

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

計算科学演習

実践編（前）: シミュレーションプログラミング

陰山 聡、政田 洋平

神戸大学大学院 システム情報学研究科 計算科学専攻

2012.07.05

今日から 3 週分の内容

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

1 実践編（前半）

シミュレーションプログラミング（陰山）

2 可視化

gnuplot による 1 次元可視化と 2 次元可視化（政田）

3 実践編（後半）

OpenMP + MPI によるハイブリッド並列化（陰山）

今日の講義

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

1 今日話すこと

- 1 誤差に関する注意
- 2 計算時間のおおざっぱな評価
- 3 無駄な計算を省くこと
- 4 コードの検証
- 5 コーディング手法
- 6 Makefile

2 今日話さないこと

- 1 コンパイラオプション
- 2 デバッグ手法
- 3 namelist（パラメータの渡し方）
- 4 最適化手法
- 5 ...

今日の演習で使うコード

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

1 マシン「scalar」

2 `cp -r /tmp/120705` 自分のディレクトリ

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

誤差の評価

例題

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$I = \frac{1}{0.25 \times 1.5} + \frac{1}{1.25 \times 3.5} + \frac{1}{2.25 \times 5.5} + \frac{1}{3.25 \times 7.5} + \cdots$$

- 次の項は？
- 一般項の形は？

この無限級数の値は？

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$I = \sum_{n=0}^{\infty} \frac{1}{(n + 0.25)(2n + 1.5)}$$

$$I = \sum_{n=0}^{\infty} \frac{1}{2n^2 + 2n + \frac{3}{8}}$$

打ち切り誤差

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$I = \frac{1}{0.25 \times 1.5} + \frac{1}{1.25 \times 3.5} + \frac{1}{2.25 \times 5.5} + \cdots$$

有限桁で打ち切る。

$$\tilde{I}(N) = \frac{1}{0.25 \times 1.5} + \frac{1}{1.25 \times 3.5} + \cdots + \frac{1}{(N + 0.25) \times (2N + 1.5)}$$

I の値を小数点以下 9 桁まで求めたい。

打ち切り誤差

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$\begin{aligned} E(N) &= I - \tilde{I}(N) \\ &= \frac{1}{(N + 0.25)(2N + 1.5)} \\ &\quad + \frac{1}{(N + 1.25)(2N + 3.5)} \\ &\quad + \frac{1}{(N + 2.25)(2N + 5.5)} + \cdots \end{aligned}$$

誤差の評価

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$\begin{aligned} E(N) &= \frac{1}{(N+1.25)(2N+3.5)} + \frac{1}{(N+2.25)(2N+5.5)} + \frac{1}{(N+3.25)(2N+7.5)} + \cdots \\ &< \frac{1}{(N+1)(2N+2)} + \frac{1}{(N+2)(2N+4)} + \frac{1}{(N+3)(2N+6)} + \cdots \\ &< \frac{1}{2(N+1)^2} + \frac{1}{2(N+2)^2} + \frac{1}{2(N+3)^2} + \cdots \\ &< \int_N^{\infty} \frac{1}{x^2} dx \\ &= \left[-\frac{1}{x} \right]_N^{\infty} \\ &= \frac{1}{N} \end{aligned}$$

