

# 数値シミュレーションの基礎 (4)

青木 孝

神奈川大学 理学部 情報科学科

Introduction to Numerical Simulation(4)

Takashi Aoki

Department of Information Science, Kanagawa University

**Abstract.** This paper notes that thermal equation can't solve by Crank-Nicolson, but can solve by CV method in Non-equal mesh step.

## 1 はじめに

熱方程式 ( $u = u(x, y)$ : 板の温度分布、 $f(x, y)$ : 熱源分布):

$$(1.1) \quad \text{div}(-\kappa \nabla u) = f$$

の拡散係数である熱伝導率  $\kappa(x, y)$  が、場所  $(x, y)$  に依存する場合に、熱分布  $u(x, y)$  を、CV(Control Volume) 法あるいは、いわゆる台形法 (クランク-ニコルソン) で解くとする。この時、空間の差分を、あみ目に段差をつけて不均等あみ目として離散化して解く場合には、たとえ拡散係数  $\kappa = \text{一定の均一}$  の場でも、CV 法では正しく解けるが、台形法では解けないことすらあることを示す (均等あみ目にすれば、両解法の解は一致する。 )。

式 (1.1) は、元は領域表面  $\Gamma$  からの熱流出量は、領域内  $\Omega$  での熱発生量と等しいという物理量 (単位面積、単位時間当たりの流量ベクトル:  $-\kappa \nabla u$ ) の保存則 :

$$(1.2) \quad \int_{\Gamma} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{\Omega} \text{div}(-\kappa \nabla u) \, dv = \int_{\Omega} f \, dv$$

の積分をガウスの発散公式を使ってはずし、微分方程式にしたものになっている ( $\kappa$  と  $\nabla u$  はくっついている。 )。  $\kappa$  が場に依存する場合には、離して  $-\kappa \nabla^2 u = f$  としてはいけない。

CV 法は、積分形式保存則 :

$$(1.3) \quad \int_{\Gamma} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{\Omega} f \, dv$$

を満足するように差分化する方法である。村田は、保存則のようなモデルの元になるような考え方を忘れてしまうような過ちを避けるために、CV 法を奨めている (文献 [1])。シミュレーションでは、収束判定値にもよるが、0.001% 以下で、保存則を満足させることはたやすい。

## 2 場の設定と CV 法

熱分布を調べる場を、Fig.1 のように、横  $x$  方向 11cm × 縦  $y$  方向 10cm 四方にとり、上の CD 境界を開放 (flux が 0 のノイマン境界:  $\kappa \nabla u = 0$ ) とし、他の 3 方 (DA, AB, BC 端) は固定境界とする。この時、簡単のため  $u=0$  [ ] とする。

Fig.1では、空間を 1cm 刻みで、 $x, y$  方向均等にメッシュに分け、差分法 (離散化) してある。この図のように、点 1 から通し番号をふれば、全部で点 100 までの差分点 (節点) となる。全節点数  $N$  は、 $N=M(=m)(y \text{ 方向}) \times M1(x \text{ 方向}) = 10 \times 10 = 100$  となる。 $i$  番目の格子点 (節点) と、隣あう十字方向の 4 点はそれぞれ、 $i+1, i-1, i+m, i-m$  点となっている。実務上は、大事な所 (熱源の境界など) はメッシュを精しくして解くので、不均等あみ目となる。

この Fig.1 の各節 (格子) 点  $i$  の  $i$  点回りに、Fig.2 のように、小領域 PQRS (Control Volume という) を作り、その周囲を式 (1.3) の領域表面  $\Gamma$  に、内側を領域内  $\Omega$  に見立てて、式 (1.3) にしたがって、CV 法による離散式を作る。格子点には、Fig.1 のように通し番号を付ける。全格子点数が離散方程式の元数となる (詳しくは、数値シミュレーションの基礎 (1) を参照のこと)。

これから、まず、CV 法による離散方程式の作り方を説明し、その次に、台形法による離散方程式の作り方を説明する。その後、離散化のところでは設定する不均等あみ目の場で、この両者の離散方程式を使い式 (1.1) を解いた解を比較評価していく。

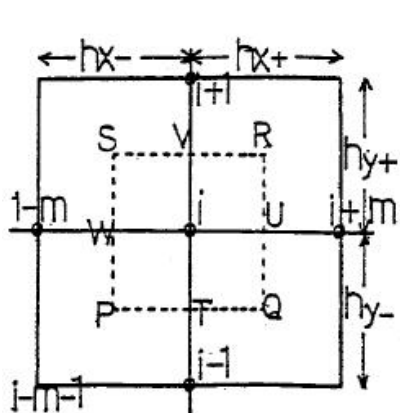


Fig.2 Mesh for CV method

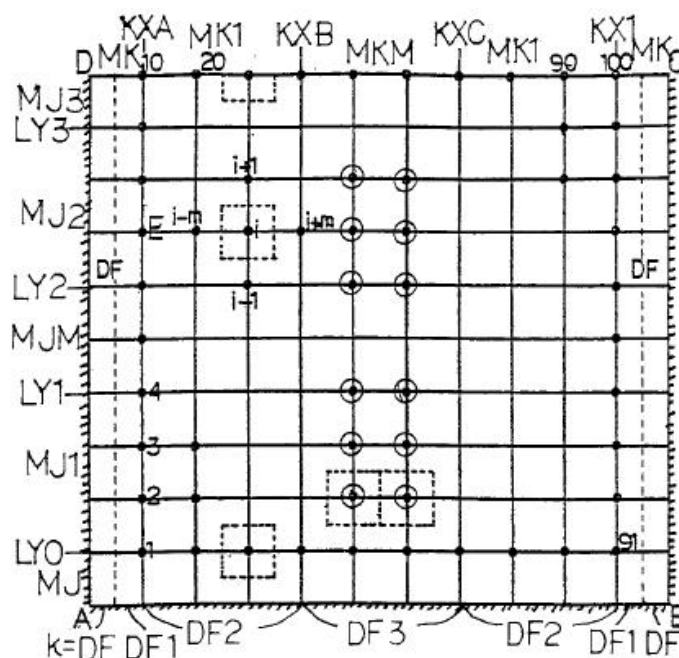


Fig.1 Simulation Field and Mesh step

式 (1.3) の左辺の線積分は、Fig.1 の各節点  $i$  の  $i$  点回りの小領域 PQRS の周囲を  $\Gamma$  と見立て (Fig.2)、

$$\int_{\Gamma} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{PQ} + \int_{QR} + \int_{RS} + \int_{SP}$$

となる。その第 1 項は、Fig.2 の左下面の熱伝導率  $\kappa$  を  $d_P$ 、右下面の  $\kappa$  を  $d_Q$  とすれば、PQ 辺の外向き法線ベクトルが  $(0, -1)^T$  であることに留意して、

$$\int_{PQ} \doteq -d_P \left( \frac{u_i - u_{i-1}}{hy_-} \right) (-1) \frac{hx_-}{2} - d_Q \left( \frac{u_i - u_{i-1}}{hy_-} \right) (-1) \frac{hx_+}{2}$$

となる。ここで、辺 PQ, QR, RS, SP に立てた外向き単位法線ベクトルが、 $(n_x, n_y)^T = (0, -1)^T, (1, 0)^T, (0, 1)^T, (-1, 0)^T$  であることに留意し、また、

$$\nabla u \cdot \mathbf{n} = \left( \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right)^T \cdot (n_x, n_y)^T$$

であることを使い、T 点での  $\frac{\partial u}{\partial y}$  は、 $\frac{\partial u}{\partial y} = \frac{u_i - u_{i-1}}{hy_-}$  とした。同様に、W 点での  $\frac{\partial u}{\partial x}$  は、 $\frac{\partial u}{\partial x} = \frac{u_i - u_{i-m}}{hx_-}$  などと、各点での  $u_i$  を使って差分化する。

線積分の第 1 項と同様に、全項を離散化して足し合わせ、 $u_{i-m}, u_{i-1}, u_i, u_{i+1}, u_{i+m}$  項毎に整理すれば、式 (1.3) を離散化した第  $i$  番方程式は、

$$(2.1) \quad a_{i,i-m}u_{i-m} + a_{i,i-1}u_{i-1} + a_{i,i}u_i + a_{i,i+1}u_{i+1} + a_{i,i+m}u_{i+m} = f_i$$

と書ける。式 (2.1) のそれぞれの係数  $(a_{i,j})$  は、Fig.2 の 4 分面で、熱伝導度  $\kappa(x, y)$  をそれぞれ、 $d_P, d_S, d_R, d_Q$  ( $\frac{\text{cal}}{\text{sec}\cdot\text{cm}}$ ) として、

$$\begin{aligned} a_{i,i} &= \frac{1}{2} \left[ d_P \left( \frac{hx_-}{hy_-} + \frac{hy_-}{hx_-} \right) + d_Q \left( \frac{hx_+}{hy_-} + \frac{hy_-}{hx_+} \right) + d_R \left( \frac{hx_+}{hy_+} + \frac{hy_+}{hx_+} \right) + d_S \left( \frac{hx_-}{hy_+} + \frac{hy_+}{hx_-} \right) \right] \\ a_{i,i-m} &= -\frac{1}{2} \left[ d_S \frac{hy_+}{hx_-} + d_P \frac{hy_-}{hx_-} \right] \\ a_{i,i-1} &= -\frac{1}{2} \left[ d_P \frac{hx_-}{hy_-} + d_Q \frac{hx_+}{hy_-} \right] \\ a_{i,i+1} &= -\frac{1}{2} \left[ d_R \frac{hx_+}{hy_+} + d_S \frac{hx_-}{hy_+} \right] \\ a_{i,i+m} &= -\frac{1}{2} \left[ d_Q \frac{hy_-}{hx_+} + d_R \frac{hy_+}{hx_+} \right] \end{aligned} \quad (2.2)$$

となる。この式 (2.1) は、 $n(=N)$  行  $n$  列で巾が  $2m+1(m=M)$  の行列方程式になっている。

一方、式 (1.3) の右辺の面積分を離散化した式 (2.1) の右辺  $f_i$  は、節点  $i$  での熱源密度を  $F_i$  とすると、Fig.2 の小領域 PQRS において、

$$(2.3) \quad f_i = \int_{\Omega} f \, d\mathbf{v} = F_i \left( \frac{hx_+ + hx_-}{2} \right) \cdot \left( \frac{hy_+ + hy_-}{2} \right)$$

となる。ここで、本当ならば、式 (2.2) の  $f_i$  は、Fig.2 の面 PQRS から発生または吸収される熱量を、4 つの部分の合算で求めなければならないのに手抜きをした。熱伝導率  $\kappa$  と同様に、面  $(i, i-m, i-m-1, i-1)$  内は一定の熱源をもつとして離散化するので、この面の熱源を、 $f_P [\frac{\text{cal}}{\text{sec}\cdot\text{cm}^3}]$  とすれば、 $\frac{1}{4}$  面 (iWPT) に当たる  $f_i$  への寄与は、 $f_P \frac{hx_-}{2} \frac{hy_-}{2} [\frac{\text{cal}}{\text{sec}}]$  (厚さが  $1[\text{cm}]$  あると思う) となる。したがって、節点  $i$  での  $f_i$  は、4 つの部分面を合算して、正しくは、

$$(2.4) \quad f_i = f_P \frac{hx_-}{2} \frac{hy_-}{2} + f_Q \frac{hx_+}{2} \frac{hy_-}{2} + f_R \frac{hx_+}{2} \frac{hy_+}{2} + f_S \frac{hx_-}{2} \frac{hy_+}{2} \quad \left[ \frac{\text{cal}}{\text{sec}} \right]$$

と求めなければならなかった。

今回のシミュレーションでは、熱源密度を節点  $i$  に対して、

$$i=42 \text{ to } 44, \quad i=52 \text{ to } 54 \text{ では: } F_i = +0.2 \left[ \frac{\text{cal}}{\text{sec}} \right]$$

$$i=46 \text{ to } 48, \quad i=56 \text{ to } 58 \text{ では: } F_i = -0.2 \left[ \frac{\text{cal}}{\text{sec}} \right]$$

