クリロフ部分空間法の計算精度依存性

長谷川秀彦（筑波大学・図書館情報学系）

1　はじめに

BiCG, CGS, Bi-CGSTAB, GPBi-CG などに代表される Krylov Subspace Methods は、クリロフ部分空間内の基底ベクトルを探索するアルゴリズムであり、計算が「正確に」行われれば高々n 回の反復で収束する（ n は行列サイズ）．ところが、コンピュータ上の数値計算では「丸め誤差 (rounding error) 」の影響によって理論通りにはいかない．本研究では、多倍長演算を利用することで「正確な数値演算」を実現し、これらの解法に丸め誤差がどう影響しているかを実験的に示す．

2　実験のあらまし

多倍長演算が可能なソフトウェアとして Omni OpenMP コンパイラ [1] を用いる．Omni OpenMP (Fortran) コンパイラに用意されている MULTIPLE PRECISION 型の変数では、コンパイル時に仮数部 (Mantissa) のビット数を指定する．これにより、同一のプログラムの演算精度を変えて実行できる．ここでは、一般の倍精度浮動小数演算相当の 53 ビット、４倍精度相当の 100 ビット、 200 ビット、 300 ビットと４種類を扱う．なお MUPLTIPLE PRECISION 型の制限により仮数部は 53 ビット以上なので、倍精度演算と同一にはならない．

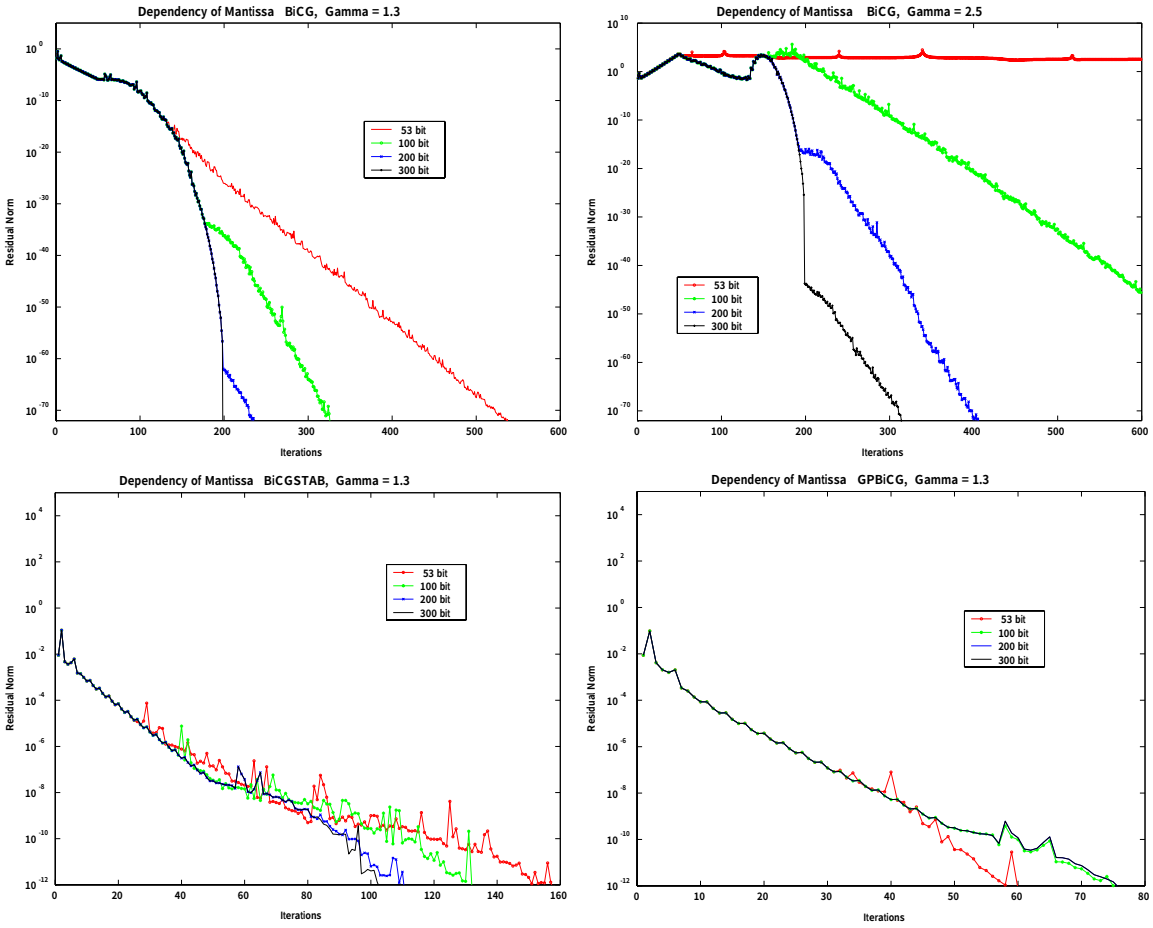
実験には BiCG, CGS, BiCGSTAB, GPBiCG を取りあげ、問題のパラメータと演算精度を変えて収束の変化をみた．テスト問題は、これらの解法の比較によく用いられる Toeplitz Matrix で、 $n = 200$ である．要素は $a_{i,i-2} = \gamma$, $a_{i,i} = 2$, $a_{i,i+1} = 1$ 、右辺は $b_i = 1$ 、収束判定は相対残差ノルムで 10^{-12} とした．γ は 1.3, 1.7, 2.1, 2.5 の４通りである．条件数は 6.4 ($\gamma = 1.3$) から 700 程度で ($\gamma = 2.5$) 、大きくはないが虚数軸方向に固有値が分布するため、これらの解法にとって収束しにくい問題と言われている．

3　実験結果

BiCG 法 $\gamma = 1.3$ （左上）では、途中まで４本のグラフが重なって、どの精度でも正しい結果が得られていることがわかる．いっぽう、仮数部 53 ビットの相対残差ノルム 10^{-15} 以下は見かけ上の収束で、正しい収束ではないこともわかる．この問題では、100 ビットで 10^{-35} まで、 200 ビットで 10^{-60} まで正しく収束する．反復過程に現れるスカラー量 α, β は 10^{-5} から 10^{5} の範囲に分布した．

$\gamma = 2.5$ （右上）では、仮数部 53 ビット、 100 ビットは収束せず、 200 ビットで 10^{-15} まで、 300 ビットで 10^{-40} まで収束する．このときにも、少数の例外はあるが、反復過程に現れる α, β は $\gamma = 1.3$ の場合と大差なかった．

CGS 法はより多くのビット数が必要だったが、BiCG 法と同様の収束を示した．



BiCGSTAB 法 $\gamma = 1.3$ （左下）のとき、仮数部 53 ビットは約 160 回で収束する．仮数部を増やすと 10^{-6} あたりからグラフは分離してより少ない反復回数で収束するが、この結果は演算精度に敏感であることを意味する．$\gamma = 2.5$ のときは 1500 ビットで 210 反復となり、事実上、収束しなかった．

GPBiCG 法 $\gamma = 1.3$ （右下）のとき、仮数部が 100 ビット以上よりも 53 ビットが良い収束を示す．53 ビットのグラフだけが分離していることから、誤差によって収束が加速されたことを表している．$\gamma = 2.5$ のときは 300 ビットで 320 反復だった．

4 むすび

これらからわかることは、(1) 多倍長演算を用いるとスムーズで速い（理論に近い）収束が得られること、(2) 必要な演算精度は問題によって異なること、(3) 必要な演算精度はアルゴリズムに強く依存すること、などである．そのいっぽう、丸め誤差の影響は収束を加速する方向に作用することもあり、単純ではない．今後、より現実的な比較・評価・検討をおこなう必要がある．

参考文献

[1] Omni OpenMP Compiler Project, http://phase.hpcc.jp/Omni/