

プラズマ数値計算¹

陰山 聡² (KAGEYAMA Akira)

神戸大学 大学院システム情報学研究所 (Graduate School of System Informatics, Kobe University)

(2011年07月17日)

1 数値計算とは

1.1 計算機の世界

最近(2011年6月)、日本のスーパーコンピュータが世界一になったというニュースが流れた。スーパーコンピュータ「京」が世界一速い計算機として認められたという内容のニュースで、最近落ち込み気味の感がある日本人の科学技術に関する自信や誇りといったようなものを回復してくれる材料として世間でも好意的に取り上げられた。

サッカーのワールドカップやオリンピックは4年に一度開催されるが、スーパーコンピュータの開発競争に関しては、スポーツの世界にもまして熾烈なスピード競争が繰り広げられており、1年に2度という早いペースでスピードコンテストが開催されている。“Top500 supercomputer sites”というリスト(<http://www.top500.org/>)がそれで、このウェブサイトを見れば、現在最も早いスーパーコンピュータ500台のリストを見ることができただけでなく、1993年以降のデータを詳しく見ることできる。

京コンピュータが最新(2011年6月)のTop500リストのトップを飾ったというのが冒頭に触れたニュースであった。では、このリストの「速さ」とは何の速さをはかっているのであろうか?そしてその速さの単位は何であろうか?もちろん、測られているのはワープロソフトの処理速度といったものではない。スーパーコンピュータは、科学技術計算のための道具なのでそれに関する計算の速度であるはずである。だが、科学技術計算と言っても流体計算や量子化学計算、建築物の構造解析計算など、様々な種類の計算がある。個人競技のスポーツでも、水泳や陸上競技、テニスのシングルなど、様々な競技がある。半年に一度、「現在、世界でもっともスポーツのできる人500人」を選ぶというのは難しいであろう。

Top500リストに乗っているスーパーコンピュータで比較されているのは、スポーツの競技種目そのものの成績ではなく、むしろ「基礎体力」のようなものに近い。それは四則演算の速度である。つまり、足し算、引き算、かけ算、割り算という小学校で習う4つの演算を、1秒あたり何回できるか、という基準で世界中のスーパーコンピュータの速度を比較しているわけである。

基礎体力の高さがそのまま特定のスポーツ競技の巧さに直結するわけではないのと同様に、四則演算の速さがそのまま直接、科学技術用の計算機としての能力につながるわけではない。しかし、これから述べるように、最新のスーパーコンピュータが持つ四則演算の速度は、他の装置や技術では比較できるようなものが簡単に見つからないほど突出している。その驚異的な四則演算能力の速さをうまく活かすことで、様々な科学技術の問題を解決することができるのである。数値計算とは一言で言えば、解決すべき課題、難問を、四則演算の問題に焼き直すことと言える。いったん四則演算の問題になれば、それがいかに大量の演算回数を必要とすることになろうとも、計算機が持つ驚異的な四則演算能力によって短時間にそれを処理することができる。

1.2 数値計算の必要性

人類がデジタル計算機を手にする以前、暗算と筆算と算盤しかない時代に、何かの数理問題—たとえば力学の3体問題—の解を見つけなければいけない状況を想像してみよう。誰かが「こうすればこの問題はただの四則演算の問題に焼き直せるぞ。」と言っても、それが例えば10億回の四則演算を必要とするのであれば誰も興味を示さなかつたであろう。ところが、現在のパソコンであれば、10億回の四則演算をわずか1秒で計算してしまう。(これがなぜ1秒と評価できるのかは後で説明する。)1秒で答えを得ることができるのであれば、パソコンを使った四則演算というのは実用的な問題解決手段であると言えるであろう。

¹第50回 プラズマ若手夏の学校(高野山大学)2011年08月09日 講義テキスト

²kage@cs.kobe-u.ac.jp

いわゆる数式処理ソフトを用いれば、計算機で微分や積分なども直接扱える。たとえば

$$\frac{du}{dx} = x^2 + u^2$$

という微分方程式を皆さんは解くことができるであろうか？すぐには解けないのではないかと思うが、これを Mathematica のような数式処理ソフトに入力すると、瞬時に

$$u(x) = -\frac{x \left(\Gamma\left(\frac{1}{4}\right) J_{-\frac{3}{4}}\left(\frac{x^2}{2}\right) + 2J_{\frac{3}{4}}\left(\frac{x^2}{2}\right) \Gamma\left(\frac{3}{4}\right) \right)}{J_{\frac{1}{4}}\left(\frac{x^2}{2}\right) \Gamma\left(\frac{1}{4}\right) - 2J_{-\frac{1}{4}}\left(\frac{x^2}{2}\right) \Gamma\left(\frac{3}{4}\right)}$$

