

レポート提出の注意

レポート課題、コンテスト課題提出方法

- ▶ **日時:**
2016年2月11日(木)24時厳守
(工学部共通科目の締切日)
- ▶ **形式:**
PDF形式(レポート本体はPDFしか認めない)
 - ▶ コンテストのソースコードはZIP形式、
コンテストの報告書はPDF
 - ▶ 名前、所属、学籍番号の記載を忘れずに！
(毎回書かない人が多い)
- ▶ **提出方法:** 電子メール: katagiri _AT_ cc.u-tokyo.ac.jp (_AT_ =@)
 - ▶ 受理メールが届かない人は催促してください。
そうしないと、未提出になるかもしれません。

レポート課題採点指標

- ▶ 基本点 (+20点)
 - ▶ 以下の1~6がそろっている場合、基本点を加算する
 - ▶ 1.表紙(名前、所属、学籍番号)、2.問題の説明(問題のレベル Lxx を明記すること)、3.結果(実行時間に関する表、および、場合によっては1PEから64PEまでの実行結果と台数効果の図)、4.結果の考察、5.講義を受けた感想、6.付録としてプログラムの主要部分
- ▶ 課題点(+1 ~)
- ▶ 独創点(+5 ~)
 - ▶ 図が綺麗、説明がうまい、説明が丁寧、いっぱい実験(実装)している、よくできた考察、など
- ▶ 出席点(~+20)
 - ▶ 切り上げ(出席日数/12) × 20 点
- ▶ 遅刻点(-10、-30、-∞)
 - ▶ 1日目:-10点、2日目:-30点、3日後以降は-∞点。
- ▶ 特に、「講義の感想」を忘れないこと！

コンテスト課題採点指標

▶ 提出物

1. コンテスト課題出力結果
2. 解法、実装法に関する説明(1ページ以上)
3. ソースコードをtarで固めたもの
 - ▶ 1と2は、PDFの形式で

▶ 入賞点(+100)

- ▶ 1位～3位に入賞した人

▶ 完走点(+20)

- ▶ 出力結果が正しいプログラムを作成した場合のみ加点

▶ Fortran版、C版、別に順位を付ける

▶ Fortran版、C版、双方に出場できる(双方加点あり)

総合評価基準

- ▶ <レポート課題点>と<コンテスト課題点>の合計で、総合評価する
- ▶ $-\infty \sim 39$
 - ▶ 不可
- ▶ $40 \sim 59$
 - ▶ 可
- ▶ $60 \sim 79$
 - ▶ 良
- ▶ $80 \sim$
 - ▶ 優

コンテスト課題発表

その前に

- ▶ コンテスト参加者のメーリングリストを作るため、参加予告メールを送ってください。
- ▶ **Subject: SPC2015wAppl** と書くこと。
- ▶ 名前、学籍番号、メールアドレス を書くこと。
- ▶ 宛先:
katagiri@cc.u-tokyo.ac.jp
- ▶ メーリングリスト名:
spc2015w@kata-lab.itc.u-tokyo.ac.jp
- ▶ 本メーリングリストで、質問などを受け付けます。
- ▶ 参加者間の情報交換にも使ってOKです。

問題説明

- ▶ 課題:「複数の右辺 b 」がある「LU分解」

- ▶ 連立一次方程式

$$A x = b$$

の解ベクトル x を求める

- ▶ ここで、解ベクトル x が1本である保証はない
- ▶ すなわち、 m 本の解ベクトルをまとめた行列 X を

$$X = (x_1 \ x_2 \ \dots \ x_m)$$

とし、 m 本の右辺ベクトル b をまとめた行列 B を

$$B = (b_1 \ b_2 \ \dots \ b_m)$$

とすると、

$$A X = B$$

の解ベクトル行列 X を、解く問題と定義する。

コンテストプログラムの実行

コンテストプログラムの実行 (C言語/ Fortran言語共通)

- ▶ 以下のコマンドを実行する

```
$ cp /home/z30082/spc2015.tar ./
```

```
$ tar xvf spc2015.tar
```

```
$ cd SPC2015
```

- ▶ 以下のどちらかを実行

```
$ cd C : C言語の人
```

```
$ cd F : Fortranの人
```

コンテストプログラムの実行 (C言語/ Fortran言語共通)

```
$ cp Makefile_debug Makefile
```

```
$ cp spcsamp.h spc.h
```

```
$ make
```

```
▶ $ pjsub spc.bash
```

```
▶ 実行が終了したら、以下を実行する
```

```
$ cat spc.out
```

コンテストプログラムの実行 (C言語)

- ▶ 以下のような結果が見えれば成功

N = 192 , M = 192

LU solve time = 0.037481 [sec.]

