Verschiedene Verfahren zur Berechnung der Exponentialabbildung einer Matrix

Anwendungen der Exponentialmatrix

- Bei linearen Differentialgleichungen

Anwendungen der Exponentialmatrix

- Bei linearen Differentialgleichungen
  - Gegeben sei das Anfangswertproblem

$$\mathbf{y}' = A\mathbf{y}$$
$$\mathbf{y}(t_0) = \mathbf{y}_0$$

Anwendungen der Exponentialmatrix

- Bei linearen Differentialgleichungen
  - Gegeben sei das Anfangswertproblem

$$\mathbf{y}' = A\mathbf{y}$$
$$\mathbf{y}(t_0) = \mathbf{y}_0$$

  - Die analytische Lösung lautet

$$\mathbf{y}(t) = e^{(t-t_0)A}\mathbf{y}_0$$

Anwendungen der Exponentialmatrix

- Bei linearen Differentialgleichungen
  - Gegeben sei das Anfangswertproblem

$$\mathbf{y}' = A\mathbf{y}$$
$$\mathbf{y}(t_0) = \mathbf{y}_0$$

  - Die analytische Lösung lautet

$$\mathbf{y}(t) = e^{(t-t_0)A}\mathbf{y}_0$$

- Im Quantum Control ist man an folgendem Produkt interessiert:

$$e^{-it_M H_M} \dots e^{-it_k H_k} \dots e^{-it_1 H_1}$$

- Ausgangspunkt: Taylorreihe der e-Funktion

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Definition der Exponentialfunktion einer Matrix

- Ausgangspunkt: Taylorreihe der e-Funktion

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

- Definition der Exponentialfunktion einer Matrix $A \in \mathbb{R}^{N \times N}$

$$\exp(A) := e^A := \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

Definition der Exponentialfunktion einer Matrix

- Ausgangspunkt: Taylorreihe der e-Funktion

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

- Definition der Exponentialfunktion einer Matrix $A \in \mathbb{R}^{N \times N}$

$$\exp(A) := e^A := \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

- Für eine Diagonalmatrix $D = diag(d_1, \ldots, d_n)$ gilt

$$e^D = diag(e^{d_1}, \ldots, e^{d_n})$$

Berechnung der Exponentialmatrix

- Naheliegend: Berechnung durch “abgeschnittene“ Taylor-Reihe:

$$\exp(A) \approx \sum_{k=0}^{n} \frac{A^k}{k!}$$

Berechnung der Exponentialmatrix

- Naheliegend: Berechnung durch “abgeschnittene“ Taylor-Reihe:

$$\exp(A) \approx \sum_{k=0}^{n} \frac{A^k}{k!}$$

- Diese Reihe konvergiert allerdings sehr langsam:

$$\| e^A - \sum_{k=0}^{n} \frac{A^k}{k!} \| \leq \left(\frac{\| A \|^{n+1}}{(n+1)!} \right) \left(\frac{1}{1 - \| A \| / (n+2)} \right) \leq \delta$$

Berechnung der Exponentialmatrix

- Naheliegend: Berechnung durch “abgeschnittene“ Taylor-Reihe:

$$\exp(A) \approx \sum_{k=0}^{n} \frac{A^k}{k!}$$

- Diese Reihe konvergiert allerdings sehr langsam:

$$\| e^A - \sum_{k=0}^{n} \frac{A^k}{k!} \| \le \left(\frac{\| A \|^{n+1}}{(n+1)!} \right) \left(\frac{1}{1 - \| A \| / (n+2)} \right) \le \delta$$

- Beispiel: $\| A \| = 100$ und $\delta = 10^{-6}$ liefert $n = 284$.

Berechnung der Exponentialmatrix

- Naheliegend: Berechnung durch “abgeschnittene“ Taylor-Reihe:

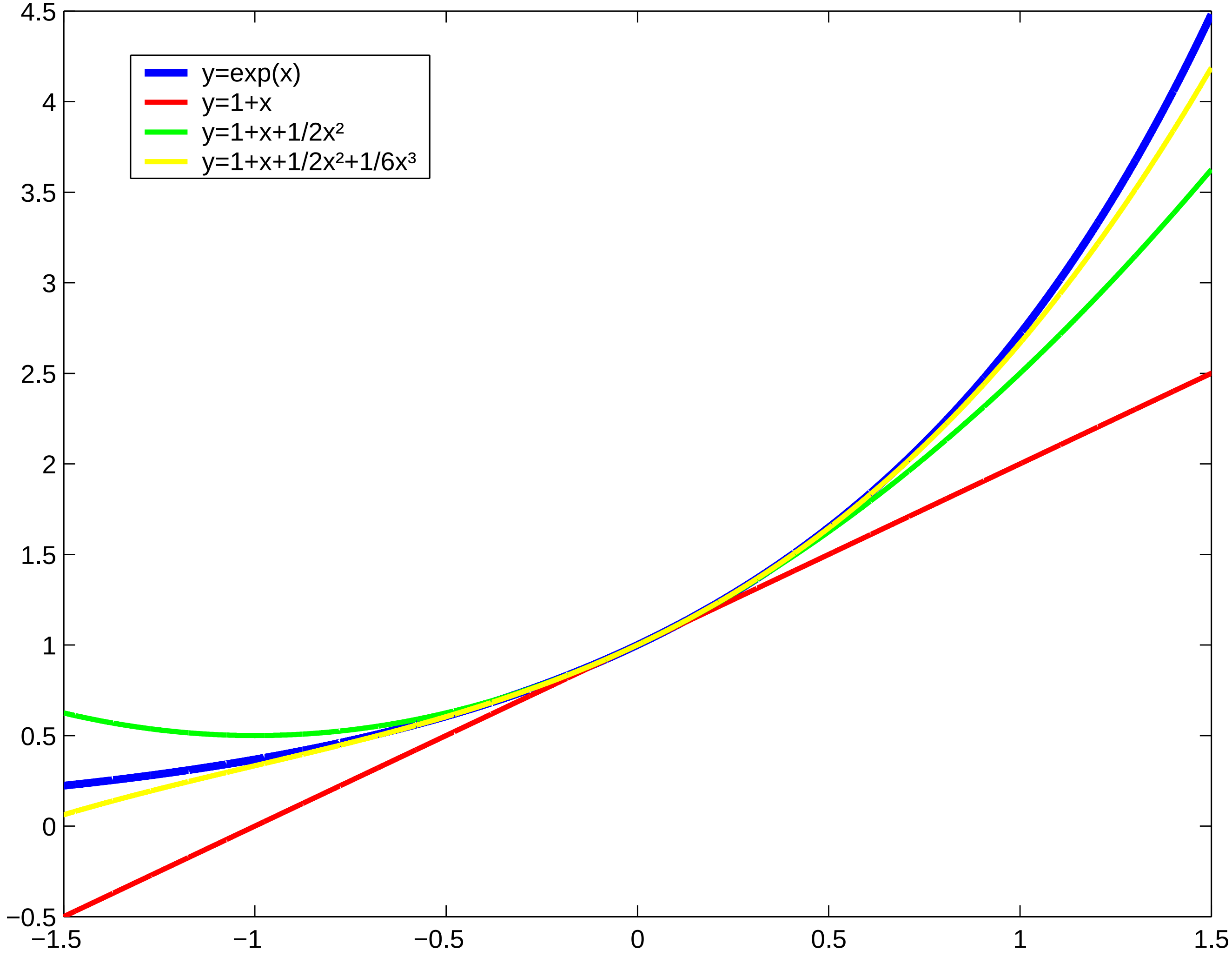
$$\exp(A) \approx \sum_{k=0}^{n} \frac{A^k}{k!}$$

- Diese Reihe konvergiert allerdings sehr langsam:

$$\| e^A - \sum_{k=0}^{n} \frac{A^k}{k!} \| \leq \left(\frac{\| A \|^{n+1}}{(n+1)!} \right) \left(\frac{1}{1 - \| A \| / (n+2)} \right) \leq \delta$$

- Beispiel: $\| A \| = 100$ und $\delta = 10^{-6}$ liefert $n = 284$.
- Weiteres Problem: Numerische Instabilität

Die ersten Taylorpolynome



- Betrachte die 2×2-Matrix $A = \begin{pmatrix} -25 & 20 \\ 20 & -20 \end{pmatrix}$

- Betrachte die 2×2-Matrix $A = \begin{pmatrix} -25 & 20 \\ 20 & -20 \end{pmatrix}$
- Die exakte Lösung lautet

$$e^A = \begin{pmatrix} 0.0420 & 0.0476 \\ 0.0476 & 0.0539 \end{pmatrix}$$

Beispiel zur Berechnung über Taylor-Reihe

- Betrachte die 2×2-Matrix $A = \begin{pmatrix} -25 & 20 \\ 20 & -20 \end{pmatrix}$
- Die exakte Lösung lautet

$$e^A = \begin{pmatrix} 0.0420 & 0.0476 \\ 0.0476 & 0.0539 \end{pmatrix}$$

- Taylor-Entwicklung liefert nach 180 Schritten

$$e^A \approx \begin{pmatrix} 41.3007 & -39.8867 \\ -78.2696 & -13.3593 \end{pmatrix}$$

Beispiel zur Berechnung über Taylor-Reihe

- Ursache: alternierende Vorzeichen in den Potenzen von A

- Ursache: alternierende Vorzeichen in den Potenzen von A
  - $A = \begin{pmatrix} -25 & 20 \\ 20 & -20 \end{pmatrix}$

- Ursache: alternierende Vorzeichen in den Potenzen von A
  - $A = \begin{pmatrix} -25 & 20 \\ 20 & -20 \end{pmatrix}$
  - $A^2 = \begin{pmatrix} 1025 & -900 \\ -900 & 800 \end{pmatrix}$