と与える (Fig.1 の節点について 印の位置)。

かくして、この式 (2.1) の対称行列である、 $n$  ( $= N = M \times M1 = 10 \times 10 = 100$ ) 元の連立一次方程式：

$$(2.5) \quad A(a_{i,l}) \cdot U(u_i) = F(f_i), \quad i, l = 1 \sim n$$

を解けば良いことになる。この時、行列  $A(100,100)$  に対しては 80kByte のメモリを必要とする。計算精度を上げるために、あみ目のきざみ幅を 10 倍精しくすれば、行列  $A$  のメモリは 950M Byte 必要になり、すぐに増大する。理論上は、ガウスの消去法で解けるが、実務上ではすぐ解けなくなり、ソルバの工夫が不可欠となる。

Fig.1 の通り番号の節点  $i$  ( $= 1 \rightarrow 100$ ) は、行列方程式 (2.1) :  $A(a_{i,l})U(u_i) = F(f_i)$  に対して、

$$i = m \times (j - 1) + k \quad (j = 1, m1, k = 1, m(= M)); \quad m1 \text{ は } x \text{ 方向の点の数}$$

と対応させる。

Fig.1 の上側境界の係数に関しては、ノイマン境界なので、Fig.2 の CV の領域は下半分の WPQU となる ( $a_{i,i+1} = 0$  とする)。また、両側の固定境界 ( $u = u_0$ ) については、左側端の対処は、 $a_{i,i-m} = 0$  としたうえで、右辺を  $f_i - a_{i,i-m} \cdot u_0$  とするなどの留意が必要。同様に、右側端の対処は、 $a_{i,i+m} = 0$  としたうえで、右辺は  $f_i - a_{i,i+m} \cdot u_0$  とするが、今回のシミュレーションでは楽をするために、 $u_0 = 0$  とするので、右辺をいじる必要はない。

あみ目としては、Fig.1 の実線の場合には、はなはだ簡単に、 $hx = hx_+ = hx_- = 1[\text{cm}]$ ,  $hy = hy_+ = hy_- = 1[\text{cm}]$  であり、均等あみ目になっている。精しく解くための、村田の奨めるあみ目の取り方は、Fig.1 の 1cm を分割する分割数を MJ として、1cm 毎に MJ の値を変えてやる方法である。x 方向も y 方向も、同じ 1cm を MJ で分割すれば、 $hx_+ = hx_- = \frac{1\text{cm}}{MJ}$ ,  $hy_+ = hy_- = \frac{1\text{cm}}{MJ}$  となり、元数  $n$  は、

$$n = N = (MJ \times 10) \times (MJ \times 11 - 1), \quad m = M = MJ \times 10, \quad M1 = MJ \times 11 - 1$$

という計算になる。MJ=1 とすれば、Fig.1 の実線と同じになる。MJ=2 とすれば、あみ目が 2 倍に精しくなり、 $n = N = 20 \times 21 = 420$  となる。これでも、均等あみ目である。

実務上は、すべての場で均等にあみ目を精しくせず、メモリ節約のために、ソースのある所など大事な所だけを精しくして、不均等あみ目を作る。Fig.1 のように、x 方向の、

D - KXA 間は、1cm を MK に分割、  
KXA - KXB 間は、1cm を MK1 に分割、

KXB - KXC 間は、1cm を MKM に分割、  
KXC - KX1 間は、1cm を MK1 に分割

KX1 - C 間は、1cm を MK に分割 (左右対称にする。デバッグには必須。)、

することにする。たとえば、境界の KXA では、 $hx_- = \frac{1}{MK} \neq hx_+ = \frac{1}{MK1}$  となり、不均等あみ目となる。同様に、y 方向でも、

A - LY0 間は、1cm を MJ に分割、  
LY0 - LY1 間は、1cm を MJ1 に分割、

LY1 - LY2 間は、1cm を MJM に分割、  
LY2 - LY3 間は、1cm を MJ2 に分割

LY3 - D 間は、1cm を MJ3 に分割、

という仕掛けを作る。境界 : LY0, LY1, LY2, LY3 では、不均等あみ目  $hy_- \neq hy_+$  となる。不

均等あみ目は、行列  $A(a_{i,j})$  を解きづらくする一因となり、あみ目の取り方が解いた精度に影響する。

この時、実務上は、特にソースのある、MKM( $x$  方向) と、MJ1,MJ2( $y$  方向) を精しく(1cmの分割数を大きく) とる。全節点数 (=方程式の元数  $n(=N)$ ) は、

$$N = m \times M1 = (MJ + 3 \times MJ1 + 2 \times MJM + 3 \times MJ2 + MJ3) \\ \times (MK + 3 \times MK1 + 3 \times MKM + 3 \times MK1 + MK - 1)$$

となる。今回のシミュレーションでは簡単のために、 $x$  方向のあみ目段差での影響を見るため、 $y$  方向は MJ=MJ1=MJM=MJ2=MJ3 とするので、全節点数は、

$$N = (MJ \times 10) \times (MK + 3 \times MK1 + 3 \times MKM + 3 \times MK1 + MK - 1)$$

となる。

また、左右両側のあみ目だけを2倍にすれば、MK=2, MK1=1, MKM=1, MJ=MJ1=MJM=MJ2=MJ3=1 となり、D - KXA 間と KX1 - C 間に1本あみ目が入り、一番安易な不均等あみ目ができ、元数は、 $N = 10 \times (12) = 120$  となる (Fig.1 に点線で示したようになる)。

両側の熱伝導率  $\kappa$  は、特に  $\kappa=DF$  としておき、 $DF=0$  とすれば、両サイドは、上側端と同様にノイマン境界と同じになるようにする。 $DF=1=DF1$  等にすれば、両サイドは固定境界のままとなる。MK=1 では、D - KXA 間が  $\kappa=DF$  となり、MK=L > 1 では、一番外側の固定端に接する領域だけが  $\kappa=DF$  となり、残りの KXA までの領域 ( $\frac{1}{L}cm \sim 1cm$ ) では  $\kappa=DF1$  となるように設定する。したがって、Fig.1 の点線のように、MK=2 とすれば、D - KXA の半分の外側部分の  $\kappa$  が  $DF$  となり、その内側半分が  $DF1$  となる。熱伝導率  $\kappa$  は、 $x$  方向にだけに変化をつける道具立てを、用意しておく ( $y$  方向は一定)。すなわち、

$$\text{KXA - KXB 間は、}\kappa=DF2、\quad \text{KXB - KXC 間は、}\kappa=DF3、 \\ \text{KXC - KX1 間は、}\kappa=DF2$$

とし、縦縞の熱伝導率  $\kappa$  を設定する。今回のシミュレーションでは、場の熱伝導率  $\kappa$  に段差をつけるが、断らなければ、 $DF$  以外の  $\kappa$  値はすべて 1 としておく。したがって、 $DF=1$  とすると、均一の熱伝導率の場 ( $\kappa=1=一定$ ) となる。

### 3 クランク-ニコルソンによる差分

微分形式の式 (1.1) :  $div(-\kappa \nabla u) = f$  は、 $\kappa$  が場内で一定であれば ( $\kappa = d$ )、

$$(3.1) \quad -\kappa \Delta u = f = -d \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

となる。Fig.2 を参照して、

$$\frac{\partial^2 u}{\partial x^2} = \frac{\left( \frac{u_{i+m} - u_i}{h_{x+}} \right) - \left( \frac{u_i - u_{i-m}}{h_{x-}} \right)}{\left( \frac{h_{x+} + h_{x-}}{2} \right)}$$

などとして、式 (3.1) のクランク-ニコルソンによる差分式を作ると、

$$(3.2) \quad -d \left( \frac{\left( \frac{u_{i+m}-u_i}{h_{x+}} \right) - \left( \frac{u_i-u_{i-m}}{h_{x-}} \right)}{\left( \frac{h_{x+}+h_{x-}}{2} \right)} + \frac{\left( \frac{u_{i+1}-u_i}{h_{y+}} \right) - \left( \frac{u_i-u_{i-1}}{h_{y-}} \right)}{\left( \frac{h_{y+}+h_{y-}}{2} \right)} \right) = f$$

となる。

$\kappa$  が場に依存する時に、場に依らないとして出発した差分式 (3.2) で、熱伝導率  $d$  を、

$$d = \left( \frac{d_S + d_P + d_Q + d_R}{4} \right), \quad d_S \text{ は } S \text{ 点での } \kappa, \quad d_P, d_Q, d_R \text{ も同じ。}$$

として与えては、大間違い。

熱伝導率  $\kappa$  が場に依存する場合には、

$$(3.3) \quad \frac{\partial}{\partial x} \left( -\kappa \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( -\kappa \frac{\partial u}{\partial y} \right) = f$$

が出发点となる。Fig.2 にしたがって離散化をすれば、式 (3.3) は、

$$(3.4) \quad - \left( \frac{\left( \frac{d_R+d_Q}{2} \right) \left( \frac{u_{i+m}-u_i}{h_{x+}} \right) - \left( \frac{d_S+d_P}{2} \right) \left( \frac{u_i-u_{i-m}}{h_{x-}} \right)}{\left( \frac{h_{x+}+h_{x-}}{2} \right)} \right) - \left( \frac{\left( \frac{d_S+d_R}{2} \right) \left( \frac{u_{i+1}-u_i}{h_{y+}} \right) - \left( \frac{d_P+d_Q}{2} \right) \left( \frac{u_i-u_{i-1}}{h_{y-}} \right)}{\left( \frac{h_{y+}+h_{y-}}{2} \right)} \right) = f$$

となる。この式 (3.4) を式 (2.1) の第  $i$  番方程式と同様に、

$$(3.5) \quad a_{i,i-m}^C u_{i-m} + a_{i,i-1}^C u_{i-1} + a_{i,i}^C u_i + a_{i,i+1}^C u_{i+1} + a_{i,i+m}^C u_{i+m} = F_i$$

の形に整理すると、それぞれの係数  $a_{i,k}^C$  は、

$$\begin{aligned} a_{i,i}^C &= \frac{1}{2} \left[ d_P \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y-}} + \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x-}} \right) \right] \\ &\quad + \frac{1}{2} \left[ d_Q \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y-}} + \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x+}} \right) \right] \\ &\quad + \frac{1}{2} \left[ d_R \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y+}} + \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x+}} \right) \right] \\ &\quad + \frac{1}{2} \left[ d_S \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y+}} + \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x-}} \right) \right] \\ a_{i,i-m}^C &= -\frac{1}{2} \left[ d_S \left( \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x-}} \right) + d_P \left( \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x-}} \right) \right] \\ a_{i,i-1}^C &= -\frac{1}{2} \left[ d_P \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y-}} \right) + d_Q \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y-}} \right) \right] \\ a_{i,i+1}^C &= -\frac{1}{2} \left[ d_R \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y+}} \right) + d_S \left( \frac{2}{(h_{y+} + h_{y-}) \cdot h_{y+}} \right) \right] \end{aligned}$$

$$(3.6) \quad a_{i,i+m}^C = -\frac{1}{2} \left[ d_Q \left( \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x+}} \right) + d_R \left( \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x+}} \right) \right]$$

となる。

式(3.5)の右辺 $F_i$ が、CV法による第 $i$ 番方程式の右辺 $f_i$ と式(2.3)より、 $f_i = F_i \left( \frac{hx_+ + hx_-}{2} \right) \cdot \left( \frac{hy_+ + hy_-}{2} \right)$ の関係があることから、式(3.5)すなわち式(3.6)の両辺に、 $\left( \frac{hx_+ + hx_-}{2} \right) \cdot \left( \frac{hy_+ + hy_-}{2} \right)$ を掛けると、CV法の第 $i$ 番方程式(2.1)すなわち式(2.2)と同じ方程式となり、同じ係数どうしを両離散化方式で比較ができることになる。

係数 $a_{i,i}$ について比べる。式(2.2)のCV法の係数：

$$(3.7) \quad a_{i,i} = \frac{1}{2} \left[ d_P \left( \frac{hx_-}{hy_-} + \frac{hy_-}{hx_-} \right) + d_Q \left( \frac{hx_+}{hy_-} + \frac{hy_-}{hx_+} \right) + d_R \left( \frac{hx_+}{hy_+} + \frac{hy_+}{hx_+} \right) + d_S \left( \frac{hx_-}{hy_+} + \frac{hy_+}{hx_-} \right) \right]$$

に対して、クランク-ニコルソン法の係数は、式(3.6)から、

$$(3.8) \quad a_{i,i}^C \cdot \left( \frac{hx_+ + hx_-}{2} \right) \cdot \left( \frac{hy_+ + hy_-}{2} \right) = \frac{1}{2} \left[ d_P \left( \frac{(hx_+ + hx_-)}{2 \cdot hy_-} + \frac{(hy_+ + hy_-)}{2 \cdot hx_-} \right) \right] \\ + \frac{1}{2} \left[ d_Q \left( \frac{(hx_+ + hx_-)}{2 \cdot hy_-} + \frac{(hy_+ + hy_-)}{2 \cdot hx_+} \right) \right] \\ + \frac{1}{2} \left[ d_R \left( \frac{(hx_+ + hx_-)}{2 \cdot hy_+} + \frac{(hy_+ + hy_-)}{2 \cdot hx_+} \right) \right] \\ + \frac{1}{2} \left[ d_S \left( \frac{(hx_+ + hx_-)}{2 \cdot hy_+} + \frac{(hy_+ + hy_-)}{2 \cdot hx_-} \right) \right]$$

となる。

式(3.7)と式(3.8)の両方式の右辺を比較して、CV法とクランク-ニコルソン法で離散化した係数が、一致するためには、

$$hx_- = \frac{hx_+ + hx_-}{2}, \quad hx_+ = \frac{hx_+ + hx_-}{2} \\ hy_+ = \frac{hy_+ + hy_-}{2}, \quad hy_- = \frac{hy_+ + hy_-}{2},$$

すなわち、

$$hx_+ = hx_-, \quad hy_+ = hy_-$$

でなければならないことが分かる。

このことは、それぞれ $x, y$ 方向で、均等あみ目でないと、両離散化法で解いた解が一致しないことを示す。均等あみ目ならば、熱伝導度 $\kappa = d$ を変えても係数 $a_{i,k}$ の値は同じになる。不均等あみ目にした場合には、両離散化法で係数 $a_{i,k}$ の値が違う値になる。熱伝導率 $\kappa$ が場に依存して $d$ に段差があると、不均等あみ目による係数 $a_{i,k}$ の値の違いが結果に顕著に現れ、クランク-ニコルソン法では、解けないこともある。

次に、均等あみ目でない状況を作り、CV法とクランク-ニコルソン法による解を比較してみる。

## 4 両側のあみ目依存

Fig.1の両側のあみ目だけを、変えてみる。まず、両側端の熱伝導率は $\kappa=DF=1.0$ で、他面の全部の熱伝導率も $\kappa=d=1.0$ とする。したがって、熱伝導率 $d$ =一定の条件で、両側端のあみ目だけ、 $MK=2$ とした不均等あみ目と、均等あみ目 $MK=1$ (他のあみ目はすべて1)で比較してみる。

すなわち、熱伝導率が $d=1$ =一定の条件で、「Fig.1の均等あみ目(Type1)」と、「Fig.1のD - KXA間、KX1 - C間に1本あみ目(Fig.1の点線)を入れた(Type2)」場合において、両離散化方式の結果がどうかわるかを比較してみる。このとき、Fig.1のKXAとKX1の境界で、あみ目に段差ができる。

また、「 $x$ 方向のあみ目だけ、すべて2倍に精しく均等あみ目(Type3)」(Fig.1で $MK, MK1, MKM=2, 2, 2$ )にした場合も、比較のためにシミュレーションしておく。このときは、CV領域は縦長になるが、それぞれの領域で $hx_+ = hx_- = \frac{1}{2}cm$ ,  $hy_+ = hy_- = 1cm$ となり均等なあみ目となる。

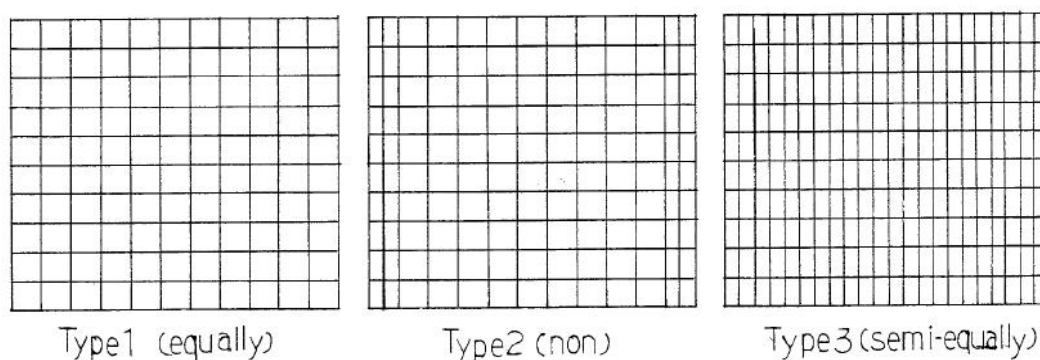


Fig.3 Mesh arrangement

( $\kappa=const.=1$ ; Type1[ $MK=1$ ], Type2[ $MK=2$ ], Type3[ $MK=MK1=MKM=2$ ])

- Fig.3のように、シミュレーション条件( $x$ 方向のあみ目だけを変える)を整理すれば、
- (1)Type1: 均等あみ目「 $MK, MK1, MKM=1, 1, 1$ 」で元数 $N=m \times M1=10 \times 10=100$ (CV領域は正方形  $hx_+ = hx_- = hy_+ = hy_-$ )
  - (2)Type2: 両側端だけ不均等あみ目「 $MK, MK1, MKM=2, 1, 1$ 」で元数 $N=m \times M1=10 \times 12=120$   
( $hx_+ \neq hx_-$ ,  $hy_+ = hy_-$ )
  - (3)Type3: 準均等あみ目( $x$ 方向だけ2倍精しく)「 $MK, MK1, MKM=2, 2, 2$ 」で元数 $N=m \times M1=10 \times 21=210$ (CV領域は縦長で均等  $hx_+ = hx_-$ ,  $hy_+ = hy_-$ )

となり。この3タイプのアミ目において、熱伝導率 $\kappa$ は一定としてCV法とクランク-ニコルソン法を比較してみる。

Table 3.1に、 $x$ 方向アミ目を不均等にしたType2、両方向均等のType1、片方向毎にあみ目を準均等にしたType3( $x$ 方向を2倍精しく)において、両離散化法でシミュレーションした、最低温度 $u_{min}(\quad)$ と最高温度 $u_{max}(\quad)$ の結果を示す。Table 3.1において、 $(\quad)$ 内には、最低温度 $u_{min}$ と最高温度 $u_{max}$ の位置を列番号( $x$ )と行番号( $y$ )で表してある。この



時に列番号 ( $x$ ) 位置は、あみ目を精しくした場合 (Type2,Type3) でも、Type1 の両方向均等の場合の列番号位置に「換算」して表記してある。その時のシミュレーション結果である、等温度線を Fig.4 に示す。

Table 3.1  $u_{max}(\quad)$ ,  $u_{min}(\quad)$ ; ( $x, y$ ) of Type1,2,3 for CV and Crank-N with  $\kappa=d=1.0=const.$ [DF=1.0]

MK,MK1,MKM	method	2,1,1(Type2)	1,1,1(Type1)	2,2,2(Type3)
$u_{min}$	CV	-0.3527; (5,8)	-0.3525; (5,8)	-0.2824; (5.5,8)
$u_{min}$	Crank-N	-0.3943; (5,8)[11.8 %]	-0.3525; (5,8)	-0.2824; (5.5,8)
$u_{max}$	CV	0.2138; (5,3)	0.2137; (5,3)	0.1793; (5.5,3)
$u_{max}$	Crank-N	0.2019; (5,3)[5.6 %]	0.2137; (5,3)	0.1793; (5.5,3)

あみ目を精しくすれば、Type1 から Type3 へと計算精度は上がる。片方向毎に均等あみ目ならば、CV 法とクランク-ニコルソン法で計算結果は、同じになる。もちろん両方向のあみ目が均等ならば、両方の離散化法で計算結果は、同じになる。ところが、熱伝導率  $\kappa=d$  =一定の場合でさえ、両側端のあみ目のみ、MK=2 (最低限のあみ目不均等) としただけで、計算結果は、熱源が「対称」であるにもかかわらず、計算した温度分布は「左右非対称」となり、Type1 の結果に近い値ではあるが、計算結果も  $u_{min}$  で 11.8 %、 $u_{max}$  で 5.6 % 違ってくる。

次に、同じあみ目 Type(Type1,2,3) をそのまま使い、すべてのあみ目 Type で、熱伝導率  $d$  を一定ではなく  $d$  も最低限変え、両側端の熱伝導率 DF だけ、DF=0.0(開放端：ノイマン境界と同じこと) として同様のシミュレーションを行なってみる。DF 以外の熱伝導率は  $d=1.0$  のままにする。type2 では、MK=2 であるので、Fig.1 の KXA 境界が最低限の不均等あみ目になると同時に、D - KXA 間の外側半分が  $d=DF=0.0$ 、内側半分が  $d=1.0$  となり熱伝導率 (場) にも最低限の段差ができる。Type1,Type3 では、あみ目に段差はできないが、両側端には熱伝導率の段差ができる。CV 法とクランク-ニコルソン法の離散化方式で離散化係数値に違いを生むのは不均等あみ目であって、熱伝導率の段差では違いを生まない。Type1,Type3 で両離散化方式の計算結果に差はでない。このように両側を開放端 DF=0.0 にして、両離散化方式で、最低温度  $u_{min}$  と最高温度  $u_{max}$  をシミュレーションした結果を Table 3.2 に示す。そのときの温度分布は、Fig.5 に示す。

Table 3.2  $u_{max}(\quad)$ ,  $u_{min}(\quad)$ ; ( $x, y$ ) of Type1,2,3 for CV and Crank-N with [DF=0.0] other  $\kappa=d=1.0$

MK,MK1,MKM	method	2,1,1(Type2)	1,1,1(Type1)	2,2,2(Type3)
$u_{min}$	CV	-0.6202; (5,8)	-0.6595; (6,8)	-0.4823; (5.5,8)
$u_{min}$	Crank-N	-0.7729; (1,6)[24.6 %]	-0.6595; (6,8)	-0.4823; (5.5,8)
$u_{max}$	CV	0.1226; (6,2)	0.1133; (5,2)	0.1088; (5.5,2)
$u_{max}$	Crank-N	0.8849; (1,10)[622 %]	0.1133; (5,2)	0.1088; (5.5,2)

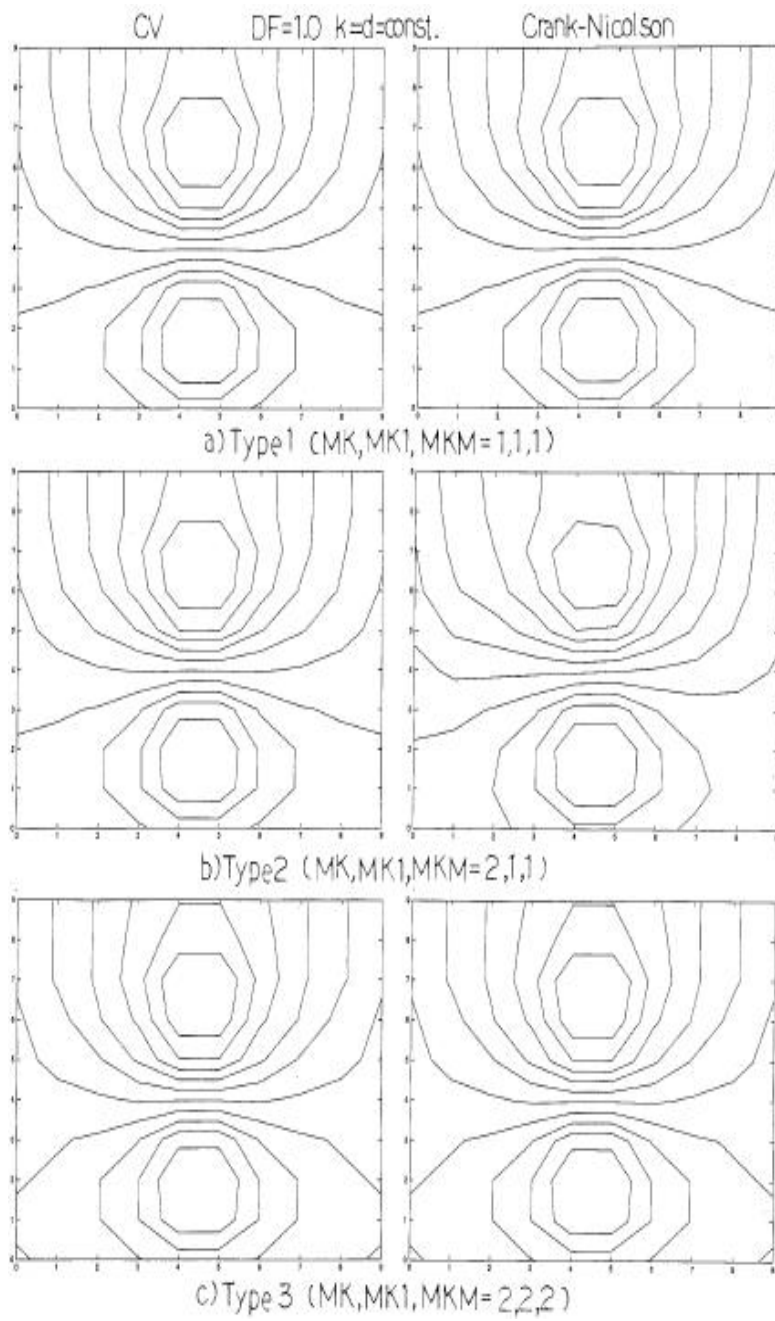


Fig.4 Temp.  $u(x,y)$  distribution of Type1,2,3 for CV and Crank-N with  $\kappa=d=1.0=\text{const.}$ [DF=1.0]

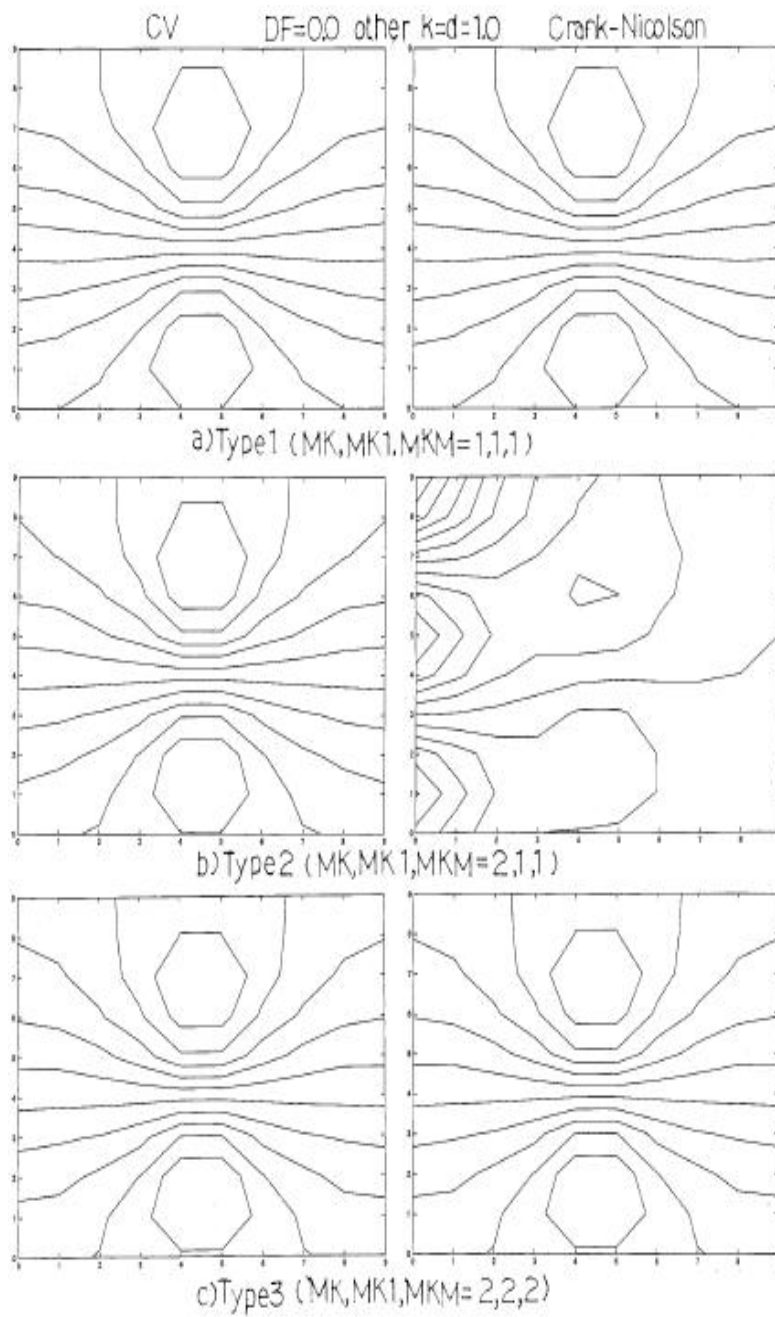


Fig.5 Temp.  $u(x,y)$  distribution of Type1,2,3 for CV and Crank-N with  $[DF=0.0]$  other  $\kappa=d=1.0$

結果として、両方向、片方向ずつ均等あみ目であれば (Type1,Type3)、両側端だけ熱伝導率  $d$  を  $DF=0.0$  と変えてみても、CV 法、クランク-ニコルソン法とも解は一致することが分かる。熱伝導率  $d$  に段差があっても、均等あみ目であれば、離散化係数値が両方法で変わらないので当然 解は一致し、問題なく解ける (Fig.4 と Fig.5 の a),c))。

両側端のあみ目だけ精しく (MK=2) して不均等あみ目とすると、両方法で離散化係数値は違ってくる。その条件 (Type2) で、さらに両側端の熱伝導率  $d$  も  $DF=0.0$  (ノイマン境界)、他の面は  $d=1.0$  として熱伝導率にも最低限の段差をつけると、クランク-ニコルソン法では、あみ目の段差だけでは CV 法の正しい結果とは少しの違いであったが (しかし非対称になってしまう。Fig.4 b))、そこに熱伝導率  $d$  の段差の両方が加わると、結果が少し崩れるどころではなく、全く解けなくなることが分かる (Fig.5 b))。CV 法ならば、あみ目に段差をつけても、熱伝導率  $d$  に段差をつけても、安定して正しく解ける。クランク-ニコルソン法では、あみ目に段差をつけると正しく物理的な解が得られず、さらに加えて熱伝導率  $d$  にも段差をつけると解の崩れがより拡大し解けない。

このクランク-ニコルソン法では解が解けない、Type2 : MK,MK1,MKM=2,1,1 の場合には、CV 法とクランク-ニコルソン法の離散化係数値が一致する条件 :  $\frac{h_{x-}}{h_{x+}} = 1$  が、Fig.1 の両側の KXA,KX1 上 (AD に平行) の格子点では、 $\frac{h_{x-}}{h_{x+}} = \frac{1}{1} = \frac{1}{2}$  となっている。

また、熱伝導率  $d$  が、 $DF=1.0$  でも  $DF=0.0$  の場合でも、CV 法による解の結果は、結果の精度には熱源 (ソース部) のあみ目の精しさが効くので、MK,MK1,MKM= < 2,1,1 > の解は、< 2,2,2 > の解よりも < 1,1,1 > の解の結果に近いことが分かる (Table 3.1, 3.2)。

あみ目に段差があっても、CV 法は安定して正確に解けるが、クランク-ニコルソン法では解けなくなり、そのあみ目の段差にさらに熱伝導率  $d$  の段差が加わると、クランク-ニコルソン法は全く使えなくなることが一例で分かった。次に、この Type2 の例:MK,MK1,MKM=2,1,1 において、クランク-ニコルソン法では解が少し非対称に崩れる  $DF=1.0$  (熱伝導率  $d$ =一定で段差なし。  $\frac{h_{x-}}{h_{x+}} = 1$ ) から、解が全く解けない  $DF=0.0$  (両側端をノイマン境界。  $\frac{h_{x-}}{h_{x+}} = 0.5$ ) まで、 $DF$  値を変えて、解の代表値とする最高温度  $u_{max}$  と最低温度  $u_{min}$  (その換算した位置も) の変化を調べてみる。このシミュレーションでは、Type2 の不均等あみ目において、熱伝導率  $DF$  値によってどのくらいから解が発散してしまうのか、正しい CV 法と発散するクランク-ニコルソン法を比べることによって調べることができる。

Fig.6 において、最高温度  $u_{max}$  と最低温度  $u_{min}$  とともに、点線が CV 法で、実線がクランク-ニコルソン法の結果を示している。Fig.6 図の内側の  $DF=1.0$  から  $DF=0.0$  までのグラフには、MK,MK1,MKM=2,1,1 の時の CV 法とクランク-ニコルソン法の結果の推移が示してある。その両外側には、左側  $DF=1.0$  と右側  $DF=0.0$  の時の、MK,MK1,MKM=1,1,1 と MK,MK1,MKM=2,2,2 の解も併記した。特に、CV 法では、解の精度には、ソース部のあみ目が効くので MK,MK1,MKM= < 1,1,1 > の解が、< 2,2,2 > の解よりも < 2,1,1 > の解に近いことが見てとれる。またこのとき、右側の  $DF=0.0$  の場合の方が、あみ目の精しさによる解の精度の差が少ない。

Fig.6 によれば、MK,MK1,MKM=2,1,1 において、熱伝導率  $DF$  を 1.0 から 0.0 まで変化させると、CV 法では安定して正確に解けるが、クランク-ニコルソン法では、左側の  $DF=1.0$  の時に、すでに解に左右非対称のずれが現れ (熱源は左右対称)、それは徐々に大きくなり

DF=0.7より熱伝導率が小さくなる(右側に行く)と、最高温度  $u_{max}$ 、最低温度  $u_{min}$  とも解が発散することが分かる。これでは、クランク-ニコルソン法は使えない。

この様子は、Fig.7に示してある。右側のクランク-ニコルソン法の解は、DF=0.7を超えるあたりから急に崩れていく。クランク-ニコルソン法では、解はDF値がDF=0.7まではDF=1.0の解に近いが、あみ目の段差により解が非対称になり(保存則を満足していない)、DF=0.9の時ですら正しいCV法の解と比べ、最高温度  $u_{max}$  で8%、最低温度  $u_{min}$  で15%も解が違う。その後、DF=0.6でひどく崩れ、DF値が0.5より小さくなってくると解が振動しだし、DF=0.0では、最高温度  $u_{max}$  が発散してしまう。

