y_l) + h \sum_{j=0}^{m-1} \widehat{w}_j f(t_{k-j}, x_{k-j}) := G(y_l)\end{aligned}$$

i.e. $y_{l+1} = G(y_l)$. Note that

$$|G(y) - G(z)| = h|\widehat{w}_{-1}|[f(t_{k+1}, y) - f(t_{k+1}, z)]$$

so that the Lipschitz condition implies

$$|G(y) - G(z)| \leq h|\widehat{w}_{-1}|M|y - z|.$$

This means that under the assumption $h|\widehat{w}_{-1}|M < 1$, G is a contraction mapping so the sequence y_l converges to a unique fixed point, x_{k+1}. □

26. Lecture 26: Stiff ODE's (systems).

In the previous lecture, we provided an error analysis for the Adams-Bashford schemes (see Theorem 25.1) and showed that Adams-Moulton schemes were well defined provided that $h \leq h_0$ (Lemma 25.1). Here h_0 depends on $|\widehat{w}_{-1}|$ and the uniform Lipschitz constant M of f. To analyze Adams-Moulton, we need the following lemma.

Lemma 26.1. *Let $\alpha, \beta, h > 0$ with $\alpha h \leq 1/2$. Then,*

$$(1-\alpha h)^{-1} \leq (1+2\alpha h),$$
$$(1-\alpha h)^{-1}(1+\beta h) \leq e^{\gamma h},$$

where $\gamma := \beta + 2\alpha$ and

$$\sum_{j=0}^{n-1} e^{j\gamma h} \leq \frac{e^{n\gamma h}}{\gamma h}.$$

Proof. The assumption $\alpha h \leq 1/2$ implies that $-1/2 \leq -\alpha h$, $1/2 \leq 1-\alpha h$ or

$$(1-\alpha h)^{-1} \leq 2.$$

Thus

$$\begin{aligned}(1-\alpha h)^{-1} &= 1+\alpha h+(\alpha h)^2+(\alpha h)^3+\ldots = 1+\alpha h+\alpha^2h^2(1+\alpha h+(\alpha h)^2+\ldots)\\ &= 1+\alpha h+\alpha^2h^2(1-\alpha h)^{-1} = 1+\alpha h+\alpha h\underbrace{(\alpha h(1-\alpha h)^{-1})}_{\leq 1}\\ &\leq 1+2\alpha h.\end{aligned}$$

Also,

$$1+\beta h \leq 1+\beta h+\frac{(\beta h)^2}{2!}+\ldots = e^{\beta h}$$

so

$$(1-\alpha h)^{-1}(1+\beta h) \leq e^{2\alpha h}e^{\beta h} = e^{\gamma h}.$$

Finally,

$$\sum_{j=0}^{n-1} e^{j\gamma h} = \sum_{j=0}^{n-1}(e^{\gamma h})^j = \frac{e^{n\gamma h}-1}{e^{\gamma h}-1} \leq \frac{e^{n\gamma h}}{1+\gamma h-1}.$$

□

We note that the local truncation error $\rho_h(k+1)$ at the $k+1$'st step is defined by

$$\begin{aligned}x(t_{k+1}) - h\widehat{w}_{-1}f(t_{k+1},x(t_{k+1})) &= x(t_k)\\ &\quad + h\sum_{j=0}^{m-1}\widehat{w}_j f(t_{k-j},x(t_{k-j})) + h\rho_h(k+1).\end{aligned} \tag{43}$$

We can now provide the error analysis for Adams-Moulton scheme.

Theorem 26.1 (Error Estimate for Adams Moulton)**.** *Assume that f satisfies a uniform Lipschitz condition, i.e.*

$$|f(t,x)-f(t,y)| \le M|x-y|, \qquad x,y\in\mathbb{R}, \qquad t\in[0,T].$$

Let $h>0$, $t_k := kh$ and $x_0, x_1, ..$ be the sequence Adams-Moulton approximates. Set $\rho_h := \max_{m-1<k\le T/h}|\rho_h(k)|$, $\overline{w} := \max_{j=0,\dots,m-1}|\widehat{w}_j|$ and $\varepsilon_k := \max_{j=0,\dots,k}|x(t_j)-x_j|$. Assume that $h\le h_0 = (2|\widehat{w}_{-1}|M)^{-1}$, then for $m\le k\le T/h$, there holds

$$\varepsilon_k \le e^{\gamma t_k}[\varepsilon_{m-1}+2\rho_h/\gamma],$$

where $\gamma := M(m\overline{w}+2|\widehat{w}_{-1}|)$.

