Chebyshev methods for the matrix exponential

Konrad Waldherr
Technische Universität München, Germany
Overview

- Gradient Flow Algorithm
- The structure of the matrices
- Methods of computing the exponential of a matrix
  - Taylor series
  - Padé approximation
  - eigendecomposition
  - Chebyshev series expansion
- Numerical results
  - accuracy
  - computation time
  - Speedup and efficiency
- Conclusions
Gradient Flow Algorithm

One iteration step in the Gradient Flow Algorithm

- Calculate the forward-propagation for all $t_1, t_2, \ldots, t_k$:
  \[
  U(t_k) = e^{-i \Delta t H_k} \cdot e^{-i \Delta t H_{k-1}} \cdots e^{-i \Delta t H_1}
  \]

- Compute the backward-propagation for all $t_M, t_{M-1}, \ldots, t_k$
  \[
  \Lambda^*(t_k) = e^{-i \Delta t H_k} \cdot e^{-i \Delta t H_{k+1}} \cdots e^{-i \Delta t H_M}
  \]

- Calculate the update
  \[
  \frac{\partial h(U(t_k))}{\partial u_j} = \text{Re} \left\{ \text{tr} \left[ \Lambda^*(t_k) (-i H_j) U(t_k) \right] \right\}
  \]
Numerical tasks

- Computation of the matrix exponentials
  \[ U_k := e^{-i \Delta t H_k} \]

- Computation of all intermediate products

\[
\begin{align*}
U_0 \\
U_1 \cdot U_0 \\
U_2 \cdot U_1 \cdot U_0 \\
\vdots \\
U_M \cdot U_{M-1} \cdot U_{M-2} \cdots U_2 \cdot U_1 \cdot U_0
\end{align*}
\]
Structure of the matrices $H$

- Pauli matrices

$$P_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad P_y = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \quad P_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- $H = H_d + H_c$, where

$$H_d = \sum_{k_1, k_2=0}^{q-1} \gamma_{k_1,k_2} I_2^{k_1} \otimes P_z \otimes I_2^{k_2} \otimes P_z \otimes I_2^{q-k_1-k_2-2}$$

$$H_c = \sum_{k=0}^{q-1} I_2^k \otimes (\alpha_k P_x + \beta_k P_y) \otimes I_2^{q-k-1}$$
Properties of $H$

- $H$ is sparse, hermitian and persymmetric
- $H$ has the following sparsity pattern

Transformation to real symmetric matrix $\tilde{H}$ possible:

$$H = V\tilde{H}V^*$$
Transformation of $H$

- $\tilde{H}$ can be transformed to two real blocks of half size:

$$
\begin{pmatrix}
I & J \\
I & -J
\end{pmatrix} \cdot \tilde{H} \cdot 
\begin{pmatrix}
I & I \\
J & -J
\end{pmatrix} = 
\begin{pmatrix}
H_1 & 0 \\
0 & H_2
\end{pmatrix} =
$$

- Result: 2 problems of half size
The exponential of a matrix

- **Definition:** The exponential of a matrix $H \in \mathbb{C}^{n \times n}$ is defined by the infinite Taylor series

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

- **Method 1:** Computation of the matrix exponential by using a partial sum of the Taylor series

$$ e^H \approx S_m(H) := \sum_{k=0}^{m} \frac{H^k}{k!} $$

- **Error Estimate:**

$$ \| e^H - S_m(H) \| \leq \left( \frac{\| H \|^{m+1}}{(m+1)!} \right) \left( \frac{1}{1 - \| H \| / (m+2)} \right) \leq \delta $$
Properties of the matrix exponential

- The functional equation $e^{x+y} = e^x e^y$ does in general not hold for matrices:
  \[ e^{A+B} = e^A \cdot e^B \iff A \cdot B = B \cdot A \]

  "The great matrix exponential tragedy"
  (Moler, van Loan, 1978)

- **Trotter Product Formula:**
  \[ \lim_{m \to \infty} \left( e^{A/m} e^{B/m} \right)^m = e^{A+B} \]