という解析解を教えてくれる。ここで $\Gamma(x)$ はガンマ関数、 $J_\nu(x)$ は第一種ベッセル関数である。数式処理ソフトの力を馬鹿にできない。だが、数式処理ソフトも万能ではない。流体の基本方程式である Navier-Stokes 方程式を Mathematica に入力すると Navier-Stokes 方程式の解析解を返してくれるというわけではないのだ。(もしも返してきたら、大事件だ。)紙とペン(と頭)では解けず、数式処理ソフトでも解けないような数理問題を計算機に解かせる。それが数値計算、あるいはもっと広く言えば計算機シミュレーションという方法である。

1.3 離散化

上で述べたとおり数値計算とは、数理モデル(その多くは微分方程式)として定式化された科学技術上の様々な課題を、四則演算の問題に焼き直すことと言える。その焼き直しには離散化と呼ばれるプロセスが含まれる。解くべき方程式(たとえば拡散方程式)には、様々な変数が現れているはずである。話を具体的にするために、時間 t 、空間座標 (x, y, z) 、そして時間と空間の関数としてのある物理量 $f(x, y, z, t)$ が現れる方程式

$$\frac{\partial f}{\partial t} = \nabla^2 f(x, y, z, t) \quad (1)$$

つまり拡散方程式を考えよう。この方程式の解析解(グリーン関数)は既知であるが、いまはそれを知らないふりをして、どうしても計算機で解く必要があるとしよう。この方程式を四則演算の問題に「焼き直す」ためには、離散化が必要がある。

時間 t と空間位置 (x, y, z) は実数、すなわち無限桁の小数である。実数を直接計算機で扱うのは難しいし、そもそも現実世界では無限に高い精度の解は不要である。そこで数値計算では、時間と空間、そして物理量の値を有限桁の小数で表現する。その桁数は2進数で52桁、10進数では15桁から16桁に相当する。($2^{52} \sim 4.5 \times 10^{15}$)。10の-15乗の精度というのは現実問題に適用するには十分小さな数である。たとえば地球半径と同じ程度の長さ $R = 6400\text{km}$ の半径を持つ円を紙の上に描くことを考える。(そのような紙があるかどうかは別にして。)その円周の長さ $L = 2\pi R$ を計算するのに、無限の桁数をもつ正しい円周率 $\pi = 0.314159265358979323846264338 \dots \times 10^1$ ではなく、 π には近いが、16桁目以降はすべてゼロとした $\pi' = 0.314159265358979000000 \dots \times 10^1$ を使って計算した $L' = 2\pi R$ の差は $L - L'$ は原子サイズである。このような誤差を気にすることは非現実的である。2進数で52桁(つまり52ビット)の仮数、そして11ビットの指数と1ビットの符号で表現した小数を倍精度浮動小数点数という。今日の多くの科学技術計算では、実数変数をこの倍精度浮動小数点数で表現する場合が多い。

数学的には、上の方程式(1)を解くというのは、任意の実数時間 t と実数位置 (x, y, z) における f の値を知ることである。 t と (x, y, z) を倍精度浮動小数点数で表現したとしても、その全ての時空点 (x, y, z, t) における解を知る必要は現実にはない。時刻が 10^{-15} 秒だけ、位置が 10^{-15} メートルだけ違う二つの時空点における f の値はたいした違いを持たないであろうからである。時間と空間は 10^{-15} よりももっとおおざっぱな間隔で、ある程度「とびとび」の時空点ごとの解が分かれば十分である。これが時間と空間の離散化である。たとえば t_a から t_b の間にある任意の時刻 $t: t_a < t < t_b$ における解を求めるかわりに、求めたい時間領域を有限の幅で切り刻んで、 $t = t_a, t_1, t_2, t_3, \dots, t_{N-1}, t_b$ という有限の数の離散的な時刻における解を求める、というのが時間の離散化であり、空間の離散化についても同様である。

時間や空間の刻み幅は $\Delta t, \Delta x, \Delta y, \Delta z$ などと書かれる場合が多い。これらは常に一定の場合もあるし、そうでない場合もある。空間方向の刻み幅 Δx 等は格子間隔と呼ばれることがある。本来実数として連続的に広がる x, y, z の空間を、有限本数の直線で格子状に分割することに相当するからである。

時間と空間の座標値が取り得る値はこのように離散化するのが普通であるが、同様に本来実数値をもつ物理量 f のとる値も同じように有限の幅で離散化するというのは自然な発想であろう。その

ような方法は超離散化と呼ばれ、最近急速に発展している新しい方法である。だが、これまでの伝統的な数値計算法、あるいはシミュレーションでは、このような超離散化は行わないのが普通である。つまり求めるべき物理量 f を倍精度浮動小数点数で表現し、その範囲でどの値をとるかは指定しない。このテキストでもこの伝統的な（通常の）離散化手法について述べる。

