

限量記号消去法による 時間モデルパラメータの決定

折居茂夫[†]

限量記号消去法(QE)による並列処理の時間モデルのパラメータ決定方法を提案する。時間モデルと測定値の差を連立不等式で表し QE によりモデルパラメータを決定する。数式処理のアルゴリズムである QE の性質を利用して、モデル式の不必要な項を判定できる。同様に新たに加えた項が必要か否かを判定できる。この性質を利用して、時間モデルの確認や改良をしながら、モデルパラメータを決定することができる。

Determination of Runtime Model Parameters by Quantifier Elimination

SHIGEO ORII[†]

We propose a method of determination of runtime model parameters for parallel processing by quantifier elimination (QE). The model parameters are determined by solving simultaneous inequalities on residuals between a runtime model and measured times. We can judge unnecessary terms of the model by the algorithmic properties of QE which is one of symbolic computation algorithms. We can also judge whether or not the need of new terms added the model. Using QE, determination of model parameters can be done through verifying and improving runtime models.

1. はじめに

ペタフロプスの計算能力を持つ並列計算機が数万台規模のプロセッサを用いて実用化され、高並列処理が日常的な時代になりつつある。このようにたくさんのプロセッサを使う並列処理では、その資源、即ちプロセッサ数(コア数)や電力等の有効利用を考慮し、各々の並列処理に見合った最適な資源を割り当てることが課題となる。

そこで並列効率メトリックを提案した[1]、並列処理時間をプロセッサ数と問題の大きさを変数とする式で表す時間モデルを構築する方法を提案してきた[2, 3]。

時間モデル構築では、アプリケーションプログラムの四則演算等の実行回数をプロセッサ数と問題の大きさを変数とする式で表現し、並列オーバーヘッド処理の実行回数を式化し、それらのモデルパラメータを実測値から決定する。これらのモデルは、アプリケーションプログラムの入力が変わるとモデルの再構築が必要となる場合がある。また同じアプリケーションプログラムでも、並列計算機が変わるとモデルパラメータを再決定する必要がある。

時間モデルを使って最適な資源を割り当てて行うとすると、モデルの再構築やモデルパラメータの再決定を少しでも早く且つ簡単にできる必要がある。従ってプログラム中に時計等のプローブを挿入することが必要な測定値を用いるようなことは避け、プログラム全体の処理時間のみからこれらの決定を行いたい。また入力や計算機が変わることにより、モデル式の中で効かなくなってしまった部分、新たに追加しなければならないモデルがある場合があるが、これには時間モデルに式を追加したり、一部の項を削除したりすることが簡単にできることが求められる。

上記の課題への考察は最小自乗法を通して行えるが、本研究では数式処理のアルゴリズムの一つ限量記号消去(Quantifier Elimination, 以後 QE と記す)を用いてモデルパラメータを決定する方法を提案する。最小自乗法ではパラメータに対する残差の自乗和の傾きから最少自乗残差となるモデルパラメータを導く。一方本研究で紹介する残差を連立不等式で表しそれを QE で解いてモデルパラメータを求める方法では、根、同根、特異点を求め、その中から解が存在する最少残差に対する解を選び出すアプローチとなる。このためこの最少残差を用いてモデルの精度が推定できる。また数式処理のアルゴリズムである QE の特性のため、モデル式の中で必要としない項のパラメータがゼロとなり、項の取舍選択ができる場合がある。

本論文では、2章で QE による時間モデルパラメータの決定方法を提案する。3章で QE を用いるメリットに言及し、QE のメリットを用いたモデルパラメータ決定手順を提案する。4章で High Performance Linpack と FUJITSU HPC2500 並列計算機システムにこの手順を適用した結果を示す。5章で最小自乗法とのメリットを議論する。

[†] 富士通(株)
Fujitsu Ltd.