誤差の話（その１）

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

N が大きければかなりよい近似。

$$E(N) \sim \frac{1}{N}$$

小数点以下 9 桁まで求めるには、 $E(N) \sim \frac{1}{N} = 10^{-9}$ 、つまり第 $N = 10^9$ 項まで足せばよい。

誤差にはさまざまな種類がある。

数値計算や計算機シミュレーションを行う前には誤差のオーダーを把握しておくべし。

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

計算時間の評価

$$\tilde{I}(10^9) = \sum_{n=0}^{10^9} \frac{1}{(n + 0.25)(2n + 1.5)}$$

これを計算するのにどの程度時間がかかるか？

- 一つの項を加えるのに
 - 足し算 2 回
 - 掛け算 2 回
 - わり算 1 回
- 全部で 10^9 個の項

全て浮動小数点数で計算されたとしたら、

- 全部で 5G 回の浮動小数点数演算 (FLOP)。
- 1GFLOPS の計算コアであれば 5 秒。
- 100MFLOPS の計算コアであれば 50 秒

この間くらいであろう。

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

【演習】

$$I = \sum_{n=0}^{\infty} \frac{1}{(n + 0.25)(2n + 1.5)}$$

この値を小数点以下 9 桁まで正しく計算する Fortran90 プログラムを書け。（並列化はしなくて良い。）

誤差の話（その2）

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

ガウス少年の話

$$G = 1 + 2 + 3 + \cdots + 100$$

$$\begin{aligned} G &= 1 + 2 + 3 + \cdots + 50 \\ &\quad + 100 + 99 + 98 + \cdots + 51 \\ &= 101 \times 50 \\ &= 5050 \end{aligned}$$

$$\begin{aligned}\tilde{I}(N) &= \frac{1}{0.25 \times 1.5} + \cdots + \frac{1}{(N + 0.25)(2N + 1.5)} \\ &= \frac{1}{0.25 \times 1.5} + \frac{1}{1.25 \times 3.5} + \cdots \\ &\quad + \frac{1}{(N + 0.25)(2N + 1.5)} + \frac{1}{(N - 0.75)(2N + 0.5)} + \cdots\end{aligned}$$

この方法は数学的にメリットがないだけでなく、数値計算的には、大変まずい。なぜか？ ヒント：

- $N = 10^9$ のときに $\frac{1}{(N+0.25)(2N+1.5)}$ はオーダーいくつかな？
- 倍精度で有効桁数は 10 進で何桁かな？

- $N = 10^9$ のときに $\frac{1}{(N+0.25) \times (2N+1.5)} = O(10^{-18})$
- 倍精度の仮数部 52 ビット: $2^{52} = 4.5 \times 10^{15}$: 15 桁と少し。

$$\begin{aligned} & \frac{1}{0.25 \times 1.5} + \frac{1}{(N + 0.25)(2N + 1.5)} \\ = & 2.666666666666667 + 0.000000000000000000 \end{aligned}$$

級数を求めるプログラム

$O(1) + \epsilon$ の形

$$\begin{aligned}\tilde{I}(N) &= \cdots + \frac{1}{(N + 0.25)(2N + 1.5)} \\ &= O(1) + \epsilon\end{aligned}$$

ではこの級数を（倍精度計算で） $N = 10^9$ まで計算することは不可能なのか？

そんなことはない。「塵も積もれば山となる。」

レポート課題

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$I = \frac{1}{0.25 \times 1.5} + \frac{1}{1.25 \times 3.5} + \frac{1}{2.25 \times 5.5} + \frac{1}{3.25 \times 7.5} + \dots$$

の値を小数点以下 9 桁まで求めよ。

(1) ソースコード

(2) 答え

(3) time コマンドで測定した計算時間
を書くこと。

time コマンドの使い方：

```
time ./a.out
```

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

熱伝導問題（復習）

問題設定のイメージ

実践編（前）

陰山・政田

はじめに

誤差の評価

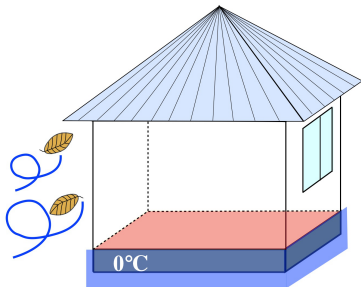
計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

- 一辺 1m の正方形の家。 外気温＝家の壁面温度＝ 0°
- 床暖房システムで床を暖める。床の熱源分布は一様
- 床（正方形）の境界上は壁と接するので 0°
- 床面全体の温度分布は？



問題設定のイメージ

実践編（前）

陰山・政田

はじめに

誤差の評価

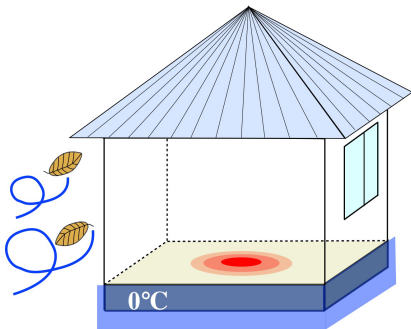
計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

- 床のどの位置が一番暖かい？ 当然中心点



座標系

実践編（前）

陰山・政田

はじめに

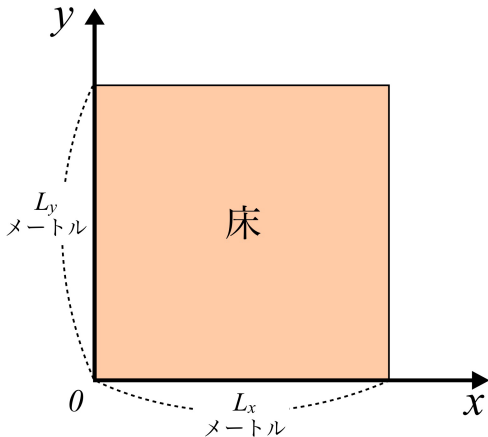
誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良



実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

プログラムの検証

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

先週使った `heat1.f90` は正しいコードであろうか？コードの「検証」。

- 実はいまの「床暖房問題」の中心点での温度（最高温度）の値は計算できる。
- 参考資料： `central_temp.pdf`
- 答え： 0.29406...

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

正解： 0.29406...

【演習】 heat1.f90 で（格子点数を変えるなどして）最高温度を調べよ。

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat1.f90 を $m=50$ で計算した場合の答え： 0.30629...

これが 中心点温度（最高温度）だと仮定すると、誤差は

誤差： $0.30629 - 0.29406 = 0.01223$

格子間隔 $h = 1/m = 0.02000$

「数値計算や計算機シミュレーションを行う前には誤差の
オーダーを把握しておくべし。」

差分誤差は $O(h^2)$ のはず。大きすぎる。なにかおかしい。

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

先週まで使っていた heat1.f90 を修正せよ。（ヒント：3カ所）

```
program heat1
  implicit none
  integer, parameter :: m=50
  ...
  real(DP) :: h, heat=1.0_DP
  allocate(u(0:m+1,0:m+1))
  allocate(un(m,m))
  h=1.0_DP/m
  ...
  if (mod(n,100)==0) print *, n, u(m/2,m/2)

  ...
```

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

```
allocate(u(0:m+1,0:m+1))
```

```
0   1   2   3   4   5   6   7  
o---+---+---+---+---+---+---o
```

```
integer, parameter :: m=6
```

```
h=1.0_DP/m
```

```
if (mod(n,100)==0) print *, n, u(m/2,m/2)
```

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

```
allocate(u(0:m+1,0:m+1))
```

```
0   1   2   3   4   5   6   7  
o---+---+---+---+---+---+---o
```

```
integer, parameter :: m=6 ! => 奇数がよい【なぜ？】  
h=1.0_DP/m                ! => h=1.0_DP/(m+1) が正しい  
if (mod(n,100)==0) print *, n, u(m/2,m/2)  
                        ! => ... これは中心点ではない。
```

verification & validation

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

```
allocate(u(0:m+1,0:m+1))
```

```
0   1   2   3   4   5   6   7   8  
o---+---+---+---+---+---+---+---o
```

```
integer, parameter :: m=7 ! => m は奇数
```

```
h=1.0_DP/(m+1)
```

```
if (mod(n,100)==0) print *, n, u(m/2+1,m/2+1)
```