- \( \left( e^{A/m} \right)^m = e^A \)

- **Scaling & Squaring**
  \[ e^A = \left( e^{A/2^k} \right)^{2^k} \]
Nineteen dubious ways

Moler, van Loan: *Nineteen dubious ways to compute the exponential of a matrix*, 1978

- Series methods (Taylor, Padé)
- ODE-methods (Single step methods, Multistep methods)
- Polynomial methods (Cayley-Hamilton, Newton interpolation, ...)
- Matrix decomposition methods (Eigen decomposition, Jordan canonical form, Schur)
- Splitting methods
Nineteen dubious ways

Moler, van Loan: *Nineteen dubious ways to compute the exponential of a matrix, 25 years later*, 2003

- Series methods (Taylor, Padé)
- ODE-methods (Single step methods, Multistep methods)
- Polynomial methods (Cayley-Hamilton, Newton interpolation, ...)
- Matrix decomposition methods (Eigen decomposition, Jordan canonical form, Schur)
- Splitting methods
- Krylov methods
Nineteen dubious ways

Moler, van Loan: *Nineteen dubious ways to compute the exponential of a matrix, 25 years later*, 2003

- Series methods (Taylor, Padé)
- ODE-methods (Single step methods, Multistep methods)
- Polynomial methods (Cayley-Hamilton, Newton interpolation, ...)
- Matrix decomposition methods (*Eigen decomposition*, Jordan canonical form, Schur)
- Splitting methods
- Krylov methods

Comparison with *Chebyshev series* method
Computation of the matrix exponential: eigendecomposition

- In the case of a diagonal matrix

\[ H = \text{diag}(d_1, \ldots, d_n) = \begin{pmatrix} d_1 \\ & \ddots \\ & & d_n \end{pmatrix} \]

it holds

\[ e^H = \text{diag}(e^{d_1}, \ldots, e^{d_n}) = \begin{pmatrix} e^{d_1} \\ & \ddots \\ & & e^{d_n} \end{pmatrix} \]

- If \( H = S D S^{-1} = S \left( \text{diag}(d_1, \ldots, d_n) \right) S^{-1} \) it follows

\[ e^H = S \left( \text{diag}(e^{d_1}, \ldots, e^{d_n}) \right) S^{-1} \]

- **Expensive part:** Computation of the eigen decomposition
Computation of the matrix exponential: Padé approximation

- For $x \in \mathbb{C}$ the Padé approximation $r_m(x)$ of $e^x$ is given by

$$r_m(x) = \frac{p_m(x)}{q_m(x)}$$

- $p_m(x) = \sum_{j=0}^{m} \frac{(2m-j)!m!x^j}{(2m)!(m-j)!j!}$

- $q_m(x) = \sum_{j=0}^{m} \frac{(2m-j)!m!(-x)^j}{(2m)!(m-j)!j!}$

- Generalization to matrices:

$$e^H \approx r_m(H) = (q_m(H))^{-1} p_m(H)$$

- Combination with Scaling & Squaring:

$$e^H \approx \left[ q_m\left(\frac{H}{2^k}\right) \right]^{-1} p_m\left(\frac{H}{2^k}\right)^{2^k}$$
Computation of the matrix exponential: Padé approximation

- Padé approximation:

\[ e^H \approx r_m(H), \text{ where } q_m(H)r_m(H) = p_m(H) \]

- Suppose \( m = 2l + 1 \) odd:

\[
p_{2l+1}(H) = b_{2l}H^{2l} + \cdots + b_2H^2 + b_0I
\]

\[
= : U + H \left( b_{2l+1}H^{2l} + \cdots + b_3H^2 + b_1I \right)
\]

\[
= : V
\]

- Because of \( p_m(-x) = q_m(x) \) we obtain \( q_m(H) = U - V \)

- Expensive part: Linear system \( (U - V) \cdot r_m(H) = U + V \)
Computation of the matrix exponential: Chebyshev series expansion

