



高度な N 体シミュレーション法

高精度時間積分、タイムステップの工夫

台坂博

一橋大学

目次

- 導入
- エルミート積分法
- タイムステップの工夫

これらは特に衝突系を扱うときに必要となるテクニック
例: 球状星団、惑星集積、惑星リングなどなど

導入

- 今回の実習では2次リープフロッグ・すべての粒子が同じ時間刻みで固定、そこそこつかえるが...
- 実際の研究では、
 - 精度良く計算したい: 信頼される結果のため
 - 長時間の進化を調べたい: でも計算時間は短く、精度は良くしたい
 - より大粒子数を扱いたい: いま扱える粒子数は実際の系にある粒子数よりずっと少ない

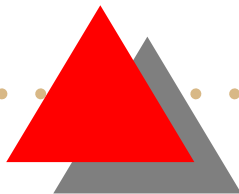
何らかの工夫が必要



例えば計算精度

- 2次リープフロッグでも、タイムステップの刻幅を小さくすれば計算精度はあがるはず(ある程度までは)
- でも、あまり得策ではない
 - 高精度を達成するのに時間に刻幅をすごく細かくする必要がある
 - 打ち切り誤差や丸め誤差の影響により精度があまり良くならない

高精度積分を使う



エルミート法

- 牧野・アーセス法が正式名称?
- 予測子・修正子法、よく使われるのは4次精度
- 力の導関数も使って補間多項式を構成, アダムス法などのように数ステップ昔の情報を必要としない
 - 出発公式が不要
 - 実装が比較的簡単
- 現在、**GRAPE** を用いて衝突系のシミュレーションを行なう際の標準的な積分法

以下、4次の公式の具体的な計算手順の説明

計算手順の簡単な説明

時刻 t で $(\mathbf{x}_0, \mathbf{v}_0)$ の粒子を考える

- 加速度、加速度の微分 $\mathbf{a}_0, \dot{\mathbf{a}}_0$ (力、力の微分) を求める

重力の場合、力の1階微分くらいなら難しくない

$$f_{ij} = -Gm_j \frac{\mathbf{r}_{ij}}{r_{ij}^3},$$

$$\dot{j}_{ij} = -Gm_j \left[\frac{\mathbf{v}_{ij}}{r_{ij}^3} - \frac{3(\mathbf{v}_{ij} \cdot \mathbf{r}_{ij})\mathbf{r}_{ij}}{r_{ij}^5} \right].$$

計算手順の例 (続き)

- 位置、速度の予測子 \mathbf{x}_p 、 \mathbf{v}_p の計算
加速度の微分まで使ったテイラー展開で与えられる。

$$\mathbf{x}_p = \frac{\Delta t^3}{6} \dot{\mathbf{a}}_0 + \frac{\Delta t^2}{2} \mathbf{a}_0 + \Delta t \mathbf{v}_0 + \mathbf{x}_0$$

$$\mathbf{v}_p = \frac{\Delta t^2}{2} \dot{\mathbf{a}}_0 + \Delta t \mathbf{a}_0 + \mathbf{v}_0,$$

\mathbf{x}_0 , \mathbf{v}_0 , \mathbf{a}_0 および $\dot{\mathbf{a}}_0$ は前の時刻での位置、速度、加速度と加速度の微分。 Δt は時間刻み $t_1 - t_0$ 。

- 上の予測子から加速度、加速度の微分 (\mathbf{a} , $\dot{\mathbf{a}}$) を計算

計算手順の例 (続きの続き)

- エルミート補間多項式から高次の展開係数を計算

$$\mathbf{a}_0^{(2)} = \frac{-6(\mathbf{a}_0 - \mathbf{a}_1) - \Delta t(4\dot{\mathbf{a}}_0 + 2\dot{\mathbf{a}}_1)}{\Delta t^2},$$

$$\mathbf{a}_0^{(3)} = \frac{12(\mathbf{a}_0 - \mathbf{a}_1) - 6\Delta t(\dot{\mathbf{a}}_0 + \dot{\mathbf{a}}_1)}{\Delta t^3}.$$

