数値解析と最適化理論　２００６年度版講義ノート

担当：小野里雅彦 (システム情報科学専攻／システム工学科)

平成１８年度前期　システム工学科２年次／電子工学科４年次

0　講義ガイダンス

0.1　本講義のねらい

数理的に解決すべき問題を前にしたときに，それを実際に解き，さらにはよりよい解を構成することは，工学において基本的でかつ大変に重要なことである．本講義ではそのための理論を学ぶ．現代の問題解決においてはコンピュータが強力な助けとなるが，それは決して万能ではない．単純な演算の組み合わせで複雑で大規模な問題を解く上には，さまざまなことに注意深く吟味し，問題の性質にあった適切なアルゴリズムを用いなくてはならない．また，そうした努力をしても実際に解くことのできる問題は実は一部でしかない.「数値解析と最適化理論」の講義においては，情報工学，制御工学，システム工学，電磁気学，物性工学などの技術的基礎の一つとして講義するので，受講者に対しては単なる知識ではなく「使える技術」として身に付けることを期待している．

0.2　本講義の実施計画

平成１８年度の本講義は以下の予定で実施する.（ただし，講義の進行や理解の度合いによって変更されることがある.）

1. 4/10: 講義ガイダンスと数値表現の基礎
2. 4/17: 関数の近似
3. 4/24: 関数の補間
4. 5/ 8: 数値微分と数値積分
5. 5/15: 線形方程式の数値解法
6. 5/22: 非線形方程式の数値解法
7. 5/29: 常微分方程式の数値解法
8. 6/ 5: 最適化理論の概要
9. 6/12: 非線形計画法（I)：制約のない問題の最適化法
10. 6/19: 非線形計画法（II)：制約付き問題の最適化法
11. 6/26: 組み合わせ最適化
12. 7/ 3: 主観的評価に基づく最適化
13. 7/10: その他の最適化法
14. 7/24：講義のまとめ

数値解析ならびに最適化の理論並びに技法は，非常に広範でかつ深い内容を持つため，この１コマ１学期の講義でそのすべてを扱うことは不可能である．そこで，本講義では諸君らに数値解析と最適化に

おいてさまざまな形式の問題が存在し，その形式のそれぞれに対して，また色々な手法が考案され，使用されているということを理解してもらいたい．また，数値解析と最適化理論について自ら学ぶ上での学問領域の「概略地図」を示すことを目指している．

0.3 本講義における評価方法等

本講義での成績は評価は，期末試験を実施し，講義への出席回数を以下の式で加味して評価する．

$$\text{成績} = (100 - 2 \times \text{出席回数}) \times \text{試験素点}/100 + (2 \times \text{出席回数})$$

この式によると，一回も出席をしなかった場合には，試験の素点で評価を行い，14 回すべてに出席した場合には，試験の素点を 72 点満点で換算し，出席点 28 点が加算される．単位取得に必要な 60 点の総合成績をとるためには，試験素点で全欠席の場合で 60 点，全出席の場合で 45 点となる．

出席の記録を公正かつ効率的に行うため，講義開始時に座席配置になった出席表を回覧する．出席は講義開始時に着席していることが基本である．出席表は講義開始後，約 20 分以内に回収する．また，原則に従い，出席表を回収した後の記入は認めない．

講義の妨げになる学生に対しては注意（イエローカード）を与え，繰り返し注意を受けるようであれば講義への出席を認めないこと（レッドカード）もある．その場合でも試験を受けることは認める．講義室内での飲食，着帽，携帯電話の使用は当然のことながら認めない．

なお，諸君からの講義に関する質問，意見，批判を歓迎する．小野里の居室は情報科学研究科棟 5F 5-14 号室，電子メールアドレスは onosato@ssi.ist.eng.hokudai.ac.jp である．

1 数値計算の基礎

1.1 数値の表現

まず，コンピュータにより数を扱うことに関して考えてみることにする．

数学で扱う実数とは，3.1415926535897 . . . のように無限に桁を続けることができるが，コンピュータの中で扱う数はある有限の長さで表現できなくてはならない．

たとえば，8bit（2 進数 8 桁) で最上位ビットを符号を表すものとして整数を表現することを考えると，-128 から 127 まで $256 = 2^8$ 個の数を表現できる．