2. 限量記号消去(QE)による時間モデルパラメータの決定方法

時間モデルが与えられたものとして、そのモデルパラメータを QE で決定する方法を提案する。時間モデルを $T(c_j, p, x)$ で表す。ここに k 個のモデルパラメータ c_j 、プロセス数 p 、問題の大きさ x である。ここで時間モデルと $i=1,2,\dots,m$ 個の測定時間 d_i とすると残差は式(1)となる。

$$T(c_j, p, x) - d_i = e_i \quad (1)$$

ここで式(1)の右辺にある値 e とし、残差がこの e より小さい場合の c_j を求めることを考えて式(2)とする。ここに $c_j = c_1, c_2, \dots, c_k$ である。

$$|T(c_j, p_i, x_i) - d_i| \leq e \quad (2)$$

モデルパラメータを式(2)と条件 $c_j \geq 0, e > 0$ を加えた論理式(3)で記述する。

$$\bigwedge_{i=1}^m (-e \leq T(c_j, p_i, x_i) - d_i) \leq e \wedge c_j \geq 0 \wedge e > 0 \quad (3)$$

ここで式(2)が解を持つ最小残差を e_{\max} とすると、式(3)の論理式から式(4)のように c_j を QE で消去することにより e_{\max} を求めることができる [4]。ここに $e_{\max} \leq e \leq \text{Max}(e_i)$ である。

$$\exists c_j \left(\bigwedge_{i=1}^m (-e \leq T(c_j, p_i, x_i) - d_i) \leq e \wedge c_j \geq 0 \wedge e > 0 \right) \stackrel{\text{QE}}{\Rightarrow} F(e_{\max}) \quad (4)$$

各モデルパラメータは式(3)の e を e_{\max} に置き換えて求めることができる。式(5)は c_1 を求めるための論理式と QE オペレーションである。

$$\exists c_2 \dots \exists c_k \left(\bigwedge_{i=1}^m (-e_{\max} \leq T(c_j, p_i, x_i) - d_i) \leq e_{\max} \wedge c_j \geq 0 \right) \stackrel{\text{QE}}{\Rightarrow} F(c_1) \quad (5)$$

3. 時間モデルパラメータをQEで決定することのメリット

2章で示したように、QE を用いると有限個の解の候補の中から、解が存在する最小残差 e_{\max} に対する解を求めることができる。このことは裏を返せば e_{\max} で解にならないものを排除できることを意味する。この QE の性質を利用してモデル式から、測定値をモデル化するのに必要な部分のみを与えられたモデル式から選び出せる場合がある。

有限個の解の候補ということをも具体的イメージするために、式(2)において $i=1,2, j=1,2$ の場合を考えると、式(6-1)はその論理式である。

$$\begin{aligned} d_1 - (a_{11} \cdot c_1 + a_{12} \cdot c_2) &\leq e \wedge -e - d_1 + (a_{11} \cdot c_1 + a_{12} \cdot c_2) \leq e \wedge \\ d_2 - (a_{21} \cdot c_1 + a_{22} \cdot c_2) &\leq e \wedge -e - d_2 + (a_{21} \cdot c_1 + a_{22} \cdot c_2) \leq e \end{aligned} \quad (6-1)$$

ここで $\Psi = \{a_i x - b_i \rho_i \mid i \in I, \rho_i \in \{=, \neq, \leq, <\}\}$ が上記論理式を構成する式となるので、QE のアルゴリズムの1つ仮想置換法[5]が適用でき、 φ の束縛変数 x は式(6-2)

に示す関係が成立する。

$$\exists x \varphi \Leftrightarrow \bigvee_{t \in S} \varphi(x // t) \quad (6-2)$$

ここに $\varphi(x // t)$ は消去される変数 x に S の元 t を形式的に代入して整理した式である。 S は消去集合と呼ばれる消去される変数を含まない式の集合で、式(6-3)で表される。

$$S = \left\{ \frac{b_i}{a_i}, \frac{b_i}{a_i} \pm 1 \mid i \in I \right\} \cup \left\{ \frac{1}{2} \cdot \left(\frac{b_i}{a_i} + \frac{b_j}{a_j} \right) \mid i, j \in I, i \neq j \right\} \quad (6-3)$$