- 速度、位置の修正子を計算する

$$\mathbf{x}_c = \mathbf{x}_0 + \frac{\mathbf{a}_0^{(2)}}{24} \Delta t^4 + \frac{\mathbf{a}_0^{(3)}}{120} \Delta t^5,$$

$$\mathbf{v}_c = \mathbf{v}_0 + \frac{\mathbf{a}_0^{(2)}}{6} \Delta t^3 + \frac{\mathbf{a}_0^{(3)}}{24} \Delta t^4.$$

補足

- 独立時間刻みと合わせるともう少し複雑
詳しくは Makino and Aarseth 1992 を参照
- 実は対称型公式
 - 反復を繰り返すと良い事がある → 永年誤差の軽減
 - 中心重力場中の運動では反復を繰り返す際に相互重力を使い回すなどして計算時間を短縮 (Kokubo and Makino)
- GRAPE-6 は $\dot{\mathbf{a}}$ もハードで計算、さらには予測子も計算 → エルミート法が GRAPE に適している理由



タイムステップの工夫

粒子によって非常に大きく軌道の時間スケールが
違うことが起こった場合

連星ができるとか、ある粒子の離心率が大きくなるとか

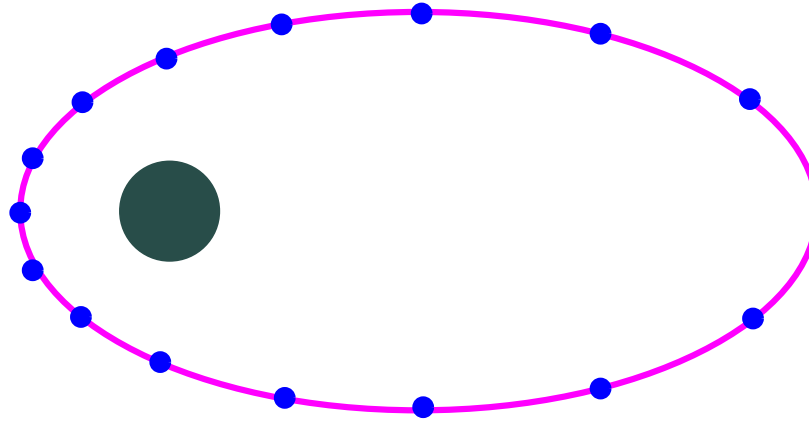
この場合に固定時間刻みを使うと...

