Lecture 2: Computing e^{At}

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6435
Adv Numerical Methods in Sci Comp

Fall 2008
September 3, 2008

Outline of Lecture 2: Computing e^{At}

- What is the need for computing e^{At}?
- Evaluation of matrix polynomials (**Horner's rule**)
- Truncation errors
- Chebyshev approximation
  - Properties
  - **Clenshaw recursion**
  - Concepts of **shifting**, **scaling**, and **doubling**
- Pade approximation
- Upper Schur transformation-based approach
- Special case: *A* is a stochastic matrix (*a la* Markov chains)
- How **not** to compute e^{At}?

References:

1. R.C. Ward, "Numerical computation of matrix exponential with accuracy estimation", SIAM J. on Numerical Analysis, Vol. 14, 600-614, 1977.
2. C.B. Moler and C.F. Van Loan., "Nineteen dubious way to compute the exponential of a matrix", SIAM Review, 801-836, Oct. 1978.
3. T.H. Kerr, "Use of idempotent matrices to validate linear system software", IEEE Trans. on Aerospace and Electronic Systems, Vol. 26, No. 6, 935-953, Nov. 1990.

What is the need for Computing e^{At} ?- 1

- What is the need for computing e^{At}?
  - e^{At} is a transcendental function that arises in a variety of applications. A representative set of examples are as follows:

I. . $\dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(t) \Rightarrow \underline{x}(t) = e^{At}\underline{x}_0 + \int_0^t e^{A(t-\sigma)}Bu(\sigma)d\sigma$

II. Discrete-time version: $u(t)$ piecewise constant over $t \in [k\delta, (k+1)\delta)$

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Gamma \underline{u}_k, \qquad \Phi = e^{A\delta}, \qquad \Gamma = \int_0^\delta e^{A\sigma}Bd\sigma$$

III. $\dot{\underline{x}} = A\underline{x}(t) + B\underline{u}(t) + E\underline{w}(t);\ \underline{x}(0) = N(\underline{x}_0, X_0)$

$\underline{w}(t)$ = zero-mean white Gaussian noise with covariance

$E\{\underline{w}(t)\underline{w}^T(\sigma)\} = W\,\delta\,(t-\sigma)$.

Define $X(t) = E\{\underline{x}(t)\underline{x}^T(t)\}$

$$\dot{X}(t) = AX(t) + X(t)A^T + EWE^T$$

$$X(t) = \int_0^t e^{A\sigma}EWE^Te^{A^T\sigma}d\sigma + e^{At}X_0e^{A^Tt}$$

IV. There exist many other control situations where e^{At} and various $\int$ arise. See the paper by Dorato and Levis, Dec. 1971 *IEEE T-AC*.

What is the need for Computing e^{At} ?- 2

V. Continuous–time Markov Chains arise in a wide variety of applications:

- Reliability/availability modeling
- Performability modeling
- Computer performance
- Manufacturing Systems
- Computer-communication network modeling.

The key equation to be solved is:

$$\dot{\underline{p}} = Q^T \underline{p}; \qquad \underline{p}(0) = \underline{p}_0 \Rightarrow \underline{p}(t) = e^{Q^T t} \underline{p}_0$$

- $Q=[q_{ij}]$ transition rate matrix, where:

  q_{ij} = rate at which the chain jumps from state i to state j
- Q is a stochastic matrix ⇒ each row of Q sums to **zero**.
- If there is a reward rate f_i associated with each state i, then the *expected cumulative reward* of the chain over an interval [0, T], termed average performability, is

$$\overline{y}_{[0 \quad T]} = \underline{f}^T \left[\int_0^T e^{Q^T t} dt \right] \underline{p}_0$$

Eigen Method for Computing e^{At}

- Most fundamental question:
  - How to compute the functions e^{At} and various $\int$ for a given A and t
- In this lecture, we focus on the problem of computing e^{At}
- Computing e^{At} is a subset of a broader problem: Compute $f(A)$ e.g., $\sin A$, $\ln A$, e^{A}, $\cos A$, etc.
  - Since $A = P\Lambda P^{-1}$, $\quad A^k = P\Lambda^k P^{-1}$

$$f(A) = Pf(\Lambda)P^{-1} = PDiag[f(\lambda_i)]P^{-1} = \sum_{i=1}^{n} f(\lambda_i)\underline{\xi}_i\underline{\eta}_i^T$$

$$P = \begin{pmatrix} \underline{\xi}_1 & \underline{\xi}_2 & \cdots & \underline{\xi}_n \end{pmatrix} \qquad P^{-1} = \begin{bmatrix} \underline{\eta}_1^T \\ \underline{\eta}_2^T \\ \underline{\eta}_n^T \end{bmatrix}$$

- We will later see that this is one of the *worst* methods for computing $f(A)$.

Example: $\quad A = \begin{bmatrix} 1+10^{-5} & 1 \\ 0 & 1-10^{-5} \end{bmatrix}; \qquad P = \begin{bmatrix} 1 & -1 \\ 0 & 2\text{x}10^{-5} \end{bmatrix}$

Computing e^{At} and Ill-conditioning

$$Pf(\Lambda)P^{-1} = \begin{bmatrix} 2.718307 & 2.75000 \\ 0.00000 & 2.718524 \end{bmatrix}; \text{ actual} = \begin{bmatrix} 2.718309 & 2.718282 \\ 0.00000 & 2.718255 \end{bmatrix}$$

- We will see in Lecture 10 that condition number of the given Eigen value problem ∝ $1/|\lambda_i - \lambda_j|$

  $\lambda_i \approx \lambda_j \Rightarrow$ trouble $\Rightarrow$ since condition number $\approx 10^5$

- Another way
  - Suppose have an approximation to $f(a)$, e.g., a polynomial series

    $$e^a \approx 1 + a + \frac{a^2}{2!} + \cdots + \frac{a^N}{N!} = \sum_{i=0}^{N} \frac{a^i}{i!} = \hat{f}(a)$$

  - Then how much error do we make by approximating $f(a)$ by $\hat{f}(a)$?

