古典的アルゴリズムによる大規模対称帯行列の固有値解析

EIGENVALUE ANALYSIS FOR LARGE SYMMETRIC BAND MATRICES WITH LEGACY ALGORITHMS

長谷川秀彦[1)]
Hidehiko HASEGAWA

1) 博士（工学）筑波大学教授　図書館情報メディア研究科（〒 305-8550 つくば市春日 1-2, hasegawa@slis.tsukuba.ac.jp）

The author shows performance results of eigenvalue analysis for large symmetric band matrices with some legacy and stable algorithms on a PC. Until now the computation speed was accelerated by CPU's clock speed, however in future it will be accelerated by Mulicore CPU with slow clock speed. It is necessary to distinguish whether these legacy and stable eigenvlaue analysis codes for the band matrices can be survive in future Multicore environment or not.
Key Words : *multicore, eigenvalue algorithms, symmetic band matrices, performance*

1. はじめに

固有値問題には、標準固有値問題 $Ax = \lambda x$ と、一般固有値問題 $Ax = \lambda Mx$ があり、$x \neq o$ となる x を固有ベクトル、 λ を固有値という。行列 A, M が実数か複素数か、対称か非対称かによって問題の数学的な難しさが大きく異なる。本発表では、A が実対称の標準固有値問題を扱う。行列を半分だけ格納すればいいことに加え、すべての固有値、固有ベクトルが実数になることが数学的に保証される。

実対称行列に対する標準固有値問題では、ハウスホルダー変換などを用いた相似変換により行列 A を３重対角化し、３重対角行列に対してスツルム２分法やQR法を用いて固有値と固有ベクトルを求め、３重対角行列の固有ベクトルに相似変換の逆変換を施して元の行列の固有ベクトルを求める。相似変換は固有値を変えないので、固有ベクトルが不要なら計算量を大幅に節約できる。しかし、この方法は N * N のメモリ領域を必要とし、３重対角化と逆変換では $O(N^3)$ の演算を必要とする。

いっぽう、数値シミュレーションで扱う行列は大規模かつ疎で、メモリ上に N $*$ N の領域を確保するのは容易でない。連立一次方程式の解法として反復法を使うように、固有値解法でも「疎な構造をそのまま保つ」アルゴリズムが必要となる。

これまで、 CPU の高速化に対応してこれらのアルゴリズムも自然と高速化されてきた [1]。いっぽう、これまでの高速化の多くがクロックスピードの高速化によるものであるのに対し、最近の高速化は CPU のマルチコア化によってなされ、クロックスピードはむしろ遅くなっている。本報告では、このような環境で古典的な対称帯行列に対するコードが有効性を保てるかという点について議論する。

2. 帯行列に対する古典的アルゴリズム

非ゼロ要素が対角要素から m1 までの範囲に限定される行列を帯行列といい、疎行列は帯行列とみなせる。対称帯行列に対する古典的な固有値解析アルゴリズムには、「村田法（mura2)」と「帯行列に対するスツルム逆反復法（eigv2)」がある [2]。

村田法では、帯の外側に発生する非ゼロ要素を複数のハウスホルダー変換によって消去し [3] 、帯構造を保ちながら３重対角化し、２分法によって３重対角行列の固有値を求める。QR法は N $*$ N の領域を必要とするためここでは使用できない。３重対角行列の固有値をシフト点として逆反復法を行い、元の行列に対する固有値と固有ベクトルを求める。３重対角化は必須で、その後の処理は固有値・固有ベクトルの組数に応じた時間とメモリが必要になる。固有ベクトルが不要なら、３重対角行列に対する２分法で打ち切ればよい。

帯行列に対するスツルム逆反復法は、３重対角行列に対する２分法・逆反復法と同様だが、スツルムカウントと逆反復の対象が対称帯行列になる。固有値カウントには、部分軸選択付きガウスの消去法と類似の Martin-Wilkinson の特殊ガウスと、軸選択なしの対称帯ガウスを使う。逆反復法のかわりに同時逆反復法を使用してもいいし、一般固有値問題への拡張もできる。

これらのアルゴリズムの高速化には、帯行列に対するハウスホルダー３重対角化と帯行列に対するガウスの消去法がキーとなる。逆反復法におけるシフト点の移動 $A - \alpha I$ では、行列が悪条件になりやすいため部分軸選択付きのガウスの消去法が必要であり、部分軸選択を回避するアルゴリズムでは数値的な問題が生じる。単一 CPU のベクトル型スーパーコンピュータ向けコードが [2] に公開されている。

3.　予備評価実験

２次元一般化ポアソン方程式を差分法で離散化した規則的疎行列（正定値）を対称帯行列とみなして、 [2] に示されているコード mura2, eigv2 をそのまま実行した。次元は $N = 2 * m1 * (m1 + 3)$ で、小さいほうから $2 * m1$ 組の固有値・固有ベクトルを求めた。

