

高次トロッター分解

背景知識

$$H = A + B$$

$$\text{通常のトロッター分解: } \exp((A+B)t) = \exp(At) \exp(Bt) + O(t^2)$$

トロッター分解の一般化

$$1\text{次の分解: } \exp((A+B)t) = \exp(At) \exp(Bt) + O(t^2)$$

$$2\text{次の分解: } \exp((A+B)t) = \exp(At/2) \exp(Bt) \exp(At/2) + O(t^3)$$

(両辺を t の2次までテーラー展開して係数を比較することで簡単に示せる.)

2次の積公式を以下で定義

$$S_2(t) := \exp(At/2) \exp(Bt) \exp(At/2)$$

一般に k 次の積公式 $S_k(t)$ が分かっているとすると

$$\exp((A+B)t) = S_k(t) + O(t^{k+1}) \quad (\text{ここで } S_k(t) = \prod_{j=1}^l \exp(c_j At) \exp(d_j Bt))$$

長い時間のシミュレーションを行うとき, 短い時間に分割して繰り返す

→繰り返し回数は次数の増加に伴って減少

→高次積公式の探索(c_j, d_j を見つけること)のモチベーション

Suzuki's fractal method

元論文

- 「Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, M. Suzuki, Physics Letters A, 1990」
- 「General theory of fractal path integrals with applications to many-body theories and statistical physics, M. Suzuki, Journal of Mathematical Physics, 1991」

任意の高次積公式の系統的な導出手法

論文内ではより一般的な形での定式化 (Theorem 1近辺, $r=3$ のとき)

$$k(=2\kappa)\text{次積公式: } S_{2\kappa}(t) = S_{2\kappa-2}(s_\kappa t) S_{2\kappa-2}((1-2s_\kappa)t) S_{2\kappa-2}(s_\kappa t)$$

としたとき $s_\kappa = 1/(2-2^{1/(2\kappa-1)})$ (←これが高次積公式探索の解)

S_2 を用いて偶数次に対して再帰的に定義

$$S_{2m}(t) = S_{2m-1}(t)$$

3次のときの簡単な証明

(準備)

$$S_2(t) = \exp((A+B)t + O(t^3)) = \exp((A+B)t + R_2 t^2 + R_3 t^3 + \dots) \text{とする (誤差は指数の肩に載せられることを用いた).}$$

$$S_2(t) \text{ と } S_2(-t) \text{ は可換なので } (e^A e^B = e^{(A+B)}) \text{ を用いると}$$

$$S_2(t) S_2(-t) = \exp((A+B)t + R_2 t^2 + R_3 t^3 \dots - (A+B)t + R_2 t^2 - R_3 t^3 \dots) = \exp(2(R_2 t^2 + R_4 t^4))$$

$$\text{また } S_2(t) S_2(-t) = I$$

$$\text{よって } R_2 = R_4 = \dots = 0$$

(証明)

$$S_4(t) = S_2(s_2 t) S_2((1-2s_2)t) S_2(s_2 t) = \dots = \exp((A+B)t + [2s^3 + (1-2s)^3] R_3 + O(t^5))$$

$$2s^3 + (1-2s)^3 = 0 \text{ を解くと } s = 1/(2-\sqrt[3]{2})$$

$$\text{デメリット: } s_\kappa, 1-2s_\kappa > 1 \rightarrow \text{誤差大}$$

デメリットを改善した公式

$$k(=2\kappa)\text{次積公式: } S_{2\kappa}(t) = S_{2\kappa-2}(u_\kappa t)^2 S_{2\kappa-2}((1-4u_\kappa)t) S_{2\kappa-2}(u_\kappa t)^2$$

$$\text{としたとき } u_\kappa = 1/(4-4^{1/(2\kappa-1)})$$



Yoshida's method

「Construction of higher order symplectic integrators, H. Yoshida, Physics Letters A, 1990」

- Suzukiの手法では指数の数が急激に増える→指数の数を抑える手法
- この手法では高次積公式のexplicitで解析的な形式がないため、複雑な連立非線形多項式方程式を導出して解く必要がある