- Theorem: Given a scalar function $f(a)$. Let $\hat{f}(a)$ be an approximation to $f(a)$ for $-c \le a \le c$ (or $|a| \le c$). Then $\hat{f}(A)$ is an approximation of $f(A)$ valid for spectral radius $\rho(A) \le c$ with truncation error:

  $$\left\| f(A) - \hat{f}(A) \right\| \le \|P\| \left\| P^{-1} \right\| \max_i |[f(\lambda_i) - \hat{f}(\lambda_i)]|$$

  where P is the similarity transformation that diagonalizes A.

Error Analysis

Proof:
$$f(A) = P\,Diag\left[f(\lambda_i)\right]P^{-i}$$
$$\hat{f}(A) = P\,Diag\left[\hat{f}(\lambda_i)\right]P^{-i} \qquad \text{valid as long as } |\lambda_i| \le c$$
$$\Rightarrow f(A) - \hat{f}(A) = P\,Diag\left[f(\lambda_i) - \hat{f}(\lambda_i)\right]P^{-1}$$
$$\left\| f(A) - \hat{f}(A) \right\| \le \|P\| \left\|P^{-1}\right\| \max_i [|f(\lambda_i) - \hat{f}(\lambda_i)|]$$

- So, the problem of computing $f(A)$ is reduced to finding a suitable, simple $f(a)$, a is a scalar
- This is a scalar numerical function approximation
- Note that most $\hat{f}$ are often polynomials or ratio of polynomials (of some type).
- Example: Maclaurin's series for $f(a)$

$$f(a) = f(0) + f'(0)a + f''(0)\frac{a^2}{2!} + \cdots = \sum_{k=0}^{\infty} f^{(k)}(0)\frac{a^k}{k!}$$

- ❑ Key sub-problem: how to evaluate (truncated) polynomials efficiently.
  - Evaluation of scalar and matrix polynomials:

consider:
$$\hat{f}(a) = p_N(a) = b_0 + b_1 a + b_2 a^2 + \ldots + b_N a^N$$
$$\hat{f}(A) = p_N(A) = b_0 + b_1 A + b_2 A^2 + \ldots + b_N A^N$$

Horner's Rule

- A method called Horner's rule or *reverse nesting* gets around round-off error problems

- Horner's Rule
  - Consider $b_0 I + b_1 A + b_2 A^2 = b_0 I + A[b_2 A + b_1 I]$
  - Initialize $p_0 = b_2 I,$
  - $p_1 = b_2 A + b_1 I,$ n^2 Multiplications and n additions
  - $p_2 = A p_1 + b_0 I,$ n^3 Multiplications and $n^3 + n$ additions
  - In general

$b_0 I + b_1 A + b_2 A^2 + \ldots + b_N A^N$ can be computed in $(N-1)$ matrix multiplications via:

$$A(A(A(A(b_N A + b_{N-1} I) + b_{N-2} I) + \ldots + b_1 I) + b_0 I)$$

  - Algorithm

initialize: $p_1 = b_N A + b_{N-1} I$

recursion: For $i = 2, \ldots, N$

$$p_i = A p_{i-1} + b_{N-i} I$$

end Do

Smarter Horner's Rule - 1

- We can do better than (N-1) matrix multiplies
  - Suppose have

$$p(A) = A^3[A^3\{b_9A^3 + (b_8A^2 + b_7A + b_6I)\} + (b_5A^2 + b_4A + b_3I)] + b_2A^2 + b_1A + b_0I$$

$$A_2 = A^2; A_3 = AA_2$$

$$S = b_9A_3 + b_8A_2 + b_7A + b_6I$$

$$S = A_3S + b_5A_2 + b_4A + b_3I$$

$$S = A_3S + b_2A_2 + b_1A + b_0I$$

$$\Rightarrow O(4n^3) \text{ vs } O(8n^3)$$

- In general, if s is any integer satisfying $1 \leq s \leq N^{1/2}$, then

$$p(A) = \sum_{k=0}^{r} B_k (A^s)^k \qquad r = floor(N/s)$$

where

$$B_k = b_{sk+s-1}A^{s-1} + \ldots + b_{sk+1}A + b_{sk}I; \qquad k = 0,1,2,\ldots,r-1$$

$$B_k = b_N A^{N-sr} + \ldots + b_{sr+1}A + b_{sr}I; \qquad k = r$$

- Compute A^2, A^3, …, A^s and apply Horner' Rule to new polynomial.
- Operation count: $\approx (s + r - 2)n^3 \pm n^3$ if $s = floor(sqrt(N))$
  $\Rightarrow$ minimal computation of $N^{1/2}n^3$.

Smarter Horner's Rule - 2

- Previous example: $s = 3$ and $r = 3$

$$B_0 = b_2 A^2 + b_1 A + b_0 I$$
$$B_1 = b_5 A^2 + b_4 A + b_3 I$$
$$B_2 = b_8 A^2 + b_7 A + b_6 I$$
$$B_3 = b_9 I$$

Example 2: 13^{th} order polynomial $s \le (13)^{1/2}$

- Pick $S = 3$ and $r = floor(13/3) = 4$
- $B_0 = b_2 A^2 + b_1 A + b_0 I$
- $B_1 = b_5 A^2 + b_4 A + b_3 I$
- $B_2 = b_8 A^2 + b_7 A + b_6 I$
- $B_3 = b_{11} A^2 + b_{10} A + b_9 I$
- $B_4 = b_{13} A + b_{12} I$
- Evaluate $A^3[A^3\{A^3(A^3[b_{13}A + b_{12}I] + b_{11}A^2 + b_{10}A + b_9 I) + b_8 A^2 + b_7 A + b_6 I\} + b_5 A^2 + b_4 A + b_3 I] + b_2 A^2 + b_1 A + b_0 I$

- Q: How do we use these concepts of function approximation and evaluation of polynomials?

