

数値シミュレーションの基礎(3)

青木 孝

神奈川大学 理学部 情報科学科

Introduction to Numerical Simulation(3)

Takashi Aoki

Department of Information Science, Kanagawa University

Abstract. I explain how to Numerical Simulation for Drift Diffusion equation. In this drift diffusion eq. , we will solve unsymmetric matrix.

- Control Volume(CV) method for drift diffusion eq.
- BCG-stab(Biconjugate Gradient method- Stabilized)

1 はじめに

前回の数値シミュレーションの基礎(1)において、板の温度分布 $u = u(x, y)$ を解く熱方程式：

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

を例にとり、一連の数値シミュレーションの手続きを村田流(文献[1])に解説した。その中で、現象のモデル化には、積分形式保存則を満足するように差分化する CV(Control Volume) 法が有用であることや、数値解法(ソルバ)において、ガウスの消去法は理論上は解けるが、メモリの制約、計算時間の問題から実務上は使えない事を示した。熱方程式(拡散系のみ)のような対称行列の場合には、実務上は反復法である ICCG 法に頼らざるを得ないことを実測で示した。

今回は、熱方程式(拡散系： $-\kappa \nabla u$)に移流項($\mu \vec{b}u$)を加えた移流拡散系の方程式を解くための数値シミュレーションの手法を解説する。この移流拡散方程式：

$$(1.2) \quad \operatorname{div}(-\kappa \nabla u + \mu \vec{b}u) = f$$

は、差分化(離散化)した時に、行列が非線形になるため、ICCG 法が使えないので、BCG-Stab 法のようなソルバを使う。

この移流項が入るだけで、方程式は拡散項とのからみで、だんぜん解きづらくなる。離散化方程式が安定に解けるように仕向けるために、メッシュサイズが関係する(セルペクレ数として知られる)など、やっかいである。この対策の決定版として指数法があり、半導体デバイス解析の分野では、Shafetter-Gummel 法として呼ばれている。

2 CV 法による熱方程式(純拡散)の離散化

この部分は前回「数値シミュレーションの基礎(1)」と同じである。Fig.1 のシミュレーション場の点 i の i 点回りに、小領域 PQRS(Control Volume という)を作り(Fig.2)、その

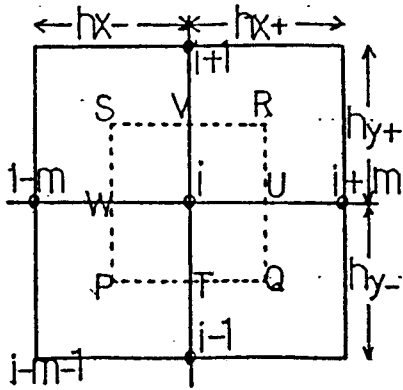


Fig.2. Mesh for CV method

周囲を Γ に見立てて、熱方程式(積分形):

$$(2.1) \quad \int_{\Gamma} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{\Omega} f \, dv$$

の CV 法による離散式を作る。左辺の線積分は次式のように細分化できる。

$$(2.2) \quad \int_{\Gamma} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{PQ} + \int_{QR} + \int_{RS} + \int_{SP}$$

右辺の面積分は次式として始末し、節点 i での f_i を求める。

$$(2.3) \quad f_i = \int_{\Omega} f \, dv = f_P \frac{hx_-}{2} \frac{hy_-}{2} + f_Q \frac{hx_+}{2} \frac{hy_-}{2} + f_R \frac{hx_+}{2} \frac{hy_+}{2} + f_S \frac{hx_-}{2} \frac{hy_+}{2} \quad \left[\frac{cal}{sec} \right]$$

最終的に、左辺を、 u_{i-m} , u_{i-1} , u_i , u_{i+1} , u_{i+m} 項毎に整理すれば、第 i 番方程式は、

$$(2.4) \quad a_{i,i-m} u_{i-m} + a_{i,i-1} u_{i-1} + a_{i,i} u_i + a_{i,i+1} u_{i+1} + a_{i,i+m} u_{i+m} = f_i$$

となり、左辺の各係数は、次式となる(前回と同じ)。

$$(2.5) \quad \begin{aligned} a_{i,i} &= \frac{1}{2} \left[d_P \left(\frac{hx_-}{hy_-} + \frac{hy_-}{hx_-} \right) + d_Q \left(\frac{hx_+}{hy_-} + \frac{hy_-}{hx_+} \right) + d_R \left(\frac{hx_+}{hy_+} + \frac{hy_+}{hx_+} \right) + d_S \left(\frac{hx_-}{hy_+} + \frac{hy_+}{hx_-} \right) \right] \\ a_{i,i-m} &= -\frac{1}{2} \left[d_S \frac{hy_+}{hx_-} + d_P \frac{hy_-}{hx_-} \right] \\ a_{i,i-1} &= -\frac{1}{2} \left[d_P \frac{hx_-}{hy_-} + d_Q \frac{hx_+}{hy_-} \right] \\ a_{i,i+1} &= -\frac{1}{2} \left[d_R \frac{hx_+}{hy_+} + d_S \frac{hx_-}{hy_+} \right] \\ a_{i,i+m} &= -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} + d_R \frac{hy_+}{hx_+} \right] \end{aligned}$$

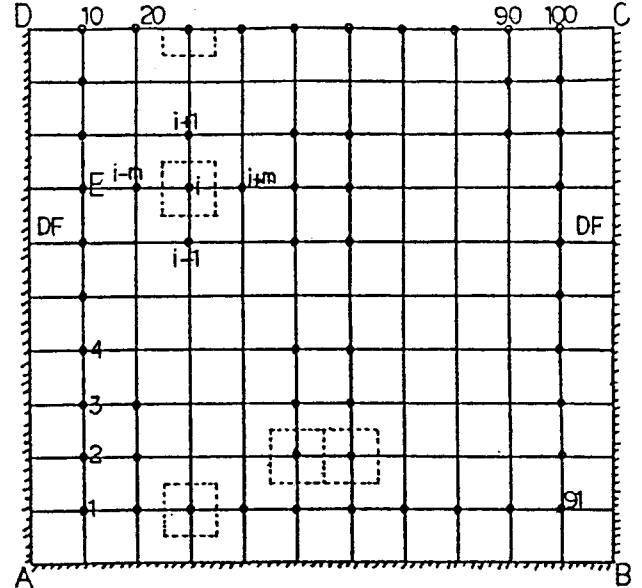


Fig.1. Simulation Field

かくして、 $A(a_{k,j})U(u_k) = F(f_k)$ の $n \times n$ 帯行列方程式 (対称) ができる (Fig.3)。Fig.1 の通し番号の節点 $i (= 1 \rightarrow 100)$ は、

