MPIによる並列アプリケーション開発「超」入門（I）

東京大学情報基盤センター
中島 研吾

お試しアカウント付き並列プログラミング講習会（試行）

本講習の目的

- 1CPU用シリアルアプリケーションの並列化（Flat MPI）
  - 並列分散データ構造の重要性
- 並列アプリケーションを使用したシミュレーションの手順を一通り体験する
  - データ生成（初期データ，領域分割）
  - 計算
- 対象
  - 差分法による熱伝導解析コード
  - 共役勾配法（CG法）による連立一次方程式求解

スケジュール（予定）

- 1300～1415　背景，一次元熱伝導解析プログラムの概要
- 1415～1545　並列データ構造の考え方
- 1600～1630　一次元熱伝導解析プログラムの並列化
- 1630～　実習

背　景

- 「T2Kオープンスパコン」，「次世代スーパーコンピュータ」等の開発を背景に，大規模並列シミュレーションへの期待は，産学において一層高まっている。
- 並列計算機を使いこなすためには，「並列プログラミング」の習得が必須である。
  - 並列計算機を使いこなす，アプリーション分野の研究者，技術者の育成

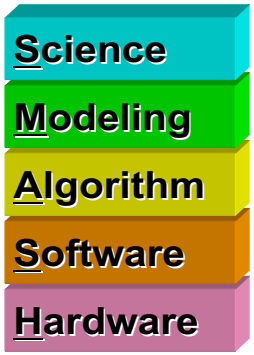
学際計算科学・工学 人材育成プログラム（東京大学）

- 平成21年度から実施
  - 平成20年度冬学期は試験的に実施
  - 並行してガイドライン策定中
- 地球惑星科学専攻における教育プログラムがモデル

4S型人材育成戦略

「4つのS」：System，Stage，Status，Style（1/2）

- **System**
  - SMASH
  - 科学技術計算の真髄



- **Stage：4つの段階**
  - 並列プログラミングへの道
  - ③が最も重要，かつ教育困難
    - 現状はアルゴリズム中心（連続体力学）

4S型人材育成戦略

「4つのS」：System，Stage，Status，Style（2/2）

- **Status：4レベルの人材**
  - **A型：「SMASH」**
    - 並列プログラムを自力開発（可視化，数値ライブラリ等は除く）
  - **B型：「SMASH」**
    - 「HPC-MW」等開発基盤により並列プログラムを自力開発
    - オープンソースコード自力改良
  - **C型：「SMASH」**
    - 既存プログラムのユーザー
  - S型（究極の人材）
    - A型に加えて・・・
    - ライブラリ，開発基盤開発に貢献
    - 真に「計算科学～計算機科学」融合に役立つ「隙間家具」的人材
  - 各レベルに対応した教育プログラムを提供することが重要
- **Style：様々な形態**
  - 講義・演習，集中講義，（遠隔）講習会，e-Learning
  - 様々なバックグラウンドの受講者の多様なニーズに柔軟に対応
  - 受講者の負担を極力増やさない：コマ数をなるべく増やさない

多圏地球COEにおける並列計算プログラミング教育

- 「並列計算プログラミング」，「先端計算機演習I・II」
- 東京大学　地球惑星科学専攻（FY.2004~）

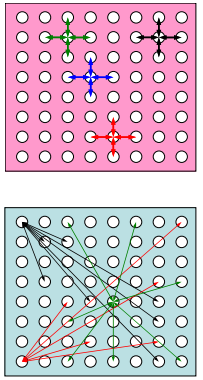
- 「地球シミュレータ」に代表されるような大型並列計算機を使いこなし，未解決の問題にチャレンジしていくような研究者を育成することが重要な目標の一つである。
- これまで，計算機科学を専門としない学生に対して科学技術シミュレーションのための並列プログラミング技術を体系的に教える授業は，日本では皆無であった。
- 今回の講習内容はこの教育内容に準拠

講義・演習の目標

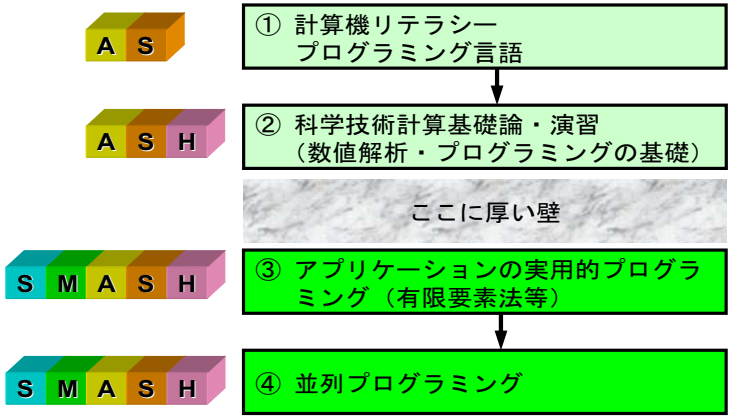
- 科学技術計算における様々な手法・アルゴリズムを理解し，各アルゴリズムにおいて並列性を引き出す様々な工夫をすることの重要性，に主眼を置いた
  - 元のアプリケーションの中身（アルゴリズム，実装）がよくわかっていれば，「並列化」は難しくない
  - MPIの関数を知っていることも重要ではあるが，最も重要なのは「並列分散データ構造」をいかに設計するか
- アルゴリズムを教えるだけでなく，プログラミングもちゃんと教えた

題 材

- 局所的手法（差分法，有限要素法等）
  - 有限体積法（ガウス・グリーン型）
  - 疎行列
- 大域的手法（境界要素法，スペクトル法等）
  - 境界要素法：FY.2004，2005
  - 粒子間熱伝導：FY.2006，2007
  - 密行列
- 幅広いバックグラウンドの受講生への配慮
  - 専攻内各大講座，天文，物理，工学系，情報理工学系
  - FORTRAN，C



並列プログラミングへの道



- ③が重要であるが，プログラミング教育はほとんど実施されていない

③（実用プログラミング）を学ぶには?

- プログラミング能力をつけるためには，徹底して実アプリケーションコードのソースを「読む」能力をつけることが重要（特に学部4年～大学院初級）
  - 英語，漢文の音読のごとく
- 並列計算で大事なのはMPI等の文法ではなく，並列データ構造等の設計。対象アプリケーション，アルゴリズムに対する深い理解が必要⇒③が重要
- これまでの経験で効果は確認済

本講習の方針

- まずは1CPU用のシリアルアプリケーションコードを徹底理解
  - アルゴリズムのやさしい一次元差分法を選択
  - 熱伝導方程式
- アプリケーションに適した並列計算のためのデータ構造（＝局所分散データ構造）を考える
- 慣れているFORTRANで解説していますが，C言語版のアプリケーションも準備，ほとんど各行解説するので，余り支障は無いでしょう

第一部　概 要

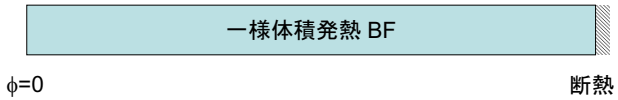
- 一次元熱伝導方程式と差分法
- 連立一次方程式の解法
- 反復法について
  - 共役勾配法（CG）法
  - 前処理について
- 共役勾配法を使用した一次元熱伝導解析プログラムについて

一次元熱伝導方程式（1/3）

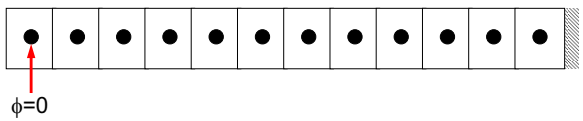
支配方程式：簡単のため熱伝導率=1

$$\frac{d^2\phi}{dx^2} + BF = 0, \quad \phi = 0 @ x = 0, \frac{d\phi}{dx} = 0 @ x = x_{\max}$$

$$\phi = -\frac{1}{2} BF\, x^2 + BF\, x_{\max}\, x$$



実際は以下のような離散化をしているので注意が必要

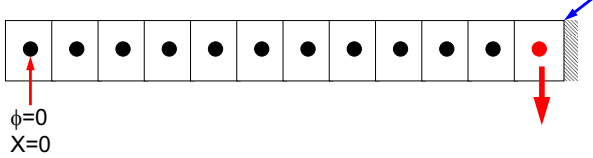


断熱となっているのはこの面，しかし温度は計算されない

一次元熱伝導方程式（2/3）

解析解

$$\phi = -\frac{1}{2} BF\, x^2 + BF\, x_{\max}\, x$$



Δx=1.d0，メッシュ数=50，とすると，X_{max}=49.5，
●の点のX座標は49.0となる。BF=1.0d0とすると●での温度は：

$$\phi = -\frac{1}{2} 49^2 + 49.5 \times 49 = -1200.5 + 9850.5 = 1225$$

ファイルコピー

```
$ cp /home/z30088/1dheat.tar .
$ tar xvf 1dheat.tar
$ cd 1dheat

このディレクトリを<$1DH>と呼ぶ

$ ls
  data    parallel    single
```


プログラムのコンパイル，実行法

- プログラムのありか

```
$ cd <$1DH>/single
$ ls heat_cg.f
```

```
$ cd <$1DH>/single
$ ls heat_cg.c
```