1.4 速度の単位

数値計算技術の具体的な話に行く前に、Top500 リストで測られている計算機の速度の単位について述べておこう。その速さの単位は FLOPS (Floating Point Operations Per Second) である。Floating Point Operations というのは、「浮動小数点数演算」の意味で、もちろんここでいう演算とは四則演算のことである。つまり FLOPS とは、10 進 15、6 桁の小数の四則演算を、1 秒あたり何回実行できるか、という単位である。最近のパソコンに搭載された CPU はだいたい 1 GFLOPS (ギガフロップス) の性能を持つ。これは 1 秒間に 10^9 回 (つまり 10 億回) の四則演算ができることを意味する。京コンピュータは、8 PFLOPS (ペタフロップス) の記録を出し、世界一になった。これは 1 秒間に 8000 兆回の四則演算を行ったことを意味する。パソコンに比べて 800 万倍も速いわけである。800 万倍というのは 130km/h で走る車と、光の速さの比である。現在のスーパーコンピュータがいかに驚異的な四則演算能力を持つか、実感できるであろう。

2 このテキストの骨組み

以下の章で考えるのは、ある関数 $f(x)$ を、限られた情報だけをもとに「推測」するかという問題である。与えられた情報といってもいろいろな可能性があるが、よくあるのは、離散化された x 座標値 $x_0, x_1, x_2, \dots, x_n$ での関数 f の値、 $f(x_0), f(x_1), f(x_2), \dots, f(x_n)$ の値だけが分かっている、という場合である。これだけの情報から、この関数の様々な性質、たとえば、

補間：格子点 $x_j, j = 0, 1, \dots, n$ からはずれた位置における関数 $f(x)$ の値

微分：格子点 $x_j, j = 0, 1, \dots, n$ 上のその関数の微分 $f'(x)$ の値、

積分：その関数のある区間で積分した定積分の値

を求める問題を考える。これらの値を厳密に決めるのは不可能なので、与えられた情報 (つまりその関数は $n+1$ 個の格子点 x_j で、 $f(x_j)$ を通るという情報) だけから「推測」するしかない。四則演算を使っていかに精度良く、そして効率よくこれらの値を推測するか、というのがこれから考える問題である。

3 補間

格子点上での関数 f の値だけが分かっているという場合に、格子点から外れたの位置における関数値をどう推測するか？我々が興味のある関数 $f(x)$ は何らかの物理量に関係した量なので、滑らかな関数であるはずであるが、ここでは関数 $f(x)$ の滑らかさという性質にこだわらず、少々乱暴に感じられるかもしれないが、以下のように考えてみよう。

x 座標上に $n+1$ 個の格子点 $x_j, j = 0, 1, 2, \dots, n$ が並んでいる。隣り合う二つの格子点からなる区間 $[x_j, x_{j+1}], j = 0, 1, 2, \dots, n$ は全部で n 個ある。関数 $f(x)$ を、この n 個の区間内それぞれで (それぞれ別々な) 一次関数で近似する。

話を具体的にするために、特定の区間 $[x_3, x_4]$ に注目しよう。この区間の両端 x_3 と x_4 で関数 $f(x)$ はそれぞれ $f(x_3)$ と $f(x_4)$ という値をとることはわかっている。区間 $[x_3, x_4]$ で $f(x)$ を一次関数で近似する場合には、この性質を満たす一次関数を作る。つまり $x-y$ 平面でグラフを描くと、二つの点 $(x, y) = (x_3, f(x_3))$ と $(x, y) = (x_4, f(x_4))$ を線分で結べばよい。区間 $[x_3, x_4]$ 以外の全ての区間 $[x_j, x_{j+1}]$ でも同様に線分をつくれれば、全体として $f(x)$ の近似関数ができる。この近似関数を $\tilde{f}(x)$ と書き、区分的線形関数と呼ぼう。各区間 $[x_j, x_{j+1}]$ 内では直線であり、格子点でグラフはつながってはいるが、そこで突然折れ曲がっている。本来滑らかな関数 $f(x)$ を折れ線グラフ $\tilde{f}(x)$ で近似していることになる。

$\tilde{f}(x)$ を使って、格子点 x_j 上からはずれた位置における関数 $f(x)$ の値を推測する。 x がどのような値であろうと、それはどれかの区間に入っているはずで、その区間では $\tilde{f}(x)$ は線形関数 (一次関数) で書かれているので、これを線形補間という。

$y = x^2$ という関数を $x = -\Delta x, 0, \Delta x$ の 3 つの離散点を使い、 x 軸上の二つの区間 $[-\Delta x, 0]$ と $[0, \Delta x]$ の間でそれぞれ線形補間すると、本来滑らかな放物線を三角形で近似することになる。その三角形の 2 辺は、3 つの点 $(x, y) = (-\Delta x, \Delta x^2), (0, 0), (\Delta x, \Delta x^2)$ を結んだものである。グラフを見れば明らかにこの線形補間には誤差がある。右側の区間の midpoint $x = \Delta x/2$ における線形補間値は三角形の右側の辺の midpoint の高さで、 $\tilde{f}_i = \Delta x^2/2$ である。一方、真の値は $f_r = f(\Delta x/2) = (\Delta x/2)^2 = \Delta x^2/4$ なので、この場合の線形補間の誤差は $|\tilde{f}_i - f_r| = \Delta x^2/2 - \Delta x^2/4 = \Delta x^2/4$ である。