$$i = m \times (j - 1) + k$$

($j = 1, m1, k = 1, m$; $m1$ は x 方向の点の数) などのように指定する。Fig.1 の場合は、はなはだ簡単に、 $hx = hx_+ = hx_- = 1[cm]$, $hy = hy_+ = hy_- = 1[cm]$ である。今、 $d_P = d_Q = d_R = d_S = 1[\frac{cal}{C \cdot sec \cdot cm}]$ とすれば、式 (2.5) はそれぞれ、 $a_{i,i} = 4.0$

$$a_{i,i-m} = -1.0 \quad a_{i,i-1} = -1.0 \quad a_{i,i+1} = -1.0 \\ a_{i,i+m} = -1.0 \text{ となる。}$$

Fig.1 の両側の伝導率 κ には、前回と同様な段差をつける。MJ=1 の場合には、

$$i = 1 \rightarrow 10 \quad \kappa_i(x, y) = DF(\text{一定値})$$

$$i = 101 \rightarrow 110 \quad \kappa_i(x, y) = DF(\text{一定値})$$

それ以外は、 $\kappa_i(x, y) = 1.0$ とする。

u_i は節点 1~100 までだが、 κ_i は 1~110 までであることに注意。この変数 DF を、 $DF = 1.0[\frac{cal}{C \cdot sec \cdot cm}]$ にすれば、 $\kappa = 1$ 一定の固定境界、 $DF = 0.0$ にすれば、両サイドはノイマン境界 ($-\kappa \nabla u = 0$) と同じになる。

右辺の熱源も、前回と同様に与える。MJ=1 の場合には、熱源を節点 i に対して、

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

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

とする。MJ が 2 以上の場合には、MJ=1 の時の点内にある補間点には同様の熱源を与える。プログラム上では、本来熱源密度は面に対して与えるべきであるが、簡単のために、各点 i の f_i にじかに格納している。以下その部分。

```
do j=5*MJ-1, 6*MJ-1
do i=0, 2*MJ
F(M*J+2*MJ+I)=0.2*hx*hy
F(M*J+6*MJ+I)=-0.2*hx*hy
```

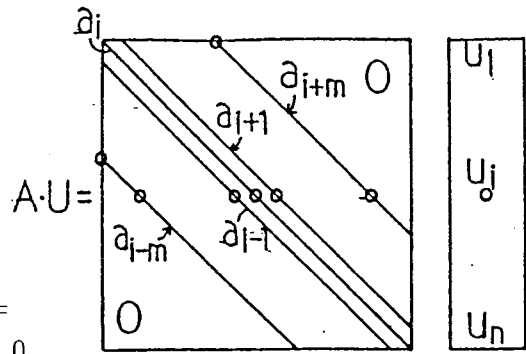


Fig.3. BandMatrix

3 移流拡散系の解法

これまでの純拡散系の方程式：

$$(3.1) \quad \text{div}(-\kappa \nabla u) = f$$

に、いわば温度 u に比例して熱が流れていくという移流項 ($+\mu \vec{b}u$) を加えた移流拡散系の方程式：

$$(3.2) \quad \text{div}(-\kappa \nabla u + \mu \vec{b}u) = f$$

の解法を考える。式 (3.2) において、解を簡単に左右対称とするために、移流ベクトル \vec{b} を、

$$\vec{b} = (0, 1)^T = (bx, by)^T \quad (b \text{ の矢印は注意のため})$$

とする。また、拡散係数 $\kappa(x, y)$ と移動度 $\mu(x, y)$ の比： $\frac{\mu}{\kappa} = C_0$ を一定値 C_0 とする。これは半導体デバイスの世界では、格子温度一定ならば、Einsteinの関係として知られる。半導体内キャリア(電子など)の移流拡散では、 $\vec{b} = \nabla\psi$ (ψ は電位)となる。

はじめに、CV法により式(4.2)の積分形(次式)を前回同様に、そぼくな中心差分によって離散化する。

$$(3.3) \quad \int_{\Gamma} (-\kappa \nabla u + \mu \mathbf{b} u) \cdot \mathbf{n} \, ds = \int_{\Omega} f \, dv$$

Fig.2の小領域PQRSの回りに、左辺は、

$$(3.4) \quad \int_{\Gamma} (-\kappa \nabla u + \mu \mathbf{b} u) \cdot \mathbf{n} \, ds = \int_{PQ} + \int_{QR} + \int_{RS} + \int_{SP}$$

となり、

$$(3.5) \quad \begin{aligned} \int_{PQ} &\doteq \left(-d_P \left(\frac{u_i - u_{i-1}}{hy_-} \right) + \mu_P \cdot b_y \left(\frac{u_i + u_{i-1}}{2} \right) \right) (-1) \frac{hx_-}{2} \\ &+ \left(-d_Q \left(\frac{u_i - u_{i-1}}{hy_-} \right) + \mu_Q \cdot b_y \left(\frac{u_i + u_{i-1}}{2} \right) \right) (-1) \frac{hx_+}{2} \quad \text{今は } by = 1 \end{aligned}$$

などとして書き下す。ここで、T点での $\frac{\partial u}{\partial y}$ は $\frac{u_i - u_{i-1}}{hy_-}$ 、同様に u_i は $\frac{u_i + u_{i-1}}{2}$ 、W点での $\frac{\partial u}{\partial x}$ は $\frac{u_i - u_{i-m}}{hx_-}$ 、同様に u_i は $\frac{u_i + u_{i-m}}{2}$ 、などと、各点での u_i を使って差分している。また、伝導度 $\kappa(x, y)$ と移動度 $\mu(x, y)$ は、メッシュで分けした面内は一定と考えて差分する。たとえば、P点での $\kappa(x, y) \equiv d_P$ および、 $\mu(x, y) \equiv \mu_P$ は、4点 $i, i-m, i-m-1, i-1$ が囲む面内一定として、プログラムレベルでは $DF(i)$ に格納するなど工夫する。配列 DF の範囲は、 $DF(N+M)$ まで必要となる。 \mathbf{b} は今、簡単のため場所に寄らない設定： $\mathbf{b} = (b_x, b_y) = (0, 1)$ とするが、あえて b_x, b_y は残しておく。

