

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

## 第2回シミュレーションスクール

### 第3日: 可視化と発展的問題

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

2010.12.08

# ディレクトリ構造

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

```
scalar計算機上で
mkdir Wed   (今日来られた方のみ)
cd Wed
mkdir data
cp -r /tmp/ss2/Wed/src-work .
今日来られた方:
cd ../Tue
cp -r /tmp/ss2/Tue/src-work .
```

# 昨日の復習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 1 ディレクトリ(フォルダ)構造
- 2 問題設定
- 3 数学的定式化
- 4 差分法による離散化
- 5 ソースコード 006.f95
- 6 コンパイルと実行 `pgf95 -o 006.exe && ./006.exe`

# 今日の目標

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 1 006.f95 をベースに
- 2 データを可視化し、さらに対象問題を発展させる
- 3 まずは復習をかねて実習
- 4 コンパイルと実行 `pgf95 -o 006.exe 006.f95 && ./006.exe`

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 熱源分布はゼロにしている。最後に温度はゼロになるはず
- 格子サイズ  $n_x, n_y$  を変えたら？
- 時間刻み  $\Delta t$  を変えて計算
- $\Delta t = \Delta t_{\text{critical}} \times 0.5$  では？
- $\Delta t = \Delta t_{\text{critical}} \times 0.99$  では？
- $\Delta t = \Delta t_{\text{critical}} \times 1.00$  では？
- $\Delta t = \Delta t_{\text{critical}} \times 1.01$  では？

# 1次元拡散方程式のCFL条件

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間

$$\frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2}$$

$$\frac{T_i(t + \Delta t) - T_i(t)}{\Delta t} = k \frac{T_{i+1}(t) - 2T_i(t) + T_{i-1}(t)}{\Delta x^2}$$

$$\begin{aligned} T_i(t + \Delta t) &= T_i(t) + \frac{k\Delta t}{\Delta x^2} \{T_{i+1}(t) - 2T_i(t) + T_{i-1}(t)\} \\ &= \frac{k\Delta t}{\Delta x^2} \{T_{i+1}(t) + T_{i-1}(t)\} + \left(1 - \frac{2k\Delta t}{\Delta x^2}\right) T_i(t) \\ &= \alpha \frac{T_{i+1}(t) + T_{i-1}(t)}{2} + (1 - \alpha) T_i(t) \\ &= \alpha \times (\text{両隣の平均値}) + (1 - \alpha) \times (\text{自分の値}) \end{aligned}$$

$$\alpha = \frac{2k\Delta t}{\Delta x^2}$$

# 1次元拡散方程式のCFL条件

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

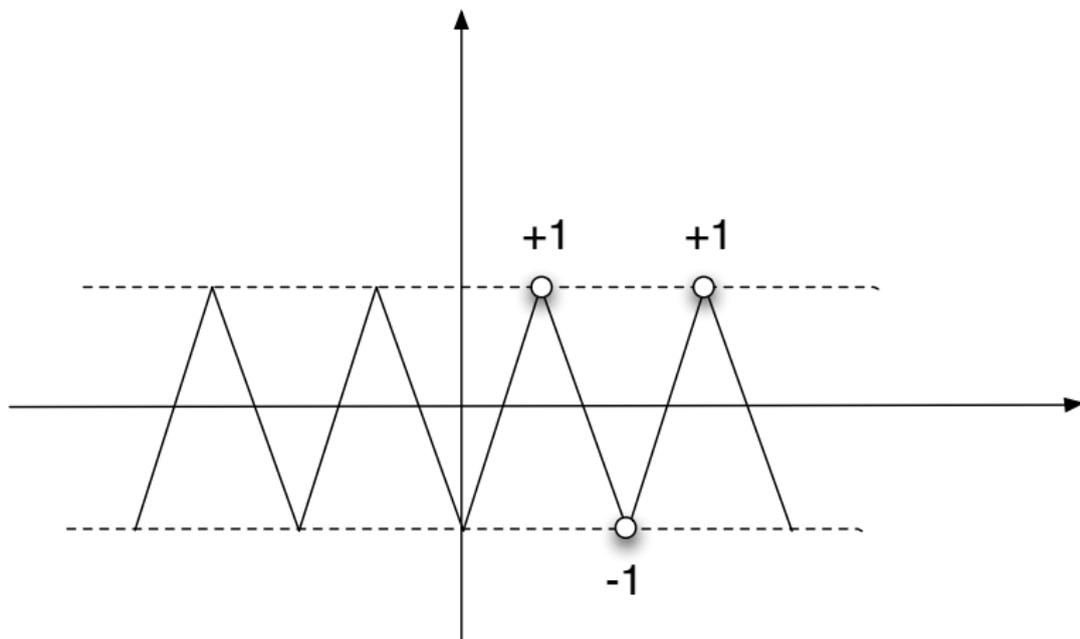
可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)