2 進	10000000	10000001	$\cdots$	11111111	00000000	00000001	$\cdots$	01111110	01111111
10 進	-128	-127	$\cdots$	-1	0	1	$\cdots$	126	127

この整数の表現の場合，自明のことではあるが，隣接する数の差（隔たり）は常に 1 である．整数においては，表現できる数の範囲（最大値，最小値）のみが問題となる．

実数を表現する方式はいろいろなものが存在するが，コンピュータの世界で広く使われているのが，浮動小数点数 (floating point number) である．これは，β 進 n 桁の計算体系では，浮動小数点数 x_f は

$$x_f = \begin{cases} 0 \\ \pm f \times \beta^m \ (L \le m \le U) \end{cases}$$

$$\text{ただし} \quad f = \frac{x_1}{\beta} + \frac{x_2}{\beta^2} + \cdots + \frac{x_n}{\beta^n},\ 1 \le x_1 \le \beta - 1,\ 0 \le x_i \le \beta - 1 \ (i = 2, \ldots, n)$$

で表現される数のことである．ここで，f を x_f の**仮数部 (mantissa)** あるいは**小数部 (fraction)**，m を x_f の**指数部 (exponent)** とよぶ．また，β を**基数**とよぶ．浮動小数点による表現形式は，数の大小によ

らず一定の有効な桁数を実現できる点にある．

【練習問題 1-1】
いま，10 進 3 桁の計算体系で，L=-2，U=2 のとき次の数はどのような浮動小数点数として表現されるか？
(a) 24.5　(b) -1.24　(c) 0.00213　(d) 123.4　(e) 0.000384

ここからは，計算機での処理で広く用いられている 2 進数に限定して考えることとする．すなわち，基数 $\beta = 2$ である．すると，上記の浮動小数点数の表現においては，常に $x_1 = 1$ となるため，その部分を表記から省略して，表現として以下のようになる．

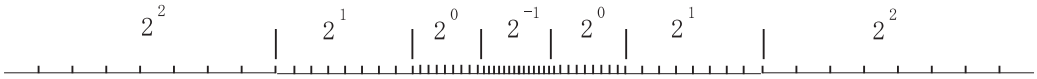
$$x_f = \begin{cases} 0 \\ \pm 1.x_1x_2x_3 \ldots x_{n(2)} \times 2^m \ (L \leq m \leq U) \end{cases}$$

浮動小数点表現の規約である IEEE-754 規格では，符号部 1bit，仮数部 23bit，指数部 8bit の合計 32bit (4byte) を用いる単精度，符号部 1bit，仮数部 52bit，指数部 11bit の合計 64bit（8byte）を用いる倍精度とすることが定められている．これらはたとえば C 言語における float 型と double 型の区別に該当している．単精度の指数部は 8bit であり，$-127 \leq m \leq 128$ としている．実際には m に 127 を加えることで 0 から 255 までの整数として指数部を表現している．このように指数部を非負の数にするために足し込まれたものを**バイアス**と呼ぶ．

1.2　数値表現と誤差

浮動小数点数として表現される数は離散的な値であり，実数軸上では間隔の開いた点として現れる．いま，簡単のため，符号部 1bit, 仮数部 3bit, 指数部 2bit で表現される体系を考えてみる．この体系で表現される数を実数軸にプロットしてみると，指数部の大きさによって，密度が違っていることがわかる．

$$-1.111_{(2)} \times 2^2, \ldots, -1.000_{(2)} \times 2^{-1}, 0, 1.000_{(2)} \times 2^{-1}, \ldots, 1.111_{(2)} \times 2^2$$



表現したい数に対して，もっとも近い浮動小数点数を選ぶ方法（丸めの方法）が”四捨五入” [1]，0 に近い側の浮動小数点数を選ぶ方法が切り捨てである．実数 x とその浮動小数点数 x_f の間に，

$$x_f = x(1 + \varepsilon_x) \quad |\varepsilon_x| \leq |\varepsilon_M|$$

が成り立ち，ε_M は浮動小数点体系と丸めの方法で決まる値で，**マシンエプシロン (machine epsilon)** と呼ばれる．このような実数を有限長の形式で表現することに起因する誤差を一般に**丸め誤差**とよぶ．

