CAAM 453/553 Review

Covers Lectures 21-36
Eigenvalues – if I ask you some question about eigenvalues, I will give you enough computing materials to expand upon the questions.

2:29
Minimax Approximations / L-infinity approximation
Given $f \in C[a,b]$ find $p \in P_n$ that minimizes $\max_{x\in[a,b]} |f(x) - p(x)|$

How to find this approximation?
-Theorem of de la Valleé Poussin
-Equioscillation Theorem (KNOW THIS THEOREM)
- $p \in P_n$ is the best L^∞ approximation to f over [a,b] if and only if there exists n+2 points $a \le x_0 \le \ldots \le x_{n+1} \le b$ such that $p(x_j) - f(x_j) = -\left[p(x_{j+1}) - f(x_{j+1})\right]$ and $\left|p(x_j) - f(x_j)\right| = \|f - p\|_{L^\infty}$

9:16
Chebyshev Polynomials
How do we motivate Chebyshev polynomials? We looked at the error term $\left|\prod_{j=0}^{n} x - x_j\right| = \left|x^{n+1} - p(x)\right|, p \in P_n$, so we approximate the polynomial of order n+1 with polynomial order n.

So we define the Chebyshev Polynomial T_k as:
$T_k(x) = \cos\left(k\cos^{-1}(x)\right), x \in [-1,1]$
Which induces the recurrence
$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$

We also discovered that we could use cosh instead of cos

Zeros, extrema of Tk….

So in general,
$T_k(x) = 2^{k-1}x^k + \ldots$
$\hat{T}_k(x) = \frac{T_k(x)}{2^{k-1}} = x^k - p(x), p \in P_{k-1}$
And that's how we derive the best approximation to the polynomial n+1, we look at the polynomial T_{n+1}. The Chebyshev polynomial, in [-1,1] is the best approximation, after scaling and shifting.

Let's spell that out.
$T_{n+1}(x) = x^{n+1} - p_n(x)$ This function has n+2 extrema on [-1,1] of oscillating sign and equal magnitude