- Ursache: alternierende Vorzeichen in den Potenzen von A
  - $A = \begin{pmatrix} -25 & 20 \\ 20 & -20 \end{pmatrix}$
  - $A^2 = \begin{pmatrix} 1025 & -900 \\ -900 & 800 \end{pmatrix}$
  - $A^3 = \begin{pmatrix} -43625 & 38500 \\ 38500 & -34000 \end{pmatrix}$

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$
- Wähle m als Zweierpotenz: $m = 2^k$

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$
- Wähle m als Zweierpotenz: $m = 2^k$
- $\left(e^{A/m}\right)^{2^k}$ durch wiederholtes (k-faches) Quadrieren

Weitere Verfahren zur Berechnung von e^A

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$
- Wähle m als Zweierpotenz: $m = 2^k$
- $\left(e^{A/m}\right)^{2^k}$ durch wiederholtes (k-faches) Quadrieren
- Wähle m, so dass $\parallel A \parallel /m \leq 1$

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$
- Wähle m als Zweierpotenz: $m = 2^k$
- $\left(e^{A/m}\right)^{2^k}$ durch wiederholtes (k-faches) Quadrieren
- Wähle m, so dass $\parallel A \parallel /m \leq 1$
- Kosten des Squarings:

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$
- Wähle m als Zweierpotenz: $m = 2^k$
- $\left(e^{A/m}\right)^{2^k}$ durch wiederholtes (k-faches) Quadrieren
- Wähle m, so dass $\| A \| /m \leq 1$
- Kosten des Squarings:
  - $k = \lceil \log_2(\| A \|) \rceil$ Matrix-Matrix-Multiplikationen

Scaling and Squaring

- Für $m \in \mathbb{N}$ gilt $e^A = \left(e^{A/m}\right)^m$
- Wähle m als Zweierpotenz: $m = 2^k$
- $\left(e^{A/m}\right)^{2^k}$ durch wiederholtes (k-faches) Quadrieren
- Wähle m, so dass $\| A \| /m \leq 1$
- Kosten des Squarings:
  - $k = \lceil \log_2(\| A \|) \rceil$ Matrix-Matrix-Multiplikationen
  - Gesamtkosten: $2 \lceil \log_2(\| A \|) \rceil N^3$

Padé-Approximation

- Padé-Approximation der Exponentialfunktion:

Padé-Approximation

- Padé-Approximation der Exponentialfunktion:

$$e^x = \frac{p_{nm}(x)}{q_{nm}(x)}$$

Padé-Approximation

- Padé-Approximation der Exponentialfunktion:

$$e^x = \frac{p_{nm}(x)}{q_{nm}(x)}$$

  - $p_{nm}(x) = \sum_{k=0}^{n} \frac{(n+m-k)!n!}{(n+m)!k!(n-k)!} x^k$

Padé-Approximation

- Padé-Approximation der Exponentialfunktion:

$$e^x = \frac{p_{nm}(x)}{q_{nm}(x)}$$

  - $p_{nm}(x) = \sum_{k=0}^{n} \frac{(n+m-k)!n!}{(n+m)!k!(n-k)!} x^k$
  - $q_{nm}(x) = \sum_{k=0}^{m} \frac{(n+m-k)!m!}{(n+m)!k!(m-k)!} (-x)^k$

