SFU
muraki

WEDNESDAY

- WEEK 5 TUTORIALS: today & THURS.
- QUIZ on FRIDAY: please be on-time for lecture

LAST DAY

- order of convergence for BiS, NM & SM.

	ACC / EFF	RoB	
BiS	Lin Conv 1 eval/iter	preserve bracket.	
NM	Quad Conv 2 eval/iter	need $\lvert \lambda_e E_n \rvert < 1$	need f'
SM	1.62 Conv 1 eval/iter	need $\lvert \lambda_e E_n \rvert < 1$	

$$\lambda_e = \frac{f''(x_e)}{2f'(x_e)}$$

some roots "easier" to find than others

- matlab's fzero
  - hybrid command
  - set/check TolX option

fzero

R2018b

Root of nonlinear function

collapse all in page

Syntax

```
x = fzero(fun,x0)
x = fzero(fun,x0,options)

x = fzero(problem)

[x,fval,exitflag,output] = fzero(__)
```

Description

`x = fzero(fun,x0)` tries to find a point x where `fun(x) = 0`. This solution is where `fun(x)` changes sign—`fzero` cannot find a root of a function such as x^2. example

`x = fzero(fun,x0,options)` uses `options` to modify the solution process. example

`x = fzero(problem)` solves a root-finding problem specified by `problem`. example

`[x,fval,exitflag,output] = fzero(__)` returns `fun(x)` in the `fval` output, `exitflag` encoding the reason `fzero` stopped, and an `output` structure containing information on the solution process. example

exitflag — Integer encoding the exit condition
integer

Integer encoding the exit condition, meaning the reason `fzero` stopped its iterations.

1	Function converged to a solution x.
-1	Algorithm was terminated by the output function or plot function.
-3	NaN or Inf function value was encountered while searching for an interval containing a sign change.
-4	Complex function value was encountered while searching for an interval containing a sign change.
-5	Algorithm might have converged to a singular point.
-6	fzero did not detect a sign change.

OTHER ROOT-FINDING METHODS

- fixed-point (§2.2)
- inverse quadratic interpolation (fzero)
- false position (§2.3)
- polynomials (§2.6) + Matlab's roots

```
>> roots([1 2 6])

ans =

 -1.0000 + 2.2361i
 -1.0000 - 2.2361i
```

$x^2 + x + \quad = 0$

complex-valued roots

$i = \sqrt{-1}$

SFU
muraki

- polynomial roots via linear algebra

The companion matrix

Consider the matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0/a_n \\ 1 & 0 & \cdots & 0 & -a_1/a_n \\ 0 & 1 & \cdots & 0 & -a_2/a_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1}/a_n \end{bmatrix}$$

Claim: the eigenvalues of this matrix are precisely the roots of the polynomial $p(x)$. $= a_n x^n + \ldots a_1 x + a_0$

- DOUBLE & MULTIPLE ROOTS
  - double roots are problematic because there is sign change
  - if you there is a root $f(x_0) = 0$

    then one can try to do root-finding on

    $g(x) \equiv$ → NEWTON

THE NEXT CHAPTER ...

- our original list of numerical issues were:

  finite-precision arithmetic ✓

  finite # of operations ✓

  finite of continuous functions *

SFU
muraki

APPROXIMATION + INTERPOLATION of FUNCTIONS. (CH 3)

- chap 2 was about finding zeros of given ( ) FUNCTIONS
- in many circumstances, the function is , or , or to calculate

BUT we do have VALUES

a of points (,)

- to apply our usual ideas of (root-finding ...)
  or (differentiating, integrating ...)

→ we produce a function
using the given points

- chap 3: POLYNOMIAL - there exists a
  with degree < through + points
- INTERPOLATION: a continuous function
  points + defined over interval
  data $\{x_k\} \leq x \leq \{x_k\}$
  (if outside data, then POLATION)

- not to be confused with " " (not through data)

WHY POLYNOMIALS?

(Weierstrass Approximation Theorem)

Suppose that f is defined and continuous on $[a, b]$. For each $\epsilon > 0$, there exists a polynomial $P(x)$, with the property that → can find.

given any

$$|f(x) - P(x)| < \epsilon, \quad \text{for all } x \text{ in } [a, b].$$

The proof of this theorem can be found in most elementary texts on real analysis (see, for example, [Bart], pp. 165–172).

... BUT not

- polynomials are simple for +
  - accuracy, efficiency + robustness ?

POLYNOMIAL BASIS FUNCTIONS

- used to define polynomials as

$$P(x) = \sum_{0}^{N}$$

MONOMIAL or BASIS SET

$$\{ \ , \ , \ \ldots \ \}$$

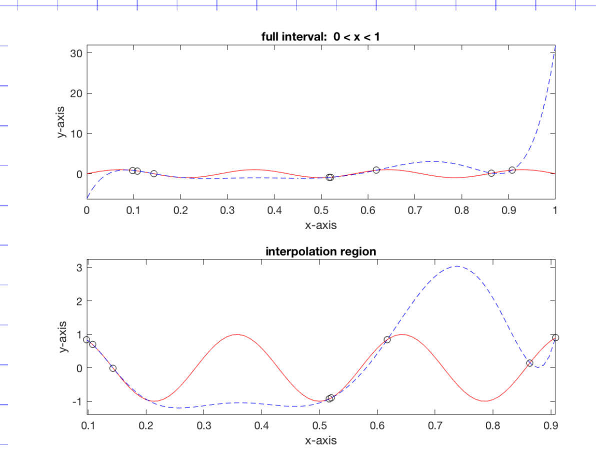
- N+1 values of can be determined from N+1 point values $\{ (\ = (\ )), k = 0 \to N \}$

- gives VANDERMONDE

$$
\begin{array}{c}
k=0 \\ \vdots \\ \vdots \\ k=N
\end{array}
\begin{bmatrix}
x_0 & \cdots & x_0 & \cdots & x_0 & 1 \\
x_1^N & & \vdots & & \vdots & 1 \\
\vdots & & \vdots & & \vdots & \vdots \\
x_k^N & & x_k^j & & \vdots & \vdots \\
\vdots & & \vdots & & \vdots & \vdots \\
x_N^N & \cdots & x_N^j & \cdots & x_N & 1
\end{bmatrix}
\begin{pmatrix} \vdots \\ a_j \\ \vdots \end{pmatrix}
=
\begin{pmatrix} \vdots \\ y_k \\ \vdots \end{pmatrix}
$$

$$
\begin{array}{ccc}
j=N & \longleftarrow & j=0
\end{array}
$$

MONOMIAL DEMO



· your observations on DEMO:

• simple curves + →

• oscillating curve + →

linear solve!

SFU
muraki

AN EXACT SOLUTION: LAGRANGE BASIS (§3.1)

- from $N+1$ points $\{(x_k, y_k) \mid k = \quad \to \quad \}$
- construct $L_k(x)$ where

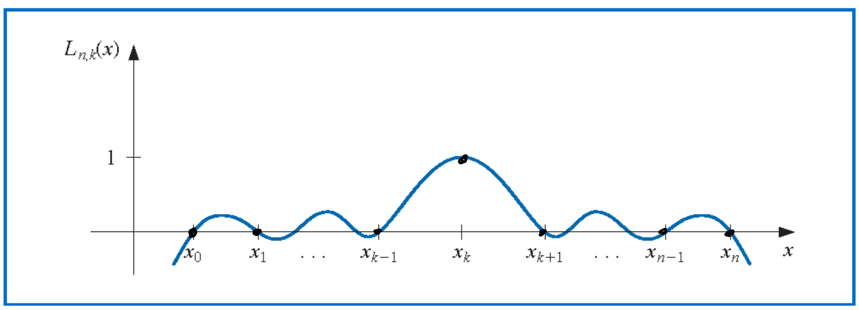
$$L_k(x_j) = \begin{cases} & x_j \neq x_k \\ & x_j = x_k \end{cases}$$

To satisfy $L_{n,k}(x_k) = 1$, the denominator of $L_{n,k}(x)$ must be this same term but evaluated at $x = x_k$. Thus

$$L_{n,k}(x) = \frac{(x - x_0)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}.$$

A sketch of the graph of a typical $L_{n,k}$ (when n is even) is shown in Figure 3.5.

$$= \prod_{\neq} \frac{(\quad)}{(\quad)}$$



- LAGRANGE INTERPOLATING POLYNOMIAL

$$P(x) = \sum_{k=0}^{N} \quad \cdot L(\quad)$$

sum over all

-deg poly

- only non-zero term for $x = x_0 \ldots x_N$

$$\Rightarrow \quad P(x_k) = \quad \square$$

- polynomial from monomials $\to$ no needed.
- addresses -conditioning BUT can still yield poor of original function
- costly in terms of operation count.

$$\text{op count for } L_k(x) = \text{op count for } \left\{ \sum_0^N y_k \underbrace{L_k(x)} \right\} = O(\quad)$$

adds   mult. +

+   −   fn

+   −   fn

- BUT idea for -dependent basis is game-changing
- framework for understanding of errors.

muraki