数理解析研究所講究録
第 553 巻 1985 年 166-170

解決すべき数値計算基本問題
—問題提起と討論—

司会 ： 筑波大学 電子・情報工学系　森　正武
Masatake Mori

５人の方に予めお願いして、それぞれの立場から問題提起をして頂き、それに基づいて活発な討論が行われた。特に

（ｉ） 多倍長計算、およびそのための数値の表現法、
（ⅱ） 算法の命名法、標準化、
（ⅲ） 数学モデル、数値計算法、計算機ハードウェアの三者の間の相互フィードバックの必要性

などが話題となった。以下、提起して頂いた問題の要約を記す。

問題提起者　１ ： 東京大学 工学部　伊理　正夫

多数の数の和における丸め誤差について、問題提起する。

$$S=\Sigma x_i$$

のような和を計算するとき、丸め誤差ΔSを最小にするには、その加え合わせの順番をどのようにきめたらよいのか。そこで、加法での順序付けを、ｂｉｎａｒｙ ｔｒｅｅで表現するとｎｏｄｅ m における部分和 S_m の誤差 ΔS_m は、その子のｎｏｄｅｓ k, l からの伝播誤差ΔS_k, ΔS_ℓと、ｎｏｄｅ m における発生誤差δS_mの和としてかける：

$$\Delta S_m=\Delta S_k+\Delta S_\ell+\delta S_m \quad .$$

ここで、　$|\delta S_m|\leqq\varepsilon|S_m|$, $\Delta x_i=0$　と仮定すると、

全体の誤差は、　$\Delta S=\Sigma_k\delta S_k$　（ｋ：ｎｏｎｌｅａｆ ｎｏｄｅｓ），

その絶対値は、　$|\Delta S|\leqq\Sigma_k|\delta S_k|\leqq\varepsilon\Sigma_k|S_k|$

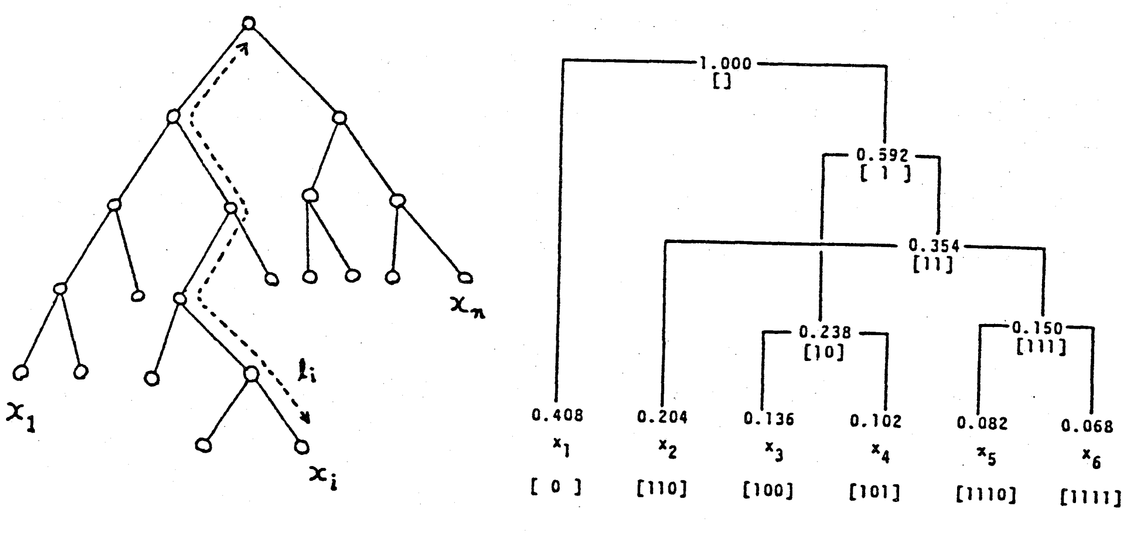
となるので、$E=\Sigma_k|S_k|$ を最小とするように、binary treeを決める問題として定式化できる。

被加数 x_i が全て正であるような場合には、x_i に対応する leaf i から root までの距離を l_i とするとき、

$$E = \Sigma_i l_i x_i \quad (i : \text{leaves})$$

と書き直すことができるので、（適当にスケーリングをして）x_i を確率と見做すことにより、Eの最小化問題は符号理論において有名な平均符号長最小化問題と等価であることがわかる。即ち、$x_i \geq 0$ の場合の最適計算順序は Huffman tree によって与えられることになる。

binary tree　　　　Huffman tree



問題：一般に負の数がある場合に、

1）　最適加算順序を決めるアルゴリズム

2）　近似解を与えるような実用的なアルゴリズム

3）　実際的には $\Sigma x_i \fallingdotseq 0$ となっている場合（方程式の解法など）が重要なので、この付加条件の下で最適加算順序を決めるアルゴリズム

を見出すこと。

問題提起者　2　：京都大学 数理解析研究所　　一松　信

問題：数学者の立場より以下の３点について

1　計算機において任意桁の（整数、実数）多倍長計算が速く実行できないか。

2　整数の厳密な計算（１が出来ればよい）。

3　整数論的関数の効率的計算法。

問題提起者　3　：日本大学 理工学部　　戸川　隼人

問題：以下の２点について

1　小さくて大きな問題（気がつきにくいもので実は数値計算全体に影響を与えるもの）。例えば、

$$\frac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n}{2}\right)}$$

の計算における ｏｖｅｒ－ｆｌｏｗ の問題には，ｌｏｇ をとることで解決できる。
例えばＵＮＩＸでは、多倍長ｌｏｇΓを組み込み関数とすることで解決している。そこで有用な多くの関数ｆ(ｘ)に対してｌｏｇｆ(ｘ)という関数副プログラムを作成すればよい。
Ｏｖｅｒ－ｆｌｏｗ，ｕｎｄｅｒ－ｆｌｏｗ など数学的でないものを無くするために、計算機に於ける浮動小数点体系を考え直してはどうか。