# 今日のサンプルコード

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- scalar上のホームディレクトリで
- `cp -r /tmp/ss2/Wed/src-work .`

# コードの信頼性

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 昨日のコード 006.f95 は本当に正しいのか？
- 計算結果を定量的に信頼できるのか？
- コードのvalidation

# コードのチェック

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 1 解析解厳密解との比較
- 2 解析的近似解との比較
- 3 保存量(あれば)が確かに保存しているか
- 4 他の計算手法を用いたコードとの比較
- 5 ベンチマーク問題(あれば)の計算
- 6 実験、観測値との比較

# 解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 1 これが最も信頼できるテスト
- 2 しかし解析解があるのは稀。非線形問題では特に。

# 熱拡散問題の解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間

- 1 一辺の長さ1の正方形領域内部の温度分布  $T(x, y)$
- 2 熱拡散係数  $k = 1$  とする。
- 3 熱源あり  $s = 4$  とする。
- 4 原点を正方形の中心にとる。
- 5  $(-\frac{1}{2}, -\frac{1}{2}) \leq (x, y) \leq (\frac{1}{2}, \frac{1}{2})$
- 6 定常解を求める
- 7  $\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + 4 = 0$
- 8 境界条件  $T(x, \pm\frac{1}{2}) = T(\pm\frac{1}{2}, y) = 0$

# 熱拡散定常問題の解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間

$$\left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) T(x, y, t) + 4 = 0 \quad (1)$$

$$T(x, y) = -(x^2 + y^2) + a_0 + \sum_{n=1}^{\infty} a_n (\cos n\pi x \cosh n\pi y + \cosh n\pi x \cos n\pi y) \quad (2)$$

この温度場  $T(x, y)$  は上の方程式(1)の解である。

# 式(2)が解であることを確認

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

1 微分してみればわかる

2 鍵となる関係式:  $\frac{d^2}{dx^2} \cos(n\pi x) = -(n\pi)^2 \cos(n\pi x)$

3 鍵となる関係式:  $\frac{d^2}{dx^2} \cosh(n\pi x) = +(n\pi)^2 \cosh(n\pi x)$

4  $\cosh x = \{\exp x + \exp(-x)\} / 2$

# 解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

式(2)をもう一度書く:

$$T(x, y) = -(x^2 + y^2) + a_0 + \sum_{n=1}^{\infty} a_n (\cos n\pi x \cosh n\pi y + \cosh n\pi x \cos n\pi y)$$

の係数  $a_j$  ( $j = 1, 2, 3, \dots$ )が未知。

- 1 境界条件から $a_j$ を決める。
- 2  $\cosh(n\pi x)$ は $n$ が大きくなると指数関数的に増加する
- 3 だから $a_n$ は $n$ が大きくなると指数関数的にゼロに近づくはず
- 4  $a_0, a_1, a_2$ の三つくらいとれば十分であろう。