Computation of $f(A)$

Computation of $f(A)$

- If approximate $f(a)$ by $\hat{f}(a)$, then $\hat{f}(A)$ approximates $f(A)$
- There are basically three approaches:
  - MacLaurin's (Taylor's) series $|a| \le 1$
  - Chebyshev polynomials
  - Pade rational approximation

MacLaurin's (Taylor's) series

- $f(a) = b_0 + b_1 a + b_2 a^2 + \cdots + b_N a^N$

$$= f(0) + f'(0)a + f''(0)\frac{a^2}{2} + \cdots + \frac{f^N(0)a^N}{N!}$$

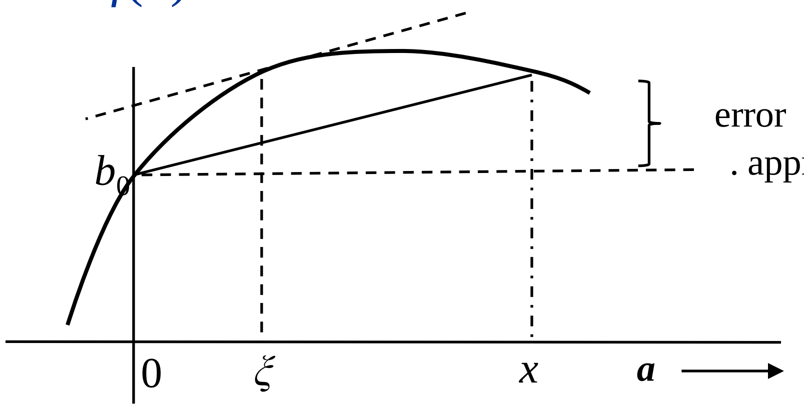
- Note that $b_k = \dfrac{f^K(0)}{K!}$

Q: What is the error involved in approximating $f(A)$ by $\hat{f}(A)$?

$$\left\| f(A) - \sum_{k=0}^{N} b_k A^k \right\|_2 \le \frac{n}{(N+1)!} \max_{0 \le x \le 1} \left\| f^{N+1}(Ax) \right\|_2$$

Approximation Error - 1

- Before we present proof, consider zeroth-order approximation to a scalar function $f(a)$ at $a = x$



- Error = $b_1x+b_2x^2+\ldots = f'(\xi).x$ (from derivative mean-value theorem)
- For n^{th} order approximation,

$$\text{error} = \frac{f^{N+1}(\xi)}{(N+1)!}\cdot x^{N+1}$$

- The result can be extended to matrices as well.
- Proof:

Let $f(Ax) = \sum_{k=0}^{N} b_k (Ax)^k + E(x); \qquad 0 \le x \le 1$

then $f_{ij}(x) = \sum_{k=0}^{N} \frac{f_{ij}^{k}(0)}{k!} x^k + \frac{f_{ij}^{N+1}(\xi_{ij})}{(N+1)!} x^{N+1} \qquad$ for some $\xi_{ij} \in [0, x]$

Approximation Error - 2

$$\Rightarrow e_{ij}(x) = \frac{f_{ij}^{N+1}(\xi_{ij})}{N+1!} x^{N+1}$$

- Now f_{ij}^{N+1} is the $(i,j)^{th}$ entry of $f^{N+1}(Ax)$ and therefore

$$e_{ij}(x) \le \max_{0\le x\le 1} \frac{f_{ij}^{N+1}}{(N+1)!}$$

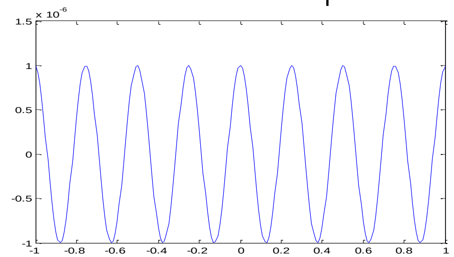
$$\le \max_{0\le x\le 1} \frac{\left\| A^{N+1} f^{N+1}(Ax) \right\|_2}{(N+1)!}$$

since $\left\| E(x) \right\|_2 \le n \max_{i,j} e_{ij}(x)$ we have

$$\left\| f(A) - \sum_{k=0}^{N} b_k A^k \right\| \le \frac{n}{(N+1)!} \max_{0\le x\le 1} \left\| A^{N+1} f^{N+1}(Ax) \right\|_2$$

Chebyshev Polynomials

- General series → Chebyshev:
  - Let $\{\Phi_k\}$ be a complete set of polynomials on $|a| \le 1$
    $\Phi_k \sim k^{th}$ order polynomial
  - Then, if $f(a) = \sum_{k=0}^{\infty} b_k \Phi_k(a)$, suggest using $\hat{f}(a) = \sum_{k=0}^{N} b_k \Phi_k(a)$
  - So error in $\hat{f}(a) \sim b_{N+1}\Phi_{N+1}(a) \approx \left| f(a) - \hat{f}(a) \right|$, an $(N+1)^{th}$ order polynomial



Want zero error at as many points as possible in (-1, 1)=N+1. Also want uniform error.

- The Chebyshev polynomials have this property:

$$T_k(a) \leftrightarrow \Phi_k(a),\ T_k(a) = \cos\left(k \cos^{-1} a\right);\ |a| \le 1$$

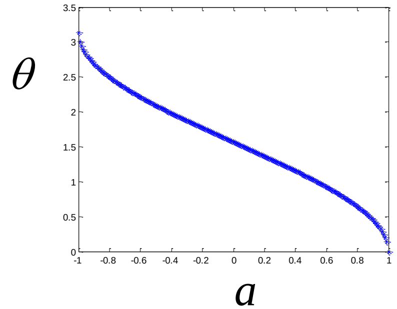
$$\text{or if } \theta = \cos^{-1} a \ \text{ or } \ \cos\theta = a \ \rightarrow \ T_k(\cos\theta) = \cos(k\theta)$$