130.646817 [MFLOPS]

Pass value: 5.493164e-04

Calculated value: 3.092829e-09

OK! Test is passed.

コンテストプログラムの実行 (Fortran言語)

- ▶ 以下のような結果が見えれば成功

NN = 192

MM = 192

LU solve time[sec.] = 6.621912399714347E-02

MFLOPS = 73.94794289654503

Pass value: 5.493164062500000E-04

Calculated value: 2.414639631369455E-09

OK! Test is passed.

サンプルプログラムの説明

- ▶ `spcsamp.h` の中身
- ▶ `#define N 192`
 - ▶ 数字を変更すると、行列サイズが変更できます
- ▶ `#define M 192`
 - ▶ 数字を変更すると、右辺ベクトルbの本数が変更できます

コンテスト課題提出方法

【コンテスト課題実行方法】

1. spc.c (spc.f) の中の関数(手続き) **spc** を並列化してください。
2. **spcFINAL.bash** を実行してください。
 - ▶ **pjsub spcFINAL.bash**

【提出物】

1. 実行後作成される、**spcFINAL1.out**、**spcFINAL2.out**、**spcFINAL3.out**、**spcFINAL4.out**が、提出用の出力リストです。
2. さらに、ソースコードの提出が必要です。

コンテスト課題プログラムの実行方法 (C言語/ Fortran言語共通)

```
$ cp Makefile_final Makefile
```

```
$ pjsub spcFINAL.bash
```

- ▶ 実行が終了したら、以下を確認する

```
$ cat spcFINAL1.out
```

```
$ cat spcFINAL2.out
```

```
$ cat spcFINAL3.out
```

```
$ cat spcFINAL4.out
```

コンテスト課題採点方法

- ▶ `spcFINAL1.out`～`spcFINAL4.out` の実行すべてが、エラーなく実行されれば、**予選通過**です。
- ▶ 予選通過者に対して、`spcFINAL1.h`、および `spcFINAL2.h` で定義された問題の各実行時間について、高速なものから以下の配点をします。
 - ▶ 1位: 10点
 - ▶ 2位: 5点
 - ▶ 3位: 2点
 - ▶ 4位: 1点
 - ▶ 5位以下: 0点
- ▶ 配点の大きい順にソートし、上位1位～3位が入賞です。

コンテストの注意点

1. spc.c (spc.f) の中の関数(手続き)
 spc 内のみコードの変更が可能です。
2. メイン関数内は、変更不可です。
3. 計測ルーチンをいじってはいけません。
4. **spcFINAL.bash** を原則変更してはいけません。
 (ただし、ハイブリッド実行のための
 記述と、キュー名の変更は可能です)

コンテストの注意点

5. 解ベクトルが1-ベクトルとなるように右辺bを生成しています。したがって、**解ベクトル行列x[][]に1を代入して終了**、というような、＜解ベクトルを知っている実装＞をしてはいけません。
 - ▶ 数値解法は、何を使っても構いません。
 - ▶ ただし、反復解法を利用する場合、初期ベクトルは0－1の一様乱数で生成してください。
6. その他、不正と判断された場合は、失格とします。
7. コンテストの注意点の内容について不明な点は、メーリングリストで公開質問してください。

その他の注意点

- ▶ C言語版では、右辺Bの収納配列は、**行列Bを転置した形式**で収納されています。
- ▶ C言語では列方向アクセスが非連続アクセスになる。そのため、Fortran言語の実装に対して、性能劣化する原因となるため。

コンテストの注意点

- ▶ 以上を除いては、何をしてしてもOKです。
 - ▶ LU分解以外の数値解法の実装
 - ▶ コンパイラオプションの変更
 - ▶ 数値計算ライブラリの利用
 - ▶ アセンブラの利用
- など