式(6-1)の論理式を $\varphi(c_2 // t)$ とすると、式(6-3)において $I=4, a_1=a_{12}, b_1=e+d_1-a_{11} \cdot c_1, a_2=-a_{12}, b_2=e-d_1+a_{11} \cdot c_1, a_3=a_{22}, b_3=e+d_2-a_{21} \cdot c_1, a_4=-a_{22}, b_4=e-d_2+a_{21} \cdot c_1$ となり、消去集 $S = \{(e+d_1-a_{11} \cdot c_1)/a_{12}, (e+d_1-a_{11} \cdot c_1)/a_{12}+1, (e+d_1-a_{11} \cdot c_1)/a_{12}-1, -(e-d_1+a_{11} \cdot c_1)/a_{12}, -(e-d_1+a_{11} \cdot c_1)/a_{12}+1, -(e-d_1+a_{11} \cdot c_1)/a_{12}-1, (e+d_2-a_{21} \cdot c_1)/a_{22}, (e+d_2-a_{21} \cdot c_1)/a_{22}+1, (e+d_2-a_{21} \cdot c_1)/a_{22}-1, -(e-d_2+a_{21} \cdot c_1)/a_{22}, -(e-d_2+a_{21} \cdot c_1)/a_{22}+1, -(e-d_2+a_{21} \cdot c_1)/a_{22}-1\} \cup \{(e+d_1-a_{11} \cdot c_1)/a_{12}-(e-d_1+a_{11} \cdot c_1)/a_{12}, (e+d_2-a_{21} \cdot c_1)/a_{22}-(e-d_2+a_{21} \cdot c_1)/a_{22}\}$ を得る。

S の元を式(6-1)の束縛変数 c_2 に形式的に代入すると、減量記号のない等価な式を得ることができる。従ってモデルパラメータは16個の消去集合の元をもとにした、有限個の解の候補から決定されることがわかる。即ち QE を用いると、解ではないパラメータ値は自動的に排除される。

QE を適用するもう一つのメリットは、モデルパラメータ値がゼロになる解の候補が存在する場合があるということである。図1の11行目の $C2=0$ はこの可能性の例である。この図は、Mathematica を用いて、式(2)における例示 $i=1,2,3, j=1,2,3$ の場合に対して、 c_3 を束縛変数として消去したものである。Mathematica では \exists は関数 `Exists` で記述し、 \exists で指定した変数やパラメータを消去する QE は関数 `Reduce` を使い、 \wedge は

&&で記述する. e 値を e_{max} に置き換え, d_i, a_{ij} の値を代入したときこの行の不等式が成り立てば, $C2=0$ の行は解の候補となり, $c1 \neq 0, c2=, c3=0$ の場合であることがわかる. このように, QE を実行するとモデルパラメータがゼロとなることにより, その項が必要かを自動的判定してくれる場合がある.

```

Reduce[Exists[{c3},
  -e ≤ d1 - a11 * c1 + a12 * c2 + a13 * c3 ≤ e &&
  -e ≤ d2 - a21 * c1 + a22 * c2 + a23 * c3 ≤ e &&
  -e ≤ d3 - a31 * c1 + a32 * c2 + a33 * c3 ≤ e &&
  c1 ≥ 0 && c2 ≥ 0 && c3 == 0 &&
  e > 0 && d1 > e && d2 > e && d3 > e
]]
...
(a11 | a12 | a13 | a21 | a22 | a23 | a31 | a32 | a33) ∈ Reals &&
d3 > 0 && 0 < e < d3 && d2 > e && d1 > e &&
( ( c2 == 0 && c1 > 0 &&  $\frac{d3 - e}{c1} \leq a31 \leq \frac{d3 + e}{c1}$  &&
 $\frac{d2 - e}{c1} \leq a21 \leq \frac{d2 + e}{c1}$  &&  $\frac{d1 - e}{c1} \leq a11 \leq \frac{d1 + e}{c1}$  ) ||
( c2 > 0 &&
( ( a32 <  $\frac{-d3 - e}{c2}$  && c1 > 0 &&
 $\frac{a32 c2 + d3 - e}{c1} \leq a31 \leq \frac{a32 c2 + d3 + e}{c1}$  &&
 $\frac{a22 c2 + d2 - e}{c1} \leq a21 \leq \frac{a22 c2 + d2 + e}{c1}$  &&
 $\frac{a12 c2 + d1 - e}{c1} \leq a11 \leq \frac{a12 c2 + d1 + e}{c1}$  ) ||
...