計測は、パソコンまたは並列計算機の 1 CPU を用い、メモリ容量を 2GB とした。「1 CPU で逐次コードを実行する」想定なので、複数 CPU のマシンでも 1 CPU のみとした（複数 CPU を使用する設定にしていないだけで、特殊な設定をしたわけではない）。

表–1　村田法 PowerPC 2GHz (sec)

N/m1	３重対角	２分法	逆反復	合計
45,450	2,690	5.01	352	3,050
(150)	(88%)		(11%)	51 分
80,600	13,000	11.6	1,310	14,300
(200)	(90%)		(9%)	4 時間
125,750	48,200	23.0	4,580	52,800
(250)	(90%)		(8%)	15 時間

表–2　村田法 m1=100/N=20,300 (sec)

	３重	２分	逆反	合計
Power3-II (400M)	962	4.23	163	1,130
Xeon (3.2G)	364	1.05	54.9	420
PowePC (2G)	392	1.40	65.8	460
Itanium2 (1.3G)	453	1.92	81.3	536

表–3　スツルム逆反復法 PowerPC 2GHz (sec)

N/m1	２分法	逆反復	合計
45,450	2,410	990	3,400
(150)	(70%)	(29%)	50 分
80,600	15,200	7,630	22,800
(200)	(67%)	(33%)	6 時間
125,750	52,800	20,700	73,500
(250)	(71%)	(28%)	20 時間

表–4　スツルム逆反復法 m1=100/N=20,300 (sec)

	２分法	逆反復	合計 (比)
Power3-II (400M)	918	373	1,290 (2.8)
Xeon (3.2G)	305	156	462 (1.0)
PowerPC (2G)	324	138	463 (1.0)
Itanium2 (1.3G)	318	148	467 (1.0)

村田法 mura2 では、ハウスホルダー３重対角化が全実行時間の 80% 以上を占め、その比率は問題が大規模になるにしたがって大きくなる。逆反復法では、十数パーセントの実行時間で 2*m1 組の固有値・固有ベクトルの組を求めているが、個々の組の処理を分離することで並列化が可能である。固有ベクトルの直交性を保つための処理は並列化の阻害要因となりうるが、それでも逐次処理になるわけではない。いっぽう、ハウスホルダー３重対角化はひとつの処理であり、内部での演算が並列化可能かを見極める必要がある。

帯行列に対するスツルム逆反復法 eigv2 の実行時間は、２分法が 70% 程度、帯行列に対する逆反復法が 30% 程度である。２分法を多分法に置き換えることも可能 [4] である。しかし、両方の処理で核となっているのは帯行列に対する部分軸選択付きのガウスの消去法なので、並列化による性能向上は部分軸選択付きガウスの消去法の並列化にかかってくる。同様なことは、村田法における逆反復法にもいえる。

対称帯行列に対する古典的な固有値解析の高速化のキーは、共有メモリにおける「対称帯行列に対するハウスホルダー３重対角化」と「帯行列に対する部分軸選択付きガウスの消去法」の並列化となる。しかし、 SMP マシンで OpenMP を用いて並列化した経験 [5] から考えると、これらの並列化は容易でない。実際、SMP とマルチコアでは処理の粒度が異なるので、測定してみないと正確なことはわからない。

4.　まとめ

近年のパソコンでは 2GB を越えるメモリが使えるため、古典的アルゴリズムでも結構な問題が解ける。しかし、最近のマルチコア化では共有メモリ上の並列化が前提となっているため、高速化にはどれだけ並列化できるかにかかってくる。一般に、限られたメモリで処理を行う帯行列アルゴリズムは逐次的にならざるをえず、並列化の効果が乏しい。今後、古典的アルゴリズムの性能で満足できるのか、それとも疎行列のままで並列処理できる新アルゴリズムが必須なのかの見極めが重要になってくる。

マルチコアでの性能は当日示す。

参考文献

1) 長谷川秀彦 :古典的アルゴリズムによる大規模対称帯行列の固有値計算，第 10 回計算工学講演会論文集 第 10 巻 第 2 号, pp. 247-248, 2005.
2) 小国力編著 :行列計算ソフトウェア，丸善，1991.
3) Kenro Murata, Kiyomi Horikoshi :A New Method for Tridiagonalization of the Symmetirc Band Matrix，Inf. Proc. Jap. 15, pp. 108-112, 1975.
4) 村田健郎他 :スーパーコンピュータ：科学技術計算への適用, 丸善, 1985.
5) 長谷川秀彦： OpenMP を用いた帯行列に対する直接解法の並列化，情報処理学会論文誌：コンピューティングシステム， Vol. 45, No. SIG 6(ACS 6), pp. 86-94, 2004.