

論文の内容の要旨

論文題目	中心値・半径方式による精度保証付き多倍長区間演算ライブラリの開発
学位申請者	松田 望

本論文では、精度保証付き多倍長演算ライブラリの性能向上のための実装法の研究、および実際のライブラリ構築を行う。精度保証付き計算における基本技術である機械区間演算の実装では、区間の下端と上端を保持する方法が一般的である。しかし、機械区間演算を多倍長に拡張した場合、実装方法を工夫することによって、中心値と半径を保持する方法の方が計算速度・メモリ効率の両面で有利になる可能性がある。ここでの工夫とは、区間の中心値を多倍長で、半径を低精度で保持することである。

一方で、中心値と半径を保持する方法では、演算過程が複雑になるなど計算負荷に関するデメリットがある。したがって、精度保証付き多倍長演算において、下端・上端方式と中心値・半径形式のどちらが真に有利であるかは、実際に演算ライブラリを実装して比較検討を行うまでは判定できない状況であった。本論文では、中心値・半径形式を採用した区間演算に関して新しい実装方法を提案し、実際に精度保証付き多倍長区間演算ライブラリの開発を行った上で、下端・上端方式と中心値・半径方式を詳細に比較検討し、上述の問題に対しての結論を導く。

まず、精度保証法と多倍長演算を併用することの意義について論じることからはじめる。多倍長演算を用いる目的は、大きく二つに分けられる。一つは、精度の高い計算結果を得ることである。もう一つは、大きな丸め誤差が発生する問題で、丸め誤差を小さく抑えることである。一回の演算で発生する丸め誤差は小さく、計算結果にはほとんど影響を与えないが、丸め誤差を含む計算結果を引き続く計算に用いると、最終的な計算結果が非常に大きな誤差を含むことがある。このような丸め誤差の累積を防ぐ汎用的な手段が、多倍長演算である。ただし、それだけでは計算結果がどれだけ改善したのかは分からない。その効果を検証するには、何らかの検算が必要である。

一方、浮動小数点数演算の丸め誤差を見積る手法として、精度保証付き数値計算がある。精度保証付き数値計算は、一般に機械区間演算によって実装される。機械区間演算では、CPUに組み込まれた丸めの方向制御命令などを用いて、発生しうる丸め誤差を包含する区間を計算結果とする。多倍長演算に精度保証付き

数値計算を組み合わせれば、丸め誤差の大きさを把握しつつその影響を抑えることが可能となり、計算の品質保証・品質管理を行う上でその意義は大きい。

現在、多倍長演算や精度保証付き数値計算を簡単に利用できる数値計算ライブラリが、多数提供されている。精度保証付き多倍長演算を行えるライブラリとしては、MPFI・kv・C-XSC・exflib・INTLABなどがある。このうちINTLAB以外は、区間の下端・上端方式を採用している。一方、INTLABでの多倍長区間演算は中心値・半径方式ではあるが、これはユーザ使用を想定していない内部的なものである。その実装も簡便ではあるものの精度や計算効率の点で無駄が多いものとなっている。本論文ではより実用に近い中心値・半径方式の精度保証付き多倍長演算ライブラリの開発を目指す。

開発を行うライブラリをLILIBと名付ける。これはC++で記述され、多倍長実数クラス・多倍長行列クラス・精度保証付き多倍長区間クラス・精度保証付き多倍長区間行列クラスの4つのクラスを提供するものである。特に、区間を中心値と半径の組み合わせで保持し、中心値には32ビットを1単位とする任意に長い仮数部を持たせ、半径には32ビットの短い仮数部を設定する。区間演算は区間半径が相対的に大きいと演算結果を過大評価する性質があるため、多倍長演算を必要とする状況においては大きな半径を使用することはない（少なくとも有意義ではない）と考えられる。したがって、半径部分に長い仮数部を設定する必要はない、というのが本ライブラリの設計思想である。また、中心値の演算に関しては、通常の高倍長演算ライブラリで用いられている高速化手法を採用していない。これは、本ライブラリの目的が下端・上端方式と中心値・半径方式の比較検討にあり、ライブラリ自体の高速化の優先度が低いからである。ただし、将来的な高速化のための方法は具体的に検討する。

中心値・半径方式の区間演算に関しては、LILIBの構造を活かした方法を提案し、計算効率を上げることを目指した。また比較演算についても考察を行い、独自の方式を提案した。さらに、ライブラリ製作を行う上で必要な情報となる内部使用関数についても記述を行い、ライブラリ全体の実装プロセスが明確となるように努めた。

数値例では、LILIBにおける区間ではない通常の高倍長演算（これを点演算と称する）と区間演算との比較を通して、下端・上端方式と中心値・半径方式との速度比較を行う。結論としては、加減算の一部と桁数が小さいときの平方根計算とを除き、中心値・半径方式の優位性が確認できている。さらに、LILIBを用いて精度保証付き多倍長演算の実効性を検証する実験を行い、他の方式との比較も踏まえてこれを確認した。また、実際的な問題への応用も試みている。