```

図1 QE 実行結果

QE のメリットの3番目は, e_{max} 値と測定値 d_i からモデルの精度が推定できることである. 良い精度の場合は $e_{max} \ll d_i$, 悪い精度では $e_{max} \geq d_i$ と考えられる. パラメータ決定に関するこれら3つのメリット即ち, 有限個の解からの選択, パラメータ値が0の判定, e_{max} 値によるモデルの良し悪しの判定から, 図2のようなモデルパラメ

ータ決定の手続きを考えることができる. なお図中 $e_{max}/d_i \leq 0.1$ としたのは, この比が有効数字を表すものと考え, 有効数字2桁以上で時間モデルを構築する手順としたためである.

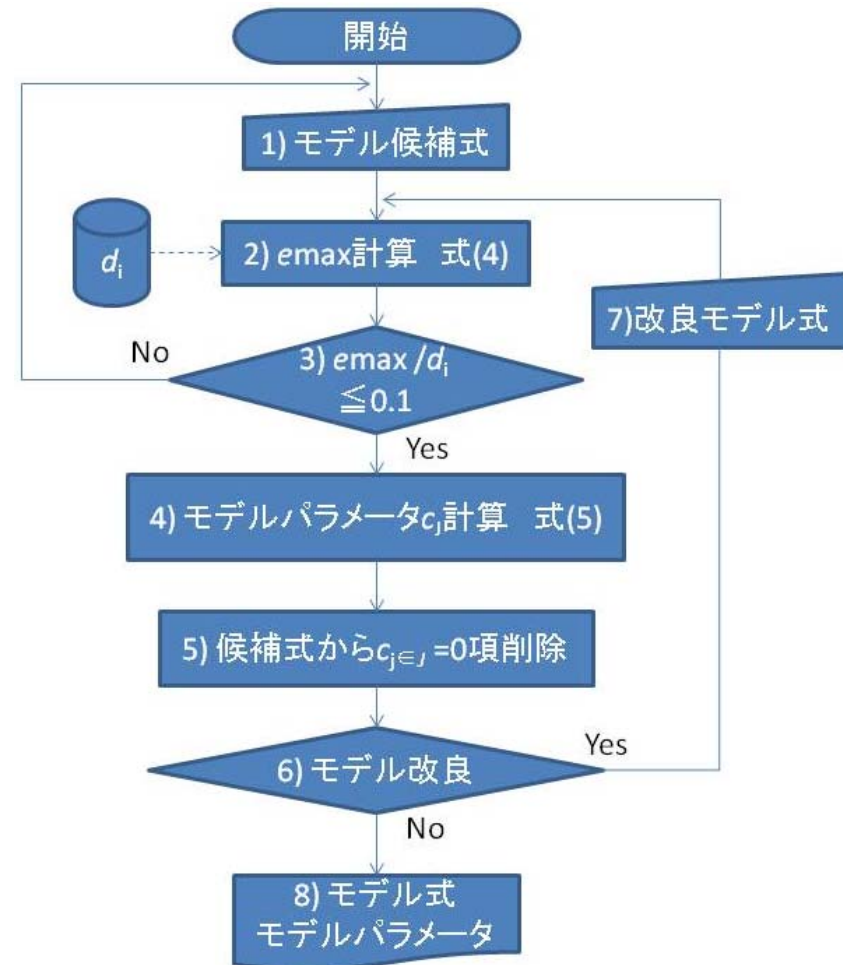


図2 QE のメリットを利用したモデルパラメータ決定手順

図の各ステップの詳細は次のようになる。

- ステップ 1) 式で記述された時間モデルの候補を用意
- ステップ 2) 式(4)を用いて $emax$ を計算
- ステップ 3) 有効数字が 2 桁になると考えられる, $emax/d_i < 0.1$ のモデルか?
(否ならば, ステップ 1)に戻って他のモデルを用意)
- ステップ 4) 式(5)と $emax$ を用いてモデルパラメータ c_j を計算
- ステップ 5) モデルから $c_{j \in J} = 0$ の項を削除
- ステップ 6) 性能予測, 精度向上等のためにモデルに改良するか?
- ステップ 7) ステップ 5)で得られたモデルを基に改良モデルを作成
- ステップ 8) 測定値 d_i と時間モデル候補式からなる連立不等式(2)の解である, 時間モデルと時間モデルパラメータを得る。

4. モデルパラメータの決定例

QE をベースにした図 2 のモデルパラメータ決定手順を用い, High Performance Linpack (以後 HPL)と FUJITSU HPC2500 からなる並列処理システムの時間モデルパラメータを決定した。

時間モデルの候補式は[6]から式(7)とする (ステップ 1)。

$$T_{HPL}(P, Q, N) = \gamma \cdot \frac{2 \cdot N^3}{3 \cdot P \cdot Q} + \beta \cdot \frac{N^2 \cdot (3 \cdot P + Q)}{2 \cdot P \cdot Q} + \alpha \cdot \frac{N \cdot ((NB + 1) \cdot \text{Log}(P) + P)}{NB} \quad (7)$$