Proof. The AM scheme reads

$$x_{k+1}-h\widehat{w}_{-1}f(t_{k+1},x_{k+1}) = x_k + h\sum_{j=0}^{m-1}\widehat{w}_j f(t_{k-j},x_{k-j}).$$

Subtracting this equation from (43), taking absolute values on both sides and proceeding as in the proof of Theorem 25.1, we get

$$\begin{aligned}|e_{k+1}-h\widehat{w}_{-1}[f(t_{k+1},x(t_{k+1}))-f(t_{k+1},x_{k+1})]| &\le |e_k| \\ &+ h\sum_{j=0}^{m-1}|\widehat{w}_j|M|e_{k-j}| + h\rho_h \le \varepsilon_k(1+mM\overline{w}h)+h\rho_h\end{aligned}$$

where $e_k := x(t_k)-x_k$. Also,

$$\begin{aligned}&|e_{k+1}-h\widehat{w}_{-1}[f(t_{k+1},x(t_{k+1}))-f(t_{k+1},x_{k+1})]| \\ &\qquad \ge |e_{k+1}| - h|\widehat{w}_{-1}||f(t_{k+1},x(t_{k+1}))-f(t_{k+1},x_{k+1})| \\ &\qquad \ge |e_{k+1}| - h|\widehat{w}_{-1}|M|e_{k+1}|.\end{aligned}$$

Combining the above two estimates we get

$$\varepsilon_{k+1}(1-h|\widehat{w}_{-1}|M) \le \varepsilon_k(1+mM\overline{w}h)+h\rho_h$$

which we rewrite as

$$\varepsilon_{k+1} \le (1-h\alpha)^{-1}(1+h\beta)\varepsilon_k + h\rho_h(1-h\alpha)^{-1}$$

where $\alpha = |\widehat{w}_{-1}|M$ and $\beta = mM\overline{w}$. Lemma 26.1 gives (since $h|\widehat{w}_{-1}|M\le 1/2$)

$$\varepsilon_{k+1} \le e^{\gamma h}\varepsilon_k + 2h\rho_h,$$

where

$$\gamma = 2\alpha+\beta = M(m\overline{w}+2|\widehat{w}_{-1}|).$$

Applying repetitively (as in the proof of Theorem 25.1) gives

$$\begin{aligned}\varepsilon_k &\leq e^{(k-m+1)\gamma h}\varepsilon_{m-1} + 2h\rho_h \sum_{j=0}^{k-m} e^{j\gamma h} \\ &\leq e^{\gamma T}(\varepsilon_{m-1} + 2\rho_h/\gamma),\end{aligned}$$

where we used Lemma 26.1 for the second inequality. This ends the proof. □

Stiff ODE's (systems): We start with two examples.

Example 26.1 (Backward Euler). *Consider $x(t) \in \mathbb{R}^2$ satisfying*

$$x'(t) = -\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & M \end{pmatrix}}_{=:A} x(t), \qquad x(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Here M is large and positive. These equations decouple:

$$x_1'(t) = -x_1, \qquad x_2'(t) = -Mx_2.$$

The solution is

$$x_1(t) = e^{-t}, \qquad x_2(t) = e^{-Mt},$$

i.e.

$$x(t) = \begin{pmatrix} e^{-t} \\ e^{-Mt} \end{pmatrix}.$$

Now consider the Backward-Euler (fixed step size h) approximation applied to the system is

$$\frac{1}{h}(x_{j+1} - x_j) = -Ax_{j+1}.$$

(Warning: here the subindices in x_j denote the approximation at $t_j = jh + t_0$, not the components of x.) Thus

$$(I + hA)x_{j+1} = x_j$$

or

$$x_{j+1} = (I + hA)^{-1}x_j = \begin{pmatrix} (1+h)^{-1} & 0 \\ 0 & (1+Mh)^{-1} \end{pmatrix} x_j.$$

This leads to

$$x_j = \begin{pmatrix} (1+h)^{-j} & 0 \\ 0 & (1+Mh)^{-j} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The approximate solutions remain bounded for any j and $h > 0$. By the way, $(1+\alpha)^{-j} \approx e^{-\alpha j}$ and so $(1+Mh)^{-j} \approx e^{-Mt_j}$.

Example 26.2 (Forward Euler)**.** *Consider again the system*

$$x'(t) = -\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & M \end{pmatrix}}_{=:A} x(t), \qquad x(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

The Forward-Euler scheme reads

$$\frac{1}{h}(x_{j+1} - x_j) = -Ax_j.$$

or

$$x_{j+1} = x_j - hAx_j = (I - hA)x_j = \begin{pmatrix} (1-h) & 0 \\ 0 & (1-Mh) \end{pmatrix} x_j$$

so

$$x_j = \begin{pmatrix} (1-h)^j & 0 \\ 0 & (1-Mh)^j \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

In this case, the solutions remain bounded only if

$$|1 - Mh| \leq 1 \qquad \textit{i.e.} \qquad Mh \leq 2.$$

When $Mh > 2$, the solutions (at least the second component) blow up exponentially (unstable).