Padé-Approximation

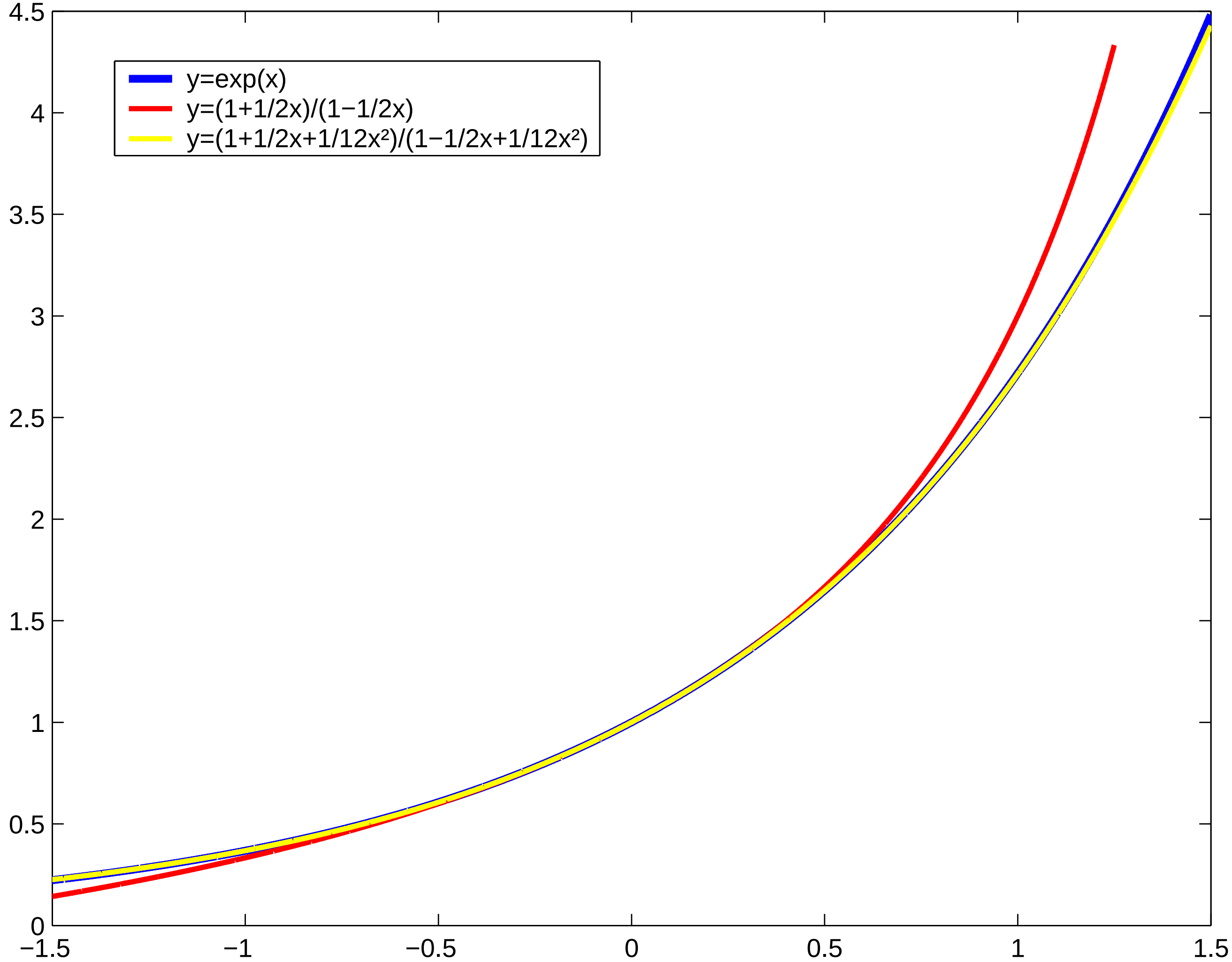
- Padé-Approximation der Exponentialfunktion:

$$e^x = \frac{p_{nm}(x)}{q_{nm}(x)}$$

  - $p_{nm}(x) = \sum_{k=0}^{n} \frac{(n+m-k)!n!}{(n+m)!k!(n-k)!} x^k$
  - $q_{nm}(x) = \sum_{k=0}^{m} \frac{(n+m-k)!m!}{(n+m)!k!(m-k)!} (-x)^k$

- $e^A \approx (q_{nn}(A))^{-1} p_{nn}(A)$

Padé-Approximation vom Index (1,1) und (2,2)



Kosten zur Berechnung von $e^A \approx (q_{nn}(A))^{-1} p_{nn}(A)$:

- Berechnung von $p_{nn}(A)$ und $q_{nn}(A)$ nach Horner:

$$a_3 A^3 + a_2 A^2 + a_1 A + a_0 E = [(a_3 A + a_2 E)A + a_1 E]A + a_0 E$$

Kosten zur Berechnung von $e^A \approx (q_{nn}(A))^{-1} p_{nn}(A)$:

- Berechnung von $p_{nn}(A)$ und $q_{nn}(A)$ nach Horner:

$$a_3 A^3 + a_2 A^2 + a_1 A + a_0 E = [(a_3 A + a_2 E)A + a_1 E]A + a_0 E$$

- Falls A dünnbesetzt, Produkte der Form Voll*dünn

Padé-Approximation

Kosten zur Berechnung von $e^A \approx (q_{nn}(A))^{-1} p_{nn}(A)$:

- Berechnung von $p_{nn}(A)$ und $q_{nn}(A)$ nach Horner:

$$a_3 A^3 + a_2 A^2 + a_1 A + a_0 E = [(a_3 A + a_2 E)A + a_1 E]A + a_0 E$$

- Falls A dünnbesetzt, Produkte der Form Voll*dünn
- A voll: $2(n-1)(2N^3)$, A dünn: $2(n-1)(2N^2 \log(N))$

Kosten zur Berechnung von $e^A \approx (q_{nn}(A))^{-1} p_{nn}(A)$:

- Berechnung von $p_{nn}(A)$ und $q_{nn}(A)$ nach Horner:

$$a_3A^3 + a_2A^2 + a_1A + a_0E = [(a_3A + a_2E)A + a_1E]A + a_0E$$

- Falls A dünnbesetzt, Produkte der Form Voll*dünn
- A voll: $2(n-1)(2N^3)$, A dünn: $2(n-1)(2N^2\log(N))$
- Invertierung von $(q_{nn}(A))^{-1}$: $2N^3$

Kosten zur Berechnung von $e^A \approx (q_{nn}(A))^{-1} p_{nn}(A)$:

- Berechnung von $p_{nn}(A)$ und $q_{nn}(A)$ nach Horner:

$$a_3A^3 + a_2A^2 + a_1A + a_0E = [(a_3A + a_2E)A + a_1E]A + a_0E$$

- Falls A dünnbesetzt, Produkte der Form Voll*dünn
- A voll: $2(n-1)(2N^3)$, A dünn: $2(n-1)(2N^2\log(N))$
- Invertierung von $(q_{nn}(A))^{-1}$: $2N^3$
- Das Produkt $(q_{nn}(A))^{-1} p_{nn}(A)$: $2N^3$

Chebyshev-Approximation

- Für $-1 \leq x \leq 1$ gilt die Entwicklung

- Für $-1 \leq x \leq 1$ gilt die Entwicklung

$$e^x = J_0(i) + 2\sum_{k=1}^{\infty} i^k J_k(-i) T_k(x)$$

Chebyshev-Approximation

- Für $-1 \leq x \leq 1$ gilt die Entwicklung

$$e^x = J_0(i) + 2 \sum_{k=1}^{\infty} i^k J_k(-i) T_k(x)$$

- T_k: Chebyshev-Polynom 1. Art der Ordnung k

Chebyshev-Approximation

- Für $-1 \leq x \leq 1$ gilt die Entwicklung

$$e^x = J_0(i) + 2\sum_{k=1}^{\infty} i^k J_k(-i) T_k(x)$$

- T_k: Chebyshev-Polynom 1. Art der Ordnung k
- J_k: Besselfunktion der Ordnung k

Chebyshev-Approximation

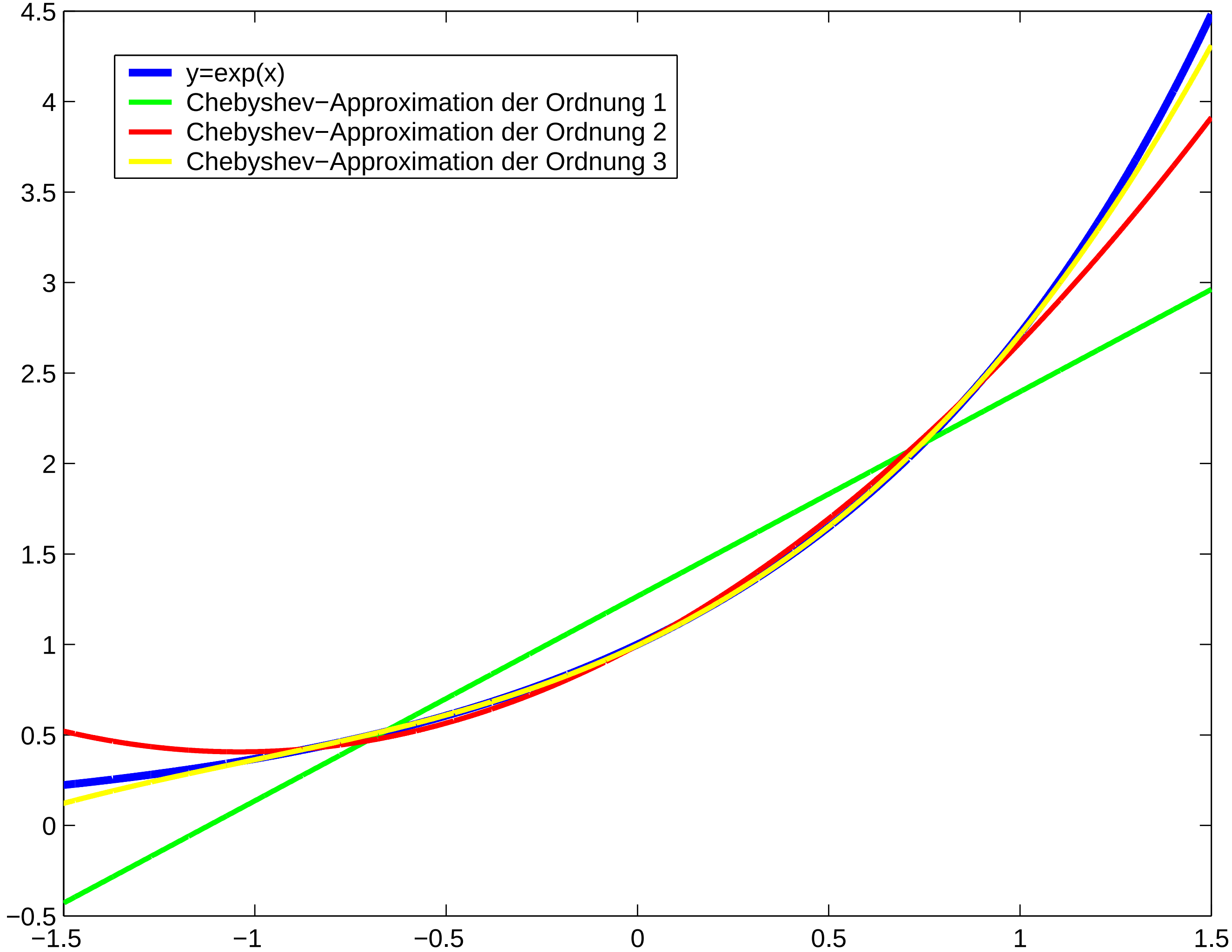
- Für $-1 \leq x \leq 1$ gilt die Entwicklung