ここに N : 行列の次元数, 2次元プロセス格子数: $P \times Q$, NB : ブロックサイズであり, α , β , γ がモデルパラメータである。

測定値 d_i は $NB=100$, $Q=1$, $P=10, 20, 30, 40, 50, 60, 70, 80$ の 8 個を用い表 1 の $emax$ を得た (ステップ 2)。

表 1 時間モデルの候補式の $emax$ とモデル (その 1)

$emax$	$\text{Min}(d_i)$	$\text{Max}(d_i)$	α	β	γ
28.7	116.	229.	0.001826	0	1.316×10^{-10}

表 1 から $emax / \text{Min}(d_i) = 0.25$, $emax / \text{Max}(d_i) = 0.13$ で, 得られた解は有効数字 1 桁以下となり十分に現象をモデル化できないと考えられる。

そこでステップ 1)に戻り式(7)において $\alpha = aP + b$ としたモデルを考える[3]。再度 $emax$ を決定しその値を表 2 に示す。表 2 の $emax$ を用いると, $emax / \text{Min}(d_i) = 0.053$,

$emax / \text{Max}(d_i) = 0.027$ であり, α = 一定としたモデルより 1 桁以上有効数字が良くなると考えられる。表 2 にこのモデルのモデルパラメータを示す (ステップ 4)。

表 2 時間モデルの候補式の $emax$ とモデル (その 2)

$emax$	a	b	β	γ
6.15	0.00001947	0	1.240×10^{-7}	4.700×10^{-10}

表 2 は $b=0$ である。従って式(8)の時間モデルを得る。

$$T_{HPL}(P, Q, N) = \gamma \cdot \frac{2 \cdot N^3}{3 \cdot P \cdot Q} + \beta \cdot \frac{N^2 \cdot (3 \cdot P + Q)}{2 \cdot P \cdot Q} + a \cdot P \cdot \frac{N \cdot ((NB + 1) \cdot \text{Log}(P) + P)}{NB} \quad (8)$$

図 3 に測定値と式(8)と表 2 で記述した時間モデルを示す。図中○印は測定値, その中の×印はモデル化に使用したデータ d_i , 実線は時間モデルである。図はモデルが測定値と一致していることを示す。従って P の変化をに対する時間モデルとそのパラメータが $N=15000$, $Q=1$ に対して得られた。