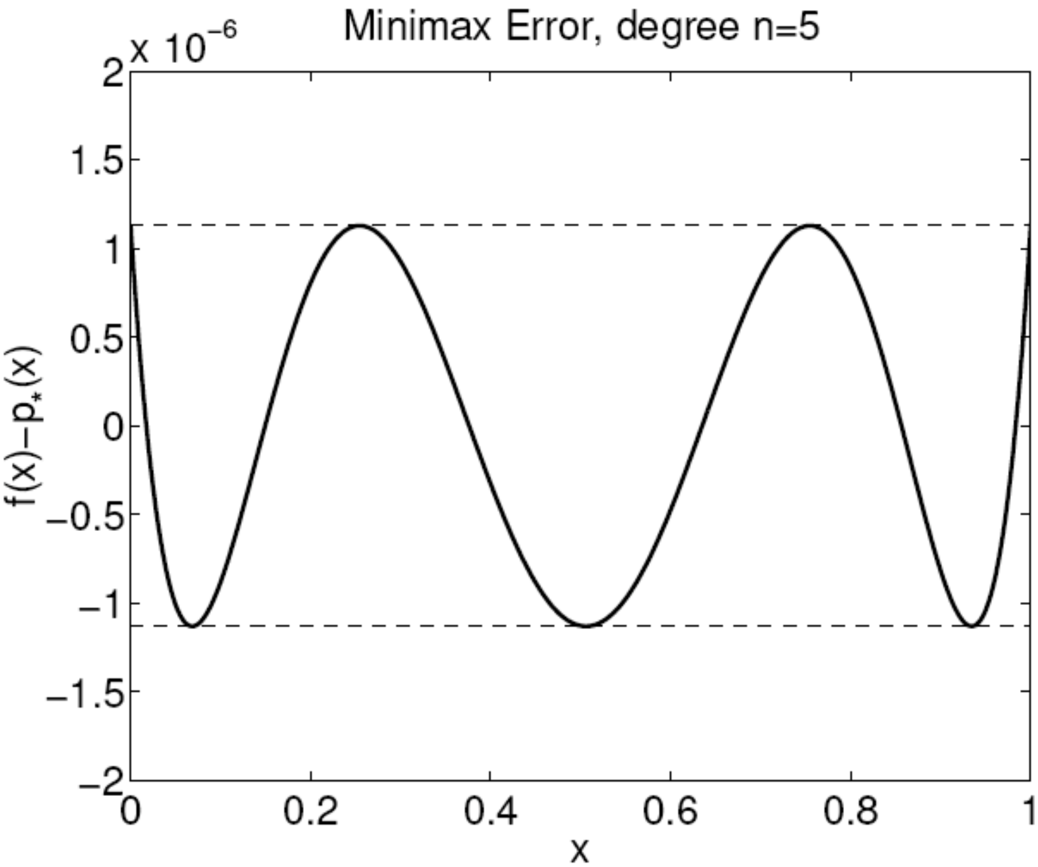

Figure 1:Something very similar to what he through on the board - Lecture 21

+ equioscillation theorem YIELDS P_n is the best approximation from P_n to x^{n+1}

20:18

Quadrature: Approximate $\int_a^b f(x)dx, f \in C[a,b]$

-Interpolatory Quadrature Rules

-Construct an interpolant to f

-Exactly integrate the interpolant

-Here's a generic interpolatory formula. Given interpolation points $x_0,...,x_n$, the interpolant to f is given by:

$$p(x) = \sum_{j=0}^{n} f(x_j) l_j(x), \;\; l_j(x) = \prod_{\substack{k=0 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}$$

The quadrature rule that follows from that:

$$\int_a^b f(x)dx \approx \int_a^b p(x)dx = \int_a^b \sum_{j=0}^{n} f(x_j) l_j(x_j) dx$$

$$= \sum_{j=0}^{n} f(x_j) \int_a^b l_j(x) dx$$

$$= \sum_{j=0}^{n} \underbrace{w_j}_{\text{Quad weights}} \underbrace{f(x_j)}_{\text{Quad nodes}}$$

From this we can see that the quadrature rule is exact for all $f \in P_n$

Error Bound: Integrate the error bound for polynomial interpolation:

$$\int_a^b f(x)dx - \int_a^b p(x)dx = \int_a^b f(x) - p(x)\, dx$$

$$= \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{j=0}^{n} (x - x_j) dx$$

32:30

Newton Cotes Quadrature

-Trapezoid Rule,Simpson's Rule

-Uniformly spaced quadrature nodes

-Trapezoid: n=1 [Integrates P_1 exactly]

-Simpsons: n=2 [Bonus degree of accuracy! Integrates P_3 cubics exactly!]

-Good to have an idea of the error

-Why don't we like using these? Because uniformly spaced points are terrible for high degree interpolants.

-There's a couple ways around this, the first being the composite rules – breaking them down to subintervals.

-Chebyshev points – aka Clenshaw-Curtis Quadrature: Nodes = Chebyshev points. Take n to be larger.

-Smooth functions die to Clenshaw – you'll get 10, 15 digits of accuracy. As your function loses regularity to analytic, so you have an ellipse, and you get fewer and fewer gains. I.e. functions that are not infinitely smooth – an example being a discontinuity in the third derivative, something like that.

40:00

Gaussian Quadrature

Interpolatory Quadrature scheme but the nodes are picked such that the rule exactly integrates polynomials of degree $2n+1$ (times a weight function).

$$\int_a^b f(x)w(x)dx \approx \sum_{j=0}^{n} w_j f(x_j),\ w_j = \int_a^b l_j(x)w(x)dx$$

Nodes = roots of orthogonal polynomial (with respect to the inner product) of f and g. $\langle f,g\rangle = \int_a^b f(x)g(x)w(x)dx, \quad w(x)\ge 0$

Possible nastiness …. $\frac{1}{\sqrt{1-x}}$

Richardson Extrapolation (Romberg Intergration)

Some numerical process converges like O(h)

Combine results for h, h/2 to get $O(h^2)$

h,h/2,h/4 to get $O(h^3)$ approximation

How do you combine the order h, h/2 to knock out the terms

54:37

Numerical Solution of differential equations

$x'(t) = f(t,x(t)),\ x(t_0) = x_0$ Initial Value Problem

Any results we have are based on how well-behaved the ODE is.