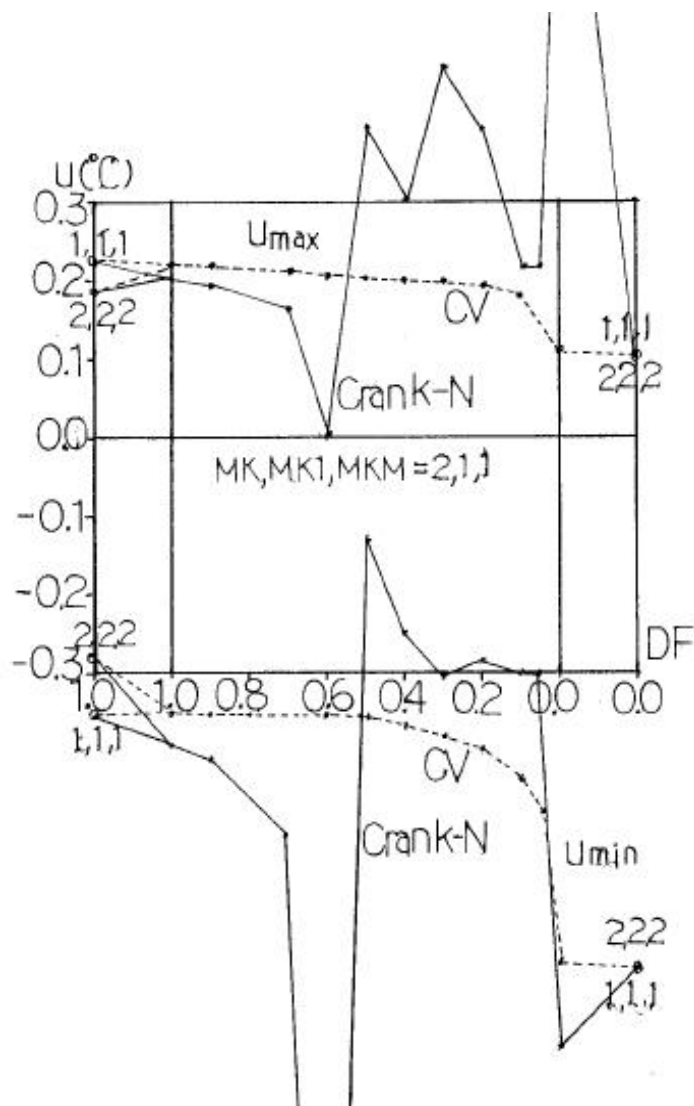


Fig.6 DF dependence(1.0→0.0) of  $u_{max}$ ,  $u_{min}$  with Type2 [MK, MK1, MKM=2, 1, 1] for CV, Crank-N

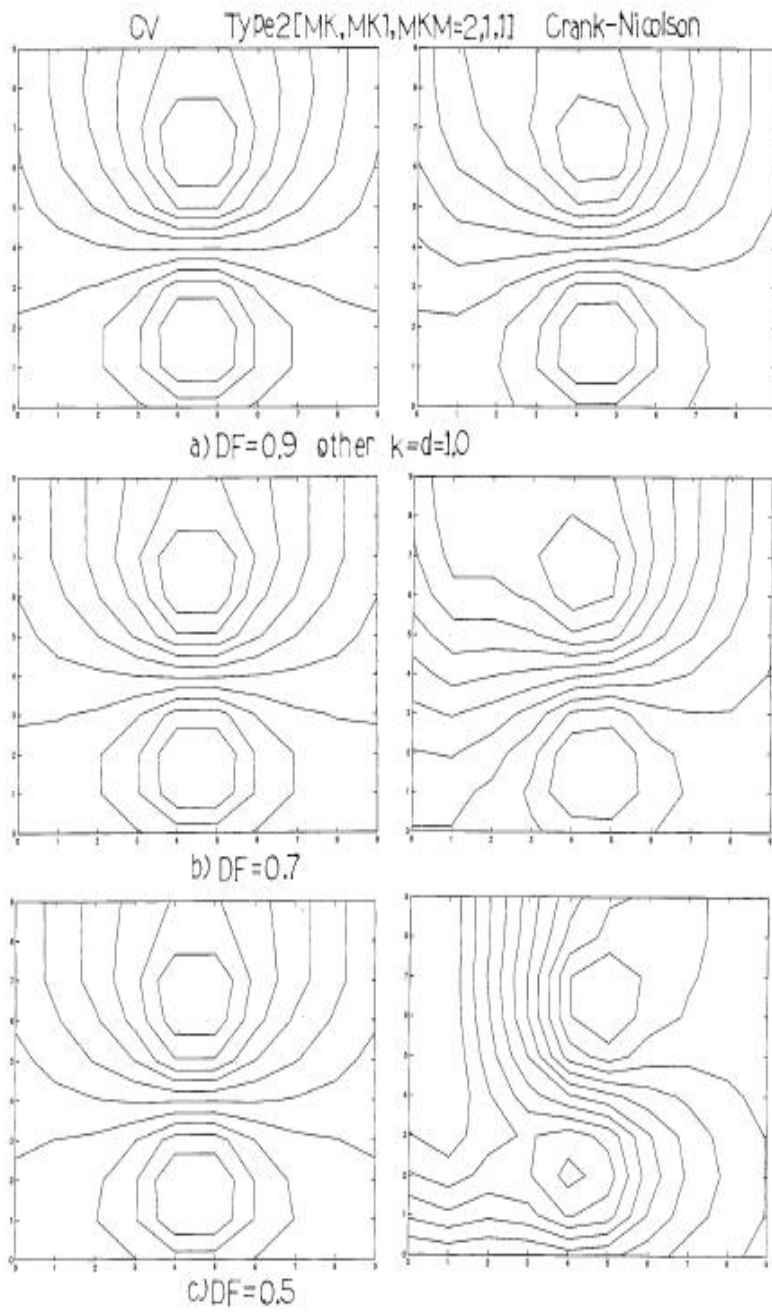


Fig.7 Temp.  $u(x,y)$  distribution of Type2 [MK,MK1,MKM=2,1,1] with DF=0.9,0.7,0.5 for CV and Crank-N

今までは、熱源(ソース)があるあみ目ではなく、熱源から遠い両側だけのあみ目と熱伝導率  $d$  を変えて様子を見た。それによって以下の示唆を得た。

- (1) CV 法はいつも安定で正しいが、クランク-ニコルソン法は、あみ目に段差:  $h_{x+} \neq h_{x-}$ ,  $h_{y+} \neq h_{y-}$  があると、保存則を満たすように離散化係数が作れず解が非対称になり正しく解けない。
- (2) あみ目の段差に加えて、熱伝導率  $d$  にも段差があると、クランク-ニコルソン法の解は急に崩れ解が解けない。

次に、実務上よくやる、ソースのある場所のあみ目を精しくして、ソース上のあみ目の段差の大きさが解に与える影響を見る。ソース上のあみ目の段差 ( $x$  方向) を精しくした MK,MK1,MKM=1,1,2 (Type4) と、これを均等あみ目にした MK,MK1,MKM=2,2,2 (Type3) と、ソース上以外を精しくした MK,MK1,MKM=2,2,1 (Type5) とを、熱伝導率  $d$ =一定となる固定境界条件の DF=1.0 ( $u=0$ ) と最低限の  $d$  の段差となる両側ノイマン境界条件の DF=0.0 の場合でシミュレーションした最高温度  $u_{max}$  と最低温度  $u_{min}$  の計算結果を、Table 3.3 (DF=1.0 の場合) と Table 3.4 (DF=0.0 の場合) に示す。ソースのある MKM のあみ目が  $u_{max}$  と  $u_{min}$  の計算精度に効くので、DF=1.0 も DF=0.0 の場合も、MK,MK1,MKM=< 1,1,2 > (Type4) の結果と < 2,2,2 > (Type3) の結果は近く、< 2,2,1 > (Type5) と < 1,1,1 > (Type1) の結果は近い。Table 3.3 と Table 3.4 において、( ) 内は ( $x,y$ ) で、 $u_{max}$  と  $u_{min}$  の Type1 の格子番号に換算した (列番, 行番) を表す。このとき Type4 と Type5 の、計算した温度分布の等高線を、Fig.8(DF=1.0) と Fig.9(DF=0.0) に示す (Type1 の結果は比較の参考)。

Table 3.3  $u_{max}$ ( ),  $u_{min}$ ( ); ( $x,y$ ) of Type3,4,5 for CV and Crank-N with  $\kappa=d=1.0=\text{const.}$ [DF=1.0]

MK,MK1,MKM	method	1,1,2(Type4)	2,2,2(Type3)	2,2,1(Type5)
$u_{min}$	CV	-0.2813; (5.5,8)	-0.2824; (5.5,8)	-0.3540; (6,8)
$u_{min}$	Crank-N	-3.3216; (7,10)	-0.2824; (5.5,8)	-3.7849; (4,10)
$u_{max}$	CV	0.1778; (5.5,3)	0.1793; (5.5,3)	0.2158; (5,3)
$u_{max}$	Crank-N	3.7161; (7,3)	0.1793; (5.5,3)	4.1360; (4,3)

Table 3.4  $u_{max}$ ( ),  $u_{min}$ ( ); ( $x,y$ ) of Type3,4,5 for CV and Crank-N with [DF=0.0] other  $\kappa=d=1.0$

MK,MK1,MKM	method	1,1,2(Type4)	2,2,2(Type3)	2,2,1(Type5)
$u_{min}$	CV	-0.5106; (5.5,8)	-0.4823; (5.5,8)	-0.6216; (6,8)
$u_{min}$	Crank-N	-1.5889; (7,3)	-0.4823; (5.5,8)	-1.8265; (4,3)
$u_{max}$	CV	0.1007; (5.5,2)	0.1088; (5.5,2)	0.1242; (5,2)
$u_{max}$	Crank-N	1.9896; (7,10)	0.1088; (5.5,2)	2.1939; (4,10)

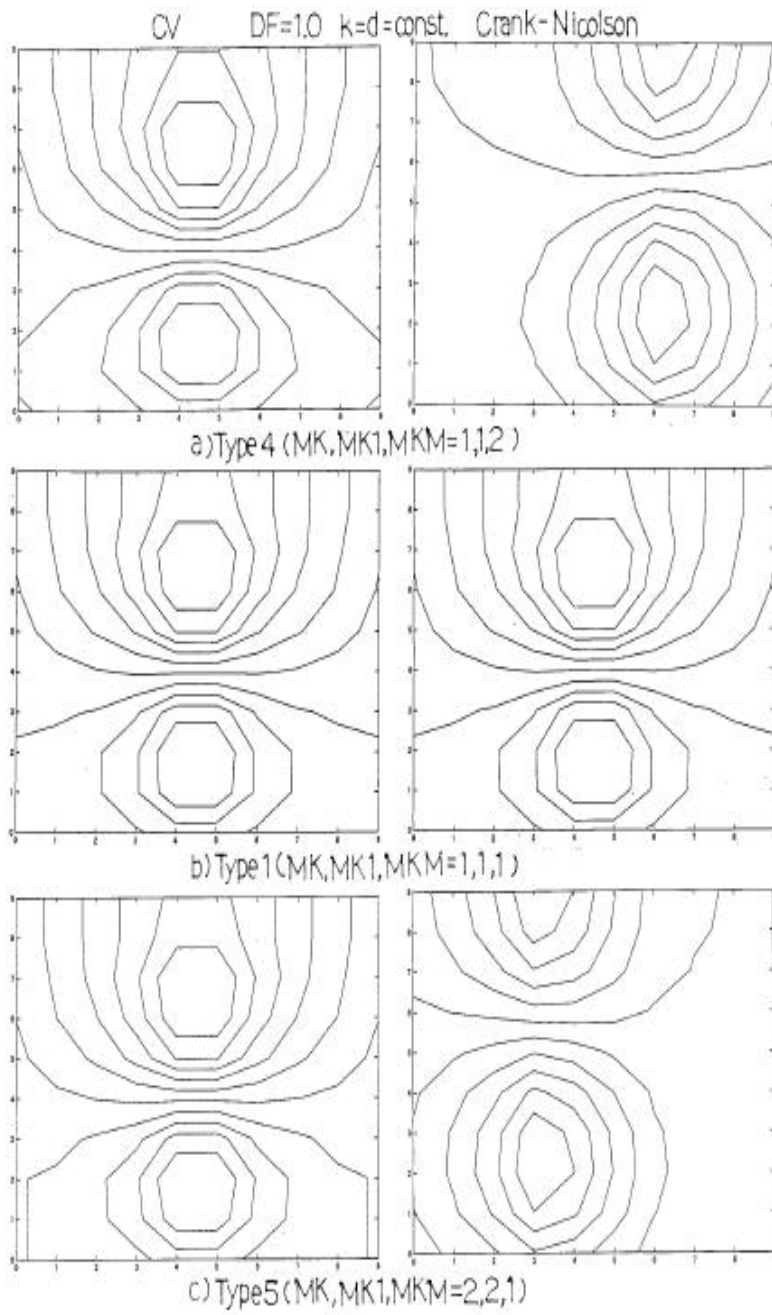


Fig.8 Temp.  $u(x,y)$  distribution of Type4,1,5 for CV and Crank-N with  $\kappa=d=1.0=\text{const.}$  [DF=1.0]