式(3.5)のように、式(3.4)の右辺4項全部を書き下だし、 $u_{i-m}, u_{i-1}, u_i, u_{i+1}, u_{i+m}$ 項毎に整理すれば、第 i 番(格子点番号)方程式：

$$(3.6) \quad a_{i,i-m} u_{i-m} + a_{i,i-1} u_{i-1} + a_{i,i} u_i + a_{i,i+1} u_{i+1} + a_{i,i+m} u_{i+m} = f_i$$

ができる。この時、式(3.6)の左辺の係数は、次式となる。

$$\begin{aligned} a_{i,i} &= \frac{1}{2} \left[d_P \frac{hx_-}{hy_-} \left(1 - \frac{\mu_P b_y hy_-}{2d_P} \right) + d_Q \frac{hx_+}{hy_-} \left(1 - \frac{\mu_Q b_y hy_-}{2d_Q} \right) \right] \\ &+ \frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} \left(1 + \frac{\mu_Q b_x hx_+}{2d_Q} \right) + d_R \frac{hy_+}{hx_+} \left(1 + \frac{\mu_R b_x hx_+}{2d_R} \right) \right] \\ &+ \frac{1}{2} \left[d_R \frac{hx_+}{hy_+} \left(1 + \frac{\mu_R b_y hy_+}{2d_R} \right) + d_S \frac{hx_-}{hy_+} \left(1 + \frac{\mu_S b_y hy_+}{2d_S} \right) \right] \\ &+ \frac{1}{2} \left[d_S \frac{hy_+}{hx_-} \left(1 - \frac{\mu_S b_x hx_-}{2d_S} \right) + d_P \frac{hy_-}{hx_-} \left(1 - \frac{\mu_P b_x hx_-}{2d_P} \right) \right] \\ a_{i,i-m} &= -\frac{1}{2} \left[d_S \frac{hy_+}{hx_-} \left(1 + \frac{\mu_S b_x hx_-}{2d_S} \right) + d_P \frac{hy_-}{hx_-} \left(1 + \frac{\mu_P b_x hx_-}{2d_P} \right) \right] \end{aligned}$$

$$\begin{aligned}
a_{i,i-1} &= -\frac{1}{2} \left[d_P \frac{hx_-}{hy_-} \left(1 + \frac{\mu_P b_y hy_-}{2d_P} \right) + d_Q \frac{hx_+}{hy_-} \left(1 + \frac{\mu_Q b_y hy_-}{2d_Q} \right) \right] \\
a_{i,i+1} &= -\frac{1}{2} \left[d_R \frac{hx_+}{hy_+} \left(1 - \frac{\mu_R b_y hy_+}{2d_R} \right) + d_S \frac{hx_-}{hy_+} \left(1 - \frac{\mu_S b_y hy_+}{2d_S} \right) \right] \\
(3.7) \quad a_{i,i+m} &= -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} \left(1 - \frac{\mu_Q b_x hx_+}{2d_Q} \right) + d_R \frac{hy_+}{hx_+} \left(1 - \frac{\mu_R b_x hx_+}{2d_R} \right) \right]
\end{aligned}$$

ここで、係数 $a_{i,t}$ は、行列 $\mathbf{A}(a_{i,t})$ の要素と対応しており、行 i は Fig.1 の場の格子点番号 $i (= m \times (j-1) + k; j = 1, m1, k = 1, m)$ を表わす。Fig.1 では、 \mathbf{A} の行列サイズは： $n = m1(x \text{ 方向}) \times m(y \text{ 方向})$ となり、均等あみ目の場合： $hx_{\pm} = hy_{\pm} = \frac{1}{MJ}(MJ \text{ は } 1cm \text{ の分割数})$ の時には、 $m = MJ \times 10$, $m1 = MJ \times 11 - 1$ である。なお、 \mathbf{b} が場に寄る量であれば、例えば $\mu_Q b_x hx_+$ の b_x は、 $b_x^U = \frac{b_{i+m} - b_i}{hx_+}$ として、 $\mu_Q b_x^U hx_+$ に置き変わる。

今、 $\frac{h}{a} = C_0$ 、等間隔あみ目： $hx_{\pm} = hy_{\pm} = h$ として、さらに $(bx, by) = (0, 1)$ を使うと、各係数は次のように簡単になる。

$$a_{i,i} = 4.0, \quad a_{i,i-m} = -1.0, \quad a_{i,i-1} = -\frac{1}{2}[2 + C_0 h], \quad a_{i,i+1} = -\frac{1}{2}[2 - C_0 h], \quad a_{i,i+m} = -1.0$$

となる。問題を簡単にするのは、 $hx = hy = 1[cm]$ とすることより、むしろ、あみ目を均等： $hx = hy$ とすることの方が主役である。

これら係数も、上側のノイマン境界の節点 (CD 上) では、左辺の積分区間を Fig.2 の下半分だけにする必要があり、扱いが異なる。線積分は、

$$(3.8) \quad \int_{\Gamma_{CD}} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{WP} + \int_{PQ} + \int_{QU}$$

となり、CD 上の節点では、係数は、式 (3.7) と同様の手順を経て次式となる。

$$\begin{aligned}
a_{i,i} &= \frac{1}{2} \left[d_P \frac{hx_-}{hy_-} \left(1 - \frac{\mu_P b_y hy_-}{2d_P} \right) + d_Q \frac{hx_+}{hy_-} \left(1 - \frac{\mu_Q b_y hy_-}{2d_Q} \right) \right] \\
&\quad + \frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} \left(1 + \frac{\mu_Q b_x hx_+}{2d_Q} \right) \right] \\
&\quad + \frac{1}{2} \left[d_P \frac{hy_-}{hx_-} \left(1 - \frac{\mu_P b_x hx_-}{2d_P} \right) \right] \\
a_{i,i-m} &= -\frac{1}{2} \left[d_P \frac{hy_-}{hx_-} \left(1 + \frac{\mu_P b_x hx_-}{2d_P} \right) \right] \\
a_{i,i-1} &= -\frac{1}{2} \left[d_P \frac{hx_-}{hy_-} \left(1 + \frac{\mu_P b_y hy_-}{2d_P} \right) + d_Q \frac{hx_+}{hy_-} \left(1 + \frac{\mu_Q b_y hy_-}{2d_Q} \right) \right] \\
a_{i,i+1} &= 0.0 \\
(3.9) \quad a_{i,i+m} &= -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} \left(1 - \frac{\mu_Q b_x hx_+}{2d_Q} \right) \right]
\end{aligned}$$

また、Fig.1 の両側の伝導率 κ には段差をつけている ($\kappa = DF$) ので、その始末が必要となる。 $\frac{\mu}{DF} = C_0$ となる。その時、MJ=1 で均等あみ目の場合には、 $i = 1 \rightarrow 9$ の各係数は、

$$(3.10) \quad \begin{aligned} a_{i,i} &= \frac{1}{2} [DF \cdot 4 + 4] \\ a_{i,i-m} &= 0.0, \quad a_{i,i-1} = -\frac{1}{2} \left[DF \left(1 + \frac{C_0 h}{2}\right) + \left(1 + \frac{C_0 h}{2}\right) \right] \\ a_{i,i+1} &= -\frac{1}{2} \left[\left(1 - \frac{C_0 h}{2}\right) + DF \left(1 - \frac{C_0 h}{2}\right) \right], \quad a_{i,i+m} = -1.0 \end{aligned}$$

となる。ここで、節点 i に隣あう点 $i-m$ には、固定境界があるので、違った扱いをする必要がある。例えば、左側に固定境界 $u = u_0 (= 0)$ を持つ E 点 (Fig.1) の場合には、 i 番方程式が、

$$(3.11) \quad a_{i,i-m} u_0 + a_{i,i-1} u_{i-1} + a_{i,i} u_i + a_{i,i+1} u_{i+1} + a_{i,i+m} u_{i+m} = f_i$$

となる。この時、未知数項でない $a_{i,i-m} u_0$ は、右辺に移項して f_i の仲間に入れる。したがって、

$$(3.12) \quad a_{i,i-1} u_{i-1} + a_{i,i} u_i + a_{i,i+1} u_{i+1} + a_{i,i+m} u_{i+m} = f_i - a_{i,i-m} u_0$$

を解くことになるので、式(3.10)において係数 $a_{i,i-m}$ は、

$$a_{i,i-m} = 0.0$$

と設定してある。 $u_0 = 0$ ならば、なおさら簡単で、右辺に手を加える必要はなくなる。

右辺の熱源の始末は、

$$f_i = \int_{\Omega} f \, dv$$

の面積分になる。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る。伝導率 κ の場合のように、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、 $f_P \left[\frac{\text{cal}}{\text{sec} \cdot \text{cm}^3} \right]$ とすれば、 $\frac{1}{4}$ 面 (iWPT) に当たる f_i への寄与は、 $f_P \frac{hx_-}{2} \frac{hy_-}{2} \left[\frac{\text{cal}}{\text{sec}} \right]$ (厚さが1[cm]あると思う)となる。

したがって、節点 i での f_i は、

$$(3.13) \quad f_i = f_P \frac{hx_-}{2} \frac{hy_-}{2} + f_Q \frac{hx_+}{2} \frac{hy_-}{2} + f_R \frac{hx_+}{2} \frac{hy_+}{2} + f_S \frac{hx_-}{2} \frac{hy_+}{2} \quad \left[\frac{\text{cal}}{\text{sec}} \right]$$

と求まる。 $hx = hy = 1$ ならば、 $f_i = \frac{1}{4}(f_P + f_Q + f_R + f_S)$ 。

当然、上側のノイマン境界上の節点 (CD 上) では、積分区間が下半分しかないので、

$$(3.14) \quad f_{i \text{ CD}} = f_P \frac{hx_-}{2} \frac{hy_-}{2} + f_Q \frac{hx_+}{2} \frac{hy_-}{2} \quad \left[\frac{\text{cal}}{\text{sec}} \right]$$

となる。熱源 f_i のシミュレーション条件の与え方は、純拡散の時と同じにする。ここまでは、移流項が加わった事を除けば、手順は純拡散の熱方程式 (1.1) の時と同じ。

こうして、積分形の移流拡散方程式 (3.3) を、保存則を満足するように離散化 (CV 法) した、 i 番方程式は、 $u_{i-m}, u_{i-1}, u_i, u_{i+1}, u_{i+m}$ 項を持つ、非対称帯行列になっている。ここで、離散化方程式 (i 番方程式) が安定に解けるように仕向けるために次の条件を課す。

すべての節点*i*につき、 $a_i > 0$ かつ $a_{i\pm 1} \leq 0$, $a_{i\pm m} \leq 0$

となるようにせよ。

そのためには、係数 a に現れる、すべての $1 \pm \frac{\mu b h}{2d}$ の形の式が > 0 すなわち、

$$(3.15) \quad \left| \frac{\mu b_x h_x}{d} \right| \leq 2, \quad \left| \frac{\mu b_y h_y}{d} \right| \leq 2$$

が成立するように、あみ目を細かくすればよい。この x, y 方向の $\left| \frac{\mu b_x h_x}{d} \right|, \left| \frac{\mu b_y h_y}{d} \right|$ を、それぞれ x, y 方向のセルペクレ数と言う。このセルペクレ数を 2 以下におさえないと、トラブルが生じる事が昔から知られている（非対称の場合の問題の 1 つ）。このシミュレーションでは、均等あみ目の場合に、セルペクレ数が、

$$\left| C_0 h_x \right|, \left| C_0 h_y \right|, \text{ または } \left| \frac{C_0}{MJ} \right|, \left| \frac{C_0}{MJ} \right|$$

となるので、 $C_0 < 2 \cdot MJ$ (ここで $MJ \geq 1$) に設定すれば、問題なしとなる。しかし、セルペクレ数が 2 を大巾に超える場合は、精度的に不安となる。そのような状況の下では、いわば決定版として、差分にこれまでの中心差分ではなく、指数法を用いなければならない。これは、半導体デバイス解析の分野では、Sharfetter-Gunnell 法と呼ばれる。この指数法については、後述する。指数法の説明の中で、中心差分の位置付けを述べるとともに、セルペクレ数が 2 を超えた場合の実害を見ることにする。しばらくは、中心差分のままいく。

係数を見ると、

$$a_{i,i} = -(a_{i,i-1} + a_{i,i+1} + a_{i,i-m} + a_{i,i+m})$$

すなわち、行列は、行対角優位になっている事が分かる。行対角優位の場合は、

A が正則 \Rightarrow 軸選択なしのガウスによって

終りまで LU 分解ができる \rightarrow ガウスで解ける。

軸選択がないと、演算量は半分、メモリは $\frac{2}{3}$ ですむ。ソルバとしては、軸選択なしの帯ガウス (対称でない) で良いことになる。プログラムレベルでは、サブルーチン：

BGLU (左辺の前進消去)、 BGSLV (右辺の前進消去と後退代入)

が受け持つ。配列は、

$$a_{i,i} : AA(i), a_{i,i+1} : AB(i), a_{i,i+m} : AC(i), a_{i,i-1} : AB1(i), a_{i,i-m} : AC1(i)$$

に対応している。以下に、プログラムリストを示す。

```

IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 CPUT
PARAMETER (MJ=1, M=MJ*10, M2=MJ*11-1, N=M*M2, EPS=1.0D-7)
DIMENSION AR (-M:M, N), AA (N), AB (-M:N), AC (-M:N), F (N), WK (M)
+, AB1 (-M:N+M), AC1 (-M:N+M)
+, FQ (N+M)
C
READ (5, *) DF
WRITE (6, *) ' % DF= ', DF
WRITE (6, *) ' % M= ', M, ' M2= ', M2, ' N= ', N
FMJ= DFLOAT (MJ)
DHX=1.0D0/FMJ
DHY=1.0D0/FMJ
CC
C0=-0.5D0
WRITE (6, *) ' % C0= ', C0
DH=1.0D0/FMJ
COH=C0*DH
COH2=0.5D0*COH

```

```

IF(DF .EQ. 0.0D0) THEN
  COH2=0.0D0
ENDIF
DO 10 I=1,M-1
  AA(I)= 0.5D0*(DF*4.0D0+4.0D0)
  AB(I)=-0.5D0*((1.0D0-COH2)+DF*(1.0D0-COH2))
  AC(I)=-0.5D0*2.0D0
  AB1(I)=-0.5D0*(DF*(1.0D0+COH2)+(1.0D0+COH2))
  AC1(I)=0.0D0
10 CONTINUE
  AB1(1)=0.0D0
*
  AA(M)= 0.5D0*(DF*(2.0D0-COH2)+(2.0D0-COH2))
  AB(M)=0.0D0
  AC(M)=-0.5D0*1.0D0
  AB1(M)=-0.5D0*(DF*(1.0D0+COH2)+(1.0D0+COH2))
  AC1(M)=0.0D0
C
  COH2=0.5D0*COH
  DO 30 J=1,M2-2
    DO 20 K=1,M-1
      AA(M*J+K)=4.0D0
      AB(M*J+K)=-0.5D0*(2.0D0-COH)
      AC(M*J+K)=-0.5D0*2.0D0
      AB1(M*J+K)=-0.5D0*(2.0D0+COH)
      AC1(M*J+K)=-0.5D0*2.0D0
20 CONTINUE
      AB1(M*J+1)=0.0D0
*
      AA(M*J+M)= 0.5D0*(4.0D0-COH)
      AB(M*J+M)=0.0D0
      AC(M*J+M)=-0.5D0*1.0D0
      AB1(M*J+M)=-0.5D0*(2.0D0+COH)
      AC1(M*J+M)=-0.5D0*1.0D0
30 CONTINUE
C
  IF(DF .EQ. 0.0D0) THEN
    COH2=0.0D0
  ENDIF
  DO 40 I=N-M+1, N-1
    AA(I)= 0.5D0*(DF*4.0D0+4.0D0)
    AB(I)=-0.5D0*(DF*(1.0D0-COH2)+(1.0D0-COH2))
    AC(I)=0.0D0
    AB1(I)=-0.5D0*((1.0D0+COH2)+DF*(1.0D0+COH2))
    AC1(I)=-0.5D0*2.0D0
40 CONTINUE
    AB1(N-M+1)=0.0D0
*
    AA(N)= 0.5D0*(DF*(2.0D0-COH2)+(2.0D0-COH2))
    AB(N)=0.0D0
    AC(N)=0.0D0
    AB1(N)=-0.5D0*((1.0D0+COH2)+DF*(1.0D0+COH2))
    AC1(N)=-0.5D0*1.0D0
C
    DO 50 I=1,N
      AR(0,I)=AA(I)
      AR(1,I)=AB(I)
      AR(M,I)=AC(I)
*
      AR(-1,I)=AB1(I)
      AR(-M,I)=AC1(I)
50 CONTINUE
C
*C Uhen ***

```



```

DO 60 J= 5*MJ-1, 6*MJ-1
  DO 70 I=0,2*MJ
    F(M*J +2*MJ+I)= 0.2D0*(DHX*DHY)
    F(M*J +6*MJ+I)=-0.2D0*(DHX*DHY)
70  CONTINUE
60  CONTINUE
C
  CALL CLOCK0
C
*C  AR*U=F wo Toku; Kotae --> F ***
  CALL BGLU( N, M, AR, EPS, IR, WK )
  CALL BGSLV( N, M, AR, F )
C
  CALL CLOCK(CPUT)
  UMIN=F(1)
  JUMIN=1
  KUMIN=1
  UMAX=F(1)
  JUMAX=1
  KUMAX=1
  DO 130 J=1,M2
    DO 120 K=1,M
      I=M*(J-1)+K
      IF(F(I).LT.UMIN) THEN
        UMIN=F(I)
        JUMIN=J
        KUMIN=K
      ENDIF
      IF(F(I).GT.UMAX) THEN
        UMAX=F(I)
        JUMAX=J
        KUMAX=K
      ENDIF
120  CONTINUE
130  CONTINUE
C
  WRITE(6,*) ' % MJ= ',MJ
  WRITE(6,*) ' % CPUT= ',CPUT
  WRITE(6,*) ' % UMIN, (x,y)= ',UMIN,' (',JUMIN,',',KUMIN,')'
  WRITE(6,*) ' % UMAX, (x,y)= ',UMAX,' (',JUMAX,',',KUMAX,')'
C
  WRITE(6,*) ' z=[ '
C
CC  DO 80 K= M,MJ,-MJ
CC  WRITE(6,2000) (F(M*J+K),J=MJ-1,MJ*10-1,MJ), ' ; '
CC 80 CONTINUE
*C  MATLAB You ***
  DO 80 K= MJ,M,MJ
    WRITE(6,2000) (F(M*J+K),J=MJ-1,MJ*10-1,MJ), ' ; '
 80  CONTINUE
2000 FORMAT(1H ,10F6.2,A3)
C
  WRITE(6,*) ' ] ; '
  WRITE(6,*) ' contour(0:1:9, 0:1:9, z,10);'
C
  STOP
  END
CCC
  SUBROUTINE BGLU ( N, M, AR, EPS, IR, WK )
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION AR(-M:M,N), WK(M)
C
C forward elimination for A
  IR = 0
  DO 100 K = 1, N

```

```

C
      IF( ABS(AR(0,K)) .LE. EPS ) THEN
          IR = IR+1
          RETURN
      END IF
C
C *****
      DO 125 J = K+1, MIN(K+M,N)
125      WK(J-K) = AR(J-K,K)
          *****
C *****
C
      DO 130 I = K+1, MIN(K+M,N)
          AR(K-I,I) = -AR(K-I,I)/AR(0,K)
          T = AR(K-I,I)
          DO 140 J = K+1, MIN(K+M,N)
140      AR(J-I,I) = AR(J-I,I)+T*WK(J-K)
          *****
C
130      CONTINUE
C
100 CONTINUE
      RETURN
      END
C
      SUBROUTINE BGLV ( N, M, AR, B )
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AR(-M:M,N), B(N)
          ****
C forward elimination for B
      DO 100 K = 1, N
C
          T = B(K)
          DO 110 I = K+1, MIN(K+M,N)
110      B(I) = B(I)+AR(K-I,I)*T
100 CONTINUE
C
C backward substitution for B
      DO 200 K = N, 1, -1
          S = B(K)
          DO 210 J = K+1, MIN(K+M,N)
210      S = S-AR(J-K,K)*B(J)
          B(K) = S/AR(0,K)
200 CONTINUE
      RETURN
      END

```

プログラム上で、 DF (Fig.1 の両側の伝導率 κ)= 0.0 と指定した時には、両側をノイマン境界 (flux=0: $-\kappa\nabla u + \mu\vec{b}u = 0$) とするために、

$i = 1 \rightarrow m$ では $C_0 = (\frac{\mu}{\kappa}) = 0.0$, $i = n - m + 1 \rightarrow n$ では $C_0 = 0.0$ と設定する。

MJ=20 の時に、対称行列となる: $div(-\kappa\nabla u) = f$ の行列方程式を、対称帯ガウス (BSGLU) と非対称帯ガウス (BGLU) を使って解くと、それぞれ計算 (CPU) 時間は次のようになる。

Table 1 CPU time performance for Solver(BSGLU, BGLU)

MJ	CPU time (s)
20	[BSGLU] Symmetric, m=200, N=43800, Memory=70MB: 9
20	[BGLU] unSymmetric, m=200, N=43800, Memory=140MB: 79

非対称帯ガウス (BGLU) では、対称帯ガウス (BSGLU) に比べ、当然メモリは2倍必要で、かかる CPU 時間は8倍強になる。

4 BCG-Stab 法

帯ガウスの計算量、メモリ上の限界 (デバッグには必要だが) は、熱方程式で見た通りであるので、反復法に頼らざるを得ない。帯ガウスの行列: AR(-M:M,N) (熱方程式では AR(0:M,N)) は、MJ=20 でメモリ 140MB 必要となり、MJ=25 では主メモリ 256MB の計算機では解けない。したがって、ICCG 法と似て A の不完全 LU 分解を利用し、Bi-CG(双共役勾配法) を改良した Bi-CGSTAB(BCG-Stab) 法を、反復法用のソルバーとして推奨し (文献 [4]) 使う。BCG-Stab 法の基本算法を Fig.4 に示す。実際には A の不完全 LU を K として、 $K^{-1}Ax = K^{-1}b$ を解く。それには上記 A のところを $K^{-1}A$ に変え、また γ_0 を $K^{-1}(b - Ax_0)$ としてスタートする。基本算法に基づくサブルーチン BCGSTB2 (=BCG-Stab (1,2)) は次のようになる。記法 (1,2) は不完全 LU 分解に起因しており、L,U の要素として、それぞれに1本ずつ多く使う所からきている。精しくは前回の「数値シミュレーションの基礎(1)」を参照のこと。

Bi-CGSTABの算法: $AX=b$ を解くのに、
 x_0 を初期解とし
 $\gamma_0 = b - Ax_0; \gamma_0^* = \gamma_0; P_0 = \gamma_0; n = 0$
while $\|\gamma_{n+1}\| < \text{eps} * \|b\|$ do
 $\alpha_n = (\gamma_0^*, \gamma_n) / (\gamma_0^*, Ap_n)$
 $e_n = \gamma_n - \alpha_n Ap_n$
 $\zeta_n = (Ae_n, e_n) / (Ae_n, Ae_n)$
 $x_{n+1} = x_n + \alpha_n p_n + \zeta_n e_n$
 $\gamma_{n+1} = e_n - \zeta_n Ae_n$
 $\beta_n = (\alpha_n \zeta_n) \cdot (\gamma_0^*, \gamma_{n+1}) / (\gamma_0^*, \gamma_n)$
 $P_{n+1} = \gamma_{n+1} + \beta_n \cdot (P_n - \zeta_n Ap_n); n = n + 1$

Fig.4. Algorithm for Bi-CGSTAB

```

SUBROUTINE BCGSTB2 (AA, AB, AC, AE, UB, DI, AB1, AC1, AE1, UB1, P, V, W, B0
+ , R0, X, AP, EPSLON, ERR, M1, N, KCOUNT, BNRM, AEE, EE, RR)
IMPLICIT REAL*8 (A-H, O-Z)
DIMENSION AA(N), AB(-M1:N), AC(-M1:N), DI(-M1:N)
+ , P(-M1:N+M1), W(-M1:N+M1), B0(N), X(-M1:N+M1)
+ , AP(N), V(-M1:N+M1), UB(-M1:N), AE(-M1:N)
+ , AB1(-M1:N+M1), AC1(-M1:N+M1), UB1(-M1:N+M1), AE1(-M1:N+M1)
+ , AEE(-M1:N+M1), EE(-M1:N+M1), RR(-M1:N+M1), R0(N)
*
DO 10 I = 1, N
10 V(I) = B0(I) - (AA(I)*X(I) + AB(I)*X(I+1) + AC(I)*X(I+M1)
+ AB1(I)*X(I-1) + AC1(I)*X(I-M1))
*** KOKO DE WA V = B - AX DE ARU ***
DO 30 I = 1, N
30 W(I) = (V(I) - AC1(I)*W(I-M1) - UB1(I)*W(I-1) - AE1(I)*W(I-M1+1))*DI(I)
DO 40 I = N, 1, -1
40 W(I) = W(I) - DI(I)*(UB(I)*W(I+1) + AC(I)*W(I+M1) + AE(I)*W(I+M1-1))
*** KOKO DE WA W = INV(LU)*(B-AX) DE ARU *****
CR = 0.0D0
DO 60 I = 1, N
P(I) = W(I)
RR(I) = W(I)
R0(I) = W(I)
60 CR = CR + W(I)*W(I)
*** ITERATION; KONO ATO, V O V, W O Q TO SHITE TSUKAU ***

```

```

DO 1000 K = 1, 400
  DO 100 I = 1, N
    AP(I)=AA(I)*P(I)+AB(I)*P(I+1)+AC(I)*P(I+M1)
    +
    +AB1(I)*P(I-1)+AC1(I)*P(I-M1)
  100 CONTINUE
  DO 110 I = 1, N
  110 W(I)=(AP(I)-AC1(I)*W(I-M1)-UB1(I)*W(I-1)-AE1(I)*W(I-M1+1))*DI(I)
  DO 120 I = N, 1, -1
  120 W(I)=W(I)-DI(I)*(UB(I)*W(I+1)+AC(I)*W(I+M1)+AE(I)*W(I+M1-1))
  ***      W = INV(LU)*AP DE ARU      ***
  SIG= 0.0D0
  DO 130 I=1, N
  130 SIG = SIG+R0(I)*W(I)
  ALFA = CR/SIG
*
  DO 140 I=1,N
  140 EE(I)= RR(I)-ALFA*W(I)
*
*
  DO 150 I = 1, N
  150 AEE(I)= AA(I)*EE(I)+AB(I)*EE(I+1)+AC(I)*EE(I+M1)
  +
  +AB1(I)*EE(I-1)+AC1(I)*EE(I-M1)
  DO 160 I = 1, N
  160 V(I)=(AEE(I)-AC1(I)*V(I-M1)-UB1(I)*V(I-1)-AE1(I)*V(I-M1+1))*DI(I)
  DO 170 I = N, 1, -1
  170 V(I)=V(I)-DI(I)*(UB(I)*V(I+1)+AC(I)*V(I+M1)+AE(I)*V(I+M1-1))
  ***      V = INV(LU)*A*E DE ARU      ***
  EEV= 0.0D0
  VV= 0.0D0
  DO 180 I= 1,N
  EEV= EEV+EE(I)*V(I)
  180 VV = VV +V(I)*V(I)
  OMEG =EEV/VV
*
  DO 190 I= 1, N
  X(I)= X(I)+ALFA*P(I)+OMEG*EE(I)
  190 RR(I)=EE(I)-OMEG*V(I)
*
  CR=0.0D0
  ERR = 0.0D0
  DO 200 I = 1, N
  CR = CR+ R0(I)*RR(I)
  200 ERR = ERR+RR(I)*RR(I)
  ERR = DSQRT(ERR)/BNRM
  IF( ERR.LT.EPSLON ) GO TO 2000
  ***** KOKOMADE *****
  BETA = CR/(OMEG*SIG)
  DO 300 I = 1, N
  300 P(I)= RR(I)+BETA*(P(I)-OMEG*W(I))
*
  1000 CONTINUE
  2000 KCOUNT = K
  RETURN
  ENDD

```

このソルバ BCG-Stab (1,2) を使い、ICCG の時と同様に、その外側の IN (プログラム上) ループで、実際に解く $AU=F$ の収束精度を評価しておく必要がある。

kct=0

x_0 を選ぶ

bnrm= $\|F\|$ の計算

CALL UPILU2 : $C = A_0$ を不完全 LU 分解 ($A_0 = A + R$)

while $\|F - Ax\| / \text{bnrm} > \text{EPSN}$ do

CALL BCGSTB2 : ($U^{-T}AU^{-1}\bar{x} = U^{-T}b$ 相当を解く) kcount 回の反復

kct=kct+kcount

BCGSTB2内の判定値:EPSICWはEPSNより小さく、場合によるがEPSN/16程度がよい。ここでの例では、1回のBCGSTB2で全体の評価値EPSNより小さくなる。EPSN=10⁻⁵で、帯ガウスの結果とBCGSTB2の結果は、7桁まで合う。

初期近似の $x_0(=BW(I))$ には、

$$x_0 = 0.0$$

を与えているが、 κ , μ , b に段差があり、複雑な場合には、それらしい初期値を与えないと収束しない。初期値の善し悪しで、収束時間も変わってくる。プログラムレベルでは、次のようになる。上記、計算主部は、プログラム上では口内である。

```

IMPLICIT REAL*8 (A-H,O-Z)
REAL*4 CPUT
PARAMETER (MJ=20,M=MJ*10,M2=MJ*11-1,N=M*M2,EPS=1.0D-7)
DIMENSION AA(N),AB(-M:N),AC(-M:N),F(N),WK(M)
+,      AB1(-M:N+M),AC1(-M:N+M)
+,      FQ(N+M)
+,      B1(N),BW(-M:N+M)
+,      DI(-M:N),P(-M:N+M),W(-M:N+M)
+,      X(-M:N+M),AP(N),      UB(-M:N),AE(-M:N)
+,      AF(-M:N),AF1(-M:N+M),AE1(-M:N+M),UB1(-M:N+M)
+,      V(-M:N+M)
+,      RRA(-M:N+M),PA(-M:N+M),RR(-M:N+M),RO(N)

DATA EPSN/1.0D-5/,NIT/128/
      EPSICW=EPSN/16.0
USHIRB=1.01D0
UU=0.98D0
      0.98(Best)***
ISTB=3

```

<離散化部>

```

CALL CLOCK0
C
*C   BW(Ui) no Syokiti ***
      DO 111 K=1,N
111  BW(K)=0.0D0
C
*C   A*U=F wo Toku; Kotae --> F ***
      BNRM=0.0
      DO 110 I=1,N
110  BNRM=BNRM+F(I)*F(I)
      BNRM=DSQRT(BNRM)
*
      IF(ISTB.EQ.3) THEN
CC   CALL UPILU4(N,M,AA,AB,AC,AE,AF,UB,DI,USHIRB
-   +      ,AB1,AC1,AE1,AF1,UB1)
      CALL ILUCM3(N,M,AA,AB,AC,AE,AF,UB,DI,UU
+   +      ,AB1,AC1,AE1,AF1,UB1)
      ELSE IF(ISTB.EQ.2) THEN
      CALL UPILU2(N,M,AA,AB,AC,AE,UB,DI,USHIRB
+   +      ,AB1,AC1,AE1,UB1)
CC   CALL ILUCM2(N,M,AA,AB,AC,AE,UB,DI,UU
CC   +      ,AB1,AC1,AE1,UB1)
      ENDIF
C
      DO 226 I=1,N
226  X(I)=BW(I)
C

```

```

KCT=0
DO 230 IN=1,NIT
IF (ISTB.EQ.3) THEN
CALL BCGSTB3 (AA,AB,AC,AE,AF,UB,DI,AB1,AC1,AE1,AF1,UB1,P,V,W,F
+
,R0 ,X,AP,EPSICW,ERR,M,N,KCOUNT,BNRM,RRA,PA,RR)
ELSE IF (ISTB.EQ.2) THEN
CALL BCGSTB2 (AA,AB,AC,AE,UB,DI,AB1,AC1,AE1,UB1,P,V,W,F
+
,R0 ,X,AP,EPSICW,ERR,M,N,KCOUNT,BNRM,RRA,PA,RR)
ENDIF
C
KCT=KCT+KCOUNT
CNR=0.0D0
DO 116 I=1,N
B1(I)=F(I)-(AC1(I)*X(I-M)+AB1(I)*X(I-1)
+
+AA(I)*X(I)+AB(I)*X(I+1)+AC(I)*X(I+M))
CC
B1(I)=F(I)-(AC(I-M)*X(I-M)+AB(I-1)*X(I-1)
CC
+
+AA(I)*X(I)+AB(I)*X(I+1)+AC(I)*X(I+M))
116
CNR=CNR+B1(I)*B1(I)
CNR=DSQRT(CNR)/BNRM
*
WRITE(6,5800) IN,X(J1),X(J2),X(J3),X(J4),CNR,KCOUNT
5800
FORMAT(3H %,X,I4,F10.5,F10.5,F10.5,F10.5,D14.6,I5)
IF(CNR.LT.EPSN) GO TO 250
EPSICW=EPSICW/2.0
C
230 CONTINUE
250 CONTINUE

```

```

DO 220 I=1,N
C
BW(I)= X(I) ***
220
F(I) = X(I)

```

BCGSTB2は、1次元配列(n個程度)を、 $AA(a_{i,i})$, $AB(a_{i,i+1})$, $AC(a_{i,i+m})$, $AB1(a_{i,i-1})$, $AC1(a_{i,i-m})$ も含め、作業領域など20本必要とするので、mが10くらい(MJ=1に相当)から、帯ガウス(AR(-m:m,n)を使う)とBCGSTB2の、CPU時間、メモリ量に差が出始める。シミュレーション条件: $C_0(=\frac{\mu}{\kappa}) = 0.5$, $DF = 1.0$ の下で、MJ=20の時、m=200, n=43800となり、メモリ量、CPU時間は、BGLU(非対称帯ガウス): 140MB (103秒)、BCGSTB2: 7MB (4秒)の差となる。この時、BCGSTB2の前処理である不完全LU分解には、サブルーチンILUCM2を使った。この資料では、後で、不完全LU分解を安定に進行させる「おまじない」として2通りの方法を比較し評価するが、ILUCM2は、前回(純拡散)のICCGの前処理に使ったICDCMPを非対称行列用に拡張したものである。

Table 1 Solver(BGLU, BCGSTB2) performance for MJ(mesh size)
ILUCM2(U=0.98), $C_0=0.5$, $DF=1.0$

MJ	N	AR(MB)	BGLU(s)	BCGSTB2(s)[MB]	kcount	$u_{min}(C)$	$u_{max}(C)$
1	100	0.016	0.	0. [0.016]	11	-0.5690	0.2532
5	2700	2.16	0.	0. [0.43]	29	-0.2562	0.1428
10	10900	18	1	1 [1.7]	25	-0.2243	0.1294
15	24600	59	5	1 [3.9]	33	-0.2140	0.1247
20	43800	140	103	4 [7]	47	-0.2089	0.1225
30	98700	474	—	11 [16]	72	-0.2039	0.1203
40	175600	1124	—	25 [28]	93	-0.2015	0.1192

Table 1 に、MJ を変えた時の各ソルバの CPU 時間、メモリ量および、BCGSTB2 内の収束

反復回数: kcount、温度 u の最小値: u_{min} 、最大値: u_{max} を示す。この計算では、ILUCM2 における後(うしろ:人の名前)のパラメータは、 $U = 0.98$ とした。この U は、プログラム上では、変数 DIV の右辺の最後にかかっており、不完全 LU 分解が安定に進行するためのまじないである。

```

SUBROUTINE ILUCM2( N,M ,AA,AB,AC,AE,UB,DI,U,AB1,AC1,AE1,UB1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AA(N),AB(-M:N),AC(-M:N),DI(-M:N),
+          AE(-M:N),UB(-M:N),
+          AB1(-M:N+M), AC1(-M:N+M),AE1(-M:N+M),UB1(-M:N+M)
DO 100 I = 1, N
*
  AE1(I) = -AC1(I)*UB(I-M)*DI(I-M)
  UB1(I) = AB1(I)-AC1(I)*AE(I-M)*DI(I-M)
*
  DIV=AA(I)-UB1(I)*UB(I-1)*DI(I-1)-AC1(I)*AC(I-M)*DI(I-M)
+      -AE1(I)*AE(I-M+1)*DI(I-M+1)
+      -(UB1(I)*AE(I-1)*DI(I-1)+AE1(I)*UB(I-M+1)*DI(I-M+1))*U
  DI(I)=1.0D0/DIV

  UB(I) = AB(I)-AE1(I)*AC(I-M+1)*DI(I-M+1)
  AE(I) = -UB1(I)*AC(I-1)*DI(I-1)

100 CONTINUE
RETURN
END

```

Table 1 の結果に基づき、MJ に対する u_{max} , u_{min} のあみ目依存性を見ておくと、Fig.5 となる。あみ目が精しくなると、 u_{max} , u_{min} は小さくなる。MJ=20 に対して 2%の精度が必要なならば、MJ=15 にしなければならない。このグラフを、横軸にセルペクレ数: $\frac{C_0}{MJ}$ をