

非対称行列向けマルチシフト QR 法の性能予測方式

山 本 有 作†

Braman らによって提案された small-bulge マルチシフト QR 法は、キャッシュマシン上で非対称密行列の固有値を求めるための効率的なアルゴリズムである。しかし、このアルゴリズムの性能を引き出すには、対象とする計算機アーキテクチャと行列サイズ n に応じて、ブロックサイズ m を適切に設定することが不可欠である。そこで本報告では、ブロックサイズの最適値を実行前に求めるための性能予測モデルを提案する。本モデルは階層型のモデリング手法に基づいており、 n 、 m の値と、レベル 3 BLAS、ダブルシフト QR 法ルーチンなどのアルゴリズム構成要素の性能データとを入力として、small-bulge マルチシフト QR 法全体の実行時間を予測する。本モデルを用いて PowerPC G5 および Opteron 上での実行時間を予測したところ、 m の値による実行時間の変化を定性的に再現することができた。

Performance Modeling of a Multishift QR Algorithm for the Nonsymmetric Eigenvalue Problem

YUSAKU YAMAMOTO†

The small-bulge multishift QR algorithm proposed by Braman, Byers and Mathias is one of the most efficient algorithms for computing the eigenvalues of nonsymmetric matrices on processors with hierarchical memory. However, to fully extract its potential performance, it is crucial to choose the block size m properly according to the target architecture and the matrix size n . In this paper, we construct a performance model for this algorithm. The model has a hierarchical structure that reflects the structure of the original algorithm and given n , m and the performance data of the basic components of the algorithm, such as the level-3 BLAS routines and the double implicit shift QR routine, predicts the total execution time. Experiments PowerPC G5 and Opteron processors show that the variation of the execution time as a function of m predicted by the model agrees with the measurements.

1. はじめに

QR 法¹⁾¹²⁾¹⁵⁾ は非対称密行列の固有値を求めるための信頼性の高いアルゴリズムとして広く使われている。近年では、シミュレーションの大規模化・精密化に伴って固有値計算が大型化するとともに、PC やワークステーション上で計算を行うケースが増えつつある。そのため、これらのマシン上で大規模行列に対する QR 法を高速に実行することが重要な課題となっている。

PC やワークステーションで使われているマイクロプロセッサは多種にわたるが、これらはいずれも数百 KB ～数 MB のキャッシュを持ち、データ参照の局所性を高めてキャッシュを有効利用することがプログラムの性能向上のために決定的に重要である。しかし、

オリジナルの QR 法はデータ参照の局所性が低い。実際、広く利用されているダブルシフト QR 法を $n \times n$ のヘッセンベルグ行列に対して適用した場合、反復 1 回あたりの演算量は $O(n^2)$ であり、かつ行列要素すべてへのアクセスが必要である。したがって、アクセス 1 回あたりの演算量は高々 $O(1)$ となる。このため、キャッシュが重要な最近のマイクロプロセッサにおいてオリジナルの QR 法を高速に実行することは困難である。

このため、QR 法の高速化に向けて様々な試みが行われている²⁾⁴⁾¹⁰⁾¹⁴⁾¹⁶⁾¹⁷⁾。中でも、Braman らによる small-bulge マルチシフト QR 法⁴⁾ はもっとも有望な手法である。このアルゴリズムは、複数個 (m 個) のシフトを用いて m 回分の反復を同時に行うマルチシフト QR 法²⁾ の一種であり、ダブルシフト QR 法と比べてデータ参照の局所性を $m/2$ 倍に高めることが可能である。従来のマルチシフト QR 法²⁾ では数値的不安定による収束特性の悪化という問題があったが¹⁰⁾¹⁸⁾、

† 名古屋大学大学院工学研究科 計算理工学専攻
Department of Computational Science and Engineering,
Nagoya University

small-bulge マルチシフト QR 法では、反復における更新演算の仕方を工夫することでこの不安定性を解消し、ダブルシフト QR 法と同等の収束性を実現している。この結果、Origin2000 による評価では、データ参照の局所性向上の効果により、LAPACK¹⁾ の 3 倍の性能が得られることが報告されている。

small-bulge マルチシフト QR 法の性能を最大限に引き出すには、シフト数（ブロックサイズ） m を適切に選ぶことが必要である。 m の値を大きくするとデータ参照の局所性は向上するが、シフトの計算のための演算量が增大する。そのため、最適な m の値は対象とする計算機アーキテクチャと行列サイズにより異なる。実験により m の最適値を求めようとすると、多大な時間を要してしまう。