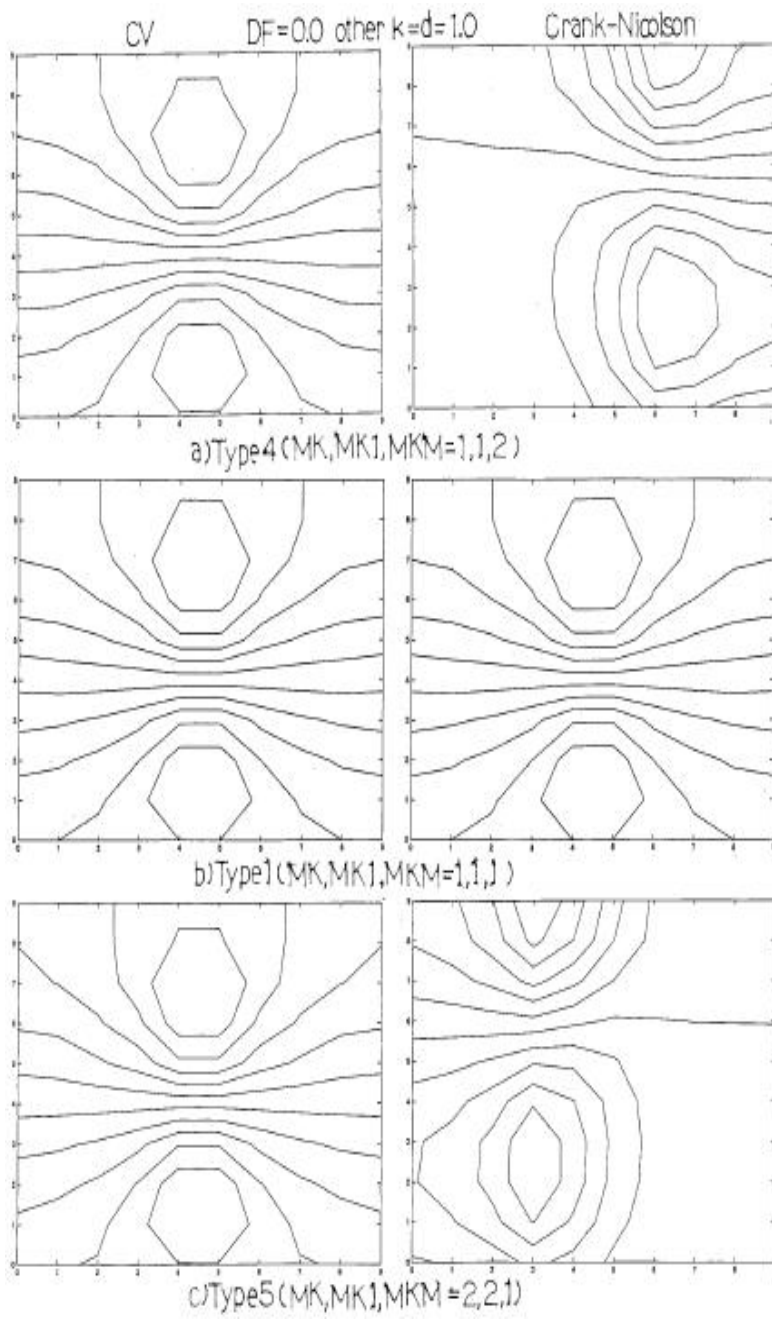


Fig.9 Temp.  $u(x, y)$  distribution of Type1,2,3 for CV and Crank-N with  $[DF=0.0]$  other  $\kappa=d=1.0$

Fig.8, Fig.9によれば、CV法は安定に解ける。一方、DF=1.0, DF=0.0ともクランク-ニコルソン法によるMK, MK1, MKM = < 1, 1, 2 > (Type4)の結果は、 $u_{max}$ と $u_{min}$ の位置が右に流れ非対称になりおかしい。同様に、DF=1.0, DF=0.0ともクランク-ニコルソン法によるMK, MK1, MKM = < 2, 2, 1 > (Type5)の結果は、 $u_{max}$ と $u_{min}$ の位置が< 1, 1, 2 > (Type4)とは逆に左に流れ非対称になりおかしい。クランク-ニコルソン法は、離散化した時に係数が保存則を満たしていないので、熱が非対称に流れこのような解になる。CV法の解の精度は、シミュレーションから0.01%のオーダーであることが分かっている。

## 5 おわりに

これまで見たように、保存則を満足するように差分化するCV法は、あみ目に段差をつけても安定に正しく解けるが、クランク-ニコルソン法では、たとえ拡散係数 $\kappa$ が一定でも正しく解けないことを示した。さらに、あみ目の段差に加え、拡散係数 $\kappa$ にも段差があると、CV法では解けるが、クランク-ニコルソン法では全く解けないことを確認した。

実は、この結論には重大な誤りがある。Fig.2を見ていただきたい。CV法において、節点 $i$ におけるCV領域周囲の1つの境界RQに対する線積分から出てくる第 $i$ 番方程式(2.1)の係数 $a_{i,i+m}$ は、右隣の節点 $i+m$ を $i$ 点と見立てた時の係数 $a_{i,i-m}$ と同じはずである(Fluxが境界RQで保存するCV法では、 $a_{i,i+m} = a_{(i+m),(i+m)-m} = a_{i+m,i}$ のはず)

節点 $i$ での $a_{i,i+m}$ は、

$$a_{i,i+m} = -\frac{1}{2} \left[ d_Q \frac{hy_-}{hx_+} + d_R \frac{hy_+}{hx_+} \right]$$

となる。右隣の節点 $l = i + m$ での係数 $a_{l,l-m} = a_{i+m,i}$ は、Fig.2から、節点 $i$ の $h_{x+}$ は節点 $i + m$ の $h_{x-}$ と等しい等を考慮して、

$$a_{i+m,i} = -\frac{1}{2} \left[ d_Q \frac{hy_-}{hx_+} + d_R \frac{hy_+}{hx_+} \right]$$

となり、同じ係数値になっている。したがって、CV法では拡散方程式(1.1)を離散化すると、行列方程式(2.1)： $A(a_{i,l})U(u_i) = F(f_i)$ の行列Aは対称行列になる。それは物理的にも正しい。対称行列ならば、行列方程式を解くソルバも対称行列の利点を生かし、メモリも計算時間も半分ですむ技法もある。

しかしながら、クランク-ニコルソン法では、確かに先の議論のように、 $x, y$ 方向それぞれに均等あみ目ならば( $h_{x+} = h_{x-}$ ,  $h_{y+} = h_{y-}$ )、係数 $a_{i,j}$ はCV法と同じになるが、不均等あみ目の場合には行列方程式のAが対称行列にならない。

クランク-ニコルソン法では、 $a_{i,i+m}^C$ は、

$$a_{i,i+m}^C = -\frac{1}{2} \left[ d_Q \left( \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x+}} \right) + d_R \left( \frac{2}{(h_{x+} + h_{x-}) \cdot h_{x+}} \right) \right]$$

となる。節点 $i + m$ での係数 $a_{(i+m),(i+m)-m}^C = a_{i+m,i}^C$ は、

$$a_{i+m,i}^C = -\frac{1}{2} \left[ d_Q \left( \frac{2}{(h_{x+1} + h_{x+}) \cdot h_{x+}} \right) + d_R \left( \frac{2}{(h_{x+1} + h_{x+}) \cdot h_{x+}} \right) \right]$$

となる。この係数  $a_{i+m,i}^C$  の中には、節点  $i$  の CV 領域外である、右隣り節点  $i+m$  の CV 領域における  $h_{x+}$  ( $h_{x+1}$  と記した) が入ってくる。したがって、実はクランク-ニコルソン法で離散化した行列  $A$  は、不均等あみ目では非対称行列になってしまう。

これまでのシミュレーション結果は、行列  $A$  が対称行列として、配列には半分の上三角部分だけを格納して対称行列用の反復法:ICCG12法を用いて行なった。このため、不均等あみ目のシミュレーション結果すべてにおいて、クランク-ニコルソン法による結果はおかしくなって解けなかったのである。クランク-ニコルソン法であっても、不均等あみ目の場合には非対称行列用のソルバを使って解けば結果は CV 法の結果と全く同じになる。ただし、クランク-ニコルソン法では、CV 法に比べメモリも CPU 時間も、わざわざほぼ倍余計にかかることになる (非対称行列用の反復法は解きづらい)。

村田は、数値シミュレーションを、次の三段階からなるとみよ、と言う (文献 [1])。

- (1) 現象のモデル化：偏微分方程式の問題に表現
- (2) 上記問題を解く：離散化と数値解法
- (3) 得られた結果を目的に照らして評価する：モデルの当否、解法の当否を含む

そして、これら一連のプロセスが完結して、はじめて数値シミュレーションも完結するという特徴を持つのである。今回のクランク-ニコルソン法による場合は、一連のプロセスを完結していなかった悪い例である。

いま、離散化した行列方程式 (2.1) :  $A(a_{i,l})U(u_i) = F(f_i)$  は、Fig.12 のようになっている。

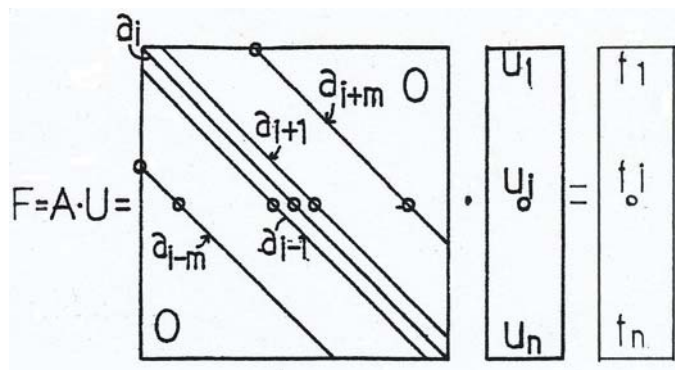


Fig.12 Band Matrix

この行列方程式を解くには、通常、直接法のガウスの消去法を使うとされる。このガウスの消去法の核 (主部) は、

「  $k$  番方程式 (軸方程式) に、 $\alpha_{ik} = -a_{ik}/a_{kk}$  をかけたものを、 $i$  番方程式 ( $i \geq k+1$ ) に加える：  $a_{ij} = a_{ij} + \alpha_{ik} \cdot a_{kj} ; j = k+1, n$  」

ところである (Fig.13)。この操作を、どういう順番でやるかには任意性がある、2つのやり方がある。先に  $j$  を動かす GLUR 型と、先に  $i$  を動かす GLU 型に分かれる。FORTRAN では、主記憶に  $a_{i,j} = A(I, J)$  として、配列  $A$  の添え字の  $I$  から先に格納されるのでメモリ参照の連続性から GLU 型がだんぜん有利で計算時間が速く良い。

[1] 行ガウス (先にjを動かす; GLUR)

```
do i=k+1,n
   $\alpha = -a_{ik}/a_{kk}$ 
  do j=k+1,n
     $a_{ij} = a_{ij} + \alpha \cdot a_{kj}$ 
```

[2] 列ガウス (先にiを動かす; GLU)

```
do j=k+1,n
   $\beta = -a_{kj}/a_{kk}$ 
  do i=k+1,n
     $a_{ij} = a_{ij} + a_{ik} \cdot \beta$ 
```