$$e^x = J_0(i) + 2 \sum_{k=1}^{\infty} i^k J_k(-i) T_k(x)$$

- T_k: Chebyshev-Polynom 1. Art der Ordnung k
- J_k: Besselfunktion der Ordnung k
- Chebyshev-Entwicklung $s_n(x)$ ist Least-Squares-Approximation bzgl. der Gewichtsfunktion $\frac{1}{\sqrt{1-x^2}}$

$$\int_{-1}^{1} [e^x - s_n(x)]^2 \frac{dx}{\sqrt{1-x^2}} \leq \int_{-1}^{1} [e^x - p(x)]^2 \frac{dx}{\sqrt{1-x^2}} \text{ für } p \in \mathcal{P}_n$$

Chebyshev-Approximation der e-Funktion



- Transfer auf Matrizen:

$$e^A = J_0(i)E + 2\sum_{k=1}^{n} i^k J_k(-i)T_k(A)$$

Chebyshev-Approximation

- Transfer auf Matrizen:

$$e^A = J_0(i)E + 2\sum_{k=1}^{n} i^k J_k(-i)T_k(A)$$

- Kosten der Auswertung:

Chebyshev-Approximation

- Transfer auf Matrizen:

$$e^A = J_0(i)E + 2\sum_{k=1}^{n} i^k J_k(-i)T_k(A)$$

- Kosten der Auswertung:
  - Auswertung der Chebyshev-Polynome nach dem Horner-Schema

- Transfer auf Matrizen:

$$e^A = J_0(i)E + 2\sum_{k=1}^{n} i^k J_k(-i) T_k(A)$$

- Kosten der Auswertung:
  - Auswertung der Chebyshev-Polynome nach dem Horner-Schema
  - A vollbesetzt: $(n-1) \cdot 2N^3$ Operationen

- Transfer auf Matrizen:

$$e^A = J_0(i)E + 2\sum_{k=1}^{n} i^k J_k(-i)T_k(A)$$

- Kosten der Auswertung:
  - Auswertung der Chebyshev-Polynome nach dem Horner-Schema
  - A vollbesetzt: $(n-1)\cdot 2N^3$ Operationen
  - A dünnbesetzt: $(n-1)\cdot 2N^2\log(N)$ Operationen

Weitere Verfahren zur Berechnung von e^A

- Zugang über Eigenwert-Zerlegung

Weitere Verfahren zur Berechnung von e^A

- Zugang über Eigenwert-Zerlegung
  - Seien $v_1, \ldots, v_N$ die Eigenvektoren von A zu den Eigenwerten $\lambda_1, \ldots, \lambda_N$

Weitere Verfahren zur Berechnung von e^A

- Zugang über Eigenwert-Zerlegung
  - Seien $v_1, \ldots, v_N$ die Eigenvektoren von A zu den Eigenwerten $\lambda_1, \ldots, \lambda_N$
  - Dann gilt $A = V diag(\lambda_1, \ldots, \lambda_N) V^{-1}$

- Zugang über Eigenwert-Zerlegung
  - Seien $v_1, \ldots, v_N$ die Eigenvektoren von A zu den Eigenwerten $\lambda_1, \ldots, \lambda_N$
  - Dann gilt $A = V diag(\lambda_1, \ldots, \lambda_N) V^{-1}$
  - Es folgt
    $e^A = e^{V diag(\lambda_1, \ldots, \lambda_N) V^{-1}} = V e^{diag(\lambda_1, \ldots, \lambda_N)} V^{-1}$

- Zugang über Eigenwert-Zerlegung
  - Seien $v_1, \ldots, v_N$ die Eigenvektoren von A zu den Eigenwerten $\lambda_1, \ldots, \lambda_N$
  - Dann gilt $A = V diag(\lambda_1, \ldots, \lambda_N) V^{-1}$
  - Es folgt
    $e^A = e^{V diag(\lambda_1, \ldots, \lambda_N) V^{-1}} = V e^{diag(\lambda_1, \ldots, \lambda_N)} V^{-1}$
  - Es ergibt sich $e^A = V diag(e^{\lambda_1}, \ldots, e^{\lambda_N}) V^{-1}$

Weitere Verfahren zur Berechnung von e^A

- Zugang über Eigenwert-Zerlegung
  - Seien $v_1, \ldots, v_N$ die Eigenvektoren von A zu den Eigenwerten $\lambda_1, \ldots, \lambda_N$
  - Dann gilt $A = V diag(\lambda_1, \ldots, \lambda_N) V^{-1}$
  - Es folgt $e^A = e^{V diag(\lambda_1, \ldots, \lambda_N) V^{-1}} = V e^{diag(\lambda_1, \ldots, \lambda_N)} V^{-1}$
  - Es ergibt sich $e^A = V diag(e^{\lambda_1}, \ldots, e^{\lambda_N}) V^{-1}$
  - Falls A normal ist, kann V orthogonal gewählt werden: $e^A = V diag(e^{\lambda_1}, \ldots, e^{\lambda_N}) V^T$