- For $|x| \leq 1$ we have

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

- $J_k$: Bessel function

$$J_k(z) = \sum_{l=0}^{\infty} \frac{(-1)^l}{2^{2l+k} l! (k + l)!} z^{2l+k}$$

- $T_k$: Chebyshev polynomial

$$T_0(x) = 1,$$
$$T_1(x) = x,$$
$$T_{k+1}(x) = 2x T_k(x) - T_{k-1}(x).$$
Computation of the matrix exponential: Chebyshev series expansion

• Generalization to matrices: For $\|H\| \leq 1$

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

• In the case of $H$ with arbitrary norm:
  Scaling & Squaring technique:

$$e^H \approx \left[ J_0(i)I + 2 \sum_{k=1}^{m} i^k J_k(-i) T_k \left( \frac{H}{2^k} \right) \right]^{2^k}$$

• $\implies$ Only elementary matrix operations required!
Evaluation of the matrix polynomial

- **Sparse case:** \( T_{k+1}(H) = 2HT_k(H) - T_{k-1}(H) \)
  \[ \implies \text{Only products of the form Sparse * Dense} \]

- **General (dense) case:** Paterson, Stockmeyer, 1973
  Efficient evaluation of matrix polynomials
Evaluation of the matrix polynomial

- **Sparse case:** \( T_{k+1}(H) = 2HT_k(H) - T_{k-1}(H) \)
  \[ \implies \text{Only products of the form Sparse * Dense} \]

- **General (dense) case:** Paterson, Stockmeyer, 1973
  Efficient evaluation of matrix polynomials

  **Example:**
  \[
p(A) = \alpha_6 A^6 + \alpha_5 A^5 + \alpha_4 A^4 + \alpha_3 A^3 + \alpha_2 A^2 + \alpha_1 A + \alpha_0 I
  \]

  - **Horner scheme** requires **5** non-scalar multiplications:
    \[
    (((((\alpha_6 A + \alpha_5 I) \cdot A + \alpha_4 I) \cdot A + \alpha_3 I) \cdot A + \alpha_2 I) \cdot A + \alpha_1 I) \cdot A + \alpha_0 I
    \]

  - **Optimal method:** (Only **3** matrix-matrix-products required)
    \[
    A_2 = A \cdot A, \quad A_3 = A_2 \cdot A
    \]
    \[
p(A) = \alpha_0 I + \alpha_1 A + \alpha_2 A_2 + \alpha_3 A_3 + A_3 \cdot (\alpha_4 A + \alpha_5 A_2 + \alpha_6 A_3)
    \]
Comparison of the methods: Computation time
Comparison of the methods: accuracy

- Chebyshev method
- Pade method
- Eigendecomposition method

Norm of the error vs Number of spins.
Numerical results on parallel platform

- System size: 2048-by-2048
- Using Intel MKL BLAS and LAPACK libraries
- Platform:
  - Dual-Socket Intel Xeon
  - X5355 processors (Quad-Cores, 2.66 GHz and 2x4 MB Level Cache, Core 2 Duo)
  - 6 GB RAM
Computation time on parallel architecture

Computation time in seconds vs. number of processors for different methods:
- Padé method
- Eigen decomposition method
- Chebyshev method

The graph shows the decrease in computation time as the number of processors increases for each method.
Speedup

- Padé method
- Eigen decomposition method
- Chebyshev method

Number of processors

![Graph showing speedup comparison for different methods with varying number of processors.]
Final conclusions

- Computation of the matrix exponential an intrinsically hard problem
- Chebyshev method very efficient (accuracy and computation time)
  - Good convergence properties
  - Only elementary matrix operations required
  - Efficient evaluation of matrix polynomials possible
  - Good in parallel
Joint work

Joint work with

- Thomas Huckle (Computer science department, TU München)
- Andreas Spörl, Thomas Schulte-Herbrüggen (Chemistry department, TU München)

Thank you very much for your attention!