- ちゃんと計算をするためには、そのことを見越して短い時間刻みを用いる必要あり
- 無駄にステップ数を要して非効率

時間刻みに何らかの工夫が必要

対処 1: 可変時間刻み

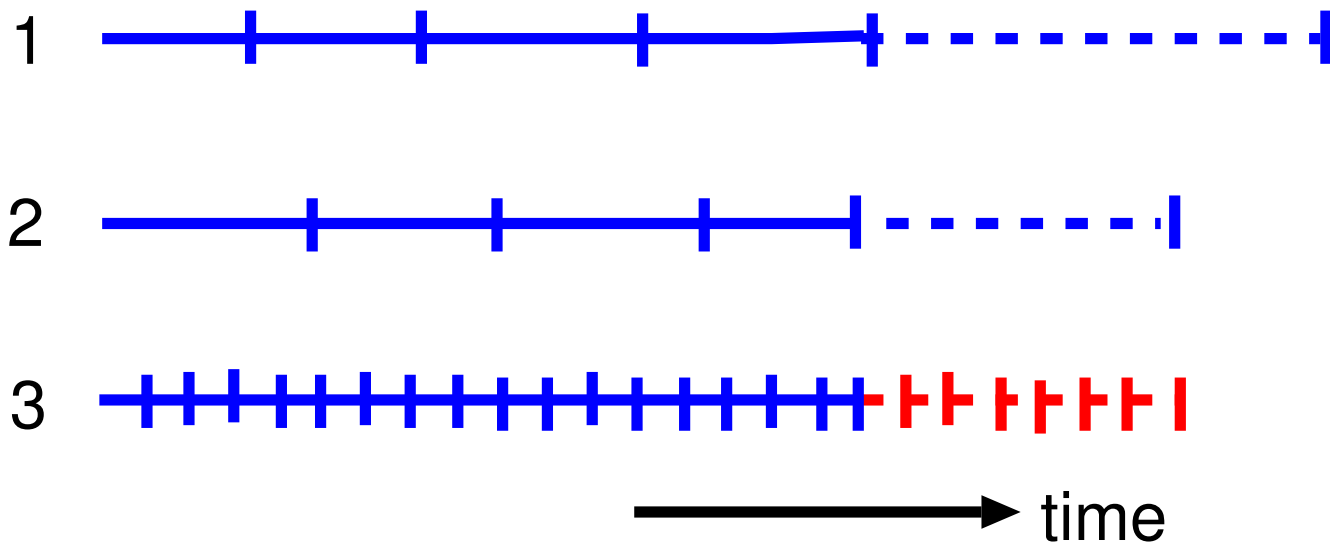
- 必要に応じて短い時間刻みを用いる



- 良く用いられる公式: アーセス公式

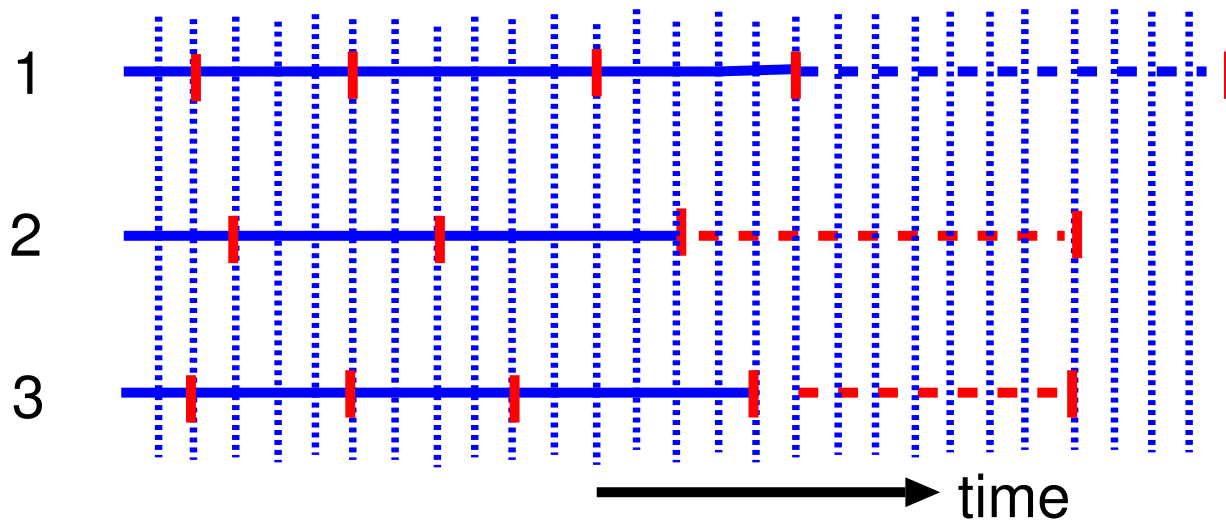
$$\delta t = \sqrt{\eta \frac{|F||F^{(2)}| + |\dot{F}|^2}{|\dot{F}||F^{(3)}| + |F^{(2)}|^2}}$$

対処 2: 独立時間刻み



- 粒子によって軌道の時間スケールが違う場合、上の方法では効率よく計算できない。粒子毎に時間刻みをバラバラに変化させる。時刻 $t_i + \delta t_i$ が最小の粒子を選んで、その粒子のみを update する。

対処 3:階層化



- 時間を 2^n に分割し粒子をまとめる
- 計算した予測子を使い回す、GRAPE の効率的利用のために必要。

実際はこれらを組み合わせ:階層化独立時間刻みとか

GRAPEへの実装

1. j 粒子のデータ $(x_j, v_j, a_j, \dot{a}_j, m_j, t_j)$ を GRAPE に送る
2. $t + dt$ が最小の粒子 (i 粒子) を選択
3. i 粒子の予測子を計算
4. i 粒子の x_p, v_p を GRAPE に送る, 計算
5. i 粒子の a_p, \dot{a}_p を GRAPE からもらう
6. $a, \dot{a}, a_p, \dot{a}_p$ を使って i 粒子の修正子を計算 (積分)
7. j 粒子の中の i 粒子に対応するデータを更新
8. 2. に戻る



まとめ

- 高精度積分法 (エルミート法)
- 時間刻みの工夫

を簡単に解説した。

これらは、衝突系など高精度が必要なシミュレーションを行なう際に役にたつテクニックであろう。