Berechnung von e^A über die Eigenvektoren

- Kosten des Verfahrens:

- Kosten des Verfahrens:
  - Kein Scaling & Squaring nötig

Berechnung von e^A über die Eigenvektoren

- Kosten des Verfahrens:
  - Kein Scaling & Squaring nötig
  - QR-Algorithmus zur Bestimmung von V benötigt $9N^3$ Operationen

- Kosten des Verfahrens:
  - Kein Scaling & Squaring nötig
  - QR-Algorithmus zur Bestimmung von V benötigt $9N^3$ Operationen
  - Das Produkt $V diag(e^{\lambda_1}, \ldots, e^{\lambda_N}) V^T$ erfordert $2N^3$ Operationen

- Kosten des Verfahrens:
  - Kein Scaling & Squaring nötig
  - QR-Algorithmus zur Bestimmung von V benötigt $9N^3$ Operationen
  - Das Produkt $V diag(e^{\lambda_1}, \ldots, e^{\lambda_N}) V^T$ erfordert $2N^3$ Operationen
  - Für dünnbesetztes A hängen die Kosten von der speziellen Struktur ab.

Weitere Verfahren zur Berechnung von e^A

- Splitting Methode

- Splitting Methode
  - Ausgangspunkt: Zerlegung $A = B + C$

- Splitting Methode
  - Ausgangspunkt: Zerlegung $A = B + C$
  - Die Funktionalgleichung $e^{x+y} = e^x e^y$ gilt für Matrizen im Allgemeinen nicht.

Weitere Verfahren zur Berechnung von e^A

- Splitting Methode
  - Ausgangspunkt: Zerlegung $A = B + C$
  - Die Funktionalgleichung $e^{x+y} = e^x e^y$ gilt für Matrizen im Allgemeinen nicht.
  - Es gilt lediglich die folgende Äquivalenz:

$$e^{B+C} = e^B e^C \Longleftrightarrow BC = CB$$

- Splitting Methode
  - Ausgangspunkt: Zerlegung $A = B + C$
  - Die Funktionalgleichung $e^{x+y} = e^x e^y$ gilt für Matrizen im Allgemeinen nicht.
  - Es gilt lediglich die folgende Äquivalenz:

$$e^{B+C} = e^B e^C \iff BC = CB$$

  - Es gilt jedoch die Trotter-(Produkt-)Formel:

$$e^{B+C} = \lim_{m\to\infty} \left(e^{B/m} e^{C/m}\right)^m$$

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß
- $\left(e^{B/m} e^{C/m}\right)^{2^k}$ durch wiederholtes Quadrieren

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß
- $\left(e^{B/m} e^{C/m}\right)^{2^k}$ durch wiederholtes Quadrieren
- Nur lohnenswert, falls $e^{B/m}$ und $e^{C/m}$ einfach zu berechnen sind.

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß
- $\left(e^{B/m} e^{C/m}\right)^{2^k}$ durch wiederholtes Quadrieren
- Nur lohnenswert, falls $e^{B/m}$ und $e^{C/m}$ einfach zu berechnen sind.
- Kosten des Verfahrens:

Splitting-Methode

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß
- $\left(e^{B/m} e^{C/m}\right)^{2^k}$ durch wiederholtes Quadrieren
- Nur lohnenswert, falls $e^{B/m}$ und $e^{C/m}$ einfach zu berechnen sind.
- Kosten des Verfahrens:
  - Berechnung von $e^{B/m}$ und $e^{C/m}$

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß
- $\left(e^{B/m} e^{C/m}\right)^{2^k}$ durch wiederholtes Quadrieren
- Nur lohnenswert, falls $e^{B/m}$ und $e^{C/m}$ einfach zu berechnen sind.
- Kosten des Verfahrens:
  - Berechnung von $e^{B/m}$ und $e^{C/m}$
  - Produkt $e^{B/m} e^{C/m}$ benötigt $2N^3$ Schritte

Splitting-Methode

$$e^{B+C} \approx \left(e^{B/m} e^{C/m}\right)^m$$

- Wähle $m = 2^k$ hinreichend groß
- $\left(e^{B/m} e^{C/m}\right)^{2^k}$ durch wiederholtes Quadrieren
- Nur lohnenswert, falls $e^{B/m}$ und $e^{C/m}$ einfach zu berechnen sind.
- Kosten des Verfahrens:
  - Berechnung von $e^{B/m}$ und $e^{C/m}$
  - Produkt $e^{B/m} e^{C/m}$ benötigt $2N^3$ Schritte
  - Squaring erfordert $k \cdot 2N^3$ Operationen

Bemerkungen

- Alle Verfahren machen massiv Gebrauch von Matrix-Matrix-Multiplikation

Bemerkungen

- Alle Verfahren machen massiv Gebrauch von Matrix-Matrix-Multiplikation
- Bisher nur klassische Verfahren vorgestellt

Bemerkungen

- Alle Verfahren machen massiv Gebrauch von Matrix-Matrix-Multiplikation
- Bisher nur klassische Verfahren vorgestellt
- Probleme sind dünnbesetzt, unitär und schlecht konditioniert

Bemerkungen

- Alle Verfahren machen massiv Gebrauch von Matrix-Matrix-Multiplikation
- Bisher nur klassische Verfahren vorgestellt
- Probleme sind dünnbesetzt, unitär und schlecht konditioniert
- Berechnung von e^{iA}?