Remark 26.1. In the previous examples $x_2(t) = e^{-Mt}$ becomes small fast so a successful scheme should approximate the first component accurately while making the second small.

27. Lecture 27.

27.1. A stiff system coming from a parabolic problem.

We start this lecture with another example of a stiff system.

Example 27.1. *[Parabolic problem] We consider the parabolic partial differential boundary value problem: find $u : [0,1] \times [0,T] \to \mathbb{R}$ such that*

$$\begin{cases} u_t(x,t) - u_{xx}(x,t) = 0, \qquad x \in (0,1), \quad t \in (0,T], \\ u(0,t) = u(1,t) = 0, \qquad t \in 00,T], \\ u(x,0) = u_0(x), \qquad x \in [0,1] \end{cases}$$

where $u_0 : [0,1] \to \mathbb{R}$ is a given initial condition.

A finite difference approximation to this system can be constructed by introducing a grid $x_j = jh$, $h = 1/N$ and an approximation

$$w(x_i,t) \approx u(x_i,t), \qquad i = 0, ..., N, \quad t \in [0,T].$$

Using the finite difference approximation to $-u_{xx}$,

$$-u_{xx}(x_i,t) = \frac{2u(x_i,t) - u(x_{i-1},t) - u(x_{i+1},t)}{h^2} + O(h^2)$$

gives

$$\begin{aligned} 0 &= u_t(x_i,t) - u_{xx}(x_i,t) \\ &= u_t(x_i,t) + \frac{2u(x_i,t) - u(x_{i-1},t) - u(x_{i+1},t)}{h^2} + O(h^2). \end{aligned}$$

We replace $u(x_i,t)$ by $w(x_i,t)$ and drop the $O(h^2)$ term to arrive at the system:

$$w_t(x_i,t) + \frac{2w(x_i,t) - w(x_{i-1},t) - w(x_{i+1},t)}{h^2} = 0 \tag{44}$$

together with $w(x_i,0) = u_0(x_i)$ and $w(x_0,t) = w(x_N,t) = 0$. Now we introduce the vector $v(t)$ defined component wise by

$$v_j(t) = w(x_j,t), \qquad j = 1, ..., N-1$$

which therefore satisfies

$$v' = -\frac{1}{h^2}Av := \widetilde{A}v,$$

where

$$A = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & \mathit{0} & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & \mathit{0} & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}. \tag{45}$$

The eigenvectors ψ_j, $j = 1, ..., N-1$, of A are known and given by

$$(\psi_j)_k = \sin\left(\frac{\pi jk}{N}\right), \qquad k = 1, ..., N-1.$$

The associated eigenvalues are

$$\lambda_j = 2 - 2\cos\left(\frac{\pi j}{N}\right).$$

This means that

$$M^{-1}AM = D$$

where the jth column of M is ψ_j and D is the diagonal matrix with entries $D_{jj} = \lambda_j$. Multiplying the ODE system by M^{-1}, we find

$$M^{-1}v'(t) = -\frac{1}{h^2}M^{-1}AMM^{-1}v$$

or using the notation $\tilde{v}(t) = M^{-1}v(t)$

$$\tilde{v}' = -\frac{1}{h^2}D\tilde{v},$$

i.e.

$$\tilde{v}_j' = -\frac{1}{h^2}\lambda_j\tilde{v}_j.$$

The solution is

$$\tilde{v}_j = e^{-\frac{\lambda_j}{h^2}t}\tilde{v}_j(0).$$

Note that

$$\lambda_1 = 2 - 2\cos\left(\frac{i\pi}{N}\right) \approx 2 - 2\left(1 - \frac{\pi^2}{2N^2}\right) = \frac{\pi^2}{N^2} = \pi^2 h^2.$$

Hence $\frac{\lambda_1}{h^2} \approx \pi^2$. Similarly,

$$\lambda_{N-1} = 2 - 2\cos\left(\frac{(N-1)\pi}{N}\right) \approx 4$$

so

$$\frac{\lambda_{N-1}}{h^2} \approx \frac{4}{h^2} = 4N^2.$$

This means that the analysis of Examples 26.1 and 26.2 of the previous lecture can be applied with 1 *and* M *replaced by* $\widetilde{\lambda}_j$*, for* $j = 1, \ldots, N-1$. *The implicit scheme will not have any stability constraints while the explicit scheme will have a constraint of the form:*

$$k \leq 2/\widetilde{\lambda}_{N-1} \approx \frac{h^2}{2}.$$

This type of behavior characterizes a stiff system.

