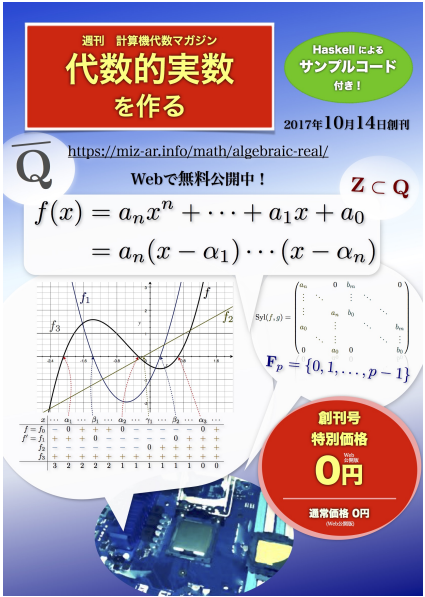
宣伝



「週刊 代数的実数を作る」
#4 まで絶賛公開中！
https://miz-ar.info/math/algebraic-real/

区間演算と精度保証

荒田 実樹

2017 年 11 月 7 日

目次

丸め

浮動小数点数では、実数やその演算結果を正確に表せないことがある

例

$1/10$ は 2 進の有限桁で正確に表せない（循環小数になる）

例

$\sqrt{2}$, e, π は正確に表せない（無理数なので）

丸め

そういう場合は「最も近い浮動小数点数」への丸めを行う
IEEE754 では

- 四則演算（FMA 含む）
- 平方根

が正しく丸められることを要請している。
それ以外の数学関数（exp とか sin とか）は正しく丸められる保証はない！

最近接丸め

最近接丸めは最もポピュラーな丸め方向。「一番近い」浮動小数点数に丸める。
ちょうど中間の場合は、最下位ビットが 0 になる方を選択する。

例

$1 + 2^{-53}$ に一番近い倍精度浮動小数点数は 1 と $1 + 2^{-52}$ の 2 つ。最下位ビットが 0 の方を選択する：

$$1 + 2^{-53} \rightsquigarrow 1$$

方向付き丸め

それ以外の丸め方向

- 正の無限大方向
- 負の無限大方向
- ゼロ方向

これらは区間演算で重要

実際のプログラミング言語では

- C 言語は実用的な言語なので、浮動小数点数の丸め方向を制御できる
  - `<fenv.h>` `fesetround/fegetround` 関数
  - `#pragma STDC FENV_ACCESS ON` で最適化を抑制
- Julia も実用的

数値計算の信頼性

浮動小数点数を使って計算した結果は信頼できない？

数値計算の信頼性

区間演算を使うと、数値計算の結果がどれだけ信頼できるのか（誤差の範囲）がわかる！

区間演算

取りうる値の上界と下界の組を使って演算する

$$
\begin{aligned}
[\underline{a}, \overline{a}] + [\underline{b}, \overline{b}] &= [\underline{a} + \underline{b}, \overline{a} + \overline{b}], \\
[\underline{a}, \overline{a}] - [\underline{b}, \overline{b}] &= [\underline{a} + \overline{b}, \overline{a} - \underline{b}], \\
[\underline{a}, \overline{a}] \cdot [\underline{b}, \overline{b}] &= [\min\{\underline{ab}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{ab}\}, \max\{\underline{ab}, \underline{a}\overline{b}, \overline{a}\underline{b}, \overline{ab}\}].
\end{aligned}
$$

区間に 0 を含まない場合は割り算もできる
浮動小数点数を使って実装する場合は、各演算において方向付き丸めを行う
（このほかに、区間の中心と半径を使う方式もある）

アルゴリズムへの適用

各種アルゴリズムを区間演算に置き換える

例

区間ガウス消去法 (Interval Gaussian elimination)

頑張れば、普通の数値計算の数倍程度の実行時間で済む

欠点

- 演算を繰り返すと値の範囲が広がってしまう
- 「演算結果は ±100 万の範囲にあります (ﾄﾞﾔｧ」と言われても嬉しくない！
- ゼロ除算が起きやすくなる

工夫が必要。平均値形式など