

Autopilot を用いた疎行列ソルバにおける実行時自動チューニング機構の設計

A Design of Run-Time Auto-Tuning Facility for Sparse Direct Solvers with Autopilot

石井良規* 片桐孝洋*** 本多弘樹* 弓場敏嗣*
Yoshinori Ishii Takahiro Katagiri Hiroki Honda Toshitugu Yuba
* 電気通信大学大学院情報システム学研究所 **科学技術振興機構さきがけ
Graduate School of Information Systems, PRESTO, JST
The University of Electro-Communications

1. はじめに

密行列ソルバに対して疎行列ソルバは、行列の数値特性や零要素の位置などによって高速化手法が異なる。ユーザにとって、適するライブラリの選択や実行パラメータの修正などに大変な負担がかかる。

そこで行列を解く際にユーザの負担を減らし、高速に、かつ安定に解を求められる機構があれば大変便利である。本研究では、この機構を Autopilot の Fuzzy logic を用いたパラメータ決定手段[1]を利用して実現する方式を提案する。

2. 疎行列ソルバ SuperLU

SuperLU[2]とは、カリフォルニア大学バークレー校の J.Demell らが開発した疎行列の直接解法ソルバである。この SuperLU で使われているパラメータで、自動チューニング可能なパラメータは以下の通りである。

行列のオーダリングの種類

- (1) Natural ordering
- (2) Minimum degree ordering on structure of $A^T \times A$
- (3) Minimum degree ordering on structure of $A^T + A$
- (4) Approximate minimum degree column ordering

ここで、このオーダリングとは fill-in をできるだけ減らし処理を高速化する手法である。

行列のタイプ

- Stype(storage type)
列方向あるいは行方向に疎行列データを圧縮するか、またはスーパーノード[3]を利用するかを選択。
- Dtype(data type)
single, double, single-complex, double-complex の選択。
- Mtype(mathematical type)
一般行列、下三角行列、上三角行列、および対称行列の選択。

3. 実行時自動チューニング機構の設計

Autopilot における Fuzzy Logic を用いた行列の切り分け方針は以下の通りである。

対角要素からの距離による判別法

対角要素から一番離れている非零要素を探し、その平均をとる。その値がバンドのようになっていればバンド行列用ソルバを、この非零要素が一直線にならない場合はスライライン行列用ソルバをそれぞれ指定する。

疎度を用いた判定法

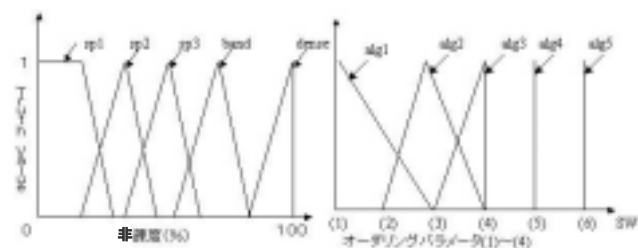
$n \times n$ の行列で非零要素数の割合に応じ、密行列か、バンド行列か、疎行列かを判断し、それに応じたソルバを選択する。ここで、非零要素数の割合を求める式(非疎度)を

$$[\text{nnz}/(n \times n)] \times 100\% \quad (\text{nnz:非零要素数}) \cdots (1)$$

のように定める。

例として、式(1)の非疎度を Fuzzy Logic の判定基準とした例を図1に示す。

ここで、図1(b)の sw の(5)および(6)は、それぞれバンドソルバ、密行列ソルバの選択を示す。



(a)ソルバ選択判定基準

(b)パラメータ判定基準

```
if(sparsity==SP1){sw=alg1}  
if(sparsity==SP2){sw=alg2}  
if(sparsity==SP3){sw=alg3}  
if(sparsity==band){sw=alg4}  
if(sparsity==dense){sw=alg5}
```

(c)推論ルール

図1 疎度を用いた設計例

4. 予備評価

120,750 × 120,750 の行列(非零の数 1,224,224)の正方形列に対して、オーダリング方式を変えて時間測定を行なうことで速度向上の効果を調べた。オーダリングの種類は前述の4種類で、デフォルトは(2)である。

表1: オーダリング方式の違いによる実行時間

手法	(1)	(2)	(3)	(4)
LU分解時間	メモリ不足	286.65(s)	メモリ不足	195.27(s)
Mflops		43.33		45.32

実験環境は SunOS ver5.8、コンパイラ gcc ver2.95.3、コンパイラオプション-O3 -c、SuperLU ver2.0で行なった。

結果として、オーダリング方式を変えただけでも性能差が見られる。したがって、本手法の疎度を用いた切り分けによるパラメータの自動チューニングを加えることで、さらなる高速化が達成できると期待される。

参考文献

- [1]Ruth Ayt ほか, “ Autopilot User’s Manual ”, 2003
<http://www-pablo.cs.uiuc.edu/>
- [2]James W.D ほか, “ SuperLU User’s Manual ”, 2003
<http://crd.lbl.gov/xiaoye/SuperLU/>
- [3]西出隆二,片桐孝洋,金田康正, “ ブロック幅を動的決定する疎行列連立一次方程式の直接解法 ”, 情報処理学会研究報告, 2001-HPC-89, 2001