改良版: heat1b.f90

実践編 (前)

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
(復習)

プログラムの
検証

プログラムの
改良

ソースコード

【演習】 格子点数 m を変更して、収束を確認。

```
pgf90 heat1b.f90
```

```
time ./a.out
```


実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

プログラムの改良

無駄な計算を省くこと

実践編 (前)

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
(復習)

プログラムの
検証

プログラムの
改良

無駄はどこにある？(どうしたら計算を速くできるか？)

【演習】

```
do n=1, nmax
  do j=1, m
    do i=1, m
      un(i,j)=(u(i-1,j)+u(i+1,j)                &
                +u(i,j-1)+u(i,j+1))/4.0_DP      &
                +heat*h*h
    end do
  end do
...

```

pgf90 -O0 でコンパイルし、time コマンドで時間を計ってみよう。

無駄な計算を省くこと

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat1c.f90 をみよ。

名前の重要性

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

- 分かりやすい変数名をつける $m \rightarrow \text{NGRID}$
- 長い変数名を使う `LOOP_MAX`
 - editor の補完機能を使う。vim なら CTRL-P
- ものごとはアルファベット順に並べるものである
 - function, subroutine

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat2.f90

【演習】 このソースコードを理解する。これからこれを変えていく。

熱伝導問題のプログラム全体構造

実践編（前）

陰山・政田

はじめに

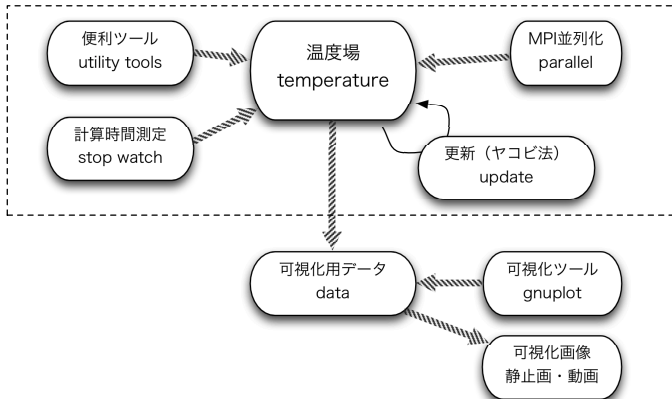
誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良



コードの変更: MPI 並列化

実践編 (前)

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
(復習)

プログラムの
検証

プログラムの
改良

heat3.f90 (＝先週のレポート問題の解答例)

【実習】 heat3.f90 をコンパイルし、MPI 並列計算せよ。
job script: heat.js (中に書いてあるディレクトリ名部分を適宜
修正すること)

```
$ mpif90 heat3.f90
```

```
$ qsub heat.js
```

コードの変更：assert の追加

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
(復習)

プログラムの
検証

プログラムの
改良

- 前提条件（絶対に正しいはずの条件）を念の為に確認する。 assert
 - 万が一条件違反 プログラム停止
 - 走り続けておかしい結果に悩むよりも、潔く止まってくれた方がマシ
 - Fortran90 には assert は組み込まれていない
 - 自分で作って実行時にも呼べばよい。
 - 一般にシミュレーションコードでは (main loop に入れな
い限り) assert のコストは無視できる。

コードの変更：module の活用

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

- ut モジュール（ユーティリティ, utility）
- ut_assert

便利なコーディングテクニック

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

- `implicit none` は当然
- 全てのルーチンは `module` の中に
- `module` に `contain` されたルーチンに `contain` されたルーチンを使うと便利
- `module public/private` を使う。
- できるかぎり `private` にする。 カプセル化
- `public` なルーチン名は `contain` するモジュール名を含ませる
 - 例：モジュール名__名前
- `module`

コードの変更:モジュール

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat4.f90

【演習】heat4.f90 を解読。実行（job 投入）しても可。
ジョブスクリプトは heat.js

コードの変更：Makefile と分割コンパイル

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat4 ディレクトリ

```
$ cd heat4
```

```
$ ls -l
```