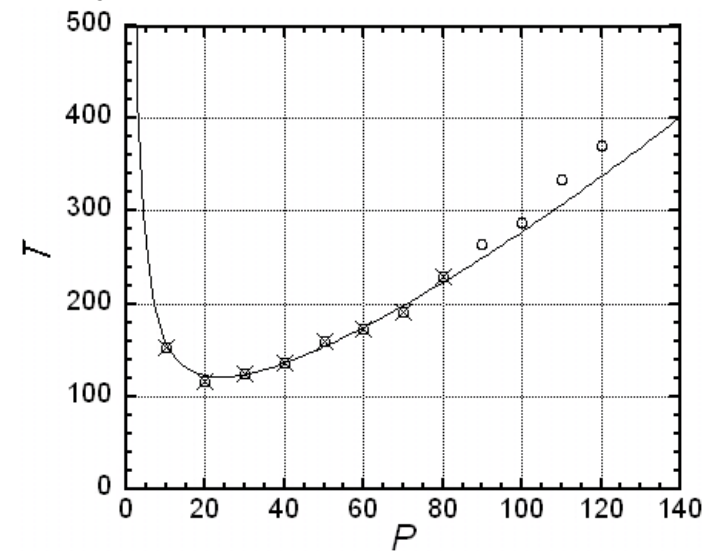


図 3 HPL&HPC2500 のモデルパラメータ決定 ($N=15000$, $Q=1$)

図3で示した時間モデルは $N=15000, Q=1$ という特殊な入力データに対するモデルである。このモデルを任意の N と Q に対する時間モデルとなるよう改良する。

式(7), (8)では通信モデルは $t_c = \alpha + \beta \cdot L$ であった。この時間が panel factorization と trailing sub matrix 計算で別々と考え、各々 $t_{cf} = \alpha_f + \beta_f \cdot L, t_{cm} = \alpha_m + \beta_m \cdot L$ として処理時間を求めると式(9)となる (ステップ6, 7)。

$$T_{HPL}(P, Q, N) = \gamma \cdot \frac{2 \cdot N^3}{3 \cdot P \cdot Q} + \beta_f \cdot \frac{N^2}{2 \cdot P} + \beta_m \cdot \frac{3 \cdot N^2}{2 \cdot Q} + a_f \cdot P \cdot \frac{N \cdot (1 + NB \cdot \text{Log}(P))}{NB} + a_m \cdot P \cdot \frac{N \cdot (P - 1 + \text{Log}(P))}{NB} \quad (9)$$

測定値 d_i を $NB=100, (P, N)=(10, 15k), (20, 20k), (30, 25k), (40, 30k), (Q, N)=(20, 35k), (30, 40k), (40, 45k)$ の順の7個を用いて、表3の結果を得た。

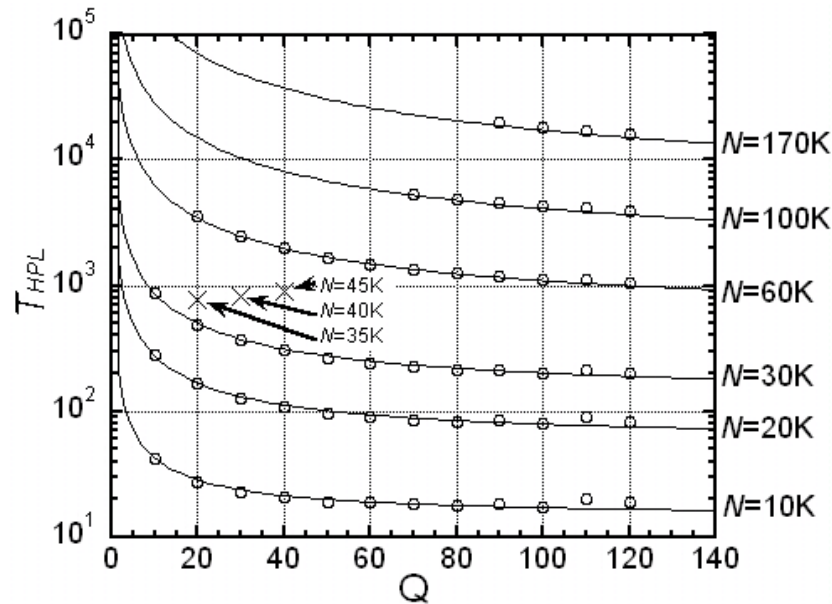


図4-1 時間モデルの Q に対するバリデーションテスト

表3 時間モデルの候補式の e_{max} とモデル (その3)

e_{max}	α_f	α_m	β_f	β_m	γ
6.75	0.00002419	0	2.882×10^{-7}	1.263×10^{-7}	3.967×10^{-10}