# 解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

正方形領域の上の辺  $y = 1/2$  上の温度 (=  $x$  の関数) を  $T^{(2)}(x)$  と書く。

$$\begin{aligned} T^{(2)}(x) &= -x^2 - \frac{1}{4} + a_0 \\ &\quad + \sum_{n=1}^2 a_n \left( \cos n\pi x \cosh \frac{n\pi}{2} + \cosh n\pi x \cos \frac{n\pi}{2} \right) \\ &= -x^2 - \frac{1}{4} + a_0 \\ &\quad + a_1 \left( \cos \pi x \cosh \frac{\pi}{2} + \cosh \pi x \cos \frac{\pi}{2} \right) \\ &\quad + a_2 \left( \cos 2\pi x \cosh \pi + \cosh 2\pi x \cos \pi \right) \\ &= -x^2 - \frac{1}{4} + a_0 + a_1 \cosh \frac{\pi}{2} \cos \pi x \\ &\quad + a_2 \left( \cos 2\pi x \cosh \pi - \cosh 2\pi x \right) \end{aligned}$$

# 解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 1 未知数  $a_0, a_1, a_2$
- 2  $x = 0, \frac{1}{3}, \frac{1}{2}$  で  $T^{(2)}(x) = 0$  という条件から決める。

# 境界条件3つ

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

$$T^{(2)}(0) = -\frac{1}{4} + a_0 + a_1 \cosh \frac{\pi}{2} + a_2 (\cosh \pi - 1) = 0, \quad (3)$$

$$T^{(2)}\left(\frac{1}{2}\right) = -\frac{1}{4} - \frac{1}{4} + a_0 - 2a_2 \cosh \pi = 0, \quad (4)$$

$$T^{(2)}\left(\frac{1}{3}\right) = -\frac{1}{9} - 1 + a_0 + \frac{a_1}{2} \cosh \frac{\pi}{2} + a_2 \left( -\frac{1}{2} \cosh \pi - \cosh \frac{2\pi}{3} \right) = 0. \quad (5)$$

# 解析解

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

式(3)から(4)を解いて

$$a_0 = 0.5889048254279103 \quad (6)$$

$$a_1 = -0.15125367793243916 \quad (7)$$

$$a_2 = 0.0038347646559121653 \quad (8)$$

$$(9)$$

中心 $(x, y) = (0, 0)$ の温度は

$$\begin{aligned} T(0, 0) &= a_0 + 2 \sum_{n=1}^{\infty} a_n \\ &\sim a_0 + 2a_1 + 2a_2 \\ &= 0.2940669988748564 \end{aligned}$$

# 解析的近似解の改良

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

数係数を $n = 2$ までとると

$$T(0,0) = 0.2940669988748564$$

だった。

数係数を $n = 3$ までとると

$$T(0,0) = 0.294832121967086$$

数係数を $n = 4$ までとると

$$T(0,0) = 0.294712632547541$$

神戸大学生 山浦君による。

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 1 熱源分布  $\text{src}(:, :) = 4.0\_DP$  として最終状態の温度の最大値が解析近似解と近いことを確かめよ。
- 2 格子点数  $n_x, n_y$  を変えて計算精度を調べよ。

# 時間計測

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- コードのどの部分でどれだけ時間がかかっているか
- `cpu_time`
- `cpu`時間を測るFortran95の組み込みルーチン
- プログラム開始時からの時間(秒単位)

```
subroutine cpu_time(time)
  real(SP), intent(out) :: time
```

# stopwatchモジュール

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- `cpu_time`ルーチンを利用した時間計測機能
- 複数のストップウォッチを用意
- それぞれのストップウォッチが`strt`と`stop`の二つのボタン
- `strt`から`stop`が押されるまでの時間を足していく
- 最後のそれぞれのストップウォッチの総計時間を出力

# ソースコード解説

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 007.f95
- module stopwatch

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 007.f95をコンパイル&実行し、時間を計測せよ。
- 格子サイズ  $nx$ ,  $ny$ を変更して時間を計測せよ。
- コードを自由に変更して時間を計測せよ。

# 可視化とは

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 何か？
- なぜ必要か？
- どうするのか？
- いつするのか？

# 可視化とは何か？なぜするのか？

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 数値 ⇒ 画像
- 数字から情報を引き出すため
- 計算結果を理解するため
- 計算結果を伝えるため

