様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　２ 年　 ５ 月 ２９ 日現在

機関番号：２５４０３
研究種目：基盤研究(C)（一般）
研究期間：2017～2019
課題番号：１７Ｋ００１０６

研究課題名（和文）高速な任意精度数値計算のための実数計算ライブラリの実現方式に関する研究

研究課題名（英文）Design and Implementation of a Fast and Efficient Arithmetic Library for Computational Real Numbers

研究代表者
川端　英之（KAWABATA, Hideyuki）

広島市立大学・情報科学研究科・講師

研究者番号：００２６４９３７

交付決定額（研究期間全体）：（直接経費）　　3,300,000円

研究成果の概要（和文）：任意精度数値計算のための高速な「実数計算」ライブラリの実現方式の追求を目的とし，開発中であったIFNライブラリの改良並びに関連技術開発を行った．
種々の最適化のほか，区間演算ライブラリMPFIを用いた近似値計算，選択的中間値消去手法の導入による所要記憶容量削減等により，Haskell版およびC版のIFNライブラリは10倍高速化された．その過程で，MPFI向けHaskellバインディングの実装や，Haskellプログラミングで直面する型クラスにまつわる型の曖昧性問題を現実的な方法で排除する手法の提案もなされた．

研究成果の学術的意義や社会的意義
数値計算プログラムの設計においては誤差の発生や伝播を如何に防ぐかが重要である．本研究で開発された任意精度数値計算のための高速な「実数計算」ライブラリは，数値計算プログラム記述における計算精度についての責任をアルゴリズム設計者に負わせるのではなく，ライブラリで吸収・隠蔽しながら，所望の精度の計算結果を得る術をユーザに与える．
HaskellおよびCにより実現された可用性の高いIFNライブラリは，敏感な問題に対する正確な数値的求解や高精度なデータを扱う実験結果の数値的検証に使用でき，通常の浮動小数点演算による高速な数値計算環境を補完するツールとして有用なものとなることが期待される．

研究成果の概要（英文）：With the aim of pursuing an efficient implementation of a high-speed "real number calculation" library for arbitrary precision numerical computation, we have improved the IFN library and developed related technologies.
The Haskell and C versions of the prototype of the IFN library we designed have achieved 10 times faster speed compared to the previous ones thanks to the use of the interval arithmetic library MPFI for calculating temporary values and the introduction of selective temporary value erasure to reduce the amount of required memory, in addition to various optimizations. During the development process, we have implemented a Haskell binding for MPFI and devised a method for eliminating the type ambiguity problem, related to the use of type class in Haskell programming, in a realistic way.

研究分野：プログラミング言語処理系

キーワード：実数計算ライブラリ　Haskell　遅延評価　無限リスト　多倍長浮動小数点演算ライブラリMPFR　区間演算ライブラリMPFI　計算精度保証

１．研究開始当初の背景

(1) 自然科学における実験技術や測定器の精度向上に伴い，実験結果の詳細な検証のために高精度な数値シミュレーションが求められ続けている．しかしながら，標準的な浮動小数点表現に基づく浮動小数点演算では，一連の演算の結果として蓄積される誤差の排除が容易ではなく，また，得られた計算結果の正確さの程度を判断するのも難しい．多倍長での浮動小数点演算が可能な MPFR 等のライブラリは桁落ちによる精度劣化の影響を抑えるために有用であるが，浮動小数点演算を用いる限りは計算誤差排除への腐心が不可欠である（この状況は現在も変わらない）．

(2) 正確な計算結果を得るための方法として，「実数計算」ライブラリ，すなわち，個々の数値を無限リストや関数合成で表現するなどオンデマンドな計算を可能にする機構の導入によって計算結果に対してユーザが任意の有効桁数を要求した計算を可能にする数値計算ライブラリ（以下，単に実数計算ライブラリと呼ぶ）に関する様々な手法が研究されている．しかし，浮動小数点演算に比べて実数計算ライブラリによる算術演算に要するコストは大きい．我々が開発中であったライブラリ（IFN ライブラリ）も，既存手法に対して優位とは言えなかった[3].

２．研究の目的

(1) 本研究では，数値計算プログラミングにおける計算誤差への配慮に要するユーザの負担低減の実現，すなわち「実数計算」ライブラリを数値計算において容易に活用できる環境の提供を目指す．そのために，我々が開発中の IFN ライブラリの可用性向上と性能改善を目的とする．

(2) 本研究ではまた，大規模計算に耐えうる実数計算ライブラリの実現可能性を追求する．そのために，IFN ライブラリと他の手法との融合による実数計算ライブラリの実現方式の検討と，より柔軟で高性能なライブラリの実現のための知見の蓄積を目的とする．

３．研究の方法

(1) IFN ライブラリは，精度保証機能付きの近似値計算，および，適応的な精度改善のための再計算制御の二つのソフトウェア階層によって実現されている[3]．本研究ではまず，IFN ライブラリを構成する各階層それぞれについて，構成要素を分析し高速化方針を検討した．そして，Haskell および C による汎用的に利用可能な API を設計，実装した．

(2) 既存の実数計算ライブラリには，Exact Real Arithmetic (ERA)や iRRAM が挙げられる．IFN ライブラリにおける計算精度制御はボトムアップで適応的であり，また，精度改善のための再計算は部分式単位で局所的に行う[3]．これに対し，ERA では First Binary Cauchy Sequences (FBCS)に基づくトップダウンな計算精度見積りによる反復的な計算の排除手法が取られ，また iRRAM では大域的な再計算機構が用いられる．これらの違いの効果の評価，および，各技法の組み合わせによる高性能な実数計算ライブラリの実現可能性を実験的に考察した．

４．研究成果

(1) IFN ライブラリに対して大幅な性能改善が得られた．Haskell 版 IFN ライブラリでは，条件数の極めて大きい係数行列を持つ敏感な連立一次方程式（以下，ヒルベルト問題と呼ぶ）の求解での比較では，従来版プロトタイプ[3]に対して 1 桁の高速化，問題サイズによっては 2 桁以上の計算時間短縮を実現した[1]．この性能改善は，数値を表すデータ型の実装最適化や各部のパラメータチューニングのほか，精度保証付き近似値演算への区間演算ライブラリ MPFI の採用，選択的中間値消去手法の導入による所要記憶容量削減等を総合して得られた効果である．

C 版（低水準言語版）IFN ライブラリは，予備評価では Haskell 版よりも高速（小規模な問題，例えば次元数 128 のヒルベルト問題求解時に約 2 割高速）であるが，大規模な問題の場合にはその差は小さくなる傾向が見られる．Haskell 版および C 版とも，語依存性を低く抑えた実装がなされているといえる．

以下の項目では，IFN ライブラリの高速化や可用性向上に関連する要素技術の成果について述べる．

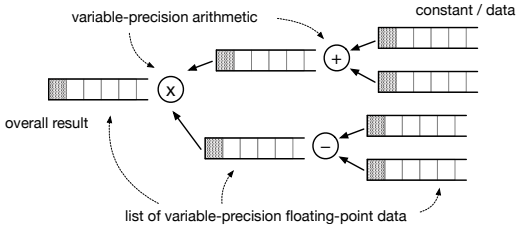

図 2　IFN のデータフローグラフのイメージ（[1]より）

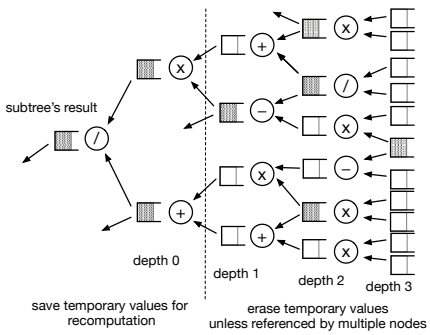

図 1　選択的中間値消去の効果（[1]より）

(2) 選択的中間値消去と呼ぶ手法により，IFN ライブラリの実行に要する所要記憶容量の大幅削減を

達成した[1]．IFN ライブラリによる算術演算は図 1 に示すようなデータフローグラフで表現され，アプリケーションプログラムにおいてループ構造による反復計算がなされる場合にはデータフローグラフは反復数分展開して連ねられたものとなる（計算結果の精度改善のための再計算に必要）．すなわち，IFN ライブラリを用いた数値計算のナイーブな実行では，総計算量に比例した記憶容量が必要となる．つまり，個々の算術演算の計算結果（暫定精度の近似値）を全て保持し続けるのは非現実的であるが，初期版の IFN ライブラリではこの点について工夫が無く，規模の大きな問題を解くことができなかった．これに対し，選択的中間値消去手法は，図 2 に示すように，グラフの末端に近くかつ複数箇所に向かう有向エッジが出ていない（共通部分式になっていない）節の計算結果の暫定値を保持する時間を短く抑えることにより，所要記憶容量の大幅な削減を実現する．

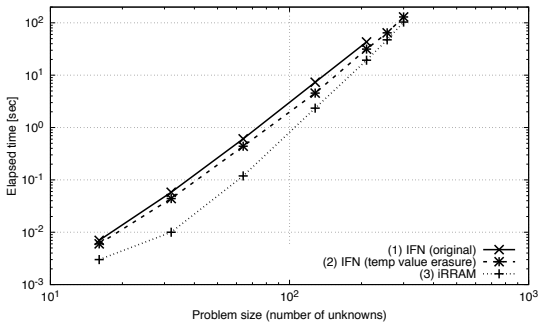

図 3　ヒルベルト問題の求解に要する時間（[1]より）

　中間値の削除は，精度改善のための再計算が必要となった場合に大きな計算量的ペナルティを生じさせ得る．しかしながら，実測結果によれば，適度なパラメータ設定で十分良好な振る舞いにできることが確認されている[1].

(3) Haskell 版 IFN ライブラリの改良は，MPFI ライブラリの Haskell バインディングの設計および実装によりなされた．IFN ライブラリにおける精度保証近似値計算に MPFI ライブラリが有効であることを確認した上で，これを Haskell プログラムから利用するために，MPFI バインディング hmpfi を設計し実装した．MPFI へのバインディングは，MPFR ライブラリに対する Haskell バインディングである hmpfr を拡張したものであるが，個々の暫定値に対応する区間の精度を評価する API の導入などの非自明な取り組みを経て実現されている．実測によると，MPFR と MPFI の処理速度の差は hmpfr と hmpfi の差とほぼ等しく，オーバヘッドを抑えた実装であることが確認されている．

(4) 既存の高性能な実数計算ライブラリ iRRAM と Haskell 版 IFN ライブラリの比較の結果，当初 1 桁以上開いていた計算速度差を大幅に短縮する高速化が達成できていることが確認された（図 3；なお図中の曲線(1)は本研究開始当初の IFN ライブラリのものではない）．iRRAM は C++を用いて実装されているが，main 関数から制御を開始する通常のプログラム記述ではなく特別な環境下での計算実行が必要であるので，Haskell プログラムからの利用は容易ではないと予想される．Haskell 版 IFN ライブラリは，iRRAM が想定していないシナリオをカバーする有用性を有するといえる．

(5) Haskell による数値計算プログラムの実装や利用においては型クラスの利用が不可欠であるが，型クラスの利用は型推論において型を決定できない場面を生じさせ得る．本研究では，型の唯一性検査によって型の曖昧性が解決される可能性を高める手法を提案し，評価した[2].

(6) IFN ライブラリの高速化のためのアプローチに関して，試行錯誤によって様々な知見を得た．例えば，精度保証近似計算の実装には多倍長区間演算ライブラリ MPFI が従来のプロトタイプより適していること，FBCS による完全なトップダウンな計算精度制御に基づく実数計算ライブラリの実装方式は部分式に対して過度な計算精度要求をしがちで高速処理には向かないであろうこと（FBCS に MPFR を組み合わせる方法も有効ではないこと）などが挙げられる．並列処理による IFN ライブラリの高速化に関しては，Haskell 処理系である GHC のランタイムライブラリを用いた細粒度並列処理導入の検討の結果，互いに独立な部分式を並列に評価するという程度の戦略では forkIO は不向きで rpar/rseq ならば効果が期待できることが確認されている．そのほか，現状の IFN ライブラリでは依然として部分式に要求する精度の見積りが粗く過度に大きな計算精度による部分式計算が行われがちで，精度改善のための再計算処理のコストの低減が課題となるなどの改善の余地が明らかになっている．

＜引用文献＞

1. 川端 英之，実数計算ライブラリ IFN の Haskell による効率的な実装，情報処理学会第 126 回プログラミング研究発表会（PRO126），Oct. 30，2019.
2. Yuya Kono, Hideyuki Kawabata, and Tetsuo Hironaka, Resolving Ambiguous Types in Haskell by Checking Uniqueness of Type Variable Assignments under Type Class Constraints, Journal of Information Processing, Vol. 27, pp.87-94, 2019.
3. Hideyuki Kawabata and Hideya Iwasaki, Improving Floating-Point Numbers: A Lazy Approach to Adaptive Accuracy Refinement for Numerical Computations, Proc. ESOP 2016, LNCS 9632, pp.390-418, 2016.

５．主な発表論文等

〔雑誌論文〕　計1件（うち査読付論文　1件／うち国際共著　0件／うちオープンアクセス　1件）

１．著者名 Kono Yuya, Kawabata Hideyuk, and Hironaka Tetsuo	４．巻 27
２．論文標題 Resolving Ambiguous Types in Haskell by Checking Uniqueness of Type Variable Assignments under Type Class Constraints	５．発行年 2019年
３．雑誌名 Journal of Information Processing	６．最初と最後の頁 87～94
掲載論文のDOI（デジタルオブジェクト識別子） 10.2197/ipsjjip.27.87	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 －

〔学会発表〕　計4件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 川端 英之
２．発表標題 実数計算ライブラリ IFN の Haskell による効率的な実装
３．学会等名 情報処理学会第126回プログラミング研究発表会
４．発表年 2019年

１．発表者名 河野 雄也，川端 英之，弘中 哲夫
２．発表標題 Haskellにおける型クラス制約を満足する型変数割当ての唯一性検査に基づく型の曖昧性解決
３．学会等名 情報処理学会第119回プログラミング研究発表会
４．発表年 2018年

１．発表者名 余頃花純，川端英之，弘中哲夫
２．発表標題 区間演算ライブラリMPFIを用いた実数計算ラ イブラリIFN-Hの記憶領域管理をHaskellのみで記述した設計と実装
３．学会等名 情報処理 学会第80回全国大会
４．発表年 2018年

１．発表者名 小林 周太郎，川端 英之，弘中 哲夫
２．発表標題 末尾再帰でない再帰プログラムの高速 化のための最適化に関する一考察
３．学会等名 日本ソフトウェア科学会第34回大会
４．発表年 2017年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考