One Step Methods

General forward: $x_{k+1} = x_k + h\Phi(t_k, x_k, h)$

Backward Euler $x_{k+1} = x_k + hf(t_k, x_k)$

Heun's $x_{k+1} = x_k + \frac{h}{2}\left(f(t_k,x_k) + f(t_k + h, x_k + hf(t_k,x_k))\right)$

Truncation Error: $T_k = \frac{x(t_{k+1}) - x(t_k)}{h} - \Phi\left(t_k, \underbrace{x(t_k)}_{\text{exact sol}}, h\right)$

some rearrangement is necessary

Ex; Forward Euler $T_k \sim O(h)$

A method is consistent if $T_k \to 0$ as $h \to 0$

Global error: $\left|X(t_k) - X_k\right| \le \frac{T}{L_\Phi}\left(e^{nhL_\Phi} - 1\right)$ where T is the maximum truncation error, L_Φ is the lipshitz constant for the method.

$t_n = t_0 + hn$, so n is fixed! This is great, because before we knew that, we were utterly screwed with exponentially growing error. As its fixed, the main factor is then T, and we know that the Global Error can be simplified as:

$\left|X(t_k) - X_k\right| \le \frac{T}{L_\Phi}\left(e^{nhL_\Phi} - 1\right) = O(T)$ as $n \to \infty$

So Consistency IMPLIES Global Convergence for 1 step methods

1:06:30
Linear Multistep Methods (LMM)

m-step method: $\sum_{j=0}^{m}\alpha_j x_{k+j} = h\sum_{j=0}^{m}\beta_j f_{k+j}$, $f_k = f\left(t_k, x_k\right)$. Explicit if $\beta_m = 0$ otherwise implicit.

What's the point of a LMM? They only need one new f evaluation per step. Therefore, in cases of difficult evaluations, this method could save much computational time.

Truncation error: $T_k = \dfrac{\sum_{j=0}^{m} a_j x\left(t_{k+1}\right) - h\sum_{j=0}^{m}\beta_j f\left(t_{k+1}, x\left(t_{k+1}\right)\right)}{h\sum_{j=0}^{m}\beta_j}$

Formulas for checking truncation error (highly probable something like this will be on exam):

$O(h)$ Consistency: $\sum a_j = 0, \sum j\alpha_j = \sum \beta_i$

$O\left(h^2\right)$ also: $\sum \frac{j^2}{2!}\alpha_j = \sum j\beta_j$

$O\left(h^3\right)$ also: $\sum \frac{j^3}{3!}\alpha_j = \sum \frac{j^2}{2!}\beta_j$

Zero Stability:

If we think about taking h to 0, getting really really small stepsizes, the idea is apply LMM to x'(t) =0.

The question here: Do solutions with inexact initial data grow or remain bounded. Dalhquist proved you only need to check with this simple problem.

To check

1) Form characteristic polynomial $\rho(z) = \sum_{j=0}^{m}\alpha_j z^j$
2) Check to see if it satisfies the root condition: all roots of p(z) =0, fall in / on the unit circle, and any roots on the unit circle are simple.

Consistency($T_k \to 0$) + Consistent Starting Data + Zero Stability => Global convergence

So these are the requirements, which can be very demanding.

Dalhquist barrier – with an m-step method, the best error you can get is O(h^{m+1}) if m is odd, $O\left(h^m\right)$ if even for a zero stable method.