そこで本報告では、small-bulge マルチシフト QR 法に対する実行時間モデルを提案する。本モデルは、Dackland ら⁷⁾ と Cuenca ら⁶⁾⁵⁾ により提案された階層型のモデリング手法に基づいており、アルゴリズムの構成要素であるレベル 3 BLAS ルーチン、ダブルシフト QR 法ルーチンなどの性能データを入力することで、行列サイズ n 、ブロックサイズ m の場合の実行時間を予測する。本モデルの利用により、 m の値を実行前に最適化することが可能となる。

以下では、まず 2 章で small-bulge マルチシフト QR 法のアルゴリズムを説明し、3 章で我々の提案するモデルの詳細を述べる。次に 4 章で、モデルによる性能予測結果を紹介する。最後に 5 章でまとめを述べる。

2. small-bulge マルチシフト QR 法

2.1 従来のマルチシフト QR 法とその問題点

$n \times n$ のヘッセンベルグ行列 A に対する QR 法を考え、QR 法の反復を l 回行って得られる行列を A_l とする。ダブルシフト QR 法では、 A_l から A_{l+2} を計算するため、 A_l の右下隅の 2×2 行列の固有値 σ_1, σ_2 を求め、これらをシフトとして QR 法の反復を 2 回分同時に行う。実際の計算では、 A_l に対してハウスホルダー変換を行うことにより、その左上隅に、2 個のシフトの情報を含む 4×4 のバルジ（ヘッセンベルグ形からの出っ張り）を導入し、それをハウスホルダー変換によって 1 行ずつ右下に移動させ、最後に右下隅から追い出すという処理を行う⁸⁾¹³⁾。これをバルジ・チェイシングと呼ぶ。このバルジ・チェイシングにより得られる行列が A_l に対して QR 法の反復を 2 回行って得られる行列に一致することは、陰的 Q 定理⁸⁾¹³⁾ により保証される。

このアルゴリズムを一般化することにより、マルチ

シフト QR 法が得られる²⁾。マルチシフト QR 法では、 A_l の右下隅の $m \times m$ 行列の固有値 $\sigma_1, \sigma_2, \dots, \sigma_m$ を求め、これらをシフトとして用いる。Bai らが提案したオリジナルのマルチシフト QR 法では、この m 個のシフトを用いて QR 法の反復を m 回分同時に行うため、 A_l の左上隅にシフト情報を含む $(m+2) \times (m+2)$ の大きなバルジを導入し、これをバルジ・チェイシングによって右下隅から追い出すことにより、 A_{l+m} を計算する。これにより、この計算では、行列の各要素に対して $O(m)$ の演算を行うことができ、ダブルシフト QR 法と比べてデータ参照の局所性を $m/2$ 倍に向上できる。このアルゴリズムは、LAPACK の DHSEQR に採用されている。しかし、このアルゴリズムでは、バルジが大きくなると数値的不安定が生じるため、 m を高々 10 程度にしか取れない¹⁸⁾。そのため、最近のマイクロプロセッサではキャッシュを十分に活用できず、性能が発揮できないという問題があった。

2.2 small-bulge マルチシフト QR 法

この問題を解決するため、Braman, Mathias, Byers は small-bulge マルチシフト QR 法と呼ばれる新しいアルゴリズムを開発した⁴⁾。この手法では、 m 個のシフトを 2 個ずつの $m/2$ 組に分け、バルジ・チェイシングにおいては 4×4 の小さいバルジを $m/2$ 個同時にチェイシングする。このアルゴリズムは、数学的には Bai らのマルチシフト QR 法と等価であるが、大きなバルジに伴う数値的不安定を回避でき、ダブルシフト QR 法と同等の収束性を達成できる。