図4-1, 図4-2は時間モデル式(10)のバリデーションテストである。図中の○はバリデーションテストのための測定値である。図中の×は d_i として用いた7個の測定値である。図4-1は $P=1$ における N と Q に関するモデルによる予測である。図4-2は $Q=1$ における N と P に関する予測である。両図から d_i を外挿した領域で良く一致していることが確認できる。また $e_{max}/\text{Min}(d_i)=0.044, e_{max}/\text{Max}(d_i)=0.008$ ゆえ、モデルの有効数字は2桁以上と推測できる。 $\alpha_m=0$ ゆえ、HPL&HPC2500の時間モデルは式(10), そのモデルパラメータは表3と決定できる。

$$T_{HPL}(P, Q, N) = \gamma \cdot \frac{2 \cdot N^3}{3 \cdot P \cdot Q} + \beta_f \cdot \frac{N^2}{2 \cdot P} + \beta_m \cdot \frac{3 \cdot N^2}{2 \cdot Q} + a_f \cdot P \cdot \frac{N \cdot (1 + NB \cdot \text{Log}(P))}{NB} \quad (10)$$

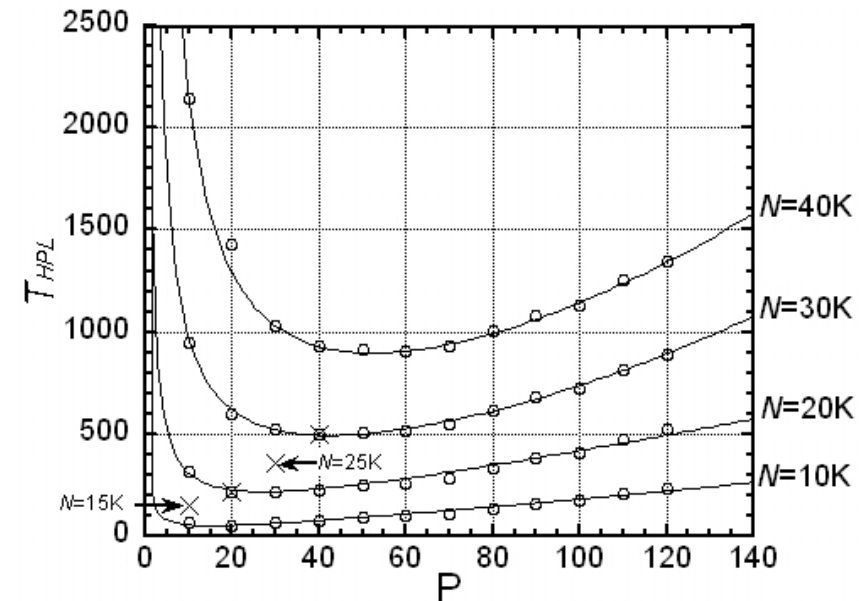


図4-2 時間モデルの P に対するバリデーションテスト

5. 議論

時間モデルが式で表現されている場合、最小自乗法でモデル式をフィッティングすることによってモデルパラメータを決定できる。例えば式(7)で $a=a \cdot P+b$ としたモデルを考え、表1と同じ測定値を用いると Mathematica8 の FindFit 関数を用いて $a=0.00001780$, $b=0.0004434$, $\beta=2.508 \times 10^{-8}$, $\gamma=4.955 \times 10^{-10}$ を得る。もしこの状態で a , b 項の取捨選択をしようとする、そのための考察が必要となる。一方 QE を用いると表2のように $b=0$ となりこの測定値に対して b に関する項は必要ないと判定できる。