- コンパイル，実行法

```
$ cd <$1DH>/single
$ f90 heat_cg.f -o cg
$ ./cg
```

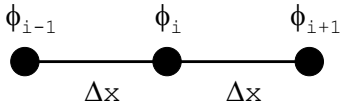
```
$ cd <$1DH>/single
$ cc heat_cg.c -o cg
$ ./cg
```

input.dat

50	N	メッシュ数
1.d0 1.e0	dx, BF	メッシュ幅，体積発熱量
5000	ITERmax	最大反復回数
1.e-7	EPS	打切誤差

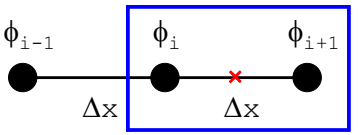
念のため・・・・差分について

- 差分法：Finite Difference Method
- マクロな微分
  - 微分係数を数値的に近似する手法
- 以下のような一次元系を考える



直感的・・・というか安易な定義

- ×（iとi+1の中点）における微分係数



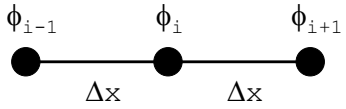
$$\left(\frac{d\phi}{dx}\right)_{i+1/2} \approx \frac{\phi_{i+1}-\phi_i}{\Delta x}$$

Δx→0となると微分係数の定義そのもの

- iにおける二階微分係数

$$\left(\frac{d^2\phi}{dx^2}\right)_i \approx \frac{\left(\frac{d\phi}{dx}\right)_{i+1/2}-\left(\frac{d\phi}{dx}\right)_{i-1/2}}{\Delta x} = \frac{\frac{\phi_{i+1}-\phi_i}{\Delta x}-\frac{\phi_i-\phi_{i-1}}{\Delta x}}{\Delta x} = \frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}$$

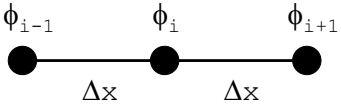
厳密な定義：Taylor展開（1/3）



$$\phi_{i+1} = \phi_i + \Delta x\left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\phi_{i-1} = \phi_i - \Delta x\left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i - \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

厳密な定義：Taylor展開（2/3）



前進差分

$$\phi_{i+1} = \phi_i + \Delta x\left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\frac{\phi_{i+1} - \phi_i}{\Delta x} = \left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + \frac{(\Delta x)^2}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

打ち切り誤差がΔxのオーダー（一次精度）

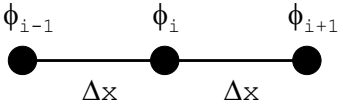
後退差分

$$\phi_{i-1} = \phi_i - \Delta x\left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i - \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\frac{\phi_i - \phi_{i-1}}{\Delta x} = \left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + \frac{(\Delta x)^2}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

打ち切り誤差がΔxのオーダー（一次精度）

厳密な定義：Taylor展開（3/3）



中央差分，中心差分

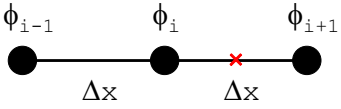
$$\phi_{i+1} = \phi_i + \Delta x\left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i + \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\phi_{i-1} = \phi_i - \Delta x\left(\frac{\partial \phi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i - \frac{(\Delta x)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

$$\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} = \left(\frac{\partial \phi}{\partial x}\right)_i + \frac{2\times(\Delta x)^2}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i \cdots$$

打ち切り誤差が$(\Delta x)^2$のオーダー（二次精度）

安易な定義：実は二次精度だった



$$\phi_{i+1} = \phi_{i+1/2} + \Delta x/2\left(\frac{\partial \phi}{\partial x}\right)_{i+1/2} + \frac{(\Delta x/2)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_{i+1/2} + \frac{(\Delta x/2)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_{i+1/2} \cdots$$

$$\phi_i = \phi_{i+1/2} - \Delta x/2\left(\frac{\partial \phi}{\partial x}\right)_{i+1/2} + \frac{(\Delta x/2)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_{i+1/2} - \frac{(\Delta x/2)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_{i+1/2} \cdots$$

$$\frac{\phi_{i+1} - \phi_i}{\Delta x} = \left(\frac{\partial \phi}{\partial x}\right)_{i+1/2} + \frac{2\times(\Delta x/2)^2}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_{i+1/2} \cdots$$

打ち切り誤差が$(\Delta x)^2$のオーダー（二次精度）

二点間の中点で二次精度，それ以外の点では一次精度・・・ということもできる。
Δxが均一でない場合も同様のことが起こる。

一次元熱伝導方程式(3/3)

要素単位の線形方程式

- 差分法による離散化

$$\left(\frac{d^2\phi}{dx^2}\right)_i \approx \frac{\left(\frac{d\phi}{dx}\right)_{i+1/2} - \left(\frac{d\phi}{dx}\right)_{i-1/2}}{\Delta x} = \frac{\frac{\phi_{i+1}-\phi_i}{\Delta x} - \frac{\phi_i-\phi_{i-1}}{\Delta x}}{\Delta x} = \frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}$$

- 各要素における線形方程式は以下のような形になる

$$\frac{d^2\phi}{dx^2} + BF = 0 \quad \rightarrow$$

$$\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2} + BF(i) = 0 \quad (1 \le i \le N)$$

$$\frac{1}{\Delta x^2}\phi_{i+1} - \frac{2}{\Delta x^2}\phi_i + \frac{1}{\Delta x^2}\phi_{i-1} + BF(i) = 0 \quad (1 \le i \le N)$$

$$A_L(i)\times\phi_{i-1} + A_D(i)\times\phi_i + A_R(i)\times\phi_{i+1} = BF(i) \quad (1 \le i \le N)$$

$$A_L(i) = \frac{1}{\Delta x^2},\ A_D(i) = -\frac{2}{\Delta x^2},\ A_R(i) = \frac{1}{\Delta x^2}$$

連立一次方程式:N=8の場合

未知数8，方程式の数8

$$i=1: A_D(1)\times\phi(1) + A_R(1)\times\phi(2) = BF(1)$$
$$i=2: A_L(2)\times\phi(1) + A_D(2)\times\phi(2) + A_R(2)\times\phi(3) = BF(2)$$
$$i=3: A_L(3)\times\phi(2) + A_D(3)\times\phi(3) + A_R(3)\times\phi(4) = BF(3)$$
$$i=4: A_L(4)\times\phi(3) + A_D(4)\times\phi(4) + A_R(4)\times\phi(5) = BF(4)$$
$$i=5: A_L(5)\times\phi(4) + A_D(5)\times\phi(5) + A_R(5)\times\phi(6) = BF(5)$$
$$i=6: A_L(6)\times\phi(5) + A_D(6)\times\phi(6) + A_R(6)\times\phi(7) = BF(6)$$
$$i=7: A_L(7)\times\phi(6) + A_D(7)\times\phi(7) + A_R(7)\times\phi(8) = BF(7)$$
$$i=8: A_L(8)\times\phi(7) + A_D(8)\times\phi(8) = BF(8)$$

連立一次方程式:N=8の場合

自分とその周囲のみに非ゼロ成分(各行3つ):疎行列

	1	2	3	4	5	6	7	8
1	AD1	AR1						
2	AL2	AD2	AR2					
3		AL3	AD3	AR3				
4			AL4	AD4	AR4			
5				AL5	AD5	AR5		
6					AL6	AD6	AR6	
7						AL7	AD7	AR7
8							AL8	AD8

- このような「疎な」行列を係数とする連立一次方程式を解く。

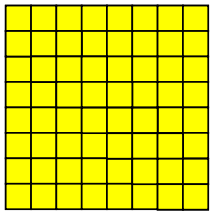
  [A]{x}={b}

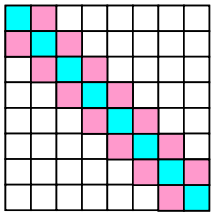
$$A_L(i)\times\phi(i-1) + A_D(i)\times\phi(i) + A_R(i)\times\phi(i+1) = BF(i) \quad (1 \le i \le 8)$$

$$A_L(i) = \frac{1}{\Delta x^2},\ A_D(i) = -\frac{2}{\Delta x^2},\ A_R(i) = \frac{1}{\Delta x^2}$$

係数行列は疎行列(Sparse Matrix)

- 0が多い
- 非ゼロ成分のみを記憶する方法
  - Compressed Row Storage, CRS(後述)
- 密行列と比較して取り扱いが困難





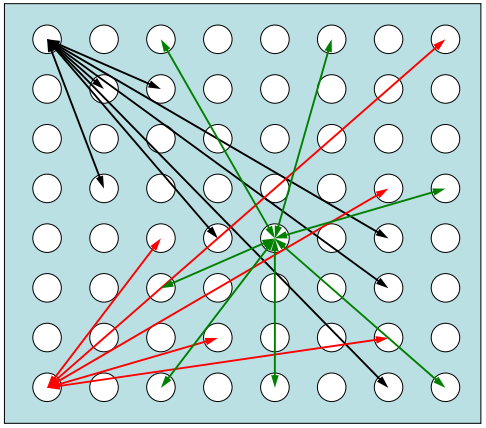
- 一次元熱伝導方程式と差分法
- 連立一次方程式の解法
- 反復法について
  - 共役勾配法（CG）法
  - 前処理について
- 共役勾配法を使用した一次元熱伝導解析プログラムについて

科学技術計算における大規模線形方程式の解法

- 多くの科学技術計算は，最終的に大規模線形方程式Ax=bを解くことに帰着される：「線形ソルバー」
  - important, expensive
- アプリケーションに応じて様々な手法が提案されている
  - 疎行列（sparse），密行列（dense）
  - 直接法（direct），反復法（iterative）
    - 本講義・演習では疎行列，反復法について扱う。
- 密行列（dense）
  - グローバルな相互作用あり：BEM，スペクトル法，MO，MD（気液）
- 疎行列（sparse）
  - ローカルな相互作用：FEM，FDM，MD（固），高速多重極展開付BEM

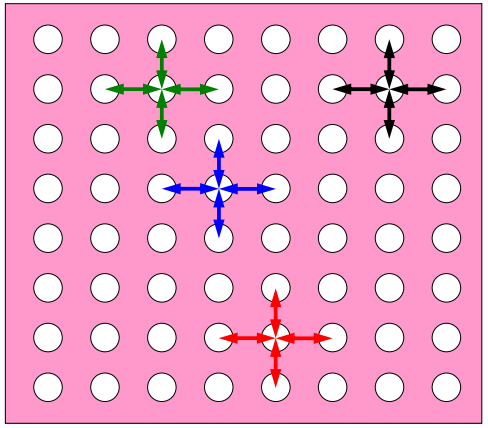
（密行列）

遠隔領域も含め，多数領域との相互作用あり
境界要素法，スペクトル法，MD法



（疎行列）

近接領域のみとの相互作用
差分法，有限要素法



直接法（Direct Method）

- Gaussの消去法，完全LU分解
  - 逆行列A^{-1}を直接求める
- 利点
  - 安定，幅広いアプリケーションに適用可能
    - Partial Pivoting
  - 疎行列，密行列いずれにも適用可能
- 欠点
  - 反復法よりもメモリ，計算時間を必要とする
    - 密行列の場合，$O(N^3)$の計算量
  - 大規模な計算向けではない
    - $O(N^2)$の記憶容量，$O(N^3)$の計算量

反復法（Iterative Method）

- 定常（stationary）法
  - 反復計算中，解ベクトル以外の変数は変化せず
  - SOR，Gauss-Seidel，Jacobiなど
  - 概して遅い
- 非定常（nonstationary）法
  - 拘束，最適化条件が加わる
  - Krylov部分空間（subspace）への写像を基底として使用するため，Krylov部分空間法とも呼ばれる
  - CG（Conjugate Gradient：共役勾配法）
  - BiCGSTAB（Bi-Conjugate Gradient Stabilized）
  - GMRES（Generalized Minimal Residual）

反復法（Iterative Method）（続き）

- 利点
  - 直接法と比較して，メモリ使用量，計算量が少ない。
  - 並列計算には適している。
- 欠点
  - 収束性が，アプリケーション，境界条件の影響を受けやすい。
  - 前処理（preconditioning）が重要。

- 一次元熱伝導方程式と差分法
- 連立一次方程式の解法
- 反復法について
  - 共役勾配法（CG）法
  - 前処理について
- 共役勾配法を使用した一次元熱伝導解析プログラムについて

代表的な反復法：共役勾配法

- Conjugate Gradient法，略して「CG」法
  - 最も代表的な「非定常」反復法
- 対称正定値行列（Symmetric Positive Definite：SPD）
  - 任意のベクトル$\{x\}$に対して$\{x\}^T[A]\{x\}>0$
  - 全対角成分>0，全固有値>0，全部分行列式>0と同値
  - （ガラーキン法）熱伝導，弾性，ねじり：本コードの場合もSPD
- アルゴリズム
  - 最急降下法（Steepest Descent Method）の変種
  - $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
    - $x^{(i)}$：反復解，$p^{(i)}$：探索ベクトル，α_i：定数）
  - 厳密解をyとするとき $\{x-y\}^T[A]\{x-y\}$を最小とするような$\{x\}$を求める。
  - 未知数N個とするとN回の反復で収束（実際は丸め誤差の影響）
  - 詳細は参考文献参照
    - 例えば：森正武「数値解析（第2版）」（共立出版）

共役勾配法のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減

$x^{(i)}$ ：ベクトル
α_i ：スカラー

共役勾配法のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減

$x^{(i)}$ ：ベクトル
α_i ：スカラー

共役勾配法のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減

$x^{(i)}$ ：ベクトル
α_i ：スカラー

共役勾配法のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

- 行列ベクトル積
- ベクトル内積
- ベクトル定数倍の加減

$x^{(i)}$:ベクトル
α_i :スカラー

前処理(preconditioning)とは?

- 反復法の収束は係数行列の固有値分布に依存
  - 固有値分布が少なく，かつ1に近いほど収束が早い(単位行列)
  - 条件数(condition number)(対称正定)＝最大最小固有値の比
    - 条件数が1に近いほど収束しやすい
- もとの係数行列 **[A]** に良く似た前処理行列 **[M]** を適用することによって固有値分布を改善する。
  - 前処理行列 **[M]** によって元の方程式 [A]{x}={b}を [A′]{x′}={b′}へと変換する。ここで[A′]=[M]$^{-1}$[A], {b′}=[M]$^{-1}${b} である。
  - [A′]=[M]$^{-1}$[A]が単位行列に近ければ良い，ということになる。
- 「前処理」は密行列，疎行列ともに使用するが，普通は疎行列を対象にすることが多い。

前処理付き共役勾配法のアルゴリズム

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    solve [M]z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
     else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```


ILU(0), IC(0)

- 最もよく使用されている前処理(疎行列用)
  - 不完全LU分解
    - Incomplete LU Factorization
  - 不完全コレスキー分解
    - Incomplete Cholesky Factorization(対称行列)
- 不完全な直接法
  - もとの行列が疎でも，逆行列は疎とは限らない。
  - fill-in
  - もとの行列と同じ非ゼロパターン(fill-in無し)を持っているのがILU(0)，IC(0)

大規模線形ソルバーの動向

- 反復法がより広く使用されるようになりつつある
  - 100コアを超えるような大規模並列システムでは直接法は並列性能が出ない：逆にそれより小さければ直接法でもOKということになる。
  - 密行列も反復法で解くような試みがなされている。
- 密行列を使わないで済ませられるようなアルゴリズムの開発
  - 高速多重極展開（Fast Multipole）
  - 遠方からの効果をクラスタリング，あるいは無視
  - 密行列
    - メモリースケーラブルではない
- 前処理付き反復法（preconditioned iterative solvers）
  - 安定した前処理の必要性
  - 安定した前処理は概して「並列化」が困難

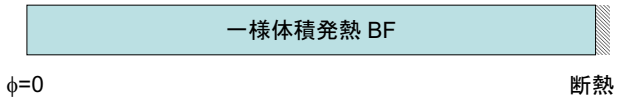
- 一次元熱伝導方程式と差分法
- 連立一次方程式の解法
- 反復法について
  - 共役勾配法（CG）法
  - 前処理について
- 共役勾配法を使用した一次元熱伝導解析プログラムについて

再び一次元熱伝導方程式（1/3）

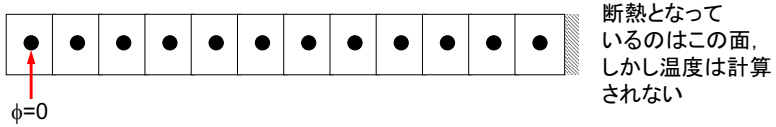
支配方程式：熱伝導率＝1（一様）

$$\frac{d^2\phi}{dx^2} + BF = 0, \quad \phi = 0 @ x = 0, \frac{d\phi}{dx} = 0 @ x = x_{\max}$$

$$\phi = -\frac{1}{2} BF\, x^2 + BF\, x_{\max}\, x$$



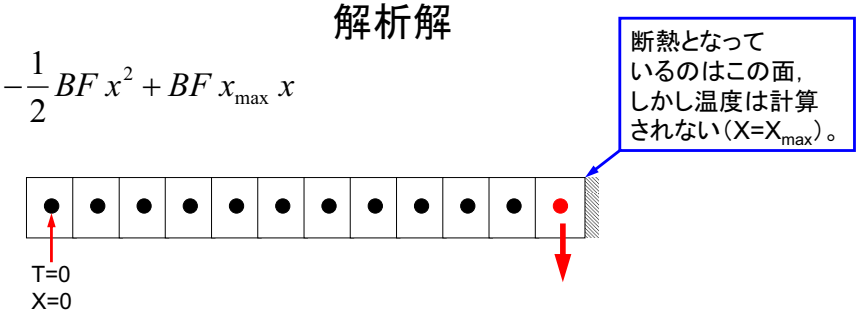
以下のような離散化（要素中心で従属変数を定義）をしているので注意が必要



一次元熱伝導方程式（2/3）

解析解

$$\phi = -\frac{1}{2} BF\, x^2 + BF\, x_{\max}\, x$$



Δx=1.d0，メッシュ数=50，とすると，X_{max}=49.5，
●の点のX座標は49.0となる。BF=1.0d0とすると●での温度は：

$$\phi = -\frac{1}{2} 49^2 + 49.5 \times 49 = -1200.5 + 9850.5 = 1225$$

一次元熱伝導方程式(3/3)

連立一次方程式

- 差分法による離散化

$$\left(\frac{d^2\phi}{dx^2}\right)_i \approx \frac{\left(\frac{d\phi}{dx}\right)_{i+1/2} - \left(\frac{d\phi}{dx}\right)_{i-1/2}}{\Delta x} = \frac{\frac{\phi_{i+1}-\phi_i}{\Delta x} - \frac{\phi_i-\phi_{i-1}}{\Delta x}}{\Delta x} = \frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}$$

- 各要素における線形方程式は以下のような形になる
  - これを共役勾配法(Conjugate Gradient法)で解く

$$\frac{d^2\phi}{dx^2} + BF = 0 \quad \rightarrow$$

$$\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2} + BF(i) = 0 \quad (1 \le i \le N)$$

$$\frac{1}{\Delta x^2}\phi_{i+1} - \frac{2}{\Delta x^2}\phi_i + \frac{1}{\Delta x^2}\phi_{i-1} + BF(i) = 0 \quad (1 \le i \le N)$$

$$A_L(i)\times\phi_{i-1} + A_D(i)\times\phi_i + A_R(i)\times\phi_{i+1} = BF(i) \quad (1 \le i \le N)$$

$$A_L(i) = \frac{1}{\Delta x^2},\ A_D(i) = -\frac{2}{\Delta x^2},\ A_R(i) = \frac{1}{\Delta x^2}$$

係数行列の格納形式

各要素における線形方程式

$$A_L(i)\times\phi_{i-1} + A_D(i)\times\phi_i + A_R(i)\times\phi_{i+1} = BF(i) \quad (1 \le i \le N)$$

$$A_L(i) = \frac{1}{\Delta x^2},\ A_D(i) = -\frac{2}{\Delta x^2},\ A_R(i) = \frac{1}{\Delta x^2}$$

	1	2	3	4	5	6	7	8
1	AD1	AR1						
2	AL2	AD2	AR2					
3		AL3	AD3	AR3				
4			AL4	AD4	AR4			
5				AL5	AD5	AR5		
6					AL6	AD6	AR6	
7						AL7	AD7	AR7
8							AL8	AD8

より一般的な形で格納すると・・・

$$DIAG(i)\times PHI(i) + \sum_{k=INDEX(i-1)+1}^{INDEX(i)} \left[AMAT(k)\times PHI(ITEM(k))\right] = RHS(i), \quad (i = 1,\ldots,N)$$

対角成分　　非対角成分

N=8の場合:連立一次方程式

自分とその周囲のみに非ゼロ成分:疎行列

	1	2	3	4	5	6	7	8			
1	$A_D(1)$	$A_R(1)$							×	ϕ(1)	BF(1)
2	$A_L(2)$	$A_D(2)$	$A_R(2)$							ϕ(2)	BF(2)
3		$A_L(3)$	$A_D(3)$	$A_R(3)$						ϕ(3)	BF(3)
4			$A_L(4)$	$A_D(4)$	$A_R(4)$					ϕ(4)	BF(4)
5				$A_L(5)$	$A_D(5)$	$A_R(5)$				ϕ(5)	BF(5)
6					$A_L(6)$	$A_D(6)$	$A_R(6)$			ϕ(6)	BF(6)
7						$A_L(7)$	$A_D(7)$	$A_R(7)$		ϕ(7)	BF(7)
8							$A_L(8)$	$A_D(8)$		ϕ(8)	BF(8)

$$A_D(1)\times\phi(1) + A_R(1)\times\phi(2) = BF(1)$$

$$A_L(2)\times\phi(1) + A_D(2)\times\phi(2) + A_R(2)\times\phi(3) = BF(2)$$

$$A_L(3)\times\phi(2) + A_D(3)\times\phi(3) + A_R(3)\times\phi(4) = BF(3)$$

$$A_L(4)\times\phi(3) + A_D(4)\times\phi(4) + A_R(4)\times\phi(5) = BF(4)$$

$$A_L(5)\times\phi(4) + A_D(5)\times\phi(5) + A_R(5)\times\phi(6) = BF(5)$$

$$A_L(6)\times\phi(5) + A_D(6)\times\phi(6) + A_R(6)\times\phi(7) = BF(6)$$

$$A_L(7)\times\phi(6) + A_D(7)\times\phi(7) + A_R(7)\times\phi(8) = BF(7)$$

$$A_L(8)\times\phi(7) + A_D(8)\times\phi(8) = BF(8)$$

$$DIAG(i)\times PHI(i) + \sum_{k=INDEX(i-1)+1}^{INDEX(i)} \left[AMAT(k)\times PHI(ITEM(k))\right] = RHS(i), \quad (i = 1,\ldots,N)$$

対角成分　　非対角成分

係数行列の格納形式(1/8)

非ゼロ成分のみを格納，疎行列向け方法

Compressed Row Storage (CRS)

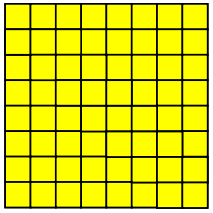
$$DIAG(i)\times PHI(i) + \sum_{k=INDEX(i-1)+1}^{INDEX(i)} \left[AMAT(k)\times PHI(ITEM(k))\right] = RHS(i), \quad (i = 1,\ldots,N)$$

DIAG(i)	対角成分(実数, i=1,N)
INDEX(i)	非対角成分数に関する一次元配列(整数, i=0,N)
ITEM(k)	非対角成分の要素(列)番号(整数, k=1, INDEX(N))
AMAT(k)	非対角成分(実数, k=1, INDEX(N))

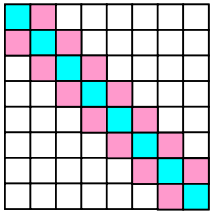
	1	2	3	4	5	6	7	8
1	$A_D(1)$	$A_R(1)$						
2	$A_L(2)$	$A_D(2)$	$A_R(2)$					
3		$A_L(3)$	$A_D(3)$	$A_R(3)$				
4			$A_L(4)$	$A_D(4)$	$A_R(4)$			
5				$A_L(5)$	$A_D(5)$	$A_R(5)$		
6					$A_L(6)$	$A_D(6)$	$A_R(6)$	
7						$A_L(7)$	$A_D(7)$	$A_R(7)$
8							$A_L(8)$	$A_D(8)$

密行列・疎行列

- 未知数：N個とすると
  - 密行列：N^2個の成分を記憶する必要あり
  - 疎行列：非ゼロの部分だけ記憶すれば，一次元問題の場合3N個の成分で済む



- $N=10^6$としたときの必要記憶容量
  - 密行列
    - $8 \times 10^6 \times 10^6 = 8 \times 10^{12} =$ 8TB
  - 疎行列（後掲のINDEX，ITEMも含む）
    - $8 \times 3 \times 10^6 + 4 \times 3 \times 10^6 = 36 \times 10^6 =$ 36MB



行列ベクトル積：密行列⇒とても簡単

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,N-1} & a_{1,N} \\ a_{21} & a_{22} & & a_{2,N-1} & a_{2,N} \\ \cdots & & \cdots & & \cdots \\ a_{N-1,1} & a_{N-1,2} & & a_{N-1,N-1} & a_{N-1,N} \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N-1} & a_{N,N} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{Bmatrix} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{Bmatrix}$$

```
{Y}= [A]{X}

do j= 1, N
  Y(j)= 0.d0
  do i= 1, N
    Y(j)= Y(j) + A(i,j)*X(i)
  enddo
enddo
```

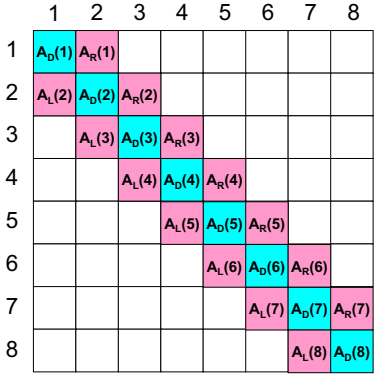

行列ベクトル積への適用

非ゼロ成分のみを格納，疎行列向け方法

DIAG(i)	対角成分（実数，i=1,N）
INDEX(i)	非対角成分数に関する一次元配列（整数，i=0,N）
ITEM(k)	非対角成分の要素（列）番号（整数，k=1, INDEX(N)）
AMAT(k)	非対角成分（実数，k=1, INDEX(N)）

$$DIAG(i) \times PHI(i) + \sum_{k=INDEX(i-1)+1}^{INDEX(i)} [AMAT(k) \times PHI(ITEM(k))] = RHS(i), \quad (i = 1, \ldots, N)$$

```
{Y}= [A]{X}

do i= 1, N
  Y(i)= D(i)*X(i)
  do k= INDEX(i-1)+1, INDEX(i)
    Y(i)= Y(i) + AMAT(k)*X(ITEM(k))
  enddo
enddo
```

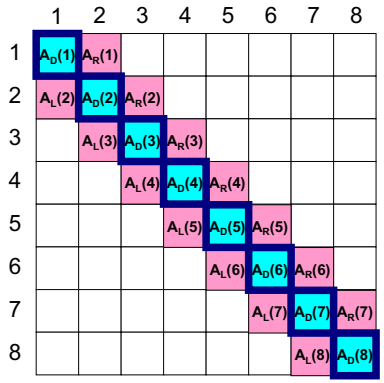


係数行列の格納形式（2/8）対角成分：DIAG

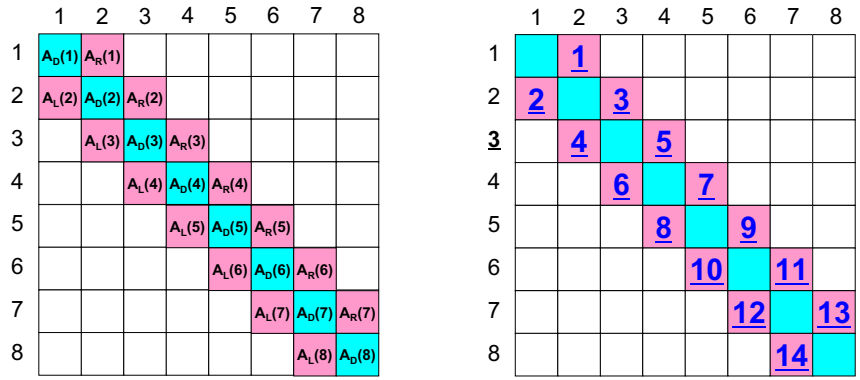
$$DIAG(i) \times PHI(i) + \sum_{k=INDEX(i-1)+1}^{INDEX(i)} [AMAT(k) \times PHI(ITEM(k))] = RHS(i), \quad (i = 1, \ldots, N)$$