線形補間の誤差について、もう少し詳しく考えてみよう。与えられた関数が二次多項式で、それを一次多項式で近似するので、その誤差の関数 $E(x)$ は、

$$E(x) = (\text{二次多項式}) - (\text{一次多項式})$$

なので、 $E(x)$ は二次多項式である。この二次多項式 $E(x)$ は格子点 x_j と x_{j+1} では定義からゼロになるはずなので、

$$E(x) = \alpha(x - x_j)(x - x_{j+1})$$

と書けるはずである。 $(\alpha$ はある定数。) $x - x_j$ や $x - x_{j+1}$ は $O(\Delta x)$ の量なので、結局

$$E(x) = O(\Delta x^2)$$

になる。

では、与えられる関数 $f(x)$ が 3 次多項式の場合を考えてみよう。この $f(x)$ を線形補間すると、その誤差は

$$E(x) = (\text{三次多項式}) - (\text{一次多項式}) = (\text{三次多項式})$$

になる。この $E(x)$ も格子点 x_j と x_{j+1} ではゼロになるはずだから、

$$E(x) = \alpha_1(x - x_j)^2(x - x_{j+1}) + \alpha_2(x - x_j)(x - x_{j+1})^2 + \alpha_3(x - x_j)(x - x_{j+1})$$

という形で書けるはずである。 $x_j \leq x \leq x_{j+1}$ の範囲の x の位置で、この式の右辺の最初の二項は $O(\Delta x^3)$ 、最後の項は $O(\Delta x^2)$ なので、結局

$$E(x) = O(\Delta x^2)$$

である。与えられる多項式 $f(x)$ が三次以上、何次の多項式であってもその関数を線形補間した時に出る誤差は、 $O(\Delta x^2)$ で評価できる。滑らかな関数はテーラー展開で多項式として表現できるので、結局 (滑らかな) 関数の線形補間の誤差は $O(\Delta x^2)$ であることがわかる。線形、つまり「一次」補間の誤差は一次ではなく二次であることに注意。

$O(\Delta x^2)$ の誤差というのは多くの場合、十分な精度と言える。 Δx がたとえば $\Delta x = 10^{-2}$ であれば線形補間の誤差が $O(10^{-4})$ 、つまり 0.01% である。それでもこれでは誤差が大きすぎるという場合は、 Δx を小さくすれば良い。 Δx を半分にするれば誤差は $1/4$ になる。だが、 Δx を半分にすると計算格子の総数が 2 倍になり、計算すべき量も 2 倍、あるいはそれ以上になる。そのような実際的な制限より Δx を小さくすることはこれ以上できないが、それでも補間誤差を小さくする必要がある場合には、線形補間ではなく、もっと高い精度の補間法を用いる必要がある。

線形補間の基本的なアイディアは、区間 $[x_j, x_{j+1}]$ 内における関数を (思い切って) 直線、つまり 1 次関数であると近似することにあつた。補間精度を上げるためには、この近似を 1 次関数から 2 次関数に向上させればよい。ただし、 x_j と x_{j+1} の二つの点だけで 2 次関数を消えることはできない。2 次多項式

$$\tilde{f}(x) = c_2 x^2 + c_1 x + c_0 \quad (2)$$

には三つの数係数 c_0, c_1, c_2 があるので、関数値のデータが 3 つ必要である。そこで、3 つの計算格子点 x_{j-1}, x_j, x_{j+1} における関数値 $f(x_{j-1}), f(x_j), f(x_{j+1})$ を使ってこの数係数を決める。これは簡単で、この三つの関数値を式 (2) に代入して連立方程式

$$f(x_{j-1}) = c_2 x_{j-1}^2 + c_1 x_{j-1} + c_0 \quad (3)$$

$$f(x_j) = c_2 x_j^2 + c_1 x_j + c_0 \quad (4)$$

$$f(x_{j+1}) = c_2 x_{j+1}^2 + c_1 x_{j+1} + c_0 \quad (5)$$

を解けばよい。実は解がなくてもその解の形はわかる。 $x = x_{j-1}, x_j, x_{j+1}$ という 3 点で、 $f(x_{j-1}), f(x_j), f(x_{j+1})$ という値をとるような二次多項式は、

$$\tilde{f}(x) = \frac{f(x_{j-1})}{2\Delta x^2} (x - x_j)(x - x_{j+1}) - \frac{f(x_j)}{\Delta x^2} (x - x_{j-1})(x - x_{j+1}) + \frac{f(x_{j+1})}{2\Delta x^2} (x - x_{j-1})(x - x_j) \quad (6)$$

という形で書けるはずである。この形の多項式を（二次の）ラグランジュ多項式という。これが解になっていることは、右辺の x に x_{j-1}, x_j, x_{j+1} の三つの値を代入してみれば確認できる。

二次ラグランジュ多項式 $\tilde{f}(x)$ を使って行う補間を、二次ラグランジュ補間という。区間 $[x_{j-1}, x_{j+1}]$ において与えられた関数を 2 次ラグランジュ補間をする場合、与えられた関数が 2 次多項式であれば上の定義からわかるようにその補間は正確（つまり誤差ゼロ）である。誤差が現れるのは、与えられた関数が三次以上の多項式の場合だけである。そこでまずは $f(x)$ が三次多項式の場合を考えてみよう。

与えられた三次の多項式を、二次の多項式で（無理矢理）表すのが二次ラグランジュ補間なので、その誤差 $E(x)$ は

$$E(x) = (\text{三次多項式}) - (\text{二次多項式}) = (\text{三次多項式})$$

である。この三次多項式 $E(x)$ は定義から 3 つの格子点 x_{j-1}, x_j, x_{j+1} でゼロになるはずなので

$$E(x) = \alpha(x - x_{j-1})(x - x_j)(x - x_{j+1})$$

という形で書ける。従って誤差は $O(\Delta x^3)$ である。与えられた関数が四次以上の多項式であっても誤差のリーディングオーダーは $O(\Delta x^3)$ である。結局二次ラグランジュ補間の誤差は $O(\Delta x^3)$ であることがわかる。

二次以上の多項式補間（ラグランジュ補間）を考えることももちろんできる。また、ここに述べたような多項式以外の方法形で関数を補間する方法も数多く存在する。多項式補間の次数のように、それぞれの補間法において精度をコントロールするパラメータがある。どの手法を選択し、それをどのパラメータ（精度）で適用するかは、問題に応じて使い分けるのが理想的である。しかし、まずはここで最初に述べた線形補間を試してみることを勧める。精度が足りなければ格子間隔 Δx を小さくする。もしもそれで十分であればファンシーな方法を採用しなければいけない理由はない。

4 微分

関数の微分を近似する方法を考えよう。再び、関数 $f(x)$ があり、この関数の値は、離散点 $x_0, x_1, x_2, \dots, x_n$ 上でだけ分かっているものとする。今の問題は関数 $f(x)$ の x に関する微分 $df/dx = f'(x)$ の値を、同じ格子点上で求めたい、というものである。今、簡単のために格子間隔 $\Delta x = x_{j+1} - x_j$ は一定としよう。

格子点 x_j における微分 $f'(x_j)$ を求めるに、 $f(x)$ を区分的線形関数 $\tilde{f}(x)$ で近似したと考えよう。つまり $\tilde{f}(x)$ は、二つの隣り合う格子点からなる区間内では直線（一次関数）である。この場合格子点 x_j における関数の微分 $f'(x_j)$ を推測するには二つの方法がある。一つは右側の区間 $[x_j, x_{j+1}]$ での $\tilde{f}(x)$ を使い、その微分 $\tilde{f}'(x_j)$ で $f'(x_j)$ を近似する方法、もう一つは、左側の区間 $[x_{j-1}, x_j]$ での $\tilde{f}(x)$ を使い、 $f'(x_j)$ を近似する方法である。 $\tilde{f}(x)$ はそれぞれの区間内ではただの一次関数なので、その微分値 $\tilde{f}'(x)$ はその直線の傾きに他ならない。

最初の方法では

$$f'(x_j) \sim \tilde{f}'(x_j) = \frac{f(x_{j+1}) - f(x_j)}{\Delta x} \quad (7)$$

二番目の方法では、

$$f'(x_j) \sim \tilde{f}'(x_j) = \frac{f(x_j) - f(x_{j-1})}{\Delta x} \quad (8)$$

として近似できる。

求めようとしている関数 $f(x)$ がもともと（全領域にわたって直線の）一次関数でない限り、この二つの値は異なるのが普通である。今行っているのは近似なので、正しい微分の値 $f'(x_j)$ との差、つまり誤差があるのは仕方がない。上の二つはその誤差の範囲で同じであり、どちらが正しいというものでもない。（その誤差がどの程度かについてはあとで見積もる。）

微分 $f'(x_j)$ は関数 $f(x)$ のグラフを描いたときの、 $x = x_j$ における接線の傾きであることを思い出そう。上の二つの式 (7) と (8) は、そのグラフで図形的な解釈ができる。 x - y 平面に描いたグラフで、点 A: $(x, y) = (x_{j-1}, f(x_{j-1}))$ 、点 B: $(x, y) = (x_j, f(x_j))$ 、点 C: $(x, y) = (x_{j+1}, f(x_{j+1}))$ を通る曲線 $y = f(x)$ の点 B における接線の傾きを、直線 AB の傾きで近似したのが式 (8) であり、直線 BC の傾きで近似したのが式 (7) である。格子間隔 Δx が小さくなればなるほど $\tilde{f}'(x_j)$ の値は、（式 (7) と (8) のどちらをとろうとも）真の微分値（接線の傾き）に近づいていく。

式 (7) の右辺の分子 $f_{j+1} - f_j$ は関数 $f(x)$ の値の差であり、分母 $\Delta x = x_{j+1} - x_j$ は格子位置の差である。微分をこのような差の商で近似する方法を差分法と言う。式 (8) も差分である。