最後に、まとめと展望を述べている。この中で、計算速度の向上のためのアイデアを記述し、実用化への道程を示した。

論文審査の結果の要旨

学位申請者氏名	松田 望
審査委員主査	山本 野人
委員	仲谷 栄伸
委員	山本 有作
委員	緒方 秀教
委員	小山 大介
委員	※渡部 善隆

第1章では、本論文の背景・目的・構成について述べている。背景として述べられていることを要約すれば以下ようになる。

精度保証付き計算における基本技術である機械区間演算の実装では、区間の下端と上端を保持する方法が一般的である。しかし、機械区間演算を多倍長に拡張した場合、実装方法を工夫することによって、中心値と半径を保持する方法の方が計算速度・メモリ効率の両面で有利になる可能性がある。ここでの工夫とは、区間の中心値を多倍長で、半径を低精度で保持することである。一方で、中心値と半径を保持する方法では、演算過程が複雑になるなど計算負荷に関するデメリットがある。したがって、精度保証付き多倍長演算において、下端・上端方式と中心値・半径形式のどちらが真に有利であるかは、実際に演算ライブラリを実装して比較検討を行うまでは判定できない状況が続いていた。

目的としては、精度保証付き多倍長演算ライブラリの性能向上のための実装法の研究、および実際のライブラリ構築を挙げている。特に、中心値・半径形式を採用した区間演算に関して新しい実装方法を提案し、実際に精度保証付き多倍長区間演算ライブラリの開発を行った上で、下端・上端方式と中心値・半径方式を詳細に比較検討し、上述の問題に対しての結論を導くとしている。

第2章では、多倍長演算と精度保証付き数値計算について解説している。

まず、精度保証法と多倍長演算を併用することの意義について論じることからはじめ、Rumpの例題の解説を通して多倍長演算での検算の難しさを示している。その上で、検算手法として精度保証付き数値計算の利用に言及する。その後、浮動小数点数の構造やcorrect rounding、区間演算などの説明に必要な概念に触れたあと、既存のライブラリをその特徴とともに紹介している。特に、中心値・半径方式を取り入れているライブラリとしてIntlabを挙げ、その構造の特徴を記述するとともに、問題点を指摘している。

第3章では、本論文で実装を行うライブラリLILIBについて解説している。そのクラス構造や演算手法について記述するとともに、入手方法も含めた使用方法も

詳細に述べ、読者が実際に使用する場合の便宜も図っている。これより、著者はライブラリ実装に「ものづくり」の側面があることを意識しているものと認められる。実装内容としては、C++で記述されること、多倍長実数クラス・多倍長行列クラス・精度保証付き多倍長区間クラス・精度保証付き多倍長区間行列クラスの4つのクラスを提供していることなどが挙げられている。特に、区間を中心値と半径の組み合わせで保持し、中心値には32ビットを1単位とする任意に長い仮数部を持たせ、半径には32ビットの短い仮数部を設定している。

第4章では実装についての詳細を記述している。区間演算は区間半径が相対的に大きいと演算結果を過大評価する性質があるため、多倍長演算を必要とする状況においては大きな半径を使用することはない（少なくとも有意義ではない）と考えられる。これを根拠として、「半径部分に長い仮数部を設定する必要はない」ということを本ライブラリの実装方針としている。また、中心値の演算に関しては、通常の高倍長演算ライブラリで用いられている高速化手法を採用していない。これは、本ライブラリの目的が下端・上端方式と中心値・半径方式の比較検討にあり、ライブラリ自体の高速化の優先度が低いからである、と説明する。ただし、将来的な高速化のための方法は具体的に後述している。

中心値・半径方式の区間演算に関しては、LILIBの構造を活かした方法を提案し、計算効率を上げることを目指している。また比較演算についても考察を行い、独自の方式を提案している。さらに、ライブラリ製作を行う上で必要な情報となる内部使用関数についても記述を行っており、これによってライブラリ全体の実装プロセスがより明確となっていると考えられる。

第5章では、数値例を扱う。LILIBにおける区間ではない通常の高倍長演算（これを点演算と称する）と区間演算との比較を通して、下端・上端方式と中心値・半径方式との速度比較を行っている。結論としては、加減算の一部と桁数が小さいときの平方根計算とを除き、中心値・半径方式の優位性が確認できたとしている。また既存のライブラリとの速度比較も行っているが、これは、本ライブラリが実用開発の途上にあることを示すものとなっている。さらに、LILIBを用いて精度保証付き多倍長演算の実効性を検証する実験を行い、区間演算以外の精度保証法との比較も踏まえてこれを確認している。また、流体力学における実際的な問題への応用も試みている。

最後に第6章でまとめと展望を述べる。この中で、計算速度の向上のためのアイデアを記述し、実用化への道程を示している。

以上によって、本論文は博士（工学）の学位論文として十分な価値を有するものと認める。