Bemerkungen

- Alle Verfahren machen massiv Gebrauch von Matrix-Matrix-Multiplikation
- Bisher nur klassische Verfahren vorgestellt
- Probleme sind dünnbesetzt, unitär und schlecht konditioniert
- Berechnung von e^{iA}?
- Wie kann man die spezielle Struktur ausnutzen?

Bemerkungen

- Alle Verfahren machen massiv Gebrauch von Matrix-Matrix-Multiplikation
- Bisher nur klassische Verfahren vorgestellt
- Probleme sind dünnbesetzt, unitär und schlecht konditioniert
- Berechnung von e^{iA}?
- Wie kann man die spezielle Struktur ausnutzen?
- Unitäre Transformation auf reelle Blockgestalt

$$e^{iH} = DU \begin{pmatrix} e^{iA} & 0 \\ 0 & e^{iB} \end{pmatrix} U^T \bar{D}$$

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode
  $e^{iH} = e^{iD+iC} \approx \left(e^{iD/m} e^{iC/m}\right)^m$

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode
  $e^{iH} = e^{iD+iC} \approx \left(e^{iD/m} e^{iC/m}\right)^m$
  - iD/m ist Diagonalmatrix

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode

  $$e^{iH} = e^{iD+iC} \approx \left(e^{iD/m} e^{iC/m}\right)^m$$

  – iD/m ist Diagonalmatrix
  – Die Eigenvektoren von iC/m sind gerade die Spalten von $F_2 \otimes \cdots \otimes F_2$

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode
  $$e^{iH} = e^{iD+iC} \approx \left(e^{iD/m} e^{iC/m}\right)^m$$
  - iD/m ist Diagonalmatrix
  - Die Eigenvektoren von iC/m sind gerade die Spalten von $F_2 \otimes \cdots \otimes F_2$
- Kosten des Verfahrens:

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode
  $e^{iH} = e^{iD+iC} \approx \left(e^{iD/m} e^{iC/m}\right)^m$
  - iD/m ist Diagonalmatrix
  - Die Eigenvektoren von iC/m sind gerade die Spalten von $F_2 \otimes \cdots \otimes F_2$
- Kosten des Verfahrens:
  - $e^{iC/m} = Fdiag(e^{\lambda_1}, ..., e^{\lambda_N})F^H$: $2N^3$

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode
  $e^{iH} = e^{iD+iC} \approx \left(e^{iD/m}e^{iC/m}\right)^m$
  - iD/m ist Diagonalmatrix
  - Die Eigenvektoren von iC/m sind gerade die Spalten von $F_2 \otimes \cdots \otimes F_2$
- Kosten des Verfahrens:
  - $e^{iC/m} = F diag(e^{\lambda_1}, ..., e^{\lambda_N})F^H$: $2N^3$
  - $e^{iD/m}e^{iC/m} = diag(d_1, ..., d_N)e^{iC/m}$

Ausnutzen der speziellen Eigenschaften

- $H = H_z + (H_x + H_y) = D + C$
- Anwendung der Splitting-Methode
  $$e^{iH} = e^{iD+iC} \approx \left(e^{iD/m}e^{iC/m}\right)^m$$
  - iD/m ist Diagonalmatrix
  - Die Eigenvektoren von iC/m sind gerade die Spalten von $F_2 \otimes \cdots \otimes F_2$
- Kosten des Verfahrens:
  - $e^{iC/m} = F diag(e^{\lambda_1}, ..., e^{\lambda_N})F^H$: $2N^3$
  - $e^{iD/m}e^{iC/m} = diag(d_1, ..., d_N)e^{iC/m}$
  - Squaring: $k \cdot 2N^3$ Operationen

Offene Fragen

- Kann die Struktur von H auch bei weiteren Verfahren (Padé,QR,...) ausgenutzt werden?

Offene Fragen

- Kann die Struktur von H auch bei weiteren Verfahren (Padé,QR,...) ausgenutzt werden?
- Wie lässt sich die Matrix-Matrix Multiplikation möglichst schnell berechnen? (Strassen,...)

Offene Fragen

- Kann die Struktur von H auch bei weiteren Verfahren (Padé,QR,...) ausgenutzt werden?
- Wie lässt sich die Matrix-Matrix Multiplikation möglichst schnell berechnen? (Strassen,...)
- Wie gut sind die Verfahren parallelisierbar?