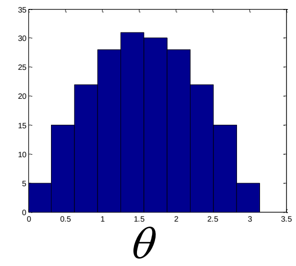
- Essentially we have made a change of variable $a = \cos\theta$

Chebyshev Polynomials - 2

- *Chebyshev functions convert periodic functions into ordinary polynomials*

$$
\begin{array}{l}
1,\ \cos\theta,\ \cos 2\theta,\ \cos 3\theta,\ \text{etc.} \\
\quad a,\ 2a^2-1,\ 4a^3-3a,\ \text{etc.} \\
T_0(a),\ T_1(a),\ T_2(a),\ T_3(a)
\end{array}
$$





uniform a → θ denser near the middle

Three-term recursion

$$\cos(k+1)\theta = \cos k\theta\cos\theta - \sin k\theta\sin\theta$$

$$\cos(k-1)\theta = \cos k\theta\cos\theta + \sin k\theta\sin\theta \quad \Rightarrow \quad T_{k+1}(a) = 2aT_k(a) - T_{k-1}(a)$$

$$\cos(k+1)\theta = 2\cos\theta\cos k\theta - \cos(k-1)\theta$$

$$\cos 2\theta = 2\cos^2\theta - 1 \quad \Rightarrow \quad T_2(a) = 2a^2 - 1$$

$$\cos 3\theta = 4\cos^3\theta - 3\cos\theta \quad \Rightarrow \quad T_3(a) = 4a^3 - 3a$$

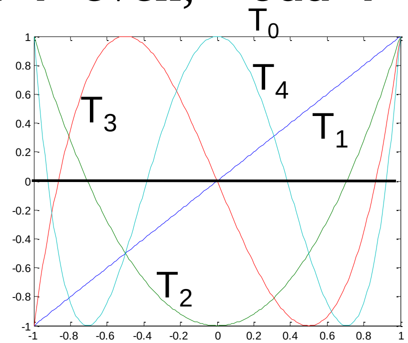
$$\cos 4\theta = 2\cos 3\theta\cos\theta - 2\cos\theta = 8\cos^4\theta - 8\cos^2\theta + 1 \quad \Rightarrow \quad T_4(a) = 8a^4 - 8a^2 + 1$$

Properties of Chebyshev Polynomials

Nice properties of Chebyshev polynomials

- The leading coefficient 2^{k-1} for $k \geq 1$ and 1 for $k = 0$.
- Symmetry $T_k(-a)=(-1)^k T_k(a)$

even → even; odd → odd



- Has k zeros in (-1,1) at $a_i = \cos\left(\frac{2i-1}{2k}\cdot\pi\right) = \cos[(i-\frac{1}{2})\frac{\pi}{k}], \quad i=1,\ 2,\cdots,\ k$
- Has (k+1) extrema (maxima and minima) at

$$a_i = \cos\left(\frac{i}{k}\cdot\pi\right), \quad i=0,\ 1,\cdots,\ k$$

$$T_k\left(\cos\frac{\pi i}{k}\right) = \cos\pi i = \begin{cases} 1 & i \text{ even} \\ -1 & i \text{ odd} \end{cases}$$

Properties of Chebyshev Polynomials

- Example: for $T_2(\cos\theta) = \cos 2\theta$ maximum at $\cos(i.\pi/2),\ i{=}0,2 \ \Rightarrow T_2(a) = +1$
  minimum at $\cos(i.\pi/2),\ i{=}1 \ \Rightarrow T_2(a) = -1$
- Chebyshev polynomials are orthogonal in the interval [-1,1] over a weight $1/\sqrt{1-a^2}$

$$\frac{2}{\pi}\int_{-1}^{1}\frac{T_k(a)T_l(a)}{\sqrt{1-a^2}}da = -\frac{2}{\pi}\int_{\pi}^{0}\cos k\theta\cos l\theta d\theta = \frac{2}{\pi}\int_{0}^{\pi}\cos k\theta\cos l\theta d\theta$$

$$= \frac{1}{\pi}\int_{0}^{\pi}[\cos\{(k+l)\theta\}+\cos\{(k-l)\theta\}]d\theta = \begin{cases} 0 & k \neq l \\ 1 & k = l,\ \ k \geq 1 \\ 2 & k = l = 0 \end{cases}$$

$\Rightarrow$ so, to find b_k, multiply $f(a)$ by $\dfrac{2}{\pi}\dfrac{T_k(a)}{\sqrt{1-a^2}}$ and integrate over $a \in (-1,\ 1)$

- Best to write $f(a) = \dfrac{b_0}{2} + \sum_{k=1}^{N} b_k T_k(a) = \sum_{k=0}^{N} b_k T_k(a) - \dfrac{b_0}{2}$

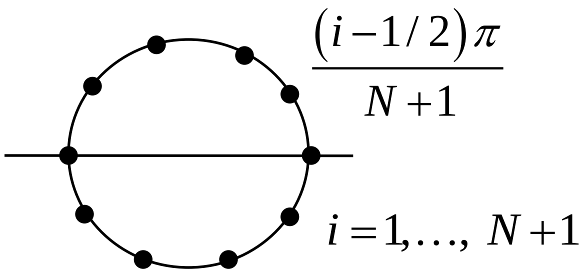
$$\Rightarrow b_k = \frac{2}{\pi}\int_{-1}^{1}\frac{f(a)T_k(a)}{\sqrt{1-a^2}}da = \frac{2}{\pi}\int_{0}^{\pi} f(\cos\theta)\cos k\theta d\theta \ \text{ for } k = 0,1,2,\ldots$$

$\Rightarrow$ b_k can be obtained from the cosine transformtion of the function $k = 0,1,$

Chebyshev Coefficients

- If we terminate at N (N^{th} order polynomial)

$$e_N(a) \sim c_{N+1} T_{N+1}(a) \le \frac{1}{2^N} \cdot \frac{\max\limits_{0 \le \xi \le a} \left| f^{N+1}(\xi) \right|}{(N+1)!}$$



or $\frac{1}{2^N}$ improvement over max. Taylor series error over interval $[-1.1]$ or same accuracy for lot less N.

- A practical method of computing b_k is to use discrete approximation at the zeros of $T_{N+1}(a)$,

$$\text{i.e., at } a_i = \cos\frac{\pi(i-1/2)}{N+1}, \quad i = 1, 2, \ldots, (N+1)$$

- So, $$b_k = \frac{2}{N+1} \sum_{k=1}^{N} f\left(\cos\frac{\pi(i-1/2)}{N+1}\right) \cos\left[\frac{\pi(i-1/2) \cdot k}{N+1}\right]$$

$\Rightarrow$ function approximation is exact at all $(N+1)$ zeros of $T_{N+1}(x)$

Why Chebyshev?

- Why Chebyshev is Good?
  - T_k is bounded between -1 and +1 $\Rightarrow \left| f(a) - \sum_{k=0}^{N} b_k T_k(a) + \frac{b_0}{2} \right| \le \sum_{k=N+1}^{\infty} |b_k|$
  - b_k^s decreases rapidly $\Rightarrow$ error is dominated by $b_{N+1}T_{N+1}(a)$, an oscillatory term with $(N+1)$ zeros and $(N+2)$ equal extrema distributed smoothly over [-1, 1] $\Rightarrow$ error spreads out evenly
  - Indeed Chebyshev is a close approximation to a minimax polynomial (of a specified degree) that optimizes $\min_{\hat{f}} \max_{|a|\le 1} \left| f(a) - \hat{f}(a) \right|$
- Application to $e^{\alpha a} = f(a)$ $|a| \le 1$: an alternate method to obtain b_k

$$b_k = \frac{2}{\pi} \int_0^{\pi} e^{\alpha \cos\theta} \cos k\theta d\theta = 2I_k(\alpha) = \text{modified Bessel function of the first kind}$$

$$I_k(\alpha) = \sum_{r=0}^{\infty} \left(\frac{\alpha}{2}\right)^{k+2r} \frac{1}{r!k+r!} \quad \text{can be precomputed or use tables or recursions, etc.}$$

$$\text{for } e^a, \; \alpha = 1 \Rightarrow e^a = I_0(1) + \sum_{k=1}^{\infty} 2I_k(1) T_k(a)$$

$$\alpha = 2 \Rightarrow e^{2a} = I_0(2) + \sum_{k=1}^{\infty} 2I_k(2) T_k(a), \quad I_k(2) = \sum_{r=0}^{\infty} \frac{1}{r!k+r!}$$

Chebyshev and e^{At}

- Computing considerations:
  - Pick $N \ni \frac{1}{2^N (N+1)!}$ sufficiently low ($\approx$ 1/10 of round-off error)
  - Note for $2x$ it is $1/(N+1)!$
  - $e^x \rightarrow N=9 \Rightarrow 1/(512.3.6.10^5) = 5 \times 10^{-9}$
  - $e^{2x} \rightarrow N=12$ for same accuracy
  - Then, evaluate $\hat{f}(a) = \frac{b_0}{2} + \sum_{k=1}^{N} b_k T_k(a)$
- Evaluation of the function in one of two ways:
  - Write out $T_k(a)$ as a k^{th} order polynomial in a and evaluate

    $\hat{f}(a) = \sum d_k a^k, \quad d_k \cong$ Taylor coefficients but <u>not exact</u>

    $\Rightarrow$ bad way: since Chebyshev exibits cancellation of terms!!
  - Better way: **<u>CLENSHAW RECURSION</u>**

$$\hat{f}(a) = \frac{b_0}{2} + \sum_{k=1}^{N} b_k T_k(a) = \frac{c_0(a) - c_2(a)}{2} = \frac{b_0}{2} + a c_1(a) - c_2(a)$$

where $c_k(a) = b_k + 2a c_{k+1}(a) - c_{k+2}(a), \; k = N-1, N-2, \ldots, 0; \; c_{N+1} = 0, \; c_N = b_N$

Clenshaw Recursion

- Proof of Clenshaw recursion:

$$\hat{f}(a)=\frac{b_0}{2}+\sum_{k=1}^{N} b_k T_k(a); \text{ s.t. } T_k(a)=2aT_{k-1}(a)-T_{k-2}(a)$$

Append with Lagrange multipliers c_k

$$\Rightarrow \hat{f}(a)=\frac{b_0}{2}+\sum_{k=1}^{N} b_k T_k(a)-\sum_{k=0}^{N} c_k\left[T_k(a)-2aT_{k-1}(a)+T_{k-2}(a)\right]$$

$$=-\frac{b_0}{2}+\sum_{k=0}^{N}(b_k-c_k)T_k(a)+\sum_{l=-1}^{N-1}2ac_{l+1}T_l(a)-\sum_{l=-2}^{N-2}c_{l+2}T_l(a)$$

$$=-\frac{b_0}{2}+(b_N-c_N)T_N(a)+(b_{N-1}-c_{N-1}+2ac_N)T_{N-1}(a)$$

$$+\sum_{k=0}^{N-2}(b_k-c_k+2ac_{k+1}-c_{k+2})T_k(a)+2a^2c_0-c_0\left(2a^2-1\right)-c_1a$$

$$k=1: T_1(a)=2aT_0(a)-T_{-1}(a)$$
$$\Rightarrow a=2a-T_{-1}(a)\Rightarrow T_{-1}(a)=a$$
$$k=0: T_0(a)=2aT_{-1}(a)-T_{-2}(a)$$
$$\Rightarrow T_{-2}(a)=2a^2-1$$

Note: use Chebyshev recursion to get $T_{-1}(a)$ and $T_{-2}(a)$.

Selecting the multiplier sequence as: $c_k=b_k+2ac_{k+1}-c_{k+2}; \quad c_{N+1}=0, \ c_N=b_N;$

we obtain $\hat{f}(a)=-\frac{b_0}{2}+c_0-c_1a=\frac{c_0-c_2}{2}$

since we are computing terms backwards, recursion is stable

Practicalities - 1

- Suppose want $f(a)$ for $x_1 \le a \le x_2$

  $\Rightarrow$ define $y = \dfrac{a - (x_1 + x_2)/2}{(x_2 - x_1)}$ $\Rightarrow$ $|y| \le 0.5$ $\Rightarrow$ $e^a = e^{y(x_2 - x_1)} e^{(x_1 + x_2)/2}$

- Computing e^{At}
  - Have $e^a = \sum_{k=0}^{N} b_k T_k(a) - \dfrac{b_0}{2}$ with error $\dfrac{1}{2^N (N+1)!}$ valid in the region $|a| \le 1$
  - Similarly $e^{At} = \sum_{k=0}^{N} b_k T_k(At) - \dfrac{b_0}{2}$. need $|\lambda_i(At)| \le 1 \ \forall i$ or $\rho(At) \le 1$ *spectral radius*

- Three step process for computing e^{At}
  - Make Eigen values cluster around zero $A \to \tilde{A}$ (SHIFTING)
  - Make $\rho\left(\tilde{A}t\right) \le 1 (0.2 - 0.5)$ through scaling (SCALING)
  - Use doubling concept (DOUBLING)

- SHIFTING

  Let $\beta = \dfrac{1}{n} \mathrm{tr}(A)$ and $\tilde{A} = A - \beta I$; $\left|\gamma I - \tilde{A}\right| = \left|(\gamma + \beta) I - A\right| = \left|\lambda I - A\right| = 0$

  $\Rightarrow \gamma_i = \lambda_i - \beta \Rightarrow \sum_{i=1}^{n} \gamma_i = \sum_{i=1}^{n} \lambda_i - n\beta = 0 \Rightarrow$ Eigen values clustered around zero.

  Note: $e^{At} = e^{\tilde{A}t} \cdot e^{\beta t}$, where $e^{\beta t}$ is a scalar

Practicalities - 2

SCALING

Find $\delta = \dfrac{t}{2^m} \ni \left\|\tilde{A}\delta\right\| \le 0.2\, to\, 0.5 \;\Rightarrow\; \rho\left(\tilde{A}\delta\right) \in [0.2\ 0.5]$

$$\Rightarrow \frac{\left\|\tilde{A}t\right\|}{2^m} \le c \;\Rightarrow\; m \ge \left\lceil \log_2 \frac{\left\|\tilde{A}t\right\|}{c} \right\rceil$$

Compute $e^{\tilde{A}\delta}$ via Chebyshev $\Rightarrow e^{\tilde{A}\delta} = \dfrac{C_0\left(\tilde{A}\delta\right) - C_2\left(\tilde{A}\delta\right)}{2}$

where $C_k\left(\tilde{A}\delta\right) = b_k I + 2\tilde{A}\delta C_{k+1}\left(\tilde{A}\delta\right) - C_{k+2}\left(\tilde{A}\delta\right),\ C_N = b_N I,\ C_{N+1} = 0$

DOUBLING

$Y = e^{\tilde{A}\delta}$

Do $i = 1, m$

$\quad Y = Y * Y$

End Do

$$e^{\tilde{A}\delta} \to e^{2\tilde{A}\delta} \to e^{4\tilde{A}\delta} \to \ldots \to e^{2^m \tilde{A}\delta} = e^{\tilde{A}t}$$

PUT THE SHIFT BACK

Finally, $e^{At} = e^{\tilde{A}t} \cdot e^{\beta t}$

Pade Approximation - 1

- Rational Function Approximation to e^a: Pade Approximation

  - Pade approximation $\hat{f}(a) = \dfrac{n_0 + n_1 a + n_2 a^2 + \ldots + n_m a^m}{1 + d_1 a + d_2 a^2 + \ldots + d_n a^n} = R(m,n)$
  - Idea is to pick (n_i, d_i) such that $\hat{f}$ agrees with the Taylor series to maximum number of terms ($2m$ in general): $b_0 + b_1 a + b_2 a^2 + \ldots + b_{2m} a^{2m}$
  - Error would be $\dfrac{a^{2m+1}}{(2m+1)!} f^{(2m+1)}(\tau)$ where $0 \le \tau \le a$
  - Pade is like Taylor series, but generally somewhat better, but not as good as Chebyshev (MATLAB uses Pade)

$$\text{Application to } e^x = \begin{cases} n_0 = 1 \\ n_i = \dfrac{m!(2m-i)!}{i!(m-i)!2m!} = \dbinom{m}{i}\dfrac{(2m-i)!}{2m!} \\ d_i = (-1)^i n_i \end{cases}$$

Pade Approximation - 2

e.g., $m=1 \Rightarrow \frac{1+a/2}{1-a/2}$; $m=2 \Rightarrow \frac{1+a/2+a^2/12}{1-a/2+a^2/12}$ as good as or better than 4 th order Taylor series i.e., error in a^5

$$\left(1+\frac{a}{2}\right)\left(1+\frac{a}{2}+\frac{a^2}{4}+\frac{a^3}{8}+\dots\right)=1+a+\frac{a^2}{2}+\frac{a^3}{4} \quad \text{need } \frac{a^3}{6} \Rightarrow \text{ error } \frac{a^3}{12}$$

- For matrix computation $\left(I+n_1A+n_2A^2+\dots\right)\left(I+d_1A+d_2A^2+\dots+\dots\right)^{-1}=N_m(A)\left[D_m(A)\right]^{-1}$
- Total computation 2(m-1) multiplications +1 inverse $\Rightarrow$ (2m-1) same as 2m^{th} order Taylor.
- Of course, can use modified Horner's rule to reduce computations