式 (7) や (8) の誤差はどの程度であろうか？ こういうときには単純な関数をテスト関数として代入し、計算しやすい位置 x を使って特殊な場合について具体的な計算をするとよい。もとの関数 $f(x)$ が一次関数であればその誤差はゼロになることは明らかなので、 $f(x)$ が二次関数の場合の誤差を見積もってみよう。

テスト関数として $f(x) = x^2$ という関数を考え、この関数の $x = 0$ での微分 $f'(x = 0)$ を差分法で計算してみる。差分式 (7) を使って評価すると、

$$f'(0) \sim \tilde{f}'(0) = \frac{f(\Delta x) - f(0)}{\Delta x} = \frac{\Delta x^2}{\Delta x} = \Delta x$$

今の場合 $f'(x) = 2x$ だから、もちろん正しい微分値は $f'(x = 0) = 0$ である。つまり $f(x)$ が二次関数のときの式 (7) の差分法による誤差は Δx である。 $f(x)$ が三次以上の一般的な多項式の時にも同様に式 (7) に従って計算してみれば、誤差のリーディングオーダーは x^2 の項で決まるので、結局この差分法の誤差は $O(\Delta x)$ であることがわかる。

これでは精度が十分でない場合は、もとの関数 $f(x)$ を区分的線形関数ではなく、(区分的) 二次多項式で近似すれば良い。つまり、三つの格子点 x_{j-1}, x_j, x_{j+1} 上で関数値 $f(x_{j-1}), f(x_j), f(x_{j+1})$ を持つ二次ラグランジュ多項式 (6) をとる。この式を x で微分するのは簡単で、

$$\tilde{f}'(x_j) = \frac{f(x_{j+1}) - f(x_{j-1}))}{2\Delta x} \quad (9)$$

を得る。これを二次の中心差分法という。計 3 回の四則演算 (1 回の引き算、1 回の割り算、1 回のかけ算) で微分という操作を近似するわけである。

x - y 平面に描いたグラフで、点 A: $(x, y) = (x_{j-1}, f(x_{j-1}))$ 、点 B: $(x, y) = (x_j, f(x_j))$ 、点 C: $(x, y) = (x_{j+1}, f(x_{j+1}))$ を通る曲線 $y = f(x)$ の点 B における接線の傾きを直線 AC の傾きで近似するのが二次中心差分法である。

二次中心差分法の誤差を評価してみよう。もとの関数 $f(x)$ が二次関数であれば、二次中心差分法は正確 (誤差がゼロ) なので、 $f(x)$ が三次以上の多項式の時に誤差が生まれるはずである。そこでテスト関数として $f(x) = x^3$ のときの、 $x = 0$ での微分値を二次中心差分で近似したときの誤差を計算してみよう。

$$\tilde{f}'(0) = \frac{\Delta x^3 - (-\Delta x)^3}{2\Delta x} = \Delta x^2.$$

正しい微分値は $f'(x) = 3x^2$ より $x = 0$ ではゼロだから、誤差は Δx^2 であることがわかる。つまり二次中心差分法の誤差は $O(\Delta x^2)$ である。

式 (9) 式とその誤差はテーラー展開を使っても求めることができる。

$$f(x_{j+1}) = f(x_j + \Delta x) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\Delta x \frac{d}{dx} \right)^k f(x_j) \quad (10)$$

同様に

$$f(x_{j-1}) = f(x_j - \Delta x) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\Delta x \frac{d}{dx} \right)^k f(x_j) \quad (11)$$

この二つの式の差をとると、

$$f(x_{j+1}) - f(x_{j-1}) = 2\Delta x \frac{d}{dx} f(x_j) + O(\Delta x^3)$$

従って

$$f'(x_j) = \frac{f(x_{j+1}) - f(x_{j-1}))}{2\Delta x} + O(\Delta x^2) \quad (12)$$

となる。

次に二階微分 $f''(x_j)$ を考えよう。二階微分は微分の微分であることを思い出すと、次のように考えるのが自然かもしれない。つまり、

$$f''(x_j) = \frac{f'(x_{j+1}) - f'(x_{j-1}))}{2\Delta x} + O(\Delta x^2)$$

なので、ここに出てくる $f'(x_{j+1})$ と $f'(x_{j-1})$ を式 (12) を使い

$$f'(x_{j+1}) = \frac{f(x_{j+2}) - f(x_j)}{2\Delta x} + O(\Delta x^2)$$

$$f'(x_{j-1}) = \frac{f(x_j) - f(x_{j-2}))}{2\Delta x} + O(\Delta x^2)$$

として代入すると、

$$f''(x_j) = \frac{f(x_{j+2}) - 2f(x_j) + f(x_{j-2}))}{4\Delta x^2} + O(\Delta x^2)$$

となる。

確かにこれも悪くはないが、この方法だと格子点 x_j での二回微分を求めるのに、すぐ隣の格子点 $x_{j\pm 1}$ での関数の値を全く使っていないので、もったいない。せっかく細かい格子間隔 Δx で空間を離散化しているのに、その2倍も粗い格子間隔 $2\Delta x$ で空間を離散化した場合の処理になっているからである。もっといい方法がある。仮にはじめから $\Delta x/2$ の格子間隔で空間を離散化していたと仮定すればつまり格子点が $\dots, x_{j-3/2}, x_{j-1}, x_{j-1/2}, x_j, x_{j+1/2}, x_{j+1}, x_{j+3/2}, \dots$ というふうに置かれていたとすれば、上の導出を (半分小さい空間スケールで) 繰り返して、

$$f''(x_j) = \frac{f'(x_{j+1/2}) - f'(x_{j-1/2}))}{\Delta x} + O(\Delta x^2)$$

$$f'(x_{j+1/2}) = \frac{f(x_{j+1}) - f(x_j)}{\Delta x} + O(\Delta x^2)$$

$$f'(x_{j-1/2}) = \frac{f(x_j) - f(x_{j-1}))}{2\Delta x} + O(\Delta x^2)$$

となるので、この3つの式から、

$$f''(x_j) = \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))}{\Delta x^2} + O(\Delta x^2) \quad (13)$$

を得る。この右辺に出てくる

$$\tilde{f}''(x_j) = \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))}{\Delta x^2} \quad (14)$$

が (二階微分を近似する) 二次中心差分である。この差分表現はテーラー展開の式 (10) と式 (11) から導出することもできる。

三階以上の微分や、格子間隔が一定でない場合 (不等間隔格子点という) の差分表現を得るためには、テーラー展開式 (注目している格子点の複数の近傍の格子点における関数値をテーラー展開で表現した式) を書き下し、それらの連立させて解けば求める微分の差分表現を得ることができる。

4.1 積分

次に関数の定積分を数値的に求める方法を考えよう。また例によって等間隔の空間格子 (格子間隔 Δx) をもつ格子点 $x_j, j = 0, 1, 2, \dots$ の上での関数の値 $f(x_j)$ だけが分かっているとす。ある区間 $[x_j, x_{j+1}]$ での関数 $f(x)$ の定積分

$$I_j = \int_{x_j}^{x_{j+1}} f(x) dx \quad (15)$$

を近似するのに、まずはこの区間を線形補間した場合を考えてみよう。区間 $[x_j, x_{j+1}]$ で $f(x)$ を線形補間すると、この区間での関数のグラフは直線になるから、定積分 I_j を台形の面積 \tilde{I}_j で近似することになる。

$$I_j \sim \tilde{I}_j = \frac{1}{2} (f(x_j) + f(x_{j+1})) \Delta x \quad (16)$$

これは格子間隔 Δx の長さをもつ小さな区間 $[x_j, x_{j+1}]$ での式であった。 x の定義域 $[x_0, x_n]$ 全体にわたる定積分は、この小区間での積分を足し合わせればよい。つまり

$$I = \int_{x_0}^{x_n} f(x) dx \quad (17)$$

$$\sim \sum_{j=0}^{n-1} \tilde{I}_j \quad (18)$$

$$= \left(\frac{1}{2}f(x_0) + \sum_{j=1}^{n-1} f(x_j) + \frac{1}{2}f(x_n) \right) \Delta x \quad (19)$$

と近似できる。これを台形公式という。

台形公式では精度が足りなければ、例によって補間法を線形補間から二次多項式補間に改善すればよい。もとの関数 $f(x)$ が、点 A: $(x, y) = (x_{j-1}, f(x_{j-1}))$ 、点 B: $(x, y) = (x_j, f(x_j))$ 、点 C: $(x, y) = (x_{j+1}, f(x_{j+1}))$ を通るとして、この $f(x)$ を、上の3点を通る二次多項式、式(6)で近似するのがこの補間法であった。便宜のためこの式をもう一度書く。

$$\tilde{f}(x) = \frac{f(x_{j-1})}{2\Delta x^2} (x-x_j)(x-x_{j+1}) - \frac{f(x_j)}{\Delta x^2} (x-x_{j-1})(x-x_{j+1}) + \frac{f(x_{j+1})}{2\Delta x^2} (x-x_{j-1})(x-x_j) \quad (20)$$

これを x_{j-1} から x_{j+1} まで積分した値

$$\tilde{I} = \int_{x_{j-1}}^{x_{j+1}} \tilde{f}(x) dx$$

を計算する。それには座標変換 $x - x_j \equiv \zeta$ を行うのが便利である。すると、 $x - x_{j-1} = \zeta + \Delta x$ 、 $x - x_{j+1} = \zeta - \Delta x$ に注意して

$$\tilde{I} = \int_{-\Delta x}^{\Delta x} \left\{ \frac{f(x_{j-1})}{2\Delta x^2} \zeta (\zeta - \Delta x) - \frac{f(x_j)}{\Delta x^2} (\zeta + \Delta x) (\zeta - \Delta x) + \frac{f(x_{j+1})}{2\Delta x^2} (\zeta + \Delta x) \zeta \right\} d\zeta$$