DIAG(i)　対角成分（実数，i=1,N）

```
DIAG(1)= AD(1)
DIAG(2)= AD(2)
DIAG(3)= AD(3)
…
DIAG(8)= AD(8)
```

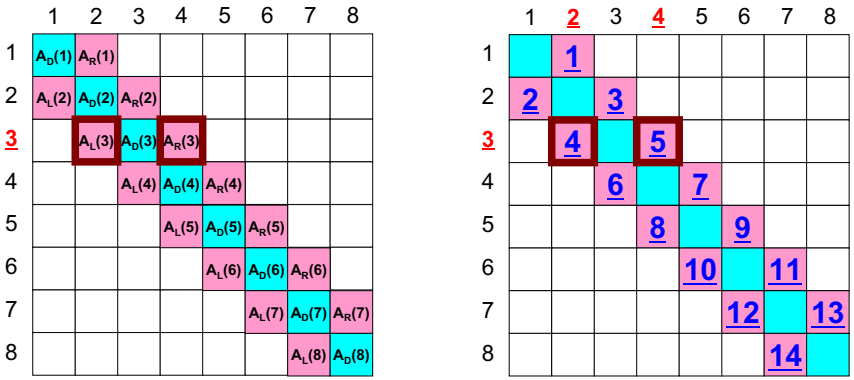


係数行列の格納形式(3/8)
INDEX, AMATの関係




```
INDEX(2)=3, INDEX(3)=5
ITEM(4)=2, AMAT(4)= AL(3)
ITEM(5)=4, AMAT(5)= AR(3)
```


係数行列の格納形式(3/8)
INDEX, AMATの関係

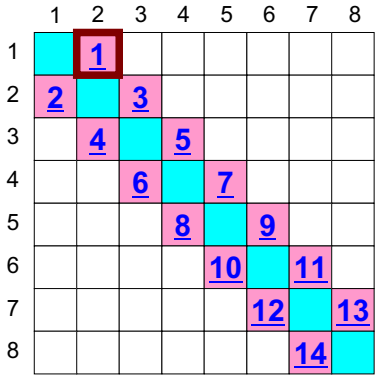



```
INDEX(2)=3, INDEX(3)=5
ITEM(4)=2, AMAT(4)= AL(3)
ITEM(5)=4, AMAT(5)= AR(3)
```


係数行列の格納形式(4/8)
非対角成分: i=1

$$DIAG(i)\times PHI(i)+\sum_{k=INDEX(i-1)+1}^{INDEX(i)}\left[AMAT(k)\times PHI(ITEM(k))\right]=RHS(i),\quad (i=1,\ldots,N)$$

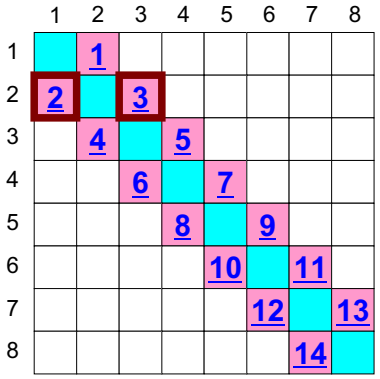
```
INDEX(i)  非対角成分数に関する一次元配列
          (整数, i=0,N)
ITEM(k)   非対角成分の要素(列)番号
          (整数, k=1, INDEX(N))
AMAT(k)   非対角成分
          (実数, k=1, INDEX(N))

INDEX(0)=0
INDEX(1)=1
          ITEM(1)=2, AMAT(1)= AR(1)
```



係数行列の格納形式(5/8)
非対角成分:i=2

$$DIAG(i)\times PHI(i)+\sum_{k=INDEX(i-1)+1}^{INDEX(i)}\left[AMAT(k)\times PHI(ITEM(k))\right]=RHS(i),\quad (i=1,\ldots,N)$$

```
INDEX(i)  非対角成分数に関する一次元配列
          (整数, i=0,N)
ITEM(k)   非対角成分の要素(列)番号
          (整数, k=1, INDEX(N))
AMAT(k)   非対角成分
          (実数, k=1, INDEX(N))

INDEX(1)=1
INDEX(2)=3
          ITEM(2)=1, AMAT(2)= AL(2)
          ITEM(3)=3, AMAT(3)= AR(2)
```

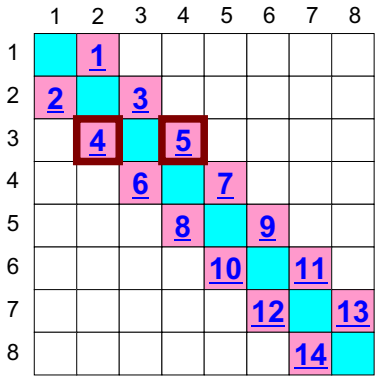


係数行列の格納形式(6/8)
非対角成分:i=3

$$DIAG(i)\times PHI(i)+\sum_{k=INDEX(i-1)+1}^{INDEX(i)}\left[AMAT(k)\times PHI(ITEM(k))\right]=RHS(i),\quad (i=1,\ldots,N)$$

INDEX(i) 非対角成分数に関する一次元配列
(整数, i=0,N)
ITEM(k) 非対角成分の要素(列)番号
(整数, k=1, INDEX(N))
AMAT(k) 非対角成分
(実数, k=1, INDEX(N))

INDEX(2)=3
INDEX(3)=5
ITEM(4)=2, AMAT(4)= A_L(3)
ITEM(5)=4, AMAT(5)= A_R(3)

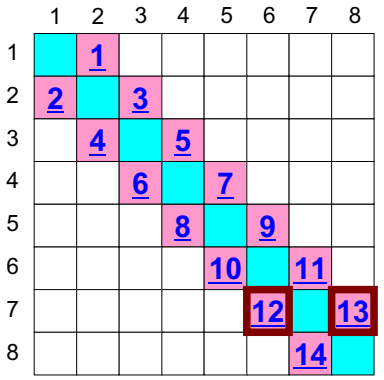


係数行列の格納形式(7/8)
非対角成分:i=7

$$DIAG(i)\times PHI(i)+\sum_{k=INDEX(i-1)+1}^{INDEX(i)}\left[AMAT(k)\times PHI(ITEM(k))\right]=RHS(i),\quad (i=1,\ldots,N)$$

INDEX(i) 非対角成分数に関する一次元配列
(整数, i=0,N)
ITEM(k) 非対角成分の要素(列)番号
(整数, k=1, INDEX(N))
AMAT(k) 非対角成分
(実数, k=1, INDEX(N))

INDEX(6)=11
INDEX(7)=13
ITEM(12)=6, AMAT(12)= A_L(7)
ITEM(13)=8, AMAT(13)= A_R(7)

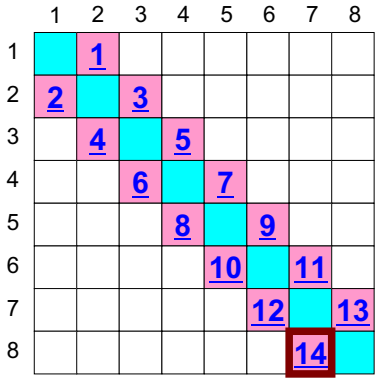


係数行列の格納形式(8/8)
非対角成分:i=8

$$DIAG(i)\times PHI(i)+\sum_{k=INDEX(i-1)+1}^{INDEX(i)}\left[AMAT(k)\times PHI(ITEM(k))\right]=RHS(i),\quad (i=1,\ldots,N)$$

INDEX(i) 非対角成分数に関する一次元配列
(整数, i=0,N)
ITEM(k) 非対角成分の要素(列)番号
(整数, k=1, INDEX(N))
AMAT(k) 非対角成分
(実数, k=1, INDEX(N))

INDEX(7)=13
INDEX(8)=14
ITEM(14)=7, AMAT(14)= A_L(8)



CG法による一次元熱伝導方程式
計算プログラム

- 前処理付き共役勾配法
  - Preconditioned Conjugate Gradient Method
  - 対称正定行列用
- 対角スケーリング(Diagonal Scaling)
  - [A]の対角成分のみからなる行列を前処理行列[M]とする。
  - 簡単(な問題にしか適用できない)。
  - 点ヤコビ(Point Jacobi)前処理とも呼ばれる。

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    solve [M]z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
    else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```

対角スケーリング，点ヤコビ前処理

- 前処理行列として，もとの行列の対角成分のみを取り出した行列を前処理行列 [M] とする。
  - 対角スケーリング，点ヤコビ（point-Jacobi）前処理

$$[M]=\begin{bmatrix} D_1 & 0 & \dots & 0 & 0 \\ 0 & D_2 & & 0 & 0 \\ \dots & & \dots & & \dots \\ 0 & 0 & & D_{N-1} & 0 \\ 0 & 0 & \dots & 0 & D_N \end{bmatrix}$$