This example is easy to understand as the eigenvalues and eigenvectors are available. In contrast, more general parabolic problems behave in a similar fashion even though their eigenvalues and eigenvectors are much more difficult to compute.

27.2. **A definition of a stiff linear system.** Let $\widetilde{A}$ be and $m \times m$ matrix (with possibly complex entries) and let $\sigma(\widetilde{A})$ denote its spectrum, i.e., $\sigma(\widetilde{A})$ is the set of eigenvalues of $\widetilde{A}$. As the eigenvalues of $\widetilde{A}$ are roots of the characteristic polynomial of $\widetilde{A}$, they are, in general, complex (even when $\widetilde{A}$ is a real valued matrix).

We consider the solution $v : [0, \infty) \to \mathbb{C}^n$ solving the linear system of ODEs:

$$\begin{aligned} v'(t) &= \widetilde{A}v(t), \qquad \text{for } t > 0, \\ v(0) &= v_0. \end{aligned} \tag{46}$$

Below, we will provide a heuristic definition of when a linear system of ODE's is stiff.

Definition 27.1. *We say that solutions of* (46) *vanish at infinity if for every* $v_0 \in \mathbb{R}^m$,

$$\lim_{t\to\infty} v(t) = \mathbf{0}.$$

We then have the following theorem:

Theorem 27.1. *[Vanishing Solutions]. The solutions of* (46) *vanish at infinity if and only if*

$$\sigma(\widetilde{A}) \subset \mathbb{C}^- := \{z \in \mathbb{C} \ : \mathfrak{Re}(z) < 0\}.$$

The proof is not difficult and is based on the Jordan Canonical Form Theorem but will not be included for brevity.

Definition 27.2. *The spectral radius of* $\widetilde{A}$ *is defined by*

$$\rho(\widetilde{A}) = \max\{|\lambda| \ : \ \lambda \in \sigma(\widetilde{A})\}.$$

Definition 27.3. *The linear system of ODEs* (46) *is stiff if the solutions vanish at infinity and* $\rho(\widetilde{A}) >> 1$.

This definition is somewhat heuristic as the precise meaning of $\rho(\widetilde{A}) >> 1$ is not clear. Note, however, Example 27.1 is a classical example of a stiff system, especially when $h = 1/N$ is small.

27.3. **Numerical ODE schemes for stiff ODEs.** We always derived numerical schemes for the scalar equation and then extended them to systems. The eigenvectors of $\widetilde{A}$ play an important role in the resulting numerical ODE schemes for (46). Let (ψ, λ) be an eigenpair for $\widetilde{A}$. For any of the numerical schemes that we have considered, the solution x_j of the numerical method applied to $v_0 = \psi$ after j steps is given by $x_j = y_j\psi$ where y_j is the corresponding numerical solution of the scalar ODE (using the same numerical ODE method) approximating the scalar equation

$$y'(t) = \lambda y(t), \quad t > 0 \qquad \text{and } y(0) = 1. \tag{47}$$

Now, if the solutions of (46) vanish at infinity, a robust numerical method for this problem should have vanishing solutions, i.e.,

$$\lim_{j\to\infty} x_j = \mathbf{0}. \tag{48}$$

This and the above discussion motivates the following definition.

Definition 27.4 (A-stable)**.** *An ODE method is A-stable if when applied to the ODE* (47)*, the sequence of approximations y_j using a fixed timestep h satisfy*

$$\lim_{j\to\infty} y_j = 0,$$

for all h and all λ with $\mathfrak{Re}(\lambda) < 0$.

Remark 27.1. Suppose that solutions of (46) vanish at infinity. The argument above the definition implies that if the numerical ODE scheme is A-stable and $\widetilde{A}$ is diagonalizable, then the numerical sequence $\{x_k\}$ approximating the solution of (46) with any positive h satisfies (48) for any starting vector v_0. Although, I believe that this result probably holds for general $\widetilde{A}$ but have never written down a proof.

$A-$stable methods are good schemes for stiff ODEs.

Example 27.2 (Backward Euler)**.** *The Backward Euler scheme applied to $u' = \lambda u$, with $u(0) = v$ given, reads*

$$\frac{u_j - u_{j-1}}{h} = \lambda u_j, \qquad u_0 = v$$

or

$$u_j = u_{j-1} + h\lambda u_j$$

i.e.

$$u_j = (1 - h\lambda)^{-1} u_{j-1} = (1 - h\lambda)^{-j} u_0.$$

Hence,

$$|u_j| = |1 - h\lambda|^{-j} |u_0|$$

but

$$|1-h\lambda|^2 = |1-h\mathfrak{Re}(\lambda)-ih\mathfrak{Im}(\lambda)|^2 = (1-h\mathfrak{Re}(\lambda))^2+h^2\mathfrak{Im}(\lambda)^2 \geq (1-h\mathfrak{Re}(\lambda))^2$$

and if $\mathfrak{Re}(\lambda) < 0$, $1 - h\mathfrak{Re}(\lambda) > 1$. *This implies that*

$$(1 - h\mathfrak{Re}(\lambda))^2 > 1$$

or

$$\frac{1}{|1 - h\lambda|^j} < 1.$$

In turn, we obtain

$$|u_j| \leq |1 - h\lambda|^{-j} |u_0| \to 0, \text{ as } j \to \infty$$

which shows that Backward Euler is A-stable.

The previous example also indicates that solutions of the Backward Euler scheme tend to zero when

$$\frac{1}{|1 - h\lambda|} < 1$$

or

$$|1 - h\lambda| > 1.$$

Note that $1 - \lambda h = 1 - \eta$ (define $\eta = \lambda h$) is a continuous function of η. Moreover,

$$|1 - \eta| = 1 \qquad \Longrightarrow \qquad 1 - \eta = e^{i\theta} \qquad \text{for some} \quad \theta \in [0, 2\pi],$$

i.e.

$$\eta = 1 - e^{i\theta}.$$

Definition 27.5 (Absolute Stability). *The region of absolute stability of a scheme is the set of* $\eta = h\lambda$ *such that the approximate solutions tend to zero.*

Example 27.3 (Backward Euler). *The region of absolute stability for Backward Euler is*

$$\mathbb{C} \setminus B_1(1) = \{z \in \mathbb{C} \ : \ |1 - z| > 1\}$$

and is depicted in Figure 11.

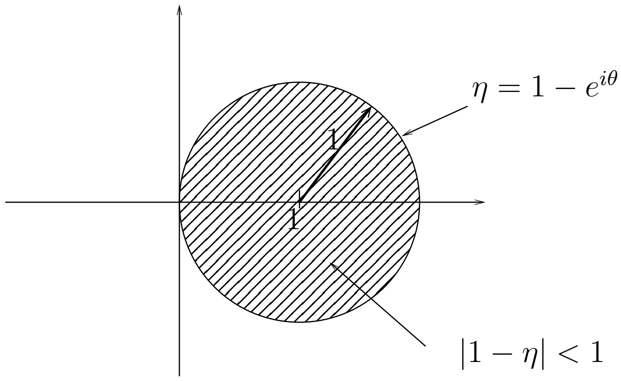

FIGURE 11. Absolute convergence region for Backward Euler (exterior of the closed shaded area).

Example 27.4 (Forward Euler)**.** *The Forward Euler scheme applied to $u' = \lambda u$ with $u(0) = u_0$ is*

$$u_j = (1 + h\lambda)u_{j-1}$$

or

$$u_j = (1 + h\lambda)^j u_0.$$

The solutions tend to zero if and only if

$$|1 + h\lambda| < 1.$$

Note that $|1 + h\lambda| = 1$ if and only if

$$1 + h\lambda = e^{i\theta}, \qquad \theta \in [0, 2\pi].$$

If $\eta := h\lambda$ then $1 + \eta = e^{i\theta}$ or $\eta = (e^{i\theta} - 1)$, see Figure 12. Forward Euler is not $A-$stable.

Remark 27.2 (Absolute and A-stability)*.* A scheme is $A-$stable if and only if the region of absolute stability contains

$$\{z \in \mathbb{C} \ : \ \mathfrak{Re}(z) < 0\}.$$

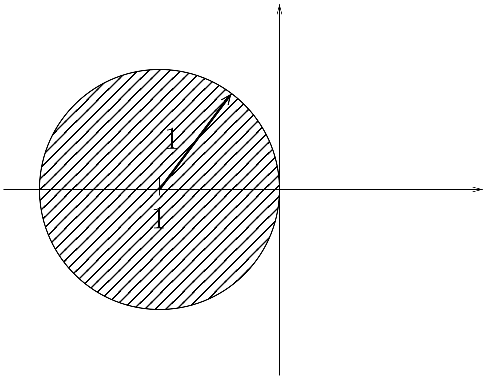

Figure 12. Absolute convergence region for Forward Euler (interior of shaded).