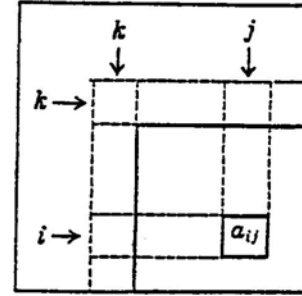


Fig.13 Gauss's elimination(GLU and GLUR)

7年前(2001年)の「日立 FLORA DX2、Pentium3,933MHz クロック、256MB メモリ、富士通 FORTRAN V3.0L10」の計算機と、現在の「日立 FLORA HX1、Pentium4,3.20GHz クロック、2GB メモリ、富士通 FORTRAN V4.0L10」の計算機で行なった CPU 時間を比較すると、Table 5.1 となる。DF=1.0 として、MJ 系列 ( $y$  方向) と MK 系列 ( $x$  方向) とも均等に 5 分割して (MJ=MK=5) 均等あみ目でシミュレーションを行なった。

Table 5.1 CPU Performance for 2001,2008 PC (DF=1.0,MJ=MK=5 equally Mesh)

PC	MJ=MK	N	A(MB)	GLU(sec)	GLUR(s)
Pen3, 933MHz, 256MB	5	2700	58	335[1.0]	860[2.6]
Pen4, 3.20GHz, 2GB	5	2700	58	25 [1.0]	151[6.0]

7年前には、GLU(列ガウス)は GLUR(行ガウス)より約3倍速く解が求まった。今の PC で同じ条件で行なった計算は、GLU が GLUR より約6倍速くなっている。これは CPU の キャッシュサイズが増え、メモリも増え、さらに FORTRAN コンパイラの内側 DO ループの最適化が良くなったためであると考えられる。もちろん、同じ GLU でも PC の性能が上がったのでクロック比以上に速くなっている。

さらに、計算精度を上げるために、均等あみ目の精しさを MJ=10 としたならば、元数  $N=10900$ 、配列 A のメモリは 968MB 必要になり、原理上ガウスの消去法は解けても、実務上はメモリの制約や、膨大な CPU 時間により解けない。そこで、離散化した行列 A の特徴を生かす。行列 A の要素が、 $a_{i,i-m}, a_{i,i-1}, a_{i,i}, a_{i,i+1}, a_{i,i+m}$  しかない、幅が  $2m+1$  (均等あみ目の場合には  $m=10 \times MJ$ ) の帯 (バンド) 行列になっていることから (Fig.12)、行列 A をサイズ「 $N^2$ 」の配列 A(N,N) に格納するのではなく、サイズが「 $(2M+1) \times N$ 」の配列 AR(-M:M,N) に係数を格納してガウスの消去法を行なう BGLU 型を使うのである。均等あみ目の場合には、元数  $N=(10 \cdot MJ)(11 \cdot MJ-1)$ 、幅  $M=10 \cdot MJ$  で、メモリは GLU の  $N^2$  語から BGLU の  $2MN$  語に大分減る。A が対称行列ならば、さらにメモリは半分となり BSGLU 型のガウスの消去法を使う。消去する過程で、帯の内側の要素: $a_{i,i-m}, a_{i,i-1}, a_{i,i}, a_{i,i+1}, a_{i,i+m}$  以外の要素も非ゼロ化する。

BGLU 型により、メモリの節約と CPU 時間の大幅な短縮ができるとはいえ、しかし、これとてコンパイラのメモリの制約から GLU では均等あみ目で MJ=7 が限度、BGLU でも MJ=24 が限度 (N=63120, 配列 AR のメモリ 243MB) である。Table 5.2 に、均等あみ目で MJ を 1 から 20 まで変化させて最低温度  $u_{min}$  値、最高温度  $u_{max}$  値の精度と CPU 時間 (sec) の推移を示す。熱伝導率  $d$  も最低限変えた例として DF=0.0 でシミュレーションを行なった。

Table 5.2 Solver performance for MJ(mesh size equally) DF=0.0

MJ	$u_{min}$ (C)	$u_{max}$ (C)	N	A(MB)	AR(MB)	BGLU(s)	GLU(s)	GLUR(s)	ER(%)
1	-0.6595	0.1133	100	0.08	0.02	0.	0.	0.	178
2	-0.4823	0.1088	420	1.4	0.14	0.	0.	0.	103
5	-0.2880	0.0825	2700	58	2.2	0.	24	187	21
7	-0.2682	0.0793	5320	226	6	0.	184	3520	12
15	-0.2427	0.0749	24600	4841	59	2	-	-	1.9
20	-0.2374	0.0739	43800	15347	140	4	-	-	0.0

通常、あみ目を精しくすると値が小さくなる。あみ目の精しさと計算精度のグラフは Table 5.2 を元に Fig14 に示す。この結果から、均等あみ目 MJ=20 の結果を基準として、計算精度が 2% 程度欲しければ、あみ目が MJ=15 は是非とも必要であることが分かる。これでは、GLU は使えない。

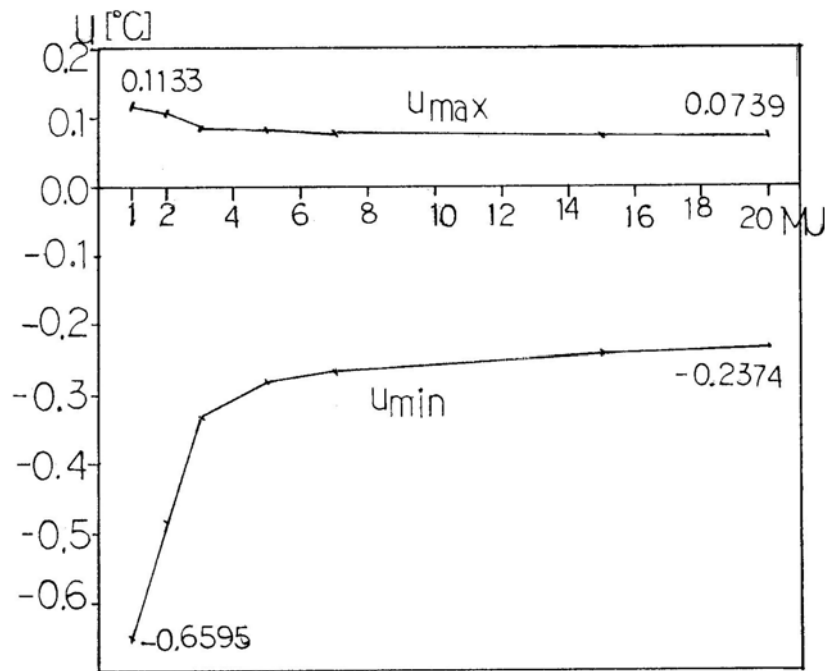


Fig.14 MJ dependence of  $u_{min}$ ,  $u_{max}$  DF=0.0(equally mesh)

さらに計算精度が必要な場合は、実務的にはメモリと計算時間の負担を減らしたい一心で、直接法の代わりに反復解法を使う。ただし、反復解法は解の精度や収束が問題となるので、帯ガウスは反復解法のデバッグにやはり必要となる。現在、反復法は、対称行列用には ICCG 法 (共役勾配法) を、非対称行列用には BCGstab (BCGSTB: 双共役勾配法の改良法) をベストな解法として使う。対称行列を BCGSTB で解いた場合には、計算時間は ICCG より約 3 倍遅くなる。

対称行列用の反復解法である ICCG 法は、前処理付き共役勾配法 (PCG: Pre-condition Conjugate Gradient) の前処理に「不完全 LU 分解」を利用した方法である。この方法は、Hestenes と Stiefel らの天才が考えた CG 法から紆余曲折を経て、20 年後に Gustafsson らにより目覚めた。行列  $A$  も、対称行列なので  $a_{i,i}, a_{i,i+1}, a_{i,i+m}$  の 3 本だけを配列に格納すればよいので、メモリは  $3N$  語だけで済み大幅な節約となる。PCG 法は、前処理に行列  $A$  に近い対称正定値行列  $\tilde{A}$  を選ぶ。Reid は、 $\tilde{A}$  として、行列  $A$  の 3 重対角部 ( $a_{i,i-1}, a_{i,i}, a_{i,i+1}$ ) の 3 本を選んだが、3 重対角部の LU 分解は完全に行なわなければならないので、それなりの CPU 時間を要した。そこで、 $\tilde{A}$  に不完全 LU 分解を使うことで、ICCG 法は格段の進歩を遂げた。行列  $A$  を LU 分解すると、行列  $L$  と  $U$  の帯の内部はほぼ完全に非ゼロ化する。そのため、BGLU 型では  $2MN$  語のメモリを要した。そのメモリの軽減のために、不完全 LU 分解を考える。

不完全 LU 分解とは、LU 分解し易い行列  $A$  に近い  $\tilde{A} = A_0 = LU$  を作ることである。積  $LU$  が行列  $A$  に等しくないのが不完全 LU 分解と言う。  $A_0$  として、行列  $A$  に加えて非ゼロの行要素が 1 本:  $a_{i,i+m-2}$  の所だけに生ずるように、 $A_0 = L \cdot U$  の上三角行列  $U$  を作るのである (Fig.15)。このとき Fig.15 のように、上三角行列  $U$  には、 $i$  行要素が  $a_{i,i}, a_{i,i+1}, a_{i,i+m}$  に加え、 $a_{i,i+m-1}$  の非ゼロの要素 1 本だけが加わるだけに仕向け、この  $A_0$  を前処理に使う ICCG を ICCG12 法と呼ぶ。  $A_0$  に行列  $A$  に加えて行要素が 1 本:  $a_{i,i-m+2}$  の所だけに非ゼロ要素を作る Fig.15 の下三角行列  $L$  には、 $a_{i,i-m+1}$  の非ゼロ要素が 1 本だけ加わるように仕向けるが、 $A_0$  は対称行列なので、行列  $U$  だけを考えればよい。

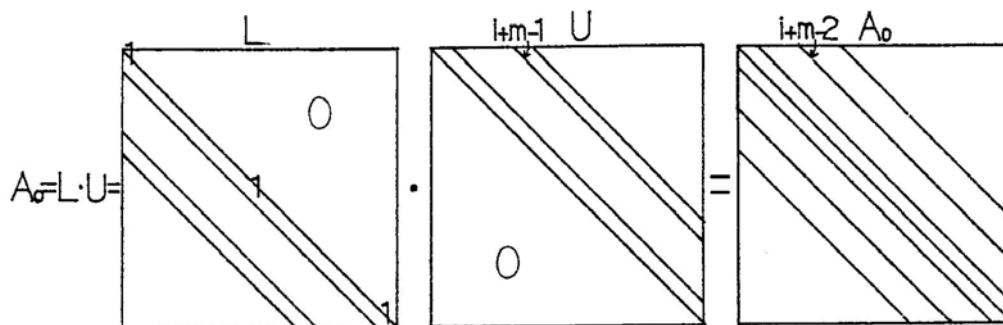


Fig.15  $A_0$  Set by Incomplete LU

また、図にはないが、 $A_0$ として、行列  $A$  に加えて非ゼロの行要素が 1 本:  $a_{i,i+m-3}$  の所にだけ加わったものを使い、上三角行列  $U$  には非ゼロ要素が  $a_{i,i}, a_{i,i+1}, a_{i,i+m}$  に加え、 $a_{i,i+m-1}$  と  $a_{i,i+m-2}$  の 2 本 (1 本増やす) が加わるだけに仕向け、その  $A_0$  を PCG 法の前処

理に使うICCGをICCG13法と呼ぶ。まとめれば、「PCG法+前処理に不完全LU分解=ICCG法」という図式になり、不完全LU分解の丁寧さにより12,13などの添え字が付く。ICCG13法の方が、メモリは1本余計に使うが、 $A_0$ が行列Aに近いのでICCG12より収束が速い。CPU時間比は、ICCG12/ICCG13=1.2見当である。

したがって、ICCG12法であれば、必要なメモリは $5N$ 語となり、BGLUの $2MN$ 語より格段に減る。BCGSTB法ならば、非対称行列用なのでメモリはほぼ2倍必要となる。Table 5.3のように必要な配列メモリは均等あみ目サイズMJに対して、直接法のGLUは $MJ^4$ のオーダー、BGLUは $MJ^3$ のオーダー、反復法のICCG12、BCGSTBとも $MJ^2$ のオーダーとなる。BGLUとても、あみ目MJが大きくなれば、原理上解けてもすぐに実務上は解けなくなる。