1:23:30

Stiff Differential Equations

$x'(t) = \lambda x(t) \rightarrow x(t) = e^{\lambda t} x(0)$

$\Rightarrow |x(t)| \rightarrow 0 \quad \forall x(0)$

if an only if $\text{Re}(\lambda) < 0$

When does LMM mimic this behavior, i.e., $|x_k| \rightarrow 0$?

$$\sum_{j=0}^{m} \alpha_j x_{k+j} = h\lambda \sum_{j=0}^{m} \beta_j x_{k+j} \Rightarrow \sum_{j=0}^{m} \left(\alpha_j - h\lambda\beta_j\right) x_{k+j} = 0$$

Stability Polynomial:

$$\rho(z) - h\lambda\sigma(z) = \sum_{j=0}^{m} \left(\alpha_j - h\lambda\beta_j\right) z^j$$

$|x_k| \rightarrow 0$ if all roots of the stability polynomial are inside the unit circle.

(Absolute) Stability Region:

The set of all $h\lambda \in \mathbb{C}$ such that $|x_k| \rightarrow 0$

Look at the plots of stability regions in the notes. The basic rule is that implicit methods have much larger regions than explicit methods. And the higher the order the method, the smaller its stability will be.

Systems:

$x' = Ax, \ \ x(0) = x_0$

If A is diagonalizable (All matrices you'll encounter in this class and final will be diagonalizable) then $A = V\Lambda V^{-1}$

$x' = V\Lambda V^{-1} x = \left(V^{-1}x\right) = \Lambda\left(V^{-1}x\right)$

$\Rightarrow y' = \Lambda y \Rightarrow y_j' = \lambda_j y_j$

$x_k \rightarrow 0$ if $h\lambda_j \in$ stability region - for all λ_j $\left(\text{eigenvalues of A}\right)$ [Lecture 33]

Certainly on the exam, be comfortable working with eigenvalues and eigenvectors. For the boundary value problems, the setting would be given. There's one example I really like – Exam 2007, ode example. Showing solutions xk.

1:42:00

Revisiting Ax = b via LU Factorization

$$A = \begin{bmatrix} \alpha & u^* \\ v & C \end{bmatrix} \underbrace{=}_{\text{1step of this}} \begin{bmatrix} 1 & 0 \\ \frac{1}{\alpha}v & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \underbrace{C - \frac{1}{\alpha}vu^*}_{\text{active submatrix}} \end{bmatrix} \begin{bmatrix} \alpha & u^* \\ 0 & I \end{bmatrix}$$

$\begin{bmatrix} 1 \\ \frac{1}{\alpha}v \end{bmatrix}$ – part of L

C-$\frac{1}{\alpha}vu^*$ – active submatrix

$\begin{bmatrix} \alpha & u^* \end{bmatrix}$ – part of U

L = unit lower triangular

U = upper triangular

This takes $\sim \frac{2}{3}n^3$ flops . It is fair game that I ask you some order operation counting.

Partial Pivoting (PA = LU): Row swapping so that $|l_{jk}| \le 1 \forall j,k$

Growth factor: $\dfrac{\max_{j,k}|u_{j,k}|}{\max_{j,k}|a_{j,k}|}$

Partial pivoting is stable if the growth factor is modest

Complete pivoting (PAQ = LU): row and column swaps => always stable, but more expensive.

1:52:00

Cholesky Factorization

If $A = A^*$

$$A = \begin{bmatrix} \alpha & v^* \\ v & C \end{bmatrix} = \begin{bmatrix} \sqrt{\alpha} & 0 \\ \frac{1}{\sqrt{\alpha}}v & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & C - \frac{1}{\alpha}vv^* \end{bmatrix} \begin{bmatrix} \sqrt{\alpha} & v^* \\ 0 & I \end{bmatrix}$$

$A = LL^*$, L=lower triangular

$\frac{1}{3}n^3$ flops

works if A is positive definite, $x^*Ax > 0, \forall x \ne 0$