Approach

$$S^{(m)}(t) = (\prod_{j=1}^m S_2(w_{m-j+1}t)) S_2(w_0t) (\prod_{j=1}^m S_2(w_jt))$$

task : $S^{(m)}$ が k 次の積公式になるような m と w_i を見つける

背景知識 : Symmetric Baker-Campbell-Haussdorff公式

$$|X| + |Y| < \ln 2, \ t \in \mathbb{R}, \ S_2(t) = \exp(Z) \text{としたとき以下が成立}$$

$$Z = \sum_{l=1}^{\infty} \alpha_l t^l$$

ここで

$$\alpha_{2n} = 0 \text{ (証明は上述済)}$$

$$\alpha_1 = X + Y$$

$$\alpha_3 = \frac{1}{12}[Y, [Y, X]] - \frac{1}{24}[X, [X, Y]]$$

$$\alpha_5 = \frac{7}{5760}[X, X, X, X, Y] - \frac{1}{720}[Y, Y, Y, Y, X] + \frac{1}{360}[X, Y, Y, Y, X] + \frac{1}{360}[Y, X, X, X, Y] - \frac{1}{480}[X, X, Y, Y, X] + \frac{1}{120}[Y, Y, X, X, Y]$$

例 $[Y, Y, X, X, Y] \equiv [Y, [Y, [X, [X, Y]]]]$

6次の積公式の場合

BCH公式を用いて $m = 1$ を計算してみる

$$\begin{aligned} &S_2(w_1\ t)\ S_2(w_0\ t)\ S_2(w_1\ t) \\ &\&= \exp \left\{ t\ w_1\ \alpha_1 + t^3\ w_1^3\ \alpha_3 + t^5\ w_1^5\ \alpha_5 + \mathcal{O}(t^7) \right\} \exp \left\{ t\ w_0\ \alpha_1 + t^3\ w_0^3\ \alpha_3 + t^5\ w_0^5\ \alpha_5 + \mathcal{O}(t^7) \right\} \\ &\&\dots \\\ &\&= \exp \left\{ t(2w_1 + w_0)\ \alpha_1 + t^3(2w_1^3 + w_0^3)\ \alpha_3 + t^5(2w_1^5 + w_0^5)\ \alpha_5 + t^5\frac{1}{6}(w_0^2\ w_1^3 - w_1^2\ w_0^3 + w_1^4\ w_0 - w_0^4\ w_1) \right\} \end{aligned}$$

ここで $\beta_5 = [\alpha_1, \alpha_1, \alpha_3]$

※この時点で($n = 1$ までの計算で)4次積公式は求められるが6次積公式は求められない

なぜか

これが6次積公式であるためには

$$(2w_1 + w_0) = 1, (2w_1^3 + w_0^3) = (2w_1^5 + w_0^5) = (w_0^2w_1^3 - w_1^2w_0^3 + w_1^4w_0 - w_0^4w_1) = 0$$

である必要があるが、変数 w_0, w_1 が2つ、非線形方程式が4つなので一般に解を持たない

これが4次積公式であるためには

$$(2w_1 + w_0) = 1, (2w_1^3 + w_0^3) = 0$$

を満たす w_0, w_1 を求めれば良い.

ちなみに $w_0 = -\frac{2^{1/3}}{2-2^{1/3}}, w_1 = \frac{1}{2-2^{1/3}}$ であり、4次積公式が求められたことになる.

6次積公式を求めるためには m をさらに増やす必要がある.

一般に m を大きくして(t^7 のオーダーまで)計算すると以下ようになる(帰納法的に示される).

$$S^{(m)}(t) = \exp tA_{1,m}\alpha_1 + t^3A_{3,m}\alpha_3 + t^5(A_{5,m}\alpha_5 + B_{5,m}\beta_5)) + O(t^7)$$

ここで $A_{j,m}, B_{5,m}$ は w_i の多項式であり、6次積公式を求めるためには $A_{1,m} = 1, A_{3,m} = 0, A_{5,m} = 0, B_{5,m} = 0$ を解くことで w_i を求めることになる. 方程式が4本あるため未知数が4個であればよい、つまり $m = 3$ まで計算したとしたときに解が求められる.

$$w_0 = 1 - 2 \sum_j w_j \text{(証明の過程で簡単に示される)}$$
を用いると計算が少し楽になる.

3つの解が発見されている(これ以上ないと言われている)

Table 1
6th order symplectic integrators.

	Solution A	Solution B	Solution C
w_1	−0.117767998417887E1	−0.213228522200144E1	0.152886228424922E−2
w_2	0.235573213359357E0	0.426068187079180E−2	−0.214403531630539E1
w_3	0.784513610477560E0	0.143984816797678E1	0.144778256239930E1

Processed product formulae

参考文献が多すぎて追いきれていないが1996–2006あたりに開発された手法

symmetricではない

$P\Sigma P^{-1}$ (Σ : kernel, P : processor)

P と P^{-1} は、時間発展によって積をとるとキャンセル→コストはkernel依存→kernelのexpの数はこれまでの手法より軽い

典型的には Σ は k より低次, P は k 次

Σ のパラメータを見つけ, P は既に知られているものを用いるので, 解くべき非線形方程式の数が少ない

詳細は省略

Greatly improved higher-order product formulae for quantum simulation

[Mauro E. S. Morales+, arXiv, 2024]

8, 10次の改善された積公式を見つけ, より正しく性能を比較する手法を導出

積公式を系統的に得るようなテクニカルな手法を編み出したというより, こうやったら計算をざり押せるというのを見つけて, 何千もの新しい解を見つけまくったという印象. そして性能を比較することで質の高い積公式を見つけたというもの.

yoshida's methodで8次のとき最小の m は7, 10次のときは15.