【練習問題 1-2】
いま，10 進数の 0.2 を 2 進数として表現してみなさい．つぎに，2 進 3 桁の計算体系で，L=-5, U=5 の浮動小数点として表現してみなさい．最後に，その数を 10 進数に変換してみなさい．

[1] 2 進数では 0 捨 1 入となる．

1.3 演算にともなう誤差

浮動小数点は様々な大きさの実数を，一定した有効桁数で表現できるという利点を持つと述べたが，大きさの違う数の和や差を演算する場合には注意が必要となる．今，わかりやすさのために 10 進 5 桁の体系を仮定して演算を考えてみよう．

【練習問題 1-3】
次の演算を行いなさい．
(a) $1.0000 \times 10^3 + 1.2345 \times 10^0$　(a) $1.0000 \times 10^3 + 1.2189 \times 10^0$　(c) $1.0000 \times 10^3 + 1.2345 \times 10^{-3}$

これにより，(a) と (b) とが同じ演算結果となること，(c) においては，加算そのものが行われないことに等しいことがわかるであろう．こうした丸め誤差が演算で集積していくことを**積み残し**や**情報落ち**とよぶ．たとえば，次のような級数を考えてみる．

$$\sum_{k=1}^{n} \frac{1}{k^2} = \frac{1}{1^2} + \frac{1}{2^2} + \cdots + \frac{1}{n^2}$$

これは $k \to \infty$ で $\pi^2/6 = 1.6449340668\cdots$ へと収束することが知られている．これを，数列の順に実際にコンピュータの単精度で計算させてみると，

$n = 100$　1.634984，　$n = 1,000$　1.643935，　$n = 10,000$　1.644725，　$n = 100,000$　1.644725

となり，n を大きくしても真値に近づいていかない．これは先程述べた積み残しが原因である．これを回避するには，n の大きい数から順に足し合わせていくと

$n = 100$　1.634984，　$n = 1,000$　1.643934，　$n = 10,000$　1.644834，　$n = 100,000$　1.644924

のように n を大きくしていくと真値に近づいていく．

上の例は，大きさの違う数の和や差を扱う際に生じる問題であったが，同じような大きさの数の差をとる場合にも注意が必要である．例えば，2 次方程式 $ax^2 + bx + c = 0$ の解の公式 $(-b \pm \sqrt{b^2 - 4ac})/2a$ を用いて $a = 1, b = 200, c = 1$ の場合の解を単精度で求めてみると，-4.997253×10^{-3}，　-1.999950×10^2 となる．この真値は -5.000125×10^{-3}，　-1.999950×10^2 であることを見ると，第一の解の精度が悪いことがわかる．これは b と $\sqrt{b^2 - 4ac}$ との大きさが近いために，仮数部の上位の桁がみな 0 になってしまい，有効な桁数が減ってしまうために生じる．これを**桁落ち**と呼ぶ．

桁落ちを防ぐためには，上の例では式を変形して $-2c/(b + \sqrt{b^2 - 4ac})$ として同じように計算すると，第一の解の値として -5.000125×10^{-3} を得ることができる．

このように，コンピュータによる数値計算においては，数学的に見れば等しいはずの値に対して，異なる結果が得られることが多くおこる．そうしたことに注意をはらって計算を行うことが重要である．

【参考情報】
数値解析に関しては，実際にコンピュータで色々な計算を行ってみることを勧める．Excel の場合には VBA でプログラミングが可能であるし，Java は無料で開発環境が入手できる．C 言語で行って見たい場合には，エル・エス・アイ ジャパン (株) 提供の LSI C-86 ver.3.30c 試食版が無料で入手できる．(http://www.lsi-j.co.jp/)．
【参考図書】
・宮下精二著：数値計算（応用数学基礎講座 7），朝倉書店 ISBN 4-254-11577-6
・伊理正夫，藤野和建：数値計算の常識，共立出版 ISBN 4-320-01343-3
・森正武，室田一雄，杉原正顕：数値計算の基礎（岩波講座 応用数学 [方法 1]）岩波書店