# 可視化の例

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

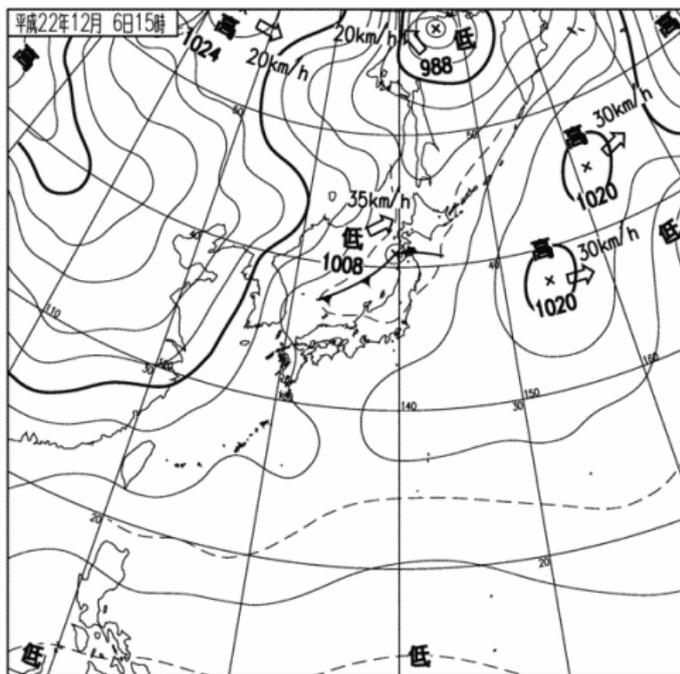
可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)



# いつ可視化をするのか？

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- ポストプロセス(計算後)が普通
- 今までは。
- 今後(計算の大規模化)は不明

# 大規模データ可視化の課題

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- データサイズ  $\Rightarrow$  1 TB  $\sim$  1 PB
- 例えばスーパーコンピュータで10時間計算
- 可視化のための出力データをネットワーク経由で転送
- データサイズ / 転送帯域
- $10 \text{ TB} / 0.1 \text{ GBps} = 10^5 \text{ sec} \sim 24 \text{ 時間}$
- 可視化処理に一日かかかるとする
- 計算時間(10時間) < ポストプロセス(転送時間 + 可視化処理時間)
- この不等式の左辺は今後も変わらず
- 右辺は増大

# 大規模データ可視化

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- ポストプロセスの限界
- インプロセス可視化は不可避
- 複雑現象(構造)の可視化はそれ自体が大規模計算

# 大規模データの可視化： データサイズ

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 時間発展の可視化＝動画ファイル
- 静止画一枚  $O(10^6)$  MB
- 5分間の動画。  $O(10^4)$  フレーム
- 動画ファイルサイズ(時間圧縮なし)  $O(10^{10}) \Rightarrow 10$  GB
- $10^3$ 個の動画ファイル  $\Rightarrow 10$  TB
- 大規模シミュレーションの出力数値データ  $> 1$  PB

# どうやって可視化するか

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 専用可視化ツール
- 汎用可視化ツール (AVS/Express, IDL 等)

今日は gnuplot (汎用グラフソフト) を使う。

# 可視化対象データの分類

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 2D, 3D
- スカラー場、ベクトル場、テンソル場

# 1次元データの可視化

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- $x$ の関数  $f(x)$
- $f(x) = x^x$ はどんな関数か？
- グラフを書く(可視化する)のが一番