2　大きくて小さい問題（数値計算全般にわたるコメント）
アルゴリズムの名称を統一してはどうか。（仕事の内容によってできるだけ細分化して、モジュール毎に名前を付けるべきである。）

問題提起者　4　：名古屋大学 工学部　　鳥居　達生

数値解析鳥瞰図と基本問題

数値解析　1)　数学的認識が主、道具は簡単なほうが良い。
（広義）　2)　数学的対象の計算機による表現。演算の機械化（プログラミング）。
　　　　　1.5)　シミュレーション（画像処理、計算幾何学）。

数値解析　　代数：有限　；　解析：無限，連続。
（狭義）　＊　用いるのは解析性，連続性，離散性など相互の規則性。

- 精度と収束の安定性の弁証法的矛盾。
- 事前誤差評価と事後誤差評価。
- 計算量（手間）。

以上を総合的に考えたプログラムを作るべきである。

問題：

1 連立一次方程式について（線形代数と関数方程式の接続を良くするために）

1) 連続問題を離散化したときの特殊性が利用できないか。

2) 行列を作らず線形作用素方程式として解けないか。
CG法は適用範囲が狭いのでなんとかできないか。

3) 無限次元帯行列：最小解をもとめる。

4) 性質の悪い問題 → 連続問題 → 物理　まで逆上って意味付けが必要。

5) 連続問題から見たときの直接法のｐｉｖｏｔｉｎｇの意味？

2 関数の解析性を利用する立場で考えて、（関数近似、積分　について）

1) 周期関数、ＦＦＴ、有限区間Ｃｈｅｂｙｓｈｅｖ近似の応用法の拡張。

2) 半無限、全無限領域における大域近似性がどうか。

3) 積分表示による大域近似性がある場合の、積分をどうするか。

4) 等角写像，逆写像（楕円積分，楕円関数）。

5) 解析接続による補外、加速問題の意味付け。

6) 有限区間の積型積分

$$\int_{-1}^{x} w(z)f(z)\,dz \qquad |x| \leq 1 \qquad w：\text{非正則},\ f：\text{正則}$$

7) 積分変換，逆変換：Ｌａｐｌａｃｅ変換，微分方程式，重積分。

3 常微分方程式について

線形の場合の解法として、Ｌａｎｃｚｏｓ法，陰Ｒｕｎｇｅ-Ｋｕｔｔａ法，コロケーション法，Ｇａｌｅｒｋｉｎ法があるが、その適用法を整理するべきである。

事前誤差評価、安定性の解析が全面的にできていない。（自動化の研究が必要）

4 非線形方程式

Ｎｅｗｔｏｎ法で複数個の解が求められないか。

代数方程式の連立型解法（Ａｂｅｒｔｈ法）を拡張できないか。

問題提起者　5　:　統計数理研究所　　田辺　国士

A　思想／戦略レベル

数値計算問題の前提となる「数理モデル」の開発が重要。
統計解析、最適制御における現実的な問題の解決に参加することも重要。

B　方法、技術レベル

1　行列計算について

線形方程式の解法

直接法：ベクトル、パラレルアルゴリズム，スパース処理（ｏｒｄｅｒｉｎｇ），
　　　シストリック・アレイ。

反復法：ＣＧ／Ｌａｎｃｚｏｓ法　多項式としての観点からの解法，
　　　　　　　　　　　　　　　　前処理の方法，非対称の場合の解法。
　　　ＳＯＲ法　　　　　　　　　一般の場合の最適パラメタの決定法。

線形最小二乗法

$\| Ax-b \|_{V^{-1}}$ 最小化のための反復法。

行列式の早い反復計算法。

特殊行列：Ｔｏｅｐｌｉｔｚ行列、確率行列の定常ベクトル。

行列方程式について：代数的Ｒｉｃｃａｔｉ方程式の計算法、条件数の推定。

固有値、特異値の計算法：Ｌａｎｃｚｏｓ法の見直し。

2　非線形方程式について

Ｎｅｗｔｏｎ法より高次収束の解法を捜す。特異問題が解けるようにする。

3　最適化法について

収束域の拡大、大域的方法を捜す。

制約条件付き最適化問題におけるＱｕａｓｉ-Ｎｅｗｔｏｎ法。

更新法とｓｔｅｐ　ｓｉｚｅ決定のための評価関数。

4　偏微分方程式について

ＰＤＥによるデータのあてはめ。

5　数値積分について

大次元Ｍｏｎｔｅ-Ｃａｒｌｏ法の適用。

6　高速解法とアルゴリズムによるデータの流れの解析

専用機の作成。Ｐｏｉｓｓｏｎ　ｓｏｌｖｅｒ。アルゴリズムの複雑度。

C　実装化レベル

行列計算用ソフトウェア（ベクトル、パラレルを含む）。数式処理と数値計算の結合。専用機用アルゴリズムの開発。グラフィク用アルゴリズムの開発および実装。