❑ We can exploit the similarity of numerator and denominator

- Compute $C=I+n_2A^2+n_4A^4+\dots=\sum \text{even powers}$

$$D=A\left[n_1I+n_3A^2+\dots\right.\quad=\sum \text{odd powers}$$

$$N_m(A)=C+D, \qquad D_m(A)=C-D$$

- Requires m multipliers and solution of $A\underline{x}_i=\underline{b}_i$ (n of them)

then $(C-D)\,\hat{f}(A)=(C+D)$

$$(C-D)\left(\underline{\hat{f}}_1\ \underline{\hat{f}}_2\ \dots\ \underline{\hat{f}}_n\right)=\left(\underline{c}_1+\underline{d}_1\quad \underline{c}_2+\underline{d}_2\quad \dots\quad \underline{c}_n+\underline{d}_n\right)$$

Pade Approximation - 3

- Algorithm:

  1. Compute $\beta = \operatorname{tr}(A)/n$; $\tilde{A} = A - \beta I$
  2. Find $\delta = t/2^M$ such that $\|\tilde{A}\|\delta \le 0.5 (\approx 0.2)$
  3. Compute $Y = e^{\tilde{A}\delta}$ via PADE
  4. Use Doubling

     Do $i = 1, M$

     $\quad Y \leftarrow Y * Y$

     End Do
  5. $e^{At} = e^{\beta t} \cdot Y$

- Use 4th order PADE for error $\approx 10^{-9}$

$$\text{Relative error} = \frac{\left\|e^{A} - Y\right\|_\infty}{\left\|e^{A}\right\|_\infty} \le \left\|\tilde{A}\right\| e^{\in \left\|\tilde{A}\right\|} \quad \text{where} \quad \in = \frac{2^{3-2m} \cdot (m!)^2}{2m!(2m+1)!}$$

- *Research Problem*: Combine PADE and CHEBYSHEV

$$\hat{f}(A) = \left[\sum_{i=0}^{m} n_i T_i(A)\right]\left[\sum_{i=0}^{m} d_i T_i(A)\right]^{-1} \quad \text{expect } d_i = (-1)^i n_i$$

Upper Schur Matrix Approach

- Transform A Matrix into Upper Schur Form (Lectures 10, 13 and 14)

$$R = Q^T AQ; Q = \text{Orthogonal} \Rightarrow A = QRQ^T \Rightarrow e^{At} = Qe^{Rt}Q^T = QGQ^T; G = e^{Rt}$$

$$R = \text{Block upper triangular matrix} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1p} \\ & R_{22} & \cdots & R_{2p} \\ & & & R_{pp} \end{bmatrix}$$

$G = e^{Rt}$ is easy to compute

where R_{ii} is 2x2 or 1 x 1

- G is relatively easy to compute

$$g_{ii} = e^{r_{ii}t}; i = 1, 2, .., n$$

$For\ k = 1, 2, .., n-1\ Do$

$For\ i = 1, 2, ..., n-k\ D0$

$Set\ j = i + k$

$$g_{ij} = \frac{1}{(r_{ii} - r_{jj})}[r_{ij}(g_{ii} - g_{jj}) + \sum_{p=i+1}^{j-1}(g_{ip}r_{pj} - r_{ip}g_{pj})]$$

End

End

Problem when Eigen values are Close to each other

e^{At} for Stochastic Matrices

- Special case: $A=Q$ a stochastic matrix
  - $Q=[q_{ij}]$ is a transition rate matrix (infinitesimal generator matrix) of the continuous-time Markov chain (CTMC)

    Important property of Q: $q_{ii} = -\sum_{\substack{j=1 \\ j\neq i}}^{N} q_{ij} \Rightarrow$ each row of Q sums to zero.
  - Moreover, we are primarily interested in solving:

    $$\dot{\underline{p}} = Q^T \underline{p}; \;\; \underline{p}(0) = \underline{p}_0 \;\; \Rightarrow \underline{p}(t) = e^{Q^T t} \underline{p}_0$$
  - One popular method is called "Uniformization."
- What is Uniformization?
  - Let $q \geq \max|-q_{ii}|$
  - Then, can construct an equivalent process for which the transition rate from each state i is q and a fraction $(1+q_{ii}/q)$ of these transitions return immediately to state i.
  - Basically, this amounts to constructing a discrete-time Markov chain (DTMC) by scaling Q by $(1/q)$ and shifting the diagonals of the scaled matrix by 1, i.e., define $Q^* = Q/q + I \;\; \Rightarrow \;\; Q = q\left(Q^* - I\right)$

Uniformization

- The process of obtaining a DTMC from a CTMC as above is called *uniformization* (= special scaling & shifting)
- Clearly, $$e^{Qt} = \sum_{n=0}^{\infty} e^{-qt} \frac{(qt)^n}{n!} \left[Q^*\right]^n$$

  then, $\underline{p}(t) = \sum_{n=0}^{N} e^{-qt} \frac{(qt)^n}{n!} \underline{\pi}_n$ where $\underline{\pi}_n = \left[Q^*\right]^T \underline{\pi}_{n-1}$ with $\underline{\pi}_0 = \underline{p}_0$

  This can be evaluated with matrix-vector operations only.
- For a specified accuracy ε, the number of terms N to be retained is computed from:

$$\varepsilon = \sum_{n=N+1}^{\infty} e^{-qt} \frac{(qt)^n}{n!} = 1 - \sum_{n=0}^{N} e^{-qt} \frac{(qt)^n}{n!}$$

- Although, this is basically Taylor series, it works for this case because of the special structure of Q.

❑ Other methods for solving Markov chain models: ODE solvers.