# 1次元データの可視化

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

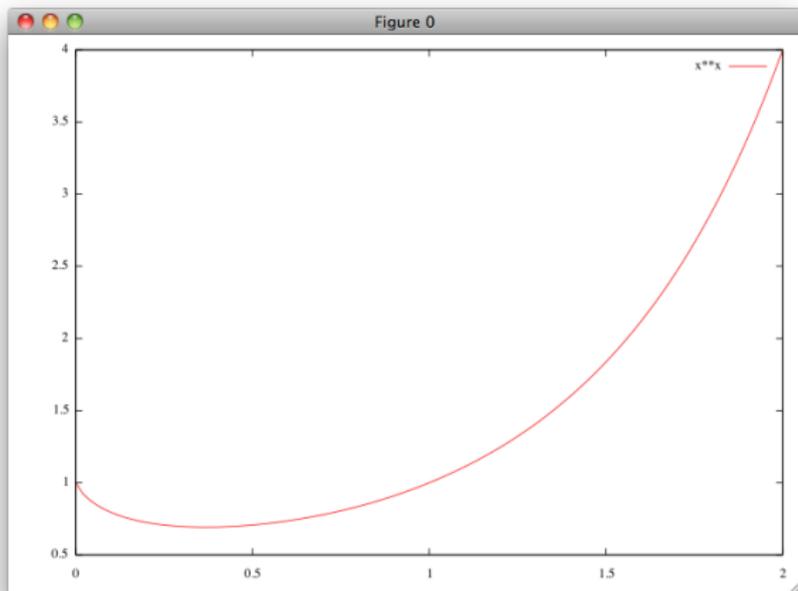
gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

$x^x$  のグラフ



# 2次元データの可視化

熱拡散

## 等高線

陰山・政田

はじめに

コードのテストと解析解

時間計測

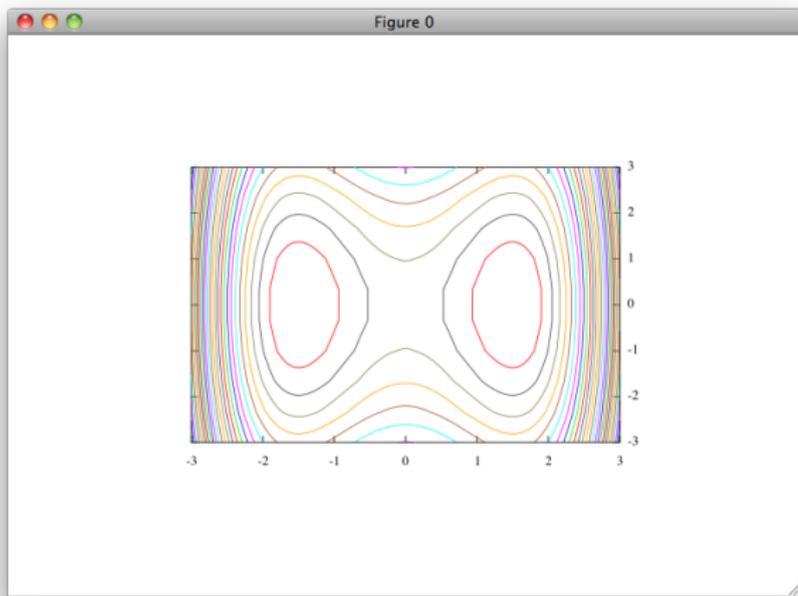
可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)



# gnuplot

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 各自の端末 (scalarではなく) で
- gnuplot
- と打てば gnuplot が立ち上がる
- quit
- で終了

path: /mnt/nfs/usr/bin/gnuplot

# gnuplot

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 各自の端末 (scalarではなく) で
- gnuplot
- と打って gnuplot が立ち上げ、
- `plot sin(x)`
- と打つ

# $\sin(x)$ のグラフ

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

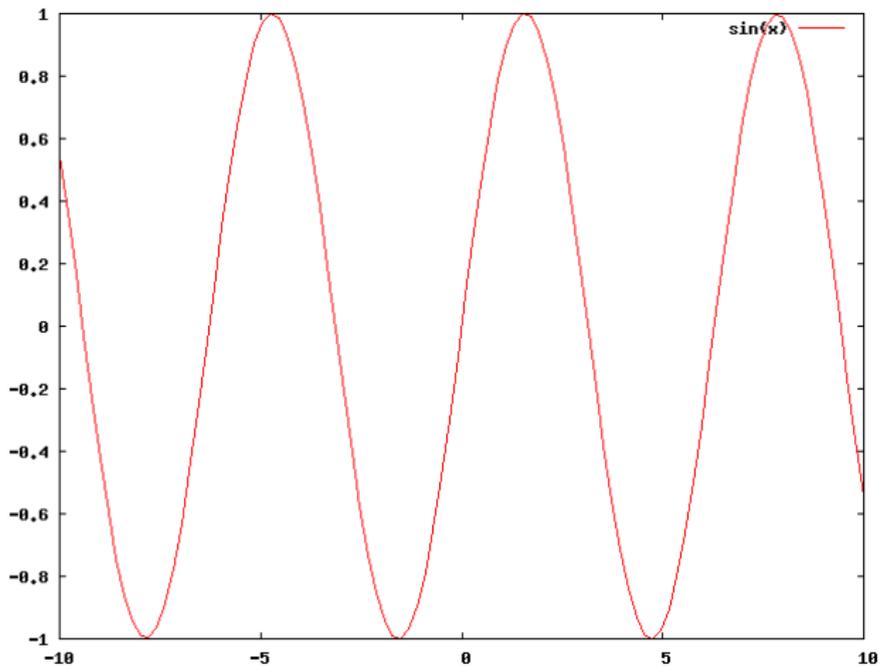
可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)



