1G-06

自動チューニングにおける d-Spline 近似の実時間のブレを考慮した手法の提案

范谷瑛† 望月大義† 藤井昭宏† 田中輝雄† 片桐孝洋††
工学院大学† 名古屋大学††

1. はじめに

ソフトウェア自動チューニングは，一般にプログラムの性能に影響を与える複数の性能パラメタ群から，最適な組み合わせを効率よく推定する手法である．様々な手法が提案され，我々は手法のひとつとして，近似関数を利用した標本点逐次追加型性能パラメタ推定法の研究を進めてきた[1]．自動チューニング基盤 ppOpen-AT [2][3][4]に実装され，pragma を追記することで利用できる[5]．いままでは，1 つの標本点に対して 1 回のみの計測結果を用いて推定を行ってきた．しかし，実際には計測結果には計測時の計算機環境状況によるブレが発生し，推定に影響を与える可能性がある．

本研究では，標本点逐次追加型性能パラメタ推定法[6]での実測を用いた推定における，実時間のブレを考慮した手法を提案する．それにより，実時間のブレが d-Spline 近似に含まれるのを軽減し，より信頼性の高いソフトウェア自動チューニングが行えることを示す．

2. 標本点逐次追加型性能パラメタ推定法

近似関数$\boldsymbol{f}(x)$をn個の離散点x_j上の値$f_j = \boldsymbol{f}(x_j)$，$1 \le j \le n$で表現する．$n$は性能パラメタのとり得る値の個数 N より十分大きくとる．性能パラメタの取り得る N 個の値から k 個の実測データ（標本点）が得られているならば，それを$y_i(1 \le i \le k)$とする．$\boldsymbol{f}$を確定するために$\boldsymbol{f}$の滑らかさを 2 階差分$|f_{j-1} - 2f_j + f_{j+1}|$，$2 \le j \le n-1$で表す．

この近似関数$\boldsymbol{f}$を評価関数$min(\|\boldsymbol{y} - E\boldsymbol{f}\|^2 + \alpha^2\|D\boldsymbol{f}\|^2)$で選ぶ．この評価関数を解くためには，$min(\|\boldsymbol{b} - Z\boldsymbol{f}\|^2)$について最小二乗問題を解けばよい．このとき，$E$は実測データ$y_i$と近似関数$f_j$の対応を，$D$は近似関数の滑らかさを表す．$\alpha$は滑らかさの強さを表し，小さく設定するほど実測データに追随する．この近似関数を d-Spline と呼ぶ．標本点逐次追加型性能パラメタ推定法は d-Spline による最適な値の推定に必要な標本点を最低限の数から始め，新たな標本点を自動選択・追加しながら，近似関数を順次更新し，最適な性能パラメタを推定する．

Proposal of Method Considering Real-Time Perturbation of D-Spline Approximation in Run-Time Automatic Tuning
Guuing Fan†, Masayoshi Mochizuki†, Akihiro Fujii†, Teruo Tanaka† and Takahiro Katagiri††
†Kogakuin University , ††Nagoya University

3. 自動チューニング基盤 ppOpen-AT

ppOpen-AT は，自動チューニング機能付きプログラムを生成する基盤であり，自動チューニング手法のひとつとして，2 章で説明した標本点逐次追加型性能パラメタ推定法も実装されている．本研究では，ppOpen-AT によって生成されたプログラムをもとにしている．

4. 提案手法

本研究では，計測結果に対して，実時間のブレを考慮し，計り直しの機会を与え推定を行う．

標本点逐次追加型性能パラメタ推定法では，1 つの標本点に対して，1 回の計測結果を用いて推定を行ってきた．しかし，実際には計測結果には計算機環境の状況によるブレの発生が推定に影響を与える可能性がある．そこで，その時点での d-Spline の最大値と最小値から暫定的な最大変化量hを取得する．近似関数f_jと追加標本点y_iの計測結果との差をh'とする．h'がhの 20%を越えているのならば，追加標本点について，再度実測し，先に得た実測値と比較を行う．計測結果が小さいほど実時間のブレが含まれる可能性が低いため，より小さい値を用いて近似関数の更新を行う．初期標本点の実測については，基準をできるかぎり正しく決めるために 2 回実測を行い，初期の近似関数を生成する．

5. 実験

5.1 実験対象

本研究では，これまでの標本点逐次追加型性能パラメタ推定法を従来手法とし，提案手法と比較を行う．実験として，AMG 法[7]の解法部での 2 つの性能パラメタである加速度係数 smoother_accel_coef と強連結成分の閾値 strong_con_threshold について推定を行う．対象とする行列

表 1 実験に使用した問題の詳細

	範囲	間隔
smoother_accel_coef	[0.50,1.00]	0.01
strong_con_threshold	[0,0.0050]	0.0001

のサイズは，67502×67502である．表 1 に性能パラメタの詳細を示す．実験環境は，CPU IntelXeon E5-2623 v3，メモリ 16GB，コンパイラ ifort version 15.0.0 を用いた．