Makefile

constants.f90

heat4.js

main.f90

ut.f90

【演習】それぞれのファイルの中身をチェック

Makefile の復習

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

```
make
make job（いきなりこう書いても可）
...
qstat
...
make clean
```

コードの変更: parallel モジュールと temperature モジュール

実践編 (前)

陰山・政田

はじめに

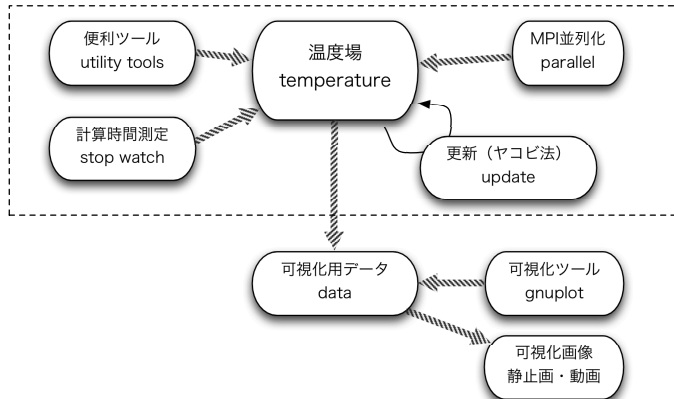
誤差の評価

計算時間の
評価

熱伝導問題
(復習)

プログラムの
検証

プログラムの
改良



コードの変更: parallel モジュールと temperature モジュール

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat5 ディレクトリ

Makefile

constants.f90

heat5.js

main.f90

parallel.f90

temperature.f90

ut.f90

【演習】ファイルの中身をチェックし、job を投入せよ。

時間計測

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

この演習で説明した計時法:

- OpenMP による `omp_get_wtime()` 関数
- MPI による `mpi_wtime()` 関数
- UNIX の `time` コマンド

今回はもう一つの方法を紹介する：

- Fortran95 言語の組み込み関数（サブルーチン）
`cpu_time()`

- 仕様（Fortran95 以降）

`subroutine cpu_time(time)`

`real(SP), intent(out) :: time` !プログラムの始まりから計測した CPU 時間（秒）

`mpi_wtime` は、`mpi_init` を呼んだ後でしか使えない。

stopwatch モジュールを使った時間計測

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

heat6 ディレクトリの

main.f90

temperature.f90

をみよ。（stopwatch が入れ子になっても良いことに注意。）

可視化データの出力

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

【演習】

heat7 ディレクトリを見よ。data ディレクトリに可視化用データが出力されている。

来週このデータを可視化する。

まとめ

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

- 1 計算時間のおおざっぱな評価
- 2 誤差に関する注意
- 3 無駄な計算を省くこと
- 4 コーディングの工夫
 - 1 分かりやすく長い名前を：変数名、ルーチン名、モジュール名
 - 2 ものことはアルファベット順に
 - 3 assert
 - 4 module の活用
 - 5 カプセル化（public/private の活用）
- 5 Makefile を使いこなそう

レポート課題

実践編（前）

陰山・政田

はじめに

誤差の評価

計算時間の
評価

熱伝導問題
（復習）

プログラムの
検証

プログラムの
改良

$$I = \frac{1}{0.25 \times 1.5} + \frac{1}{1.25 \times 3.5} + \frac{1}{2.25 \times 5.5} + \frac{1}{3.25 \times 7.5} + \cdots$$

の値を小数点以下 9 桁まで求めよ。

(1) ソースコード

(2) 答えの数値

(3) time コマンドで測定した計算時間
を書くこと。

提出先： kageyama.lecture@gmail.com

メールタイトル：120705 学籍番号 名字

例：120705 120x227x Yamada

締めきり：2012 年 7 月 12 日午前 9 時