# 温度分布データを出力するプログラム

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 008.f95
- 1次元プロファイル: 長方形の  $y$  が中間の値  $x$  の関数 ( $j=ny/2$  上)
- 2次元プロファイル: 長方形全体  $x, y$  の関数
- 出力ルーチンは
- `call ut__print_1d_prof(nx,tmp(:,ny/2))`
- `call ut__print_2d_prof(nx,ny,tmp)`

# コード解説

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

`$HOME/Wed/src-work/008.f95`

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 008.f95を使う
- データ出力用のディレクトリ `../data` があることを確認
- コンパイル&実行 `pgf95 -o 008.exe 008.f95 && ./008.exe`
- `../data` ディレクトリにファイルが作られたことを確認
- それらのデータファイル(テキストデータ)の中身をエディタ等で確認
- 008.f95を自由に変更して実行。データファイル確認。

# 実習1次元グラフ

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

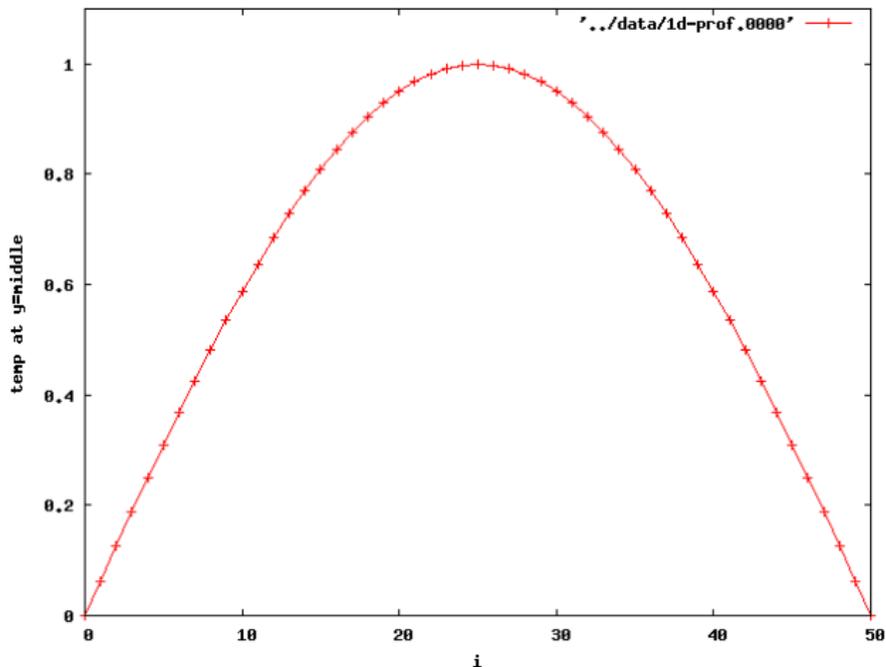
gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

## 初期状態の温度分布を見る



# gnuplotスクリプト

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

gnuplotのプロンプトに直接コマンドを打ち込むことも可能だが、スクリプトを使ったほうが簡単で間違いが少ない。

前ページの図を作るためのスクリプト `$HOME/Wed/`

`008-plot1d.gp`

# 実行方法

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

(scalarではなく)目の前のパソコンから src-work に cd して、  
gnuplot 008-plot1d.gp

# 演習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 008-plot1d.gp を編集し、
- `plot '../data/1d-prof.0000' w lp`
- の部分の数字0000を変更して他の時刻のグラフを描け