5.2 実験結果

smoother_accel_coef と strong_con_threshold に関して，最適な性能パラメタを推定するまでを 1 回の試行とし，従来手法，提案手法ともに 50 回試行を行った．最終的に生成された近似関数がそれぞれ 50 個あり，近似関数間の類似度を近似関数間の距離$\sqrt{\sum_{i-1}^{n}(f_1(x_i)-f_2(x_i))^2}$により定義しZで表す．Zは値が 0 に近いほど近似関数が類似しているとする．

図 1，2 は smoother_accel_coef について従来手法でパラメタ推定をし，近似関数の組み合わせを示したものである．図 1 の近似関数は頻出した近似関数の形状であり，追加標本点も同じである．一方，図 2 では，近似関数の形状が大きく異なり，追加標本点も安定して選ばれていない．これは，実時間のブレの影響によるものであり，図 2 のような近似関数のブレを軽減することでより安定した近似関数を生成できる．

図 3, 4 は, すべての組み合わせのZを算出し，ソーティングを施したものである．従来手法より提案手法は，Zの分布面積が小さくなっており，ともに 34%削減できた． Zの詳細を表 2 示す．smoother_accel_coef について，提案手法で

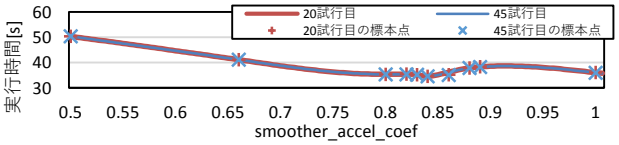

図 1　Z の小さい近似関数の組み合わせ

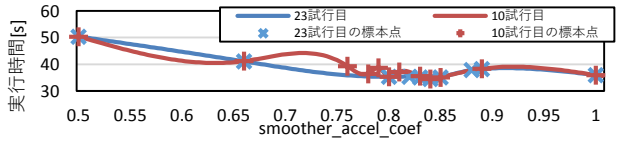

図 2　Z の大きい近似関数の組み合わせ

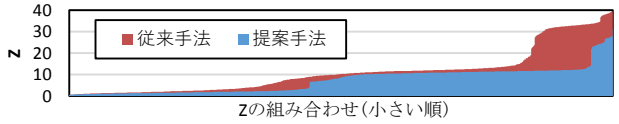

図 3　smoother_accel_coef についてのZの値

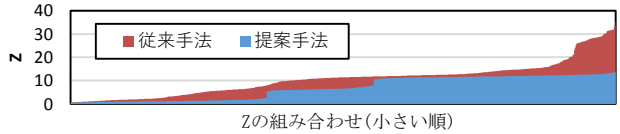

図 4　strong_con_threshold についてのZの値

表 2　Z の最大値，最小値，平均値，標準偏差

	smoother_accel_coef		strong_con_threshold	
	従来手法	提案手法	従来手法	提案手法
最大値	39.44	31.13	35.48	13.81
最小値	0.55	0.43	0.39	0.47
平均値	11.23	7.35	10.67	6.96
標準偏差	10.38	5.80	7.11	4.78

は, 平均値は 34%, 標準偏差は 42%削減でき，同様に strong_con_threshold について平均値は 34%， 標準偏差は 32%削減できた．これは，図 2 のような近似関数の発生を軽減できたからであり，実時間のブレの影響を軽減できた．

6. おわりに

本研究では，実時間のブレを考慮したパラメタ推定に近似関数の値と近似関数f_jと追加標本点y_iの計測結果との差が大きい場合に，再度実測を行うことで実時間のブレの影響を軽減する手法を提案した．実験の結果，提案手法は従来手法よりZの分布面積を 34%削減できた．これは，実時間のブレの影響を軽減できたことを示し, より高信頼な近似関数の生成を実現できた．

今後は他の手法を取り入れて，より高信頼なソフトウェア自動チューニングを目指す．

謝辞　本研究の一部は JSPS 科学研究費 16H02823 の助成を受けて行われた．

参考文献

[1] J. Bilmes, K. Asanovic, C.-W. Chin, J. Demmel, Opti-Mizing Matrix Multiply using PHiPAC : a Portable, High-Performance, ANSI C Coding Methodology, in: Proc. of the 11 th ICS, Vol. 97, pp. 340-347 (1997).

[2] T. Katagiri, S. Ohshima, M. Matsumoto, Auto-tuning of Computation Kernels from an FDM code with ppOpen-AT, in: Proc. of MCSoC2014, pp. 91-98 (2014).

[3] ppOpen-HPC Project, http://ppopenhpc.cc.u-tokyo.ac.jp/ppopemhpc.

[4] R. Murata, J. Irie, A. Fujii, T. Tanaka, T. Katagiri, Enhancement of Incremental Performance Parameter Estimation on ppOpen-AT, in: proc. Of MCSoC2015, pp. 203-210 (2015).

[5] T. Tanaka, S. Ito, S. Ohshima, Early Experiences for Adaptation of Auto-tuning by ppOpen-AT to an Explicit Method, in: Proc. of MCSoC2013, pp. 153-158 (2013).

[6] T. Tanaka, R. Otsuka, A.Fujii, T.Katagiri, T. Imamura, Implementation of d-Spline-based Incremental Perform-ance Parameter Estimation Method with ppOpen-AT, Scientific Programing 22, pp. 299-307 (2014).

[7] P. Vanek, J. Mandel, M. Brezina, Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems, TR UCD-CCM-036 (1995).