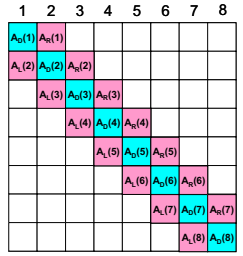
- **solve [M]z$^{(i-1)}$= r$^{(i-1)}$**という場合に逆行列を簡単に求めることができる。

CG法による一次元熱伝導方程式 プログラム概要（1/7）

```
!C
!C   1D Poisson Equation Solver by
!C   CG (Conjugate Gradient) Method
!C
!C   d/dx(dPHI/dx) + BF = 0
!C   PHI=0@x=0
!C
      program CG_poi
      implicit REAL*8 (A-H,O-Z)

      integer :: N, ITERmax
      integer :: R, Z, P, Q, DD

      real(kind=8) :: dx, OMEGA, RESID, dPHI, dPHImax, BF, EPS
      real(kind=8), dimension(:), allocatable :: PHI, RHS
      real(kind=8), dimension(:  ), allocatable :: DIAG, AMAT
      real(kind=8), dimension(:,:), allocatable :: W

      integer, dimension(:  ), allocatable :: INDEX, ITEM

!C
!C-- INIT.
      open  (11, file='input.dat', status='unknown')
       read (11,*) N
       read (11,*) dX, BF
       read (11,*) ITERmax
       read (11,*) EPS
      close (11)
```


CG法による一次元熱伝導方程式 プログラム概要（2/7）

```
      allocate (PHI(N), DIAG(N), AMAT(2*N-2), RHS(N))
      allocate (INDEX(0:N), ITEM(2*N-2), W(N,4))
       PHI= 0.d0
      AMAT=  1.d0/dX
      DIAG= -2.d0/dX
       RHS=  -BF * dX
```



$$\frac{d^2\phi}{dx^2}+BF=0,\quad \phi=0@x=0,\ \frac{d\phi}{dx}=0@x=x_{\max}$$

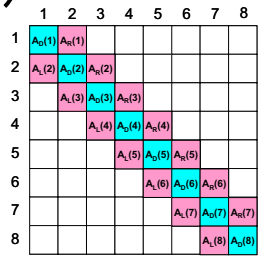
$$\left(\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}\right)\times V+BF\times V=0$$

非対角成分の数は「2」
ただし，i=1，i=Nのときは 「1」

したがって，非対角成分の総数は「2*N-2」

CG法による一次元熱伝導方程式 プログラム概要（2/7）

```
      allocate (PHI(N), DIAG(N), AMAT(2*N-2), RHS(N))
      allocate (INDEX(0:N), ITEM(2*N-2), W(N,4))
       PHI= 0.d0
      AMAT=  1.d0/dX
      DIAG= -2.d0/dX
       RHS=  -BF * dX
```



$$\left(\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}\right)\times V+BF\times V=0$$

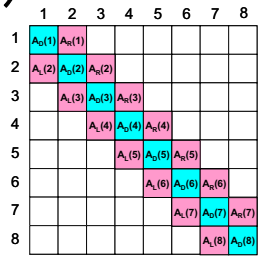
$$\left(\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}\right)\times \Delta x+BF\times \Delta x=0$$

非対角成分の数は「2」
ただし，i=1，i=Nのときは 「1」

したがって，非対角成分の総数は「2*N-2」

CG法による一次元熱伝導方程式 プログラム概要(2/7)

```
      allocate (PHI(N), DIAG(N), AMAT(2*N-2), RHS(N))
      allocate (INDEX(0:N), ITEM(2*N-2), W(N,4))
       PHI= 0.d0
      AMAT=  1.d0/dX
      DIAG= -2.d0/dX
       RHS=  -BF * dX
```



$$\left(\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x^2}\right)\times\Delta x+BF\times\Delta x=0$$

$$\frac{\phi_{i+1}-2\phi_i+\phi_{i-1}}{\Delta x}=\left(\frac{1}{\Delta x}\right)\phi_{i-1}-\left(\frac{2}{\Delta x}\right)\phi_i+\left(\frac{1}{\Delta x}\right)\phi_{i+1}=-BF\times\Delta x$$

$A_L(i)$ $A_D(i)$ $A_R(i)$ RHS(i)

非対角成分の数は「2」
ただし，i=1，i=Nのときは 「1」

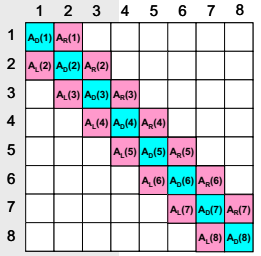
したがって，非対角成分の総数は「2*N-2」

CG法による一次元熱伝導方程式 プログラム概要(3/7)

```
      INDEX= 2
!C
!C-- CONNECTIVITY
       INDEX(0)= 0
       INDEX(1)= 1
       INDEX(N)= 1

       do i= 1, N
         INDEX(i)= INDEX(i) + INDEX(i-1)
       enddo

       do i= 1, N
         jS= INDEX(i-1)
         if (i.eq.1) then
           ITEM(jS+1)= i+1
           AMAT(jS+1)= 0.d0
           DIAG(i   )= 1.d0
            RHS(i   )= 0.d0
         else if
     &     (i.eq.N) then
           ITEM(jS+1)= i-1
           DIAG(i   )= -1.d0/dX
         else
           ITEM(jS+1)= i-1
           ITEM(jS+2)= i+1
           if (i-1.eq.1) then
             AMAT(jS+1)= 0.d0
           endif
         endif
       enddo
```

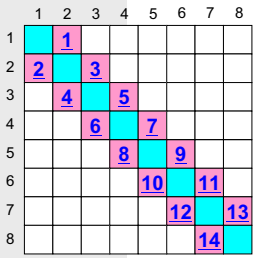
非対角成分の数は「2」
ただし，i=1，i=Nのときは 「1」

このルールに従って「INDEX」生成



CG法による一次元熱伝導方程式 プログラム概要(3/7)

```
      INDEX= 2
!C
!C-- CONNECTIVITY
       INDEX(0)= 0
       INDEX(1)= 1
       INDEX(N)= 1

       do i= 1, N
         INDEX(i)= INDEX(i) + INDEX(i-1)
       enddo

       do i= 1, N
         jS= INDEX(i-1)
         if (i.eq.1) then
           ITEM(jS+1)= i+1
           AMAT(jS+1)= 0.d0
           DIAG(i   )= 1.d0
            RHS(i   )= 0.d0
         else if
     &     (i.eq.N) then
           ITEM(jS+1)= i-1
           DIAG(i   )= -1.d0/dX
         else
           ITEM(jS+1)= i-1
           ITEM(jS+2)= i+1
           if (i-1.eq.1) then
             AMAT(jS+1)= 0.d0
           endif
         endif
       enddo
```



```
INDEX(0)=0
INDEX(1)=1
                ITEM(1)=2, AMAT(1)= AR1
INDEX(2)=3
                ITEM(2)=1, AMAT(2)= AL2
                ITEM(3)=3, AMAT(3)= AR3
INDEX(3)=5
                ITEM(4)=2, AMAT(4)= AL3
                ITEM(5)=4, AMAT(5)= AR3
…
INDEX(N-1)= 2*N-3
                ITEM(2*N-4)= N-2, AMAT(2*N-4)= AL(N-1)
                ITEM(2*N-3)= N  , AMAT(2*N-3)= AR(N-1)
INDEX(N)  = 2*N-2
                ITEM(2*N-2)= N-1, AMAT(N*N-2)= AL(N)
```

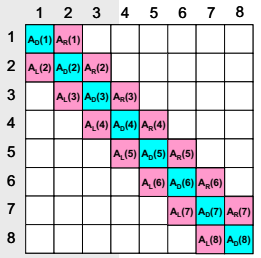

CG法による一次元熱伝導方程式 プログラム概要(3/7)

```
      INDEX= 2
!C
!C-- CONNECTIVITY
       INDEX(0)= 0
       INDEX(1)= 1
       INDEX(N)= 1

       do i= 1, N
         INDEX(i)= INDEX(i) + INDEX(i-1)
       enddo

       do i= 1, N
         jS= INDEX(i-1)
         if (i.eq.1) then
           ITEM(jS+1)= i+1
           AMAT(jS+1)= 0.d0
           DIAG(i   )= 1.d0
            RHS(i   )= 0.d0
         else if
     &     (i.eq.N) then
           ITEM(jS+1)= i-1
           DIAG(i   )= -1.d0/dX
         else
           ITEM(jS+1)= i-1
           ITEM(jS+2)= i+1
           if (i-1.eq.1) then
             AMAT(jS+1)= 0.d0
           endif
         endif
       enddo
```

境界条件(i=1):後述

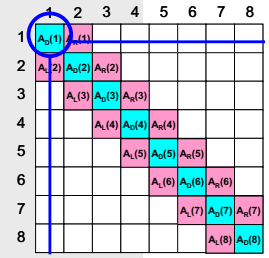
境界条件(i=N)

それ以外



CG法による一次元熱伝導方程式 プログラム概要(3/7)

```
      INDEX= 2
!C
!C-- CONNECTIVITY
       INDEX(0)= 0
       INDEX(1)= 1
       INDEX(N)= 1

       do i= 1, N
         INDEX(i)= INDEX(i) + INDEX(i-1)
       enddo

       do i= 1, N
         jS= INDEX(i-1)
         if (i.eq.1) then
           ITEM(jS+1)= i+1
           AMAT(jS+1)= 0.d0
           DIAG(i   )= 1.d0
            RHS(i   )= 0.d0
         else if                                    &
     &     (i.eq.N) then
           ITEM(jS+1)= i-1
           DIAG(i   )= -1.d0/dX
         else
           ITEM(jS+1)= i-1
           ITEM(jS+2)= i+1
           if (i-1.eq.1) then
             AMAT(jS+1)= 0.d0
           endif
         endif
       enddo
```



境界条件(i=1):後述

境界条件(i=N)

それ以外

固定境界条件が指定されている点を非対角成分として持っている場合非対角成分をゼロクリアする。

境界条件の処理:i=N

```
i = N
jS= INDEX(i-1)

ITEM(jS+1)= i-1
AMAT(jS+1)= +1.d0/dx  デフォルト値
DIAG(i   )= -1.d0/dX
```

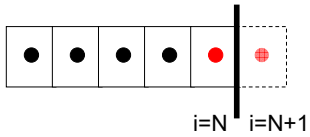
$$\frac{d\phi}{dx}=0@x=x_{\max}\Rightarrow\frac{\phi_{N+1}-\phi_N}{\Delta x}=0$$

$$\left(\frac{\phi_{N+1}-2\phi_N+\phi_{N-1}}{\Delta x^2}\right)\times\Delta x+BF\times\Delta x=0$$

$$\Rightarrow\left(\frac{-\phi_N+\phi_{N-1}}{\Delta x^2}\right)\times\Delta x+BF\times\Delta x=0$$

$$\Rightarrow(0)\phi_{N+1}+\left(\frac{-1}{\Delta x}\right)\phi_N+\left(\frac{1}{\Delta x}\right)\phi_{N-1}=-BF\times\Delta x$$

DIAG(i)　AMAT(jS+1)　RHS



境界面で断熱条件が成立するためには，$\phi_{N+1}=\phi_N$を満たすような仮想的な要素があると都合が良い

境界条件の処理:i=1 (1/4)

```
ITEM(jS+1)= i+1
AMAT(jS+1)= 0.d0
DIAG(i   )= 1.d0
RHS (i   )= 0.d0
```

$$\phi=0@x=0\Rightarrow\phi_1=0$$

$$\Rightarrow(0)\phi_2+(1)\phi_1+(0)\phi_0=0$$

DIAG　　RHS

	1	2	3	4	5	6	7	8
1	1	0						
2	a	-2a	a					
3		a	-2a	a				
4			a	-2a	a			
5				a	-2a	a		
6					a	-2a	a	
7						a	-2a	a
8							a	-a

$$a=\frac{1}{\Delta x}$$

- これでは係数行列が対称とならないため共役勾配法が使えない。

境界条件の処理:i=1(2/4)

```
ITEM(jS+1)= i+1
AMAT(jS+1)= 0.d0
DIAG(i   )= 1.d0
RHS (i   )= 0.d0
```

$$\phi=0@x=0\Rightarrow\phi_1=0$$

$$\Rightarrow(0)\phi_2+(1)\phi_1+(0)\phi_0=0$$

DIAG　　RHS

	1	2	3	4	5	6	7	8
1	1	0						
2	a	-2a	a					
3		a	-2a	a				
4			a	-2a	a			
5				a	-2a	a		
6					a	-2a	a	
7						a	-2a	a
8							a	-a

$$a=\frac{1}{\Delta x}$$

- これでは係数行列が対称とならないため共役勾配法が使えない。

境界条件の処理：i=1（3/4）

- i=2における方程式

$$(a)\phi_3+(-2a)\phi_2+(a)\phi_1=RHS_2$$

- ここでϕ_1は既知の値 $\overline{\phi_1}$

	1	2	3	4	5	6	7	8
1	1	0						
2	a	-2a	a					
3		a	-2a	a				
4			a	-2a	a			
5				a	-2a	a		
6					a	-2a	a	
7						a	-2a	a
8							a	-a

$$a=\frac{1}{\Delta x}$$

境界条件の処理：i=1（4/4）

- i=2における方程式

$$(a)\phi_3+(-2a)\phi_2+(a)\phi_1=RHS_2$$

- ここでϕ_1は既知の値 $\overline{\phi_1}$

$$(a)\phi_3+(-2a)\phi_2+(0)\phi_1=RHS_2-(a)\overline{\phi_1}$$

- このケースの場合は，$\overline{\phi_1}=0$であるため，右辺（RHS）の修正は不要である。

	1	2	3	4	5	6	7	8
1	1	0						
2	0	-2a	a					
3		a	-2a	a				
4			a	-2a	a			
5				a	-2a	a		
6					a	-2a	a	
7						a	-2a	a
8							a	-a

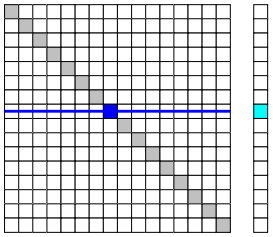
$$a=\frac{1}{\Delta x}$$

固定境界条件の処理：消去

i=kkの点でPHIxの値に固定されている場合

```
      do i= 1, N
        if (i.eq.kk) then
          DIAG(i)= 1.d0
          RHS (i)= PHIx
          do j= INDEX(i-1)+1, INDEX(i)
            AMAT(j)= 0.d0
          enddo
        endif
      endif

      do i= 1, N
        if (i.ne.kk) then
          do j= INDEX(i-1)+1, INDEX(i)
            jj= ITEM(j)
            if (jj.eq.kk) then
              RHS (i)= RHS(i) - AMAT(j)*PHIx
              AMAT(j)= 0.d0
            endif
          enddo
        enddo
      enddo
```

ゼロクリア

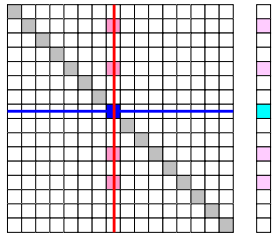


固定境界条件の処理：消去

i=kkの点でPHIxの値に固定されている場合

```
      do i= 1, N
        if (i.eq.kk) then
          DIAG(i)= 1.d0
          RHS (i)= PHIx
          do j= INDEX(i-1)+1, INDEX(i)
            AMAT(j)= 0.d0
          enddo
        endif
      endif

      do i= 1, N
        if (i.ne.kk) then
          do j= INDEX(i-1)+1, INDEX(i)
            jj= ITEM(j)
            if (jj.eq.kk) then
              RHS (i)= RHS(i) - AMAT(j)*PHIx
              AMAT(j)= 0.d0
            endif
          enddo
        enddo
      enddo
```

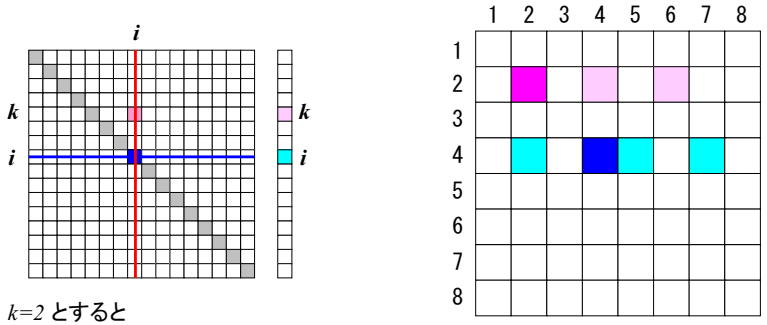
ゼロクリア



今回の問題ではPHIX(jj)に相当するものが0だったのでこの右辺の処理は不要であった。

固定境界条件の処理：具体例

点「i」で固定境界条件を適用しているものとする $\phi=\tilde{\phi}_i$

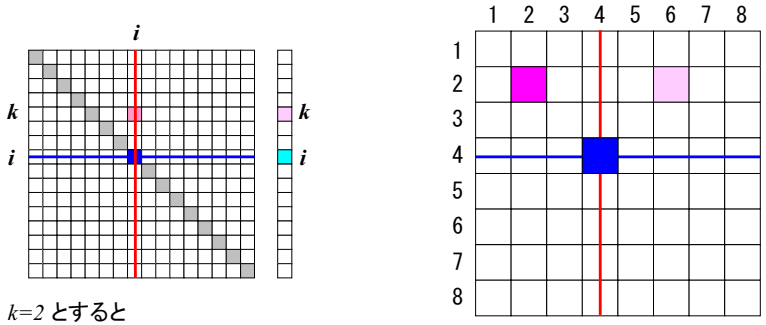


$i=4,\ k=2$ とすると

$$DIAG_2\phi_2 + AMAT_{24}\phi_4 + AMAT_{26}\phi_6 = RHS_2$$
$$DIAG_4\phi_4 + AMAT_{42}\phi_2 + AMAT_{45}\phi_5 + AMAT_{47}\phi_7 = RHS_4$$

固定境界条件の処理：具体例

点「i」で固定境界条件を適用しているものとする $\phi=\tilde{\phi}_i$



$i=4,\ k=2$ とすると

$$DIAG_2\phi_2 + AMAT_{24}\phi_4 + AMAT_{26}\phi_6 = RHS_2$$
$$DIAG_4\phi_4 + AMAT_{42}\phi_2 + AMAT_{45}\phi_5 + AMAT_{47}\phi_7 = RHS_4$$

$$\Rightarrow \quad DIAG_2\phi_2 + AMA_{26}\phi_6 = RHS_2 - AMAT_{24}\tilde{\phi}_4$$
$$\phi_4 = \tilde{\phi}_4$$