# 時間発展の可視化

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- アニメーション
- 静止画を次々とplotすればよい
- コマとコマの間に適当な秒数の休止 pause を入れる

# アニメーション用gnuplotスクリプト

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- コマ数が多いと手で編集するのは大変
- スクリプト自動生成プログラム
- `$HOME/Wed/src-work/008-plot1d-anime-gp-generator.f95`

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

```
$HOME/Wed/src-work/008-plot1d-anime-gp-generator.f95
```

コンパイルして実行

```
pgf95 -o runme 008-plot1d-anime-gp-generator.f95  
./runme > animator
```

animator ファイルの中身をエディタ等で確認

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

gnuplot animator

アニメーションが見えるはず

# gnuplotの2次元データ形式

熱拡散

x00 y00 関数値

陰山・政田

x01 y00 関数値

はじめに

x02 y00 関数値

コードのテストと解析解

•

時間計測

•

可視化とは

•

gnuplotによる1次元可視化

x09 y00 関数値  
(空行)

gnuplotによる2次元可視化

x00 y01 関数値

x01 y01 関数値

x02 y01 関数値

応用編

•

格子点数と計算時間)

•

•

x09 y01 関数値  
(空行)

# gnuplotの2次元データ形式

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

../data/2d-prof.0000 等がこのファイル形式になっていることを確認。

# 等高線

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

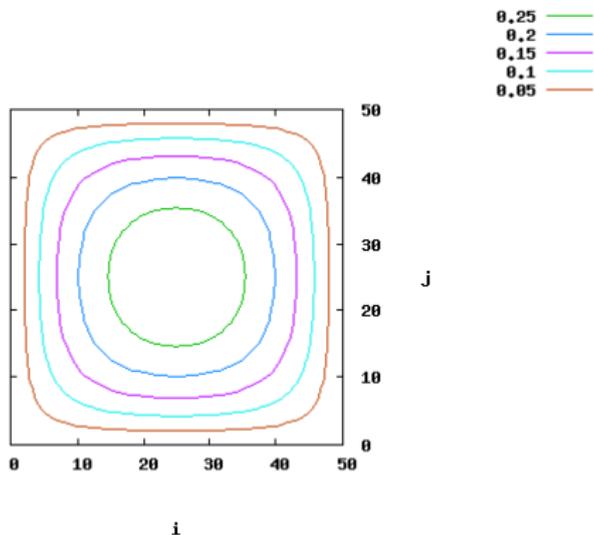
gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

Temperature at 0000



# 等高線

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

gnuplotスクリプト

`$Home/Wed/src-work/008-plot2d.gp`

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

008-plot2d.gp 中の描画対象データファイルを指定してる部分を変更し、別の時刻の等高線を描け。

# 鳥瞰図

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

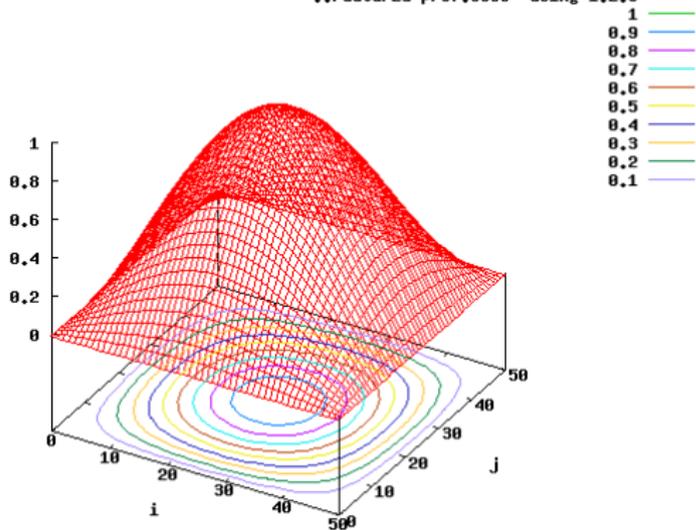
gnuplotによる2次元可視化

応用編

格子点数と計算時間)