大きい m をとるほどエラーを減らせる

手法1

連立非線形方程式を解くためにMatlabのfsolve(Levenverg-Marquardtアルゴリズム)を用いるとより高速に解が得られる. 小さい値を取る解ほど性能が高い. 大きな値を解は捨てる.

手法2

テイラー展開を用いる手法. 正確な $\exp(t(X+Y))$ と, w_j で置いた積公式 $S^{(m)}(t, w_1, \dots, w_m)$ を適当な次数までテイラー展開し, 係数を比較した等式を解くことで解 w_j を得る. (本当に今まで誰も思いつかなかったのか?)

具体的には以下ようになる ($T_k[\cdot]$ は $t=0$ の周りで k 次までテイラー展開をする写像)

正確な $\exp(t(X+Y))$ のテイラー展開

$$T_k[e^{t(X+Y)}] = \sum_{p=0}^k \frac{t^p}{p!} (X+Y)^p = \sum_{p=0}^k \frac{t^p}{p!} \sum_{r_1, \dots, r_p=0}^1 X^{r_1} Y^{1-r_1} \dots X^{r_p} Y^{1-r_p}$$

Yoshida'sは以下のように書ける.

$$S^{(m)}(t, w_1, \dots, w_m) = e^{tw_m X/2} e^{tw_m Y} e^{t(w_m+w_{m-1})X/2} e^{t(w_{m-1})Y} e^{t(w_{m-1}+w_{m-2})X/2} \dots e^{tw_0 Y} e^{t(w_1+w_0)X/2} e^{tw_1 Y} \dots e^{tw_m X/2} = e^{tc_1 X} e^{tc_2 Y} \dots e^{tc_{4m+3} X}$$

ここで $c_1 = w_m/2, c_2 = w_m, c_3 = (w_m + w_{m-1})/2, \dots, c_{4m+3} = w_m/2$

これをテイラー展開すると以下ようになる.

$$T_k[S^{(m)}(t, w_1, \dots, w_m)] = \sum_{r_1, \dots, r_{4m+3}=0, r_1+\dots+r_{4m+3} \leq k}^{4m+3} \frac{t^{r_1+\dots+r_{4m+3}}}{r_1! \dots r_{4m+3}!} c_1^{r_1} \dots c_{4m+3}^{r_{4m+3}} X^{r_1} Y^{r_2} \dots X^{r_{k-1}} Y^{r_{4m+3}}$$

係数を自動的に管理するためのデータ構造を構成

結果

8次

[S. Blanes+, Mathematics of Computation, 2013]などで用いられた性能比較: エラー $\delta(t)$ は $O(t^8)$ なので $\delta(t) = \chi t^8$ と書ける(χ は定数). random Hamiltonianに対して, χ を計算する. エラーはスペクトルノルム(得られた行列の差)や固有値の差を用いて計算.

$m=7$ のときの8次の解は以下

	Best 8th order with $m = 7$	Best 8th order with $m = 8$
w_1	0.315293092396766596632056663811	0.29137384767986663096528500968049
w_2	0.33462491824529818378495797988218	0.26020394234904150277316667709864
w_3	0.2990641813036559238444635406886	0.18669648149540687549831902999911
w_4	−0.57386247111608226665638772663554	−0.40049110428180105319963667975074
w_5	0.19075471029623837995387625645037	0.15982762208609923217390166127256
w_6	−0.40910082580003159399730009589356	−0.38400573301491401473462588779099
w_7	0.74167036435061295344822780178381	0.56148845266356446893590729572808
w_8	NA	0.12783360986284110837857554950443

TABLE I. Our best-performing 8th order solutions with $m = 7$ and 8.

ここで $\chi = 5.8 \times 10^{-6}$. yoshidaの解で最も良かったもので $\chi = 9.7 \times 10^{-4}$

$m = 8$ のときはunderdetermined(未知数よりも方程式の数が少ない)な方程式になる. 連続な解を持つため, 手法1(恐らく)を使えばエラーを減らせる

yoshida形式だけでなく, prosessed形式に対しても解を探索している.

10次に対しては今まで見つけられていなかった解を数多く発見

性能評価手法の提案

とりあえず保留