CG法による一次元熱伝導方程式 プログラム概要(4/7)

```
!C
!C +---------------+
!C | CG iterations |
!C +---------------+
!C===
      R = 1
      Z = 2
      Q = 2
      P = 3
      DD= 4

      do i= 1, N
        W(i,DD)= 1.0D0 / DIAG(i)
      enddo
!C
!C-- {r0}= {b} - [A]{xini} |

      do i= 1, N
        W(i,R) = DIAG(i)*PHI(i)
        do j= INDEX(i-1)+1, INDEX(i)
          W(i,R) = W(i,R) + AMAT(j)*PHI(ITEM(j))
        enddo
      enddo

      BNRM2= 0.0D0
      do i= 1, N
        BNRM2 = BNRM2  + RHS(i)  **2
        W(i,R)= RHS(i) - W(i,R)
      enddo
!C***********************************************************************
```

対角成分の逆数（前処理用）
その都度，除算をすると効率が悪いため，予め配列に格納

CG法による一次元熱伝導方程式 プログラム概要(4/7)

```
!C
!C +---------------+
!C | CG iterations |
!C +---------------+
!C===
      R = 1
      Z = 2
      Q = 2
      P = 3
      DD= 4

      do i= 1, N
        W(i,DD)= 1.0D0 / DIAG(i)
      enddo
```

W(i,1)= W(i,R) ⇒ {r}
W(i,2)= W(i,Z) ⇒ {z}
W(i,2)= W(i,Q) ⇒ {q}
W(i,3)= W(i,P) ⇒ {p}

W(i,4)= W(i,DD) ⇒ 1/DIAG

Compute $r^{(0)}= b-[A]x^{(0)}$
for i= 1, 2, …
  solve $[M]z^{(i-1)}= r^{(i-1)}$
  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
  if i=1
    $p^{(1)}= z^{(0)}$
  else
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i)}$
  endif
  $q^{(i)}= [A]p^{(i)}$
  $\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence |r|
end

CG法による一次元熱伝導方程式 プログラム概要(4/7)

```
!C
!C +---------------+
!C | CG iterations |
!C +---------------+
!C===
      R = 1
      Z = 2
      Q = 2
      P = 3
      DD= 4

      do i= 1, N
        W(i,DD)= 1.0D0 / DIAG(i)
      enddo
!C
!C-- {r0}= {b} - [A]{xini} |

      do i= 1, N
        W(i,R) = DIAG(i)*PHI(i)
        do j= INDEX(i-1)+1, INDEX(i)
          W(i,R) = W(i,R) + AMAT(j)*PHI(ITEM(j))
        enddo
      enddo

      BNRM2= 0.0D0
      do i= 1, N
        BNRM2 = BNRM2  + RHS(i)   **2
        W(i,R)= RHS(i) - W(i,R)
      enddo
```

BNRM2： 収束判定用

Compute $r^{(0)}= b-[A]x^{(0)}$
for i= 1, 2, …
  solve $[M]z^{(i-1)}= r^{(i-1)}$
  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
  if i=1
    $p^{(1)}= z^{(0)}$
  else
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i)}$
  endif
  $q^{(i)}= [A]p^{(i)}$
  $\alpha_i\ = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence |r|
end

CG法による一次元熱伝導方程式 プログラム概要(5/7)

```
      do iter= 1, ITERmax
!C
!C-- {z}= [Minv]{r}

      do i= 1, N
        W(i,Z)= W(i,DD) * W(i,R)
      enddo

!C
!C-- RHO= {r}{z}

      RHO= 0.d0
      do i= 1, N
        RHO= RHO + W(i,R)*W(i,Z)
      enddo

!C
!C-- {p} = {z} if      ITER=1
!C   BETA= RHO / RHO1  otherwise

      if ( iter.eq.1 ) then
        do i= 1, N
          W(i,P)= W(i,Z)
        enddo
      else
        BETA= RHO / RHO1
        do i= 1, N
          W(i,P)= W(i,Z) + BETA*W(i,P)
        enddo
      endif
```

Compute $r^{(0)}= b-[A]x^{(0)}$
for i= 1, 2, …
  solve $[M]z^{(i-1)}= r^{(i-1)}$
  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
  if i=1
    $p^{(1)}= z^{(0)}$
  else
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i)}$
  endif
  $q^{(i)}= [A]p^{(i)}$
  $\alpha_i\ = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence |r|
end

CG法による一次元熱伝導方程式 プログラム概要(6/7)

```
!C
!C-- {q}= [A]{p}

      do i= 1, N
        W(i,Q) = DIAG(i)*W(i,P)
        do j= INDEX(i-1)+1, INDEX(i)
          W(i,Q) = W(i,Q) + AMAT(j)*W(ITEM(j),P)
        enddo
      enddo

!C
!C-- ALPHA= RHO / {p}{q}

      C1= 0.d0
      do i= 1, N
        C1= C1 + W(i,P)*W(i,Q)
      enddo
      ALPHA= RHO / C1

!C
!C-- {x}= {x} + ALPHA*{p}
!C   {r}= {r} - ALPHA*{q}

      do i= 1, N
        PHI(i)   = PHI(i) + ALPHA * W(i,P)
        W  (i,R)= W(i,R) - ALPHA * W(i,Q)
      enddo
```

Compute $r^{(0)}= b-[A]x^{(0)}$
for i= 1, 2, …
  solve $[M]z^{(i-1)}= r^{(i-1)}$
  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
  if i=1
    $p^{(1)}= z^{(0)}$
  else
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i)}$
  endif
  $q^{(i)}= [A]p^{(i)}$
  $\alpha_i\ = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence |r|
end

CG法による一次元熱伝導方程式 プログラム概要(7/7)

```
      DNRM2 = 0.0
      do i= 1, N
        DNRM2= DNRM2 + W(i,R)**2
      enddo

        RESID= dsqrt(DNRM2/BNRM2)

        write (*, 1000) iter, RESID
 1000   format (i5, 1pe16.6)

        if ( RESID.le.EPS) goto 900
        RHO1 = RHO

      enddo
!C***************************************************

      IER = 1

  900 continue
!C===

!C
!C-- OUTPUT
      do i= 1, N
        write (*,'(i8, 1pe16.6)') i, PHI(i)
      enddo

      end program CG_poi
```

Compute $r^{(0)}= b-[A]x^{(0)}$
for i= 1, 2, …
  solve $[M]z^{(i-1)}= r^{(i-1)}$
  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
  if i=1
    $p^{(1)}= z^{(0)}$
  else
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i)}$
  endif
  $q^{(i)}= [A]p^{(i)}$
  $\alpha_i\ = \rho_{i-1}/p^{(i)}q^{(i)}$
  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
  check convergence |r|
end

CG法による一次元熱伝導方程式 プログラム概要(7/7)

```
      DNRM2 = 0.0
      do i= 1, N
        DNRM2= DNRM2 + W(i,R)**2
      enddo

        RESID= dsqrt(DNRM2/BNRM2)

 1000                                , RESID

                                    oto 900

      en
!C****** RESID= |r|/|b| ************************

      IER = 1

  900 continue
!C===

!C
!C-- OUTPUT
      do i= 1, N
        write (*,'(i8, 1pe16.6)') i, PHI(i)
      enddo

      end program CG_poi
```

BNRM2： 無次元化に使用

r= b-[A]x
DNRM2=|r|²
BNRM2=|b|²

```
Compute r(0)= b-[A]x(0)
for i= 1, 2, …
    solve [M]z(i-1)= r(i-1)
    ρi-1= r(i-1) z(i-1)
    if i=1
      p(1)= z(0)
    else
      βi-1= ρi-1/ρi-2
      p(i)= z(i-1) + βi-1 p(i)
    endif
    q(i)= [A]p(i)
    αi  = ρi-1/p(i)q(i)
    x(i)= x(i-1) + αip(i)
    r(i)= r(i-1) - αiq(i)
    check convergence |r|
end
```


反復法の収束判定 「解を得られた」という判定

解の推定値 $\mathbf{x}^{(k)}$

$$\mathbf{x}^{(i)} = \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix}$$

- 適切な条件のもとで，$i \Rightarrow i+1$のプロセスを繰り返すことによって，$\mathbf{x}^{(i)}$は正しい解に収束していく。
- $[\mathbf{A}]\{\mathbf{x}\}=\{\mathbf{b}\}$という方程式を解いているので，$\|\mathbf{b}-\mathbf{A}\mathbf{x}\| \sim 0$となれば収束したとみなすことができる。
- 通常は$\|b\|$で無次元化した「残差ノルム」が予め設定した値 ε より小さくなった場合に収束したとみなす。ε の値は要求される精度によって異なる。

残差ノルム

$$\frac{\left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(i)}\right\|}{\|\mathbf{b}\|} < \varepsilon \qquad \left\|\mathbf{b}-\mathbf{A}\mathbf{x}^{(i)}\right\| = \sqrt{\sum_{j=1}^{n}\left|b_j - \sum_{k=1}^{n} a_{jk} x_k^{(i)}\right|^2}, \quad \|\mathbf{b}\| = \sqrt{\sum_{j=1}^{n}\left|b_j\right|^2}$$