Temperature at 0000

'../data/2d-prof.0000' using 1:2:3



# gnuplotスクリプト

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

`$Home/Wed/src-work/008-plot2d-birdseyeview.gp`

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

gnuplotスクリプト `$Home/Wed/src-work/`

`008-plot2d-birdseyeview.gp` を変更し、別の時刻のデータを表示するようにせよ。

- これまで熱源分布が一様 ( $x$ と $y$ に依らない)としてきた。
- 床暖房を考えると、熱源分布が一様でない場合の定常温度分布を知りたい。
- 熱源  $\text{src}(:, :)$  が市松模様になっているときの温度分布を解け。
- $x > 0$   $y > 0$  なら  $\text{src} = +1$
- $x > 0$   $y < 0$  なら  $\text{src} = -1$
- $x < 0$   $y > 0$  なら  $\text{src} = -1$
- $x < 0$   $y < 0$  なら  $\text{src} = +1$

# 実習

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

## 2次元可視化アニメーションプログラム スクリプト生成プログラム

```
/tmp/ss2/Wed/src-sample/008-plot2d-anime-gp-generator.f95  
[on scalar] cp /tmp/ss2/Wed/src-sample/  
008-plot2d-anime-gp-generator.f95 .  
[on scalar] pgf95 -o runme 008-plot2d-anime-gp-generator.f95  
[on scalar] ./runme > animator  
[on PC] gnuplot animator
```

# 格子点数を大きく取る

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

計算時間が増える。  
どれくらいの割合で増えるであろうか？

実習:  $n_x(=n_y)$  を 10, 20, 40, 80, 160, ... と増やして計算する。

# 単一プロセッサ計算の限界

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- これまでの例題ではプロセッサコア一つしか使っていない。
- 理想的には、二つのコアを使えば2倍、四つのコアを使えば4倍早く計算時間が終わるはず。
- プログラムの並列化の必要性

# 明日以降の講義(並列化) への準備

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

- 対象問題は同じ
- モジュールによるカプセル化を駆使したプログラムは実用的だが、コードが長くなる
- 並列化の問題に注意を集中するために計算手法とコードを限界まで単純化する
- 計算手法の単純化: 時間発展の刻み幅 $\Delta t \Rightarrow$  CFL条件の限界にとる
- この場合、「ヤコビ法」と呼ばれる。
- コードの単純化: 必要最小限の機能にまでコードをそぎ落とす。

# ヤコビ法

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間

これまで使っていた式

$$\begin{aligned} T_{i,j}(t + \Delta t) = & \Delta t \{ \alpha_x (T_{i+1,j}(t) + T_{i-1,j}(t)) \\ & + \alpha_y (T_{i,j+1}(t) + T_{i,j-1}(t)) \\ & + s_{i,j} \} \\ & + (1 - \beta \Delta t) T_{i,j}(t) \end{aligned}$$

$\Delta t$ をCFL条件の限界( $\Delta t_{\text{critical}} = 1/\beta$ )にとる。

右辺の最後の項( $T_{i,j}$ に直接依存する項)がゼロとなる。

# ヤコビ法

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

$$\begin{aligned} T_{i,j}(t + \Delta t) = & \frac{\alpha_x}{\beta} \{T_{i+1,j}(t) + T_{i-1,j}(t)\} \\ & + \frac{\alpha_y}{\beta} \{T_{i,j+1}(t) + T_{i,j-1}(t)\} \\ & + \frac{1}{\beta} s_{i,j} \end{aligned}$$

⇒ ある格子点の新しい温度は、周りの格子点の温度の平均値

さらに  $\Delta x = \Delta y$  ならもっと簡単になる。

# 定常熱伝導問題をヤコビ法で解く短いコード

熱拡散

陰山・政田

はじめに

コードのテストと解析解

時間計測

可視化とは

gnuplotによる1次元可視化

gnuplotによる2次元可視化

応用編

格子点数と計算時間)

009.f95