式(10)と表3に用いた測定値で最小自乗法からモデルパラメータを決定すると表4となる。これを図5の赤色の実線で示す。○印の測定値と比較すると、 P が大きくなりモデルパラメータの決定に使用した×印の測定値から離れるに従って、予測値が低くなるのがわかる。これに対し QE から決定した表3のモデルパラメータ値で表した黒の実線は、測定値から離れた領域でも予測値と一致している。この違いは、最小自乗法が測定点の残差平方和を最小にするようにモデルパラメータを決定しているのに対し、QE が残差の絶対値の最小値 e_{max} に対する解を選んでいることの原因しているものと思われる。

表4 最小自乗法で求めた式(8)のモデルパラメータ値

α_f	β_f	β_m	γ
0.00002253	2.497×10^{-7}	1.239×10^{-7}	4.190×10^{-10}

QE 計算には時間がかかる。この時間を最小自乗法と比較すると、上記の式(10)のパラメータ決定において、2.20GHz の AMD Athlon 64 Processor 3500+, メモリ 1.96GB の Windows XP では QE で 16.7 秒、最小自乗法は 0.016 秒である。この差は変数と測定値の数が増すのに伴い大きくなるので、測定値が増えた場合は QE と最小自乗法との併用等が視野に入る。

謝辞 富士通株式会社テクニカルコンピューティング・ソリューション事業本部の奥田基エグゼクティブアーキテクト、樋口哲二シニアマネージャの援助に感謝します。

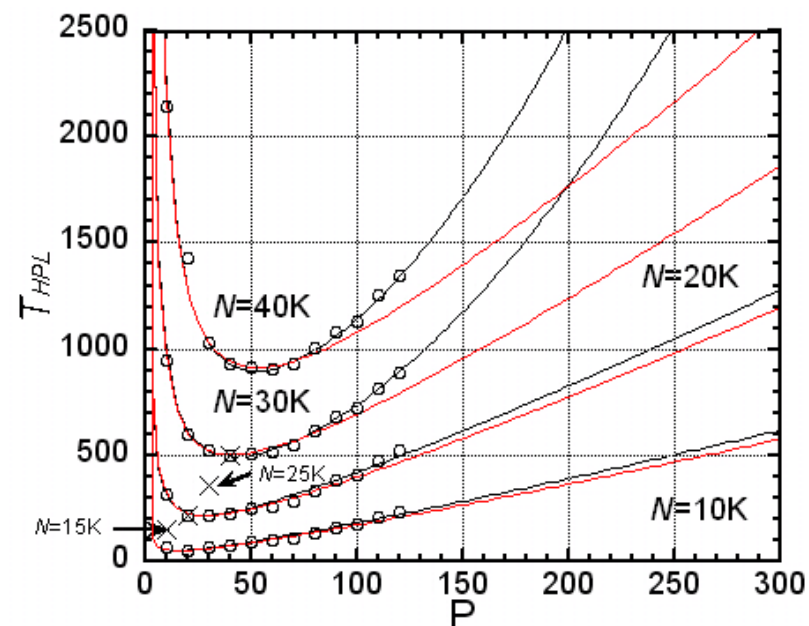


図5 QE と最小自乗法によるモデルパラメータ決定の比較

参考文献

- 1) S. Orii: Metrics for evaluation of parallel efficiency toward highly parallel processing, Parallel Comput, 36 (2010) 16-25.
- 2) 折居茂夫: 数値計算のための並列計算機性能評価方法, 情報処理学会論文誌, Vo.39, No.3 (1998).
- 3) 折居茂夫: 時間モデルを用いた並列処理の性能評価 - 並列化部に隠れた並列オーバーヘッド-, 情報処理学会研究報告, 2011-HPC-130, Vol.2011, No.1 (2011).
- 4) Shigeo Orii, Hirokazu Anai: Application of Quantifier Eliminator to Symbolic-Numeric Optimization in Biochemical Model, Research Communications in Biochemistry and Cell & Molecular Biology, Vol. 12, No. (1&2) (2008) pp.73-89.
- 5) V. Weispfenning, The complexity of linear problems in fields, J. Symbolic Comput. (1988) pp. 3-27.
- 6) A. Petitet, R. C. Whaley, J. Dongarra and A. Cleary: HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers. <http://www.netlib.org/benchmark/hpl/>