Dubious Methods - 1

- Bad (Dubious) Methods:
  - Do not use exact formulae or Eigen value-based methods (unless by hand). May be OK if $A=A^T$
  - Caley-Hamilton Theorem

$$A^n + \alpha_n A^{n-1} + \ldots + \alpha_2 A + \alpha_1 I = 0 \;\Rightarrow\; A^k = -\sum_{i=0}^{n-1} \beta_{ik} A^i$$

$$e^{At} = \sum_{k=0}^{\infty} \frac{t^k \cdot A^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \sum_{i=0}^{n-1} \beta_{ik} A^i = \sum_{i=0}^{n-1} \left(\sum_{k=0}^{\infty} \beta_{ik} \frac{t^k}{k!} \right) A^i = \sum_{i=0}^{n-1} \gamma_i(t) \cdot A^i$$

  - Lagrange interpolation (SYLVESTER's THEOREM)

$$e^{At} = \sum_{i=1}^{n} e^{\lambda_i t} \prod_{\substack{k=1 \\ k \neq i}} \frac{(A - \lambda_k I)}{(\lambda_i - \lambda_k)}$$

Alternate: $e^{At} = \sum_{i=1}^{n} e^{\lambda_i t} \underline{\xi}_i \underline{\eta}_i^T$

Dubious Methods - 2

- Inverse Laplace Transform: *Leverrier and Faddeva* or *Sourian and Frame* Algorithm

$$(sI - A)^{-1} = \sum_{k=1}^{n} \frac{s^{k-1}}{d(s)} B_k$$

$$d(s) = |sI - A| = s^n + \alpha_n s^{n-1} + \ldots + \alpha_2 s + \alpha_1 = 0$$

$$B_n = I \quad \rightarrow \quad \alpha_n = -\mathrm{tr}(AB_n)/1$$

$$B_{n-1} = AB_n + \alpha_n I \quad \rightarrow \quad \alpha_{n-1} = -\mathrm{tr}(AB_{n-1})/2$$

$$B_{n-2} = AB_{n-1} + \alpha_{n-1} I \quad \rightarrow \quad \alpha_{n-2} = -\mathrm{tr}(AB_{n-2})/3$$

$$B_1 = AB_2 + \alpha_2 I \quad \rightarrow \quad \alpha_1 = -\mathrm{tr}(AB_1)/n$$

Check: $AB_1 + \alpha_1 I = 0$; $A^{-1} = -\frac{B_1}{\alpha_1}$; $\det(A) = (-1)^n \alpha_1$; $e^{At} = L^{-1}\{\sum_{k=1}^{n-1} \frac{s^{k-1}}{d(s)} B_k\}$

❑ Slick Test cases for e^{At} subroutines

Suppose A is an idempotent matrix, $A^2=A$

$$\text{Then,} \quad e^{At} = \sum_{k=1}^{\infty} \frac{A^k t^k}{k!} = I + \sum_{k=1}^{\infty} \frac{At^k}{k!} = I + A(e^t - 1)$$

How to Test e^{At} Software?

- How to construct idempotent matrices
  - Consider any m x n matrix C and its pseudo inverse $C^{\dagger}$. Then $(I_n - C^{\dagger} C)$ and $(I_m - C C^{\dagger})$ are idempotent.
- Tests on the accuracy of computed e^{At}

1) Known e^{At} as in idempotent case

2) $e^{At}.e^{-At} = I_n$ or $\left\| e^{At} \cdot e^{-At} - I_n \right\|$

3) $\left| e^{At} \right| = | P e^{\Lambda t} P^{-1} | = | P || e^{\Lambda t} || P^{-1} | = e^{\sum_{i=1}^{n} \lambda_i t} = e^{\mathrm{tr}(At)}$

- Test Examples:

1) Idempotent matrices (e.g., Kerr, 1990)
2) Bad problem for Taylor series (Moler and Van Loan, SIAM Review, 1978) (assume t=1)

$$A = \begin{bmatrix} -49 & 24 \\ -64 & 31 \end{bmatrix}$$

Test Examples

3) Other test cases from Moler and Van Loan

$$A=\begin{bmatrix}0&6&0&0\\0&0&6&0\\0&0&0&6\\0&0&0&0\end{bmatrix};\quad A=\begin{bmatrix}1-\in&1\\0&1+\in\end{bmatrix},\quad \in\ =10^{-5},\,10^{-6},\,10^{-7}$$

4) Some other test cases

$$A=\begin{bmatrix}0&1\\-0.5&-1\end{bmatrix};\quad A=\begin{bmatrix}4&2&0\\1&4&1\\1&1&4\end{bmatrix};A=n\,by\,n\,controllability\,matrix\,with\,zero\,n^{th}\ row$$

5) Terrestrial Navigation example

- Consider the local-level terrestrial navigator (Britting, K.S., 1971, "Inertial Navigation Systems Analysis," New York, John Wiley) which has no vertical accelerometer.
- This type of system consists of a three-axis inertial platform and two accelerometers mounted orthogonally in the east and north directions

Test Examples

- The error equations for this class of system can be written as follows:

$$\underline{\dot{x}} = A\underline{x} + \underline{b}u$$

where $\underline{x}^T = \left[\in_N,\ \in_E,\ \in_D,\ dL,\ dl,\ d\dot{L},\ d\dot{l}\right]$

$$A = \begin{bmatrix} 0 & -W_{ie}\sin L & 0 & -W_{ie}\sin L & 0 & 0 & \cos L \\ -W_{ie}\sin L & 0 & -W_{ie}\sin L & 0 & 0 & -1 & 0 \\ 0 & -W_{ie}\cos L & 0 & -W_{ie}\cos L & 0 & 0 & -\sin L \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & W_s^2 & W_s^2 & 0 & 0 & 0 & 0 \\ -W_s\sec L & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and $\underline{b}u = \left[q_1,\ q_2,\ q_3,\ 0,\ 0,\ q_{4/r},\ q_{5/r}\cos L\right]^T$

- The transition matrix for the above system must be evaluated for the following parameters.

$$W_{ie} = \frac{15\pi}{180^\circ}\ \text{rad/h},\quad L = 45^\circ,\quad W_s = \sqrt{20.1}\ \text{rad/h}$$

Summary

- What is the need for computing e^{At}?
- Evaluation of matrix polynomials (**Horner's rule**)
- Truncation errors
- Chebyshev approximation
  - Properties
  - **Clenshaw recursion**
  - Concepts of **shifting**, **scaling**, and **doubling**
- Pade approximation
- Upper Schur transformation-based approach
- Special case: A is a stochastic matrix (*a la* Markov chains)
- How **not** to compute e^{At}?