となる。これを計算するのは簡単ではあるが退屈なので楽をしよう。右辺の積分をじっと見れば、

$$\tilde{I} = \Delta x \times (\alpha f(x_{j-1}) + \beta f(x_j) + \gamma f(x_{j+1})) \quad (21)$$

という形に書けることはすぐ分かるので、あとは数係数 α, β, γ を決めればよい。そのために三つのテスト関数 $f_0 = 1, f_1 = \zeta, f_2 = \zeta^2$ をいれれば

$$(\text{左辺}) = \int_{-\Delta x}^{\Delta x} 1 d\zeta = \Delta x + \Delta x = 2\Delta x = (\text{右辺}) = \Delta x (\alpha + \beta + \gamma) \quad (22)$$

$$\int_{-\Delta x}^{\Delta x} \zeta d\zeta = 0 = \Delta x (-\alpha + \gamma) \quad (23)$$

$$\int_{-\Delta x}^{\Delta x} \zeta^2 d\zeta = \left[\frac{\zeta^3}{3} \right]_{-\Delta x}^{\Delta x} = \frac{2\Delta x^3}{3} = \Delta x (\alpha \Delta x^2 + \gamma \Delta x^2) \quad (24)$$

式(22)-(24)より、

$$\begin{cases} \alpha + \beta + \gamma = 2 \\ -\alpha + \gamma = 0 \\ \alpha + \gamma = \frac{2}{3} \end{cases} \quad (25)$$

を得る。これは簡単に解けて、 $\alpha = \gamma = 1/3, \beta = 4/3$ である。式(21)に代入して、結局求める積分は

$$\tilde{I} = \frac{\Delta x}{3} \{f(x_{j-1}) + 4f(x_j) + f(x_{j+1})\} \quad (26)$$

となる。これをシンプソンの公式という。

上の式(26)は長さ $2\Delta x$ にわたる区間 $[x_{j-1}, x_{j+1}]$ での定積分であったが、もっと大きな範囲での積分を求める場合にはそれを $2\Delta x$ ごとの区間にわけてそれぞれにシンプソンの公式を適用すればよい。

5 まとめ

このテキストでは数値計算において最も基本的な手法といえる、補間法、差分法、数値積分法の初歩について述べた。「与えられたデータが格子点上の関数の値だけである」という条件の下、(格子点以外の位置での)その関数の値や、関数の微分の値、そして関数の定積分値を推測するために、区分的線形関数(折れ線グラフ)でまずは関数を近似し、次に区間内の近似関数を二次関数に改善することで精度を上げるという考え方で、これらの手法をある程度系統的に説明しようと試みたが、分かりやすかったであろうか。この考え方で非線形方程式の求解法(ニュートン法)や微分方程式の解法などについても理解できる。4次ルンゲ=クッタ法とシンプソン公式とのつながりなども解説したかったが、時間と紙幅の制限からそれは諦めることにする。

数値計算に関してはここで述べた補間法、差分法、数値積分法以外に、上で少し触れた微分方程式の解法や線形計算(行列計算)など重要なトピックが数多くあり、それぞれの課題について様々な手法が山のように開発されている。それらを全てを最初から深く、網羅的に理解するのは、百科事典を一ページ目から読み通すようなもので、大変である。むしろそれぞれのトピックについてまずはもっとも初等的なものだけに絞り、背後にある基本的なアイデアが何であるかを把握することが大事だと思う。基本的なアイデアさえつかめば、技術的な詳細は自分で再現できるはずである。初歩的なものよりも高度とされる手法の多くは、鍵となる基本アイデアを発展させたり、ひねりを加えたりしただけである。数値計算は本来、自由なものである。道具が必要となれば自分で作ればよい。そういった気楽な姿勢を持つ方が、網羅的な知識よりも応用が利くのではないかと思う。

とはいえ、このテキストで述べたことは余りに少ない。補間法にはここで述べた多項式による方法以外にも様々な手法があるし、微分を含む方程式を離散化する方法についても有限差分法以外に、有限体積法、有限要素法、境界要素法、スペクトル法など、様々な手法が開発され、実用化されている。これらがどんな手法であるかを知っておくことは必須といえる。(講義では解説する予定である。)

数値計算の教科書としては最近出版された日本語の教科書 [1] がお勧めである。英語で書かれた本としては [2] が昔から有名である。[3] は古い本だが、数値計算の達人が、様々な面白い話題について16章にわたって書いた楽しい(そして勉強になる)本である。[4] は複数の著者が書いた本で、これもいろいろな話題が述べられているが、長年数値計算をしている人間でもヒヤリとするような落とし穴などについても書かれているので、数値計算に慣れたと感じた頃に読むことを勧める。

References

- [1] 『理工学のための数値計算法』水島 二郎、柳瀬 眞一郎 著、数理工学社 第2版、2009年
- [2] “Numerical Recipes: The Art of Scientific Computing”, 3rd ed. (2007), Cambridge Univ. Press
- [3] 『数値計算術』森口繁一著、共立出版、1987年
- [4] 『数値計算のつぼ』二宮市三 編、共立出版、2004年