修 士 論 文 概 要 書

CD

2007 年　2 月提出　　　学籍番号 3605U045-1

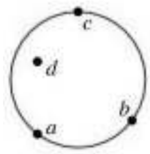
専攻名（専門分野）	情報・ネットワーク専攻	氏 名	川崎　文裕	指 導 教 員	大石　進一　　印
研究指導名	尾崎　克久				
研 究 題 目	Incircle 問題における高精度数値計算				

1　概要

計算幾何の問題の一つである点の内外判定問題から Incircle 問題に着目し，問題の難しさにより，適宜高精度計算を用いながら高速にロバストな結果を得る．また，Rump, Ogita, Oishi らによる新しい和の演算アルゴリズム Newsum を計算幾何に応用することで，先行研究を行った Jonathan Richard Shewchuk より高速に解を導くことを目標とする．

2　Incircle 問題

4 点 A,B,C,D が与えられたとき，3 点 A,B,C を通る円を境界線とし，残りの点 D がその境界の内側，外側，境界線上のどこに位置しているかを判定する．判定結果は行列式の符合に依存．



$$\det(A) = \begin{vmatrix} a_x & a_y & a_x^2+a_y^2 & 1 \\ b_x & b_y & b_x^2+b_y^2 & 1 \\ c_x & c_y & c_x^2+c_y^2 & 1 \\ d_x & d_y & d_x^2+d_y^2 & 1 \end{vmatrix} = \begin{vmatrix} a_x-d_x & a_y-d_y & (a_x-d_x)^2+(a_y-d_y)^2 \\ b_x-d_x & b_y-d_y & (b_x-d_x)^2+(b_y-d_y)^2 \\ c_x-d_x & c_y-d_y & (c_x-d_x)^2+(c_y-d_y)^2 \end{vmatrix}$$

図 1　Incircle とその行列式

3　高精度計算アルゴリズム

3.1 Dekker のアルゴリズム Split

定数 $N = 2^{\frac{t}{2}} + 1$

```
Split(a) = aH + aL
  c = fl(N * a);
  d = fl(c - a);
  aH = fl(c - d);
  aL = fl(a - aH);
```

4回の浮動小数点演算を行っているので，コストは4flopsである．Splitによりデータ $a \in F$ に対して上位27bit，下位26bitに分解する．

$a = a_H + a_L$．　但し $|a_H| \geqq |a_L|$．

3.2 Dekker のアルゴリズム TwoProduct

積を和の形にエラーフリーに変換．

```
[x + y] = TwoProduct(a,b)
  x = fl(a * b);
  [aH, aL] = Split(a);
  [bH, bL] = Split(b);
  err1 = fl(x - aH * bH);
  err2 = fl(err1 - aL * bH);
  err3 = fl(err2 - aH * bL);
  y = fl(aL * bL - err3);
```

以上より，17回の浮動小数点演算を行っているので，コストは17flops.

積を全て和に変換することで優れた和の演算アルゴリズム Newsum を利用する．

4　関数Newsum

4.1 Rump, Ogita, Oishiらによる新しい和の演算アルゴリズムを計算幾何に応用する．

```
u : unit roundoff = 2^-53
for(i = 0 , t = 0.0 ; i < n ; i++){
    temp = (sigma + p[i]) - sigma;
    w[i] = p[i] - temp;
    t += temp;
}
```

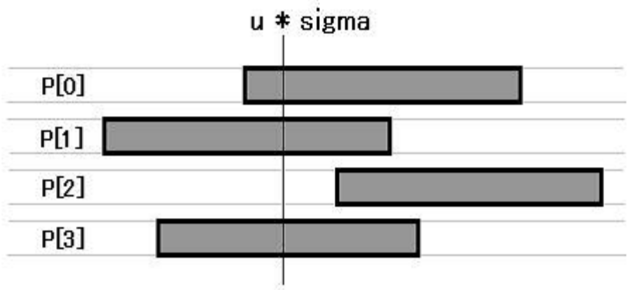


図2　NewsumとP[i]

4.2データの構成法

$$\begin{aligned} abcd &= (x+y)cd & &//[x,y] = TwoProduct(a,b) \\ &= (xc+yc)d & &//|x| > |y| \\ &= (t_1+t_2+t_3+t_4)d & &//|t_1| > |t_2| \\ &= s_1+s_2+s_3+s_4+s_5+s_6+s_7+s_8 & &//|s_1| > |s_2| \end{aligned}$$

TwoProductの性質上，s_1からs_8のデータにおける絶対値の関係は，図3 のように配置される．一区間はIEEE 754規格，double型53bit.

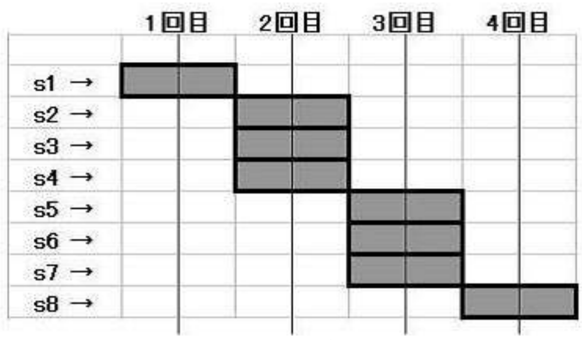


図3　Newsumとデータの関係

5　数値実験

5.1　実行環境

・CPU　　: Intel(R) Pentium(R)M 1.20GHz
・Memory : 760 MB
・OS　　　: Windows XP Professional
・Conpiler : Visual C++ 6.0

5.2 事前誤差評価

点 D に[-1,1]のランダム座標を与えた場合

	誤差の上限
(4.1)式	8.16889E-16
(4.2)式	1.96449E-15

点 D(100000,-1000000)を与えた場合

	誤差の上限
(4.1)式	1.33226E+10
(4.2)式	1.41388E-3

5.3 高精度計算の速度比較

アルゴリズムの特徴

Incircleexact : 常に全てのデータを扱い無誤差の演算を行う. 因数分解を多用したアルゴリズムにより 0 要素に対して著しい効果を発揮する．よって，実行時間は与えられた座標の疎密に大きく依存し，疎であるほど速い演算を実現.

提案手法 Newsum:符号が保証されるまでの間データを適応的に扱うので，実行時間は座標の悪条件さ(位置判定の難しさ)に依存する．つまり，疎密に対しての影響はない.

比較は，Shewchuk の Incircleexact と提案手法 Newsum の二つのアルゴリズムに対し，与える座標の悪条件さ，疎密の関係性など，場合分けをして行う．以下，単位は全て[micro seconds].

(1)悪条件かつ，密な座標を与えた場合
A(8.855E-17, 1.251E-3) B(-1.251E-3, 8.855E-17)
C(-8.855E-17, -1.251E-3)D(1.251E-3, -1.091E-48)

Incircleexact	21.990
提案手法 Newsum	11.5(見通し)

(2)悪条件かつ，疎な座標を与えた場合
A(0,1) B(-1,0) C(0,-1)　D(1, 1.233E-32)

Incircleexact	4.406
提案手法 Newsum	11.754

(3)悪条件かつ，やや疎な座標を与えた場合
A(0, 1.251E-3) B(-1.251E-3, 0)
C(0,-1.251E-3)　D(1.251E-3, 1.233E-32)

Incircleexact	6.470
提案手法 Newsum	11.534

以上より，与えられる座標が密である時ほど，今回の提案手法は効果を発揮し，Shewchuk のアルゴリズムの約 2 倍の速度まで実現した.

6　考察

6.1 事前誤差評価

位置判定の対象となる点Dに大きな座標を与えた場合,誤差の上限がとても大きな値となる.この条件では,誤差の過大評価を招き,結果として符号が保証されないことがある.この問題の解消のため,計算の過程で,aを消去した行列式をプログラム上で実現すると, 判定対象となる点Dの座標によらずある程度の精度を保つ.

このことより,他の座標よりも相対的に大きな行を消去して,事前誤差評価を用いては損であることが分る.

6.2 高精度計算アルゴリズムの速度比較

Shewchukのアルゴリズムを上回る速度を計測したものは, 悪条件, 良条件ともに密な座標を与えたときであった. その要因はShewchukのアルゴリズムは0要素に対して特に強いアルゴリズムであるが, 密な座標に対しては効果を発揮しないことからである. 点A,B,C,Dも密な座標を与えた場合, 最大で約2倍の計算速度を実現した. しかし, このアルゴリズムは次の展望で挙げる通り, さらなる発展の余地を残している.

7 展望

・0 要素への対応
・データ計算部分のさらなる改良
・コンパイルオプションの変更

この3点の改良を加えることによって, いかなる条件においてもさらなる速度向上が見込めるはずである.

8 参考文献

1. 大石進一:”応用数学セミナー 数値計算”,裳華房, (1999),
2. 大石進一:”Linux数値計算ツール”,コロナ社,(2000),
3. 大石進一:”精度保証付き数値計”,コロナ社,(2000),
4. ANSI/IEEE:IEEE Standard for Binary Floating Point Arithmetic, Std 754-1985 edition, IEEE, New York,(1985),
5. 矢島徹・及川正行:”工学基礎 線形代数”,サイエンス社, (2004),
6. S.M. Rump, T. Ogita, S. Oishi: Accurate Floating-point Summation, submitted for publication.(2006),
7. Adaptive Precision Floating-Point Arithmetic and Fast Robust Predicates for Computational Geometry. (1997),
8. Jonathan Richard Shewchuk Home Page. http://www.cs.berkeley.edu/~jrs/
9. Fast Robust Predicates for Computational Geometry. http://www.cs.cmu.edu/~quake/robust.html