Table 5.3 Solver and using memory(equally mesh)

type	matrix	solver	A(Byte)	N(M=10·MJ)
Gauss	Nonsym	GLU	$8 \cdot N^2$	$(10 \cdot MJ)(11 \cdot MJ - 1)$
Gauss	Nonsym	GLUR	$8 \cdot N^2$	$(10 \cdot MJ)(11 \cdot MJ - 1)$
Gauss	Nonsym	BGLU	$8 \cdot (2M + 1) \cdot N$	$(10 \cdot MJ)(11 \cdot MJ - 1)$
Iteration	Sym	ICCG12	$8 \cdot 5N$	$(10 \cdot MJ)(11 \cdot MJ - 1)$
Iteration	Nonsym	BCGSTB12	$8 \cdot 9N$	$(10 \cdot MJ)(11 \cdot MJ - 1)$

Table 5.4に、DF=0.0として反復法:ICCG12とSCGSTB12で計算した $u_{min}$ ,  $u_{max}$ の結果(同じ)と各反復法で使用するメモリ量を示す。CV法で離散化した行列A(対称)はICCG12のソルバで解き、クランク-ニコルソン法で離散化した行列A(非対称)はBCGSTB12で解いている。あみ目の精しさMJが増えると直接法で使用する行列AやARのメモリは膨大になるが、たとえBCGSTB12を使っても均等あみ目でMJ=109が限界である。

Table 5.4 Iteration Solver and using Memory for MJ(mesh size equally) DF=0.0

MJ	$u_{min}(C)$	$u_{max}(C)$	N	A(MB)	AR(MB)	ICCG12(MB)	BCGSTB12(MB)
5	-0.2880	0.0825	2700	58	2	0.1	0.2
7	-0.2682	0.0792	5320	226	6	0.2	0.4
15	-0.2427	0.0749	24600	4841	59	1	2
24	-0.2347	0.0734	63120	31873	242	3	5
30	-0.2320	0.0729	98700	78000	474	4	7
60	-0.2267	0.0719	395400	1250729	3799	16	29
109	-0.2244	0.0714	1305820	13641326	22763	52	94

非対称行列方程式 $Ax=b$ を解くためにBCG法は、それと双対な方程式 $A^T x^* = b^*$ を組み合わせて、対称行列用のCG法に似せて計算する方法である。BCG法は収束が遅く計算量も増えるので、概ね、同様の不完全LU分解を使ったICCG法の2倍のCPU時間がかかる。したがって、対称行列にはICCG12、非対称行列にはBCGSTB13を使うのが適当である。BCGSTB12/BCGSTB13のCPU時間比は1.2程度である。BCGSTB13の不完全LU

分解で使う行列  $A_0$  の方が BCGSTB12 のそれに比べ A に近いので、収束が速いのである。メモリも BCGSTB13 は、BCGSTB12 より 2 本 : 2N 語増えるだけである。

以上、これらの直接法、反復法に対して、DF=0.0(両端をノイマン境界にする)で均等あみ目(MJ系=MK系)として、それぞれのソルバのCPU時間(sec)を比較した。Table 5.5 に示す。反復法の収束判定値は、ICCGとBCGSTBとも残差がEPSN=1.0D-5より小さくならしたら終わりにする。残差は  $\frac{\|f-Aw\|}{\|f\|}$  とする。Table 5.5を見ると、MJ=109で「CV法(離散化)+ICCG12(ソルバ)」によるCPU時間は56(sec)で求まっている。「クランク-ニコルソン+BCGSTB12」による結果はわざわざ離散化が非対称行列になるので、計算時間も余計に159(sec):3倍かかっている。非対称行列の方が解が収束しにくい(解きづらい)ので、CV法の約3倍も計算時間がかかる。BCGSTB13にすれば、計算時間は124(sec):2.4倍になる。CV法による対称の離散化行列をわざわざBCGSTB12(非対称行列用のソルバ)で解くと、計算時間が52(sec)から145(sec)になり、約3倍余計に計算時間がかかる。これらは、離散化と数値解法(ソルバも含めて)は最適なものを選ばねばならないことを強く示している。あみ目が精しくなるほど、ICCGとBCGSTBとも収束時間がかかる。また、直接法のGLUとGLURとの計算時間比は、あみ目が精しくなるほど開く。MJ=7の場合には、キャッシュにない配列メモリ参照が頻繁に起こるために、約20倍もGLUの方がGLURに比べ速い。

Table 5.5 Solver performance:CPU time(sec) for MJ(mesh size equally) DF=0.0

meth.	MJ	N	ICCG12(s)[MB]	BCGSTB12(s)[MB]	GLUR(s)	GLU(s)	BGLU(s)
CV	30	98700	1 [4]	3 [7]	-	-	-
CV	60	395400	5 [16]	60 [29]	-	-	-
CV	109	1305820	52 [52]	145 [94]	-	-	-
Cr-N	5	2700	×	0. [0.2]	187	24	0.
Cr-N	7	5320	×	0. [0.4]	3520	184	0.
Cr-N	15	24600	×	0. [2]	-	-	2
Cr-N	24	63120	×	1 [5]	-	-	9
Cr-N	30	98700	×	3 [7]	-	-	-
Cr-N	60	395400	×	24 [29]	-	-	-
Cr-N	109	1305820	×	159 [94]	-	-	-

同じDF=0.0のシミュレーションを、メモリ節約のために実務上行なう、ソースのある所など大事な所のあみ目を精しく、その他を粗くする不均等あみ目で行なってみた。不均等あみ目のルールとしては、次のようにソースのある部分のあみ目を精しく、それ以外を周辺に向け段階的に減らすことにする。ここでは、

$$MJ1=MJ2=MKM(\text{基準最大あみ目})$$

$$MK1=MJM=\frac{1}{2}MJ1$$

$$MK=MJ=MJ3=\frac{1}{4}MJ1$$

とする。Table 5.6に、DF=0.0で、不均等あみ目のMJ1(基準最大あみ目)を変えてシミュレーションした場合の各ソルバのCPU時間比較を示す。



Table 5.6 Solver performance:CPU time(sec) for MJ1(Not equally mesh) DF=0.0

meth.	MJ1	N	ICCG12(s)[MB]	BCGSTB12(s)[MB]	GLUR(s)	GLU(s)	BGLU(s)
CV	150	1093652	45 [44]	122 [79]	-	-	-
CV	164	1309950	68 [52]	146 [94]	-	-	-
Cr-N	19	5600	×	0. [0.4]	5456	216	0.
Cr-N	35	57720	×	1 [4]	-	-	9
Cr-N	150	1093652	×	134 [79]	-	-	-
Cr-N	164	1309950	×	204 [94]	-	-	-

Table 5.6 によれば、不均等あみ目にした効果は見てとれる。不均等あみ目ならば、ソースなどの大事な所のあみ目(計算精度に影響する)を精しくできる。不均等あみ目と均等あみ目とにおいて、あみ目の精しさによる計算精度の違いを見るために、Table 5.7にまとめてみた(DF=0.0の場合)。Table 5.7において、MJ1=MJならば均等あみ目を表し、MJ1≠MJならば不均等あみ目を表す。また、BCGSTB13による結果も併記した。CV法により離散化した対称行列Aを、わざわざ非対称行列用のBCGSTB系列でシミュレーションした結果には、これまで通りの記法： を付けた。

Table 5.7 Solver performance and precision for MJ1,MJ(mesh variation) DF=0.0

meth.	MJ1	MJ	N	ICCG12(s)	BCGSTB12(s)	BCGSTB13(s)	$u_{min}(C)$	$u_{max}(C)$
CV	109	109	1305820	52	145	145	-0.2244	0.0714
Cr-N	109	109	1305820	×	159	124	-0.2244	0.0714
Cr-N	150	37	1093652	×	134	129	-0.2241	0.0712
Cr-N	164	41	1309950	×	204	167	-0.2239	0.0712
CV	150	37	1093652	44	122	110	-0.2241	0.0712
CV	164	41	1309950	69	145	145	-0.2239	0.0712

Table 5.7 によれば、不均等あみ目を使えばメモリを節約した上で計算精度を上げることができることが分かる。MJ1=164, MJ= $\frac{1}{4}$ MJ1=41の不均等あみ目の計算精度は、 $u_{min}$ 値において、同程度のメモリを使用する均等あみ目のMJ=109よりも0.2%向上する。不均等あみ目ならば、ソース部分のあみ目を精しくし周辺を粗くして全体の元数を減らしてCPU時間とメモリを節約しても、計算精度は望めることが分かる。MJ1=150, MJ=37で元数を、均等あみ目MJ=109の約2割減らしてCPU時間も同じ約2割速く計算しても、計算精度は、不均等あみ目のMJ1=164を基準にして0.1%の差で求まる(Table 5.8)。

Table 5.8 Solver performance and precision for MJ1,MJ(mesh variation) DF=0.0(Crank-N)

method	MJ1	MJ	N	BCGSTB12(s)	$u_{min}(C)$	$u_{max}(C)$
Crank-N	109	109	1305820 [1.0]	159[1.0]	-0.2244[0.2%]	0.0714
Crank-N	150	37	1093652 [0.8]	134[0.8]	-0.2241[0.1%]	0.0712
Crank-N	164	41	1309950[ $\approx$ 1.0]	204[1.3]	-0.2239 [0.0%]	0.0712

同じ数値シミュレーションを行なうのに、CV法の差分式は物理的に納得のいく対称行列になり対称行列用ソルバICCGが使えて、一方クランク-ニコルソンでは差分式が非対称になり、非対称行列用のソルバBCGSTBで解くしかなく、メモリも余計に必要で計算も断然解きづらく時間もかかる(均等あみ目ならば、CV法もクランク-ニコルソン法も差分式は同じ対称行列にはなる)。Table 5.7は、その数値例を表にまとめたわけである。

以上、拡散系の方程式(1.1)において、物理保存則を満足するように差分化するCV法が、クランク-ニコルソン法よりも、離散化(差分化)、ソルバ(数値解法)の選択においても断然良いことが分かった。

## 参考文献

- [1] 小国 力編著, 村田 健郎、三好 俊郎、ドンガラ J,J、長谷川 秀彦著、行列計算ソフトウェア (WS、スーパーコン、並列計算機), 丸善, pp.252-275, Nov.1991
- [2] 村田 健郎, CV法と、手作り数値シミュレーションシステムの奨め, 日本計算工学会, 計算工学 vol.3, No.1, pp.44-53, 1998
- [3] 村田 健郎, CV法と、手作り数値シミュレーションシステムの奨め (第2回)-非線形純拡散問題と割線反復法-, 日本計算工学会, 計算工学 vol.3, No.3
- [4] 村田 健郎, CV法と、手作り数値シミュレーションシステムの奨め (第3回)-移流拡散系の離散化: 特に指数法について-, 日本計算工学会, 計算工学 vol.3, No.4, pp.246-253, Dec.1998
- [5] 村田 健郎, CV法と、手作り数値シミュレーションシステムの奨め (第4回)-連立非線形移流拡散系: 半導体デバイス解析の場合-, 日本計算工学会, 計算工学 vol.4, No.2, 1999
- [6] 村田 健郎, 「BASIC 数学」連載: 線形数値計算法とその応用, 現代数学社, 1992
- [7] 村田 健郎, 線形代数と線形計算法序節, サイエンス社, 1986
- [8] 小国 力著, MATLAB と利用の実際 [第2版], サイエンス社, 2001
- [9] 村田 健郎, 名取 亮, 唐木 幸比古著, 岩波書店, pp.56-137, 1990
- [10] 砂川重信著, 精講物理, 学生社, pp.342-344, Apr.1975
- [11] 松平升, 大槻義彦, 和田正信共著, 理工教養 物理学 I, 培風館, pp.103-105, Dec.1975