Small-bulge マルチシフト QR 法におけるバルジ・チェイシングは、次に述べる通り 3 つのフェーズに分けられる。

2.2.1 フェーズ I：バルジ群の導入

まずフェーズ I では、行列の左上隅より、 4×4 のバルジ $m/2$ 個を連続して導入する。バルジ間での演算の干渉を避けるには、隣り合うバルジは 3 行以上離す必要があるため⁴⁾、このフェーズが終了すると、行列の第 1 行目から第 $3m/2+1$ 行目にかけて $m/2$ 個のバルジが並ぶ（図 1）。フェーズ I におけるバルジ・チェイシングは、 A_l の左上隅の $(3m/2+1) \times (3m/2+1)$ 行列（これを $A_{l,1:3m/2+1,1:3m/2+1}$ と書く）のみを使って行えることに注意すると、フェーズ I の演算を次の 2 つの部分に分けることができる（図 2 参照）。

(a) $A_{l,1:3m/2+1,1:3m/2+1}$ に対してハウスホルダー変換を繰り返してバルジ・チェイシングを行い、 $m/2$ 個のバルジを第 1 行目から第 $3m/2+1$ 行目に持ってくる。同時に、これらのハウスホルダー変換の積を $(3m/2+1) \times (3m/2+1)$ 行列 U として蓄

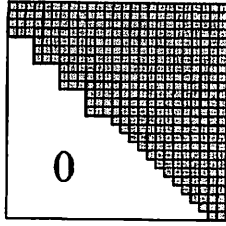


図1 フェーズ I 終了後の行列 ($m = 8$)

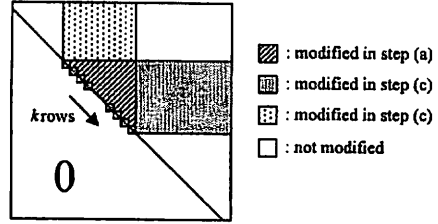


図3 フェーズ II における処理の分割

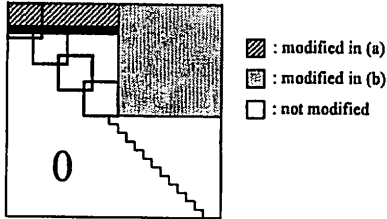


図2 フェーズ I における処理の分割

積する。

(b) 第1行～第 $3m/2 + 1$ 行の残りの部分

$A_{i,1:3m/2+1,3m/2+2:n}$ (非対角ブロック) に U を左側から掛けることにより、この部分を更新する。

この2つのうち、ステップ (b) は完全にレベル3 BLAS、すなわち行列乗算を用いて実行できる。ステップ (a) はレベル3 BLAS を用いては行えないが、 $m \ll n$ ならば、この部分がフェーズ I の演算量に占める割合は小さい。したがって、演算のほとんどをレベル3 BLAS を用いて行うことが可能となる。

2.2.2 フェーズ II：中心部の演算

フェーズ II では、行列の最初の $3m/2 + 1$ 行にあるバルジ群をチェイシングし、最後の $3m/2 + 1$ 行にまで持ってくる。いま、ある整数 k を決め、バルジ群を k 行下へ移動させる処理を一まとまりの処理として考える (図3)。 $m/2$ 個のバルジは $3m/2 + 1$ 本の行を占めるため、この処理に関わる行・列はそれぞれ $3m/2 + k$ 本である。フェーズ I の場合と同様、この処理を3つの部分に分けることができる。

- (a) $(3m/2 + k) \times (3m/2 + k)$ の対角ブロック内部でバルジ・チェイシングを行い、そこで用いたハウスホルダー変換の積を $(3m/2 + k) \times (3m/2 + k)$ の行列 U として蓄積する。
- (b) $3m/2 + k$ 本の行の非対角ブロックに左から U を掛け、この部分を更新する。
- (c) $3m/2 + k$ 本の列の非対角ブロックに右から U を掛け、この部分を更新する。

フェーズ I と同様に、ステップ (b)、(c) はレベル3

BLAS のみを用いて行うことができるため、フェーズ II も演算のほとんどをレベル3 BLAS で実行できる。なお、Braman らは演算量の解析により、 $k \sim \frac{3}{2}m$ とするのが演算量最小化の面から最適であることを示している。

2.2.3 フェーズ III：バルジ群の追い出し

フェーズ III では、行列の最後の $3m/2 + 1$ 行にあるバルジ群をチェイシングして右下隅から追い出す。この処理もフェーズ I と同様、2つの部分に分けることができ、演算量のほとんどを占める非対角ブロックの更新はレベル3 BLAS を用いて行える。

2.2.4 その他の処理

small-bulge マルチシフト QR 法の演算としては、以上のフェーズ I ~ III の処理に加え、シフト $\sigma_1, \sigma_2, \dots, \sigma_m$ を求めるための演算が必要である。また、計算が進行して行列が十分に小さくなった場合は、ダブルシフト QR 法に切り替えて残りの行列の固有値計算を行う。しかし、全計算量の中でこれらの計算の占める割合は小さい。このため、small-bulge マルチシフト QR 法では演算の大部分をレベル3 BLAS で行うことができ、キャッシュの有効利用が可能となる。Braman らは、Origin2000 上での評価において、本アルゴリズムが LAPACK DHSEQR の3倍高速であると報告している⁴⁾。

2.3 アルゴリズム中で用いられる演算ルーチン

前節での説明から明らかなように、small-bulge マルチシフト QR 法は次の4種類の演算ルーチンから構成される。

- (A) 対角ブロック内部でのバルジ・チェイシング。以下では、フェーズ I, II, III のためのバルジ・チェイシングのルーチンをそれぞれ BCHASE1, BCHASE2, BCHASE3 と書く。
- (B) 非対角ブロックの更新を行うためのレベル3 BLAS ルーチン。具体的には、密行列乗算 $C = AB$ を行うルーチン DGEMM を用いる。行ブロックの更新では、 A も B も非転置の場合のルーチン (DGEMM-NN)、列ブロックの更新では A が非転

置、 B が転置の場合のルーチン (DGEMM-NT) が必要である。

- (C) 非対角ブロックの更新 (行列 U による乗算) の結果を元の行列に書き込むためのコピールーチン。行ブロック、列ブロックのコピーのためのルーチンをそれぞれ COPY1, COPY2 と書く。
- (D) シフトの計算、および A_l のサイズが十分小さくなったときの A_l の固有値の計算に用いるダブルシフト QR 法のルーチン。これには、EISPACK の HQR を用いる。

次章で述べる階層型の性能モデル作成においては、これらのルーチンを基本演算ルーチンとして用いる。

2.4 最適ブロックサイズ

small-bulge マルチシフト QR 法の性能を引き出すには、ブロックサイズ m を適切に選ぶことが必要である。一般に、 m が大きいほどデータの再利用性は向上し、レベル 3 BLAS 部分の性能を高くできるが、その反面、シフト計算の演算量、および BCHASE1~3 の演算量は m とともに増大する。最適な m はこのトレードオフによって決まり、対象とする計算機や行列サイズ n により大きく異なる。たとえば Braman らは、Origin2000 上での評価において、 $1000 \leq n \leq 1999$ の場合は $m = 60$ 、 $2000 \leq n \leq 2499$ の場合は $m = 116$ 、 $2500 \leq n \leq 3999$ の場合は $m = 150$ という値を用いている⁴⁾。本研究では、性能予測モデルを用いることにより、最適な m の値を実行前に予測することを目的とする。

3. 性能予測モデル

3.1 階層型の性能モデリング手法

small-bulge マルチシフト QR 法の性能予測モデルを構築するため、本研究では階層型の性能モデリング手法⁵⁾⁶⁾⁷⁾を用いる。この手法では、まずレベル 3 BLAS ルーチン、ダブルシフト QR 法ルーチンなどの基本演算ルーチンに対し、実測性能に基づく性能モデルを構築する。次に、アルゴリズム中でのこれらのルーチンのコール 1 回 1 回に対して、コール時の入力パラメータと性能モデルに基づき、その実行時間を予測する。最後に、各コールに対する予測時間を積み上げることにより、アルゴリズム全体の実行時間を予測する。この手法は、LU 分解⁶⁾、QR 分解⁷⁾、レベル 3 BLAS に基づく三重対角化アルゴリズム²⁰⁾などの性能モデリングに適用され、予測精度と予測コストの両面で満足できる結果を与えることが示されている。

3.2 基本演算ルーチンの性能モデリング

先に 2.3 で述べた通り、基本演算ルーチンとして

は BCHASE1, BCHASE2, BCHASE3, DGEMM-NN, DGEMM-NT, COPY1, COPY2, HQR の 8 個がある。ここでは、BCHASE2 を例としてモデリングの実際を示す。このルーチンは、フェーズ II において対角ブロック内部のバルジ・チェイシングを行うルーチンであり、シフト数 m とチェイシングの行数 k の 2 つの入力パラメータを持つ。そこで、 m, k を与えたときのモデルによる実行時間の予測値を $f_{\text{BCHASE2}}(m, k)$ で表す。

モデルを構築するには、まず (m, k) 平面の格子点において、BCHASE2 の性能を測定する。ここでは、 m を $10 \leq m \leq 150$ の範囲で 10 おきに動かし、 k は $\frac{1}{5}m, \frac{2}{5}m, \frac{3}{5}m, \frac{4}{5}m, m$ の 5 通りに動かして測定を行った。そして、各 k に対する $f_{\text{BCHASE2}}(m, k)$ の関数形として m の 3 次関数

$$\begin{aligned} f_{\text{BCHASE2}}(m, k) &= f_{\text{BCHASE2}}^{(k)}(m) \\ &= a_3^{(k)}m^3 + a_2^{(k)}m^2 + a_1^{(k)}m + a_0^{(k)}. \end{aligned} \quad (1)$$

を仮定し、係数 $a_3^{(k)}, a_2^{(k)}, a_1^{(k)}$ and $a_0^{(k)}$ を最小 2 乗法により求めた。これ以外の k の値に対する関数値は、 k に関する補間を用いて求めることにした。補間としては、もっとも簡単な 1 次補間を用いた。これは、近似すべき関数がキャッシュミスなどの影響で滑らかでない可能性があり、その場合、高次の補間は不安定となる可能性があるからである。

他の基本演算ルーチンに対しても、同様にして性能モデリングを行った。COPY1, COPY2, DGEMM-NN, DGEMM-NT は入力パラメータが 2 つ (ここでの DGEMM では行列の一方は正方行列) なので、モデリング手法は BCHASE2 と同じであり、BCHASE1, BCHASE3, HQR はパラメータが 1 つ (m のみ) なのでモデリングはより容易である。

3.3 アルゴリズム全体の性能モデリング

以上の基本演算ルーチンに対する性能モデルを用いて、small-bulge マルチシフト QR 法に対する性能モデルを構築する。そのため、まず、BCHASE1.TIME, DGEMM.TIME, COPY1.TIME, HQR.TIME などという名前のルーチンを作成した。これらはそれぞれ BCHASE1, DGEMM, COPY1, HQR と同じ引数を持つが、計算は行わず、代わりに与えられた入力パラメータに対するそのルーチンの予測実行時間を返すルーチンである。次に、small-bulge マルチシフト QR 法のプログラムにおいて、基本演算ルーチンすべてを .TIME の付いた時間予測ルーチンで置き換え、各ルーチンが返す予測時間を累積するようにした。これにより、このプログラムを実行すると、与えられた

入力パラメータ m , n に対する small-bulge マルチシフト QR 法の予測実行時間が得られることになる。この手法は、レベル 3 BLAS を用いた三重対角化アルゴリズムの性能予測にも適用され、様々なマシン、入力パラメータに対して 5 ~ 10% の精度で予測を行えることが示されている²⁰⁾。

なお、QR 法は反復法であるため、プログラム中で収束判定が必要であるが、ここで作成した実行時間予測プログラムでは実際の計算を行わないため、収束判定ができない。そこで、マルチシフト QR 法では多くの場合に 4 反復で 1 個の固有値が計算できるという Kressner の観察に基づき、収束までの全反復回数を 4n 回と仮定した。

4. 実験結果

4.1 実験環境

ここで提案した性能モデルを評価するため PowerPC G5 (2.0GHz) と Opteron (1.8GHz) の 2 つのプロセッサ上で実験を行った。G5 では IBM XL Fortran コンパイラを用い、コンパイルオプションは -O3 -qsmpp=omp -qarch=ppc970 -qtune=ppc970 とした。また、Opteron では GNU f77 コンパイラを用いた。BLAS は両方とも GOTO BLAS を用いた。small-bulge マルチシフト QR 法の入力としては、 $[-1, 1]$ の一様乱数を要素とする密行列を生成し、ハウスホルダー変換によりヘッセンベルグ形に変形して用いた。行列サイズ n は 1000, 2000, 4000, 8000 の 4 通りとし、 m は 30, 60, 90, 120 の 4 通りとした。

4.2 実験結果

PowerPC G5 での予測実行時間と実測実行時間 (単位は秒) を表 1 に示す。表から明らかなように、本モデルで予測した実行時間はかなりの過大評価となっている。これは主として、固有値 1 個の計算に平均して 4 反復が必要という見積もりが過大だったことによると思われる。実際、この行列に対して、固有値 1 個の計算のために必要な反復回数は、 n によっても異なるが、3.5 回前後であった。ただし、 m を変えたときの実行時間の増減については、本モデルにより定性的に再現できている。実際、 $n = 1000, 2000, 4000$ については、モデルから予測される最適な m の値は真の最適値と一致している。 $n = 8000$ の場合については、正しい予測はできていないが、モデルにより予測された最適値 $m = 90$ を用いた場合の実行時間は、真の最適値 $m = 120$ を用いた場合に比べて 3% 弱増加するだけである。

一方、Opteron での予測実行時間と実測実行時間と

を表 2 に示す。PowerPC G5 の場合と同様、予測実行時間は過大評価となっているが、 m による実行時間の変化の傾向はほぼ再現できており、予測した最適値は、真の最適値に等しいか、あるいはその隣の値となっている。

以上より、本モデルは、small-bulge マルチシフト QR 法において最適な m の値を予測するために使用できる見込みがあると考えられる。ただし、三重対角化アルゴリズムの例²⁰⁾ と比べると、モデルの誤差はまだ大きく、その原因の解明が必要である。

表 1 PowerPC G5 での予測実行時間 (下段) と実測実行時間 (上段)

n	$m = 30$	$m = 60$	$m = 90$	$m = 120$
1000	5.84	5.27	5.12	6.75
	8.28	6.79	6.77	7.82
2000	42.21	33.75	32.83	34.47
	53.40	40.53	39.59	46.09
4000	356.85	267.65	236.20	260.97
	393.86	288.60	267.22	296.86
8000	3073.24	2496.06	2270.95	2129.30
	3075.62	2180.10	1969.56	2116.30

表 2 Opteron での予測実行時間 (下段) と実測実行時間 (上段)

n	$m = 30$	$m = 60$	$m = 90$	$m = 120$
1000	6.56	6.79	6.77	7.85
	7.90	7.45	7.71	8.37
2000	51.97	45.28	47.78	44.26
	59.83	52.62	51.54	54.56
4000	438.81	354.59	325.94	329.58
	461.69	401.12	376.20	384.17
8000	3549.57	2802.24	2620.92	2573.04
	3641.13	3108.63	2888.67	2919.31

5. 結論

本報告では非対称密行列の固有値を求めるための small-bulge マルチシフト QR 法に対し、性能予測モデルを提案した。本モデルは階層型のモデリング手法に基づいており、行列サイズ n 、シフト数 m と、レベル 3 BLAS、ダブルシフト QR 法ルーチンなどのアルゴリズム構成要素の性能データとを入力として、small-bulge マルチシフト QR 法全体の実行時間を予測する。PowerPC G5 および Opteron 上での実行時間の予測を行ったところ、 m の値による実行時間の変化を定性的に再現することができた。これにより、本モデルを small bulge マルチシフト QR 法において最適な m の値を予測するために使用できる見込みを

得た。

今後の課題としては、(i) 予測誤差の原因を解明し、モデルの精度を向上させること、(ii) より多くのプロセッサおよび例題を用いてモデルの評価を行うこと、(iii) 本モデルを共有メモリ型並列計算機上での small-bulge マルチシフト QR 法に適用することなどが挙げられる。

謝辞 日頃から有益な議論をして頂いている自動チューニング研究会のメンバーに感謝いたします。本研究は名古屋大学 21 世紀 COE プログラム「計算科学フロンティア」の補助を受けている。

参考文献

- 1) E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK User's Guide*. SIAM, 1992.
- 2) Z. Bai and J. Demmel. On a block implementation of hessenberg QR iteration. *Int. J. of High Speed Computing*, 1:97–112, 1989.
- 3) C. Bischof, B. Lang, and X. Sun. Parallel tridiagonalization through two-step band reduction. Technical Report 17, PRISM Working Note, 1994.
- 4) K. Braman, R. Byers, and R. Mathias. The multishift QR algorithm. part I: Maintaining well-focused shifts and level 3 performance. *SIAM Journal on Matrix Analysis and Applications*, 23(4):929–947, 2002.
- 5) J. Cuenca, L-P. Garcia, and D. Gonzalez Gimenez. Empirical modelling of parallel linear algebra routines. In *Proceedings of the 5th International Conference on Parallel Processing and Applied Mathematics (PPAM2003)*, number 3019 in Lecture Notes in Computer Science, pages 169–174. Springer-Verlag, 2004.
- 6) J. Cuenca, D. Gimenez, and J. Gonzalez. Architecture of an automatically tuned linear algebra library. *Parallel Computing*, 30:187–210, 2004.
- 7) K. Dackland and B. Kågström. A hierarchical approach for performance analysis of ScaLAPACK-based routines using the distributed linear algebra machine. In *Proceedings of Workshop on Applied Parallel Computing in Industrial Computation and Optimization (PARA96)*, number 1184 in Lecture Notes in Computer Science, pages 187–195. Springer-Verlag, 1996.
- 8) J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.
- 9) Jack Dongarra, S. J. Hammarling, and D. C. Sorensen. Block reduction of matrices to condensed forms for eigenvalue computations. *Journal of Computational and Applied Mathematics*, 27:215–227, 1989.
- 10) A. Dubrulle. The multishift QR algorithm: Is it worth the trouble? Palo Alto Scientific Center Report G320-3558x, IBM Corp., 1991.
- 11) J. G. F. Francis. The QR transformation. a unitary analogue to the LR transformation. I. *Comput. J.*, 4:265–271, 1961/1962.
- 12) J. G. F. Francis. The QR transformation. II. *Comput. J.*, 4:332–345, 1961/1962.
- 13) G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.
- 14) G. Henry, D. S. Watkins, and J. Dongarra. A parallel implementation of the nonsymmetric QR algorithm for distributed memory architectures. *SIAM J. Sci. Comput.*, 24(1):284–311, 2002.
- 15) V. N. Kublanovskaya. On some algorithms for the solution of the complete eigenvalue problem. *U.S.S.R. Comput. Math. and Math. Phys.*, 3:637–657, 1961.
- 16) D. S. Watkins. Bidirectional chasing algorithms for the eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 14:166–179, 1993.
- 17) D. S. Watkins. Shifting strategies for the parallel QR algorithm. *SIAM J. Sci. Comput.*, 15:953–958, 1994.
- 18) D. S. Watkins. The transmission of shifts and shift blurring in the QR algorithm. *Linear Algebra and Its Applications*, 241/243:877–896, 1996.
- 19) Y.-J. J. Wu, P. A. Alpatov, C. Bischof, and R. A. van de Geijn. A parallel implementation of symmetric band reduction using PLAPACK. In *Proceedings of the Scalable Parallel Libraries Conference*, 1996.
- 20) Y. Yamamoto. Performance modeling and optimal block size selection for a blas-3 based tridiagonalization algorithm. In *Proceedings of HPC-Asia 2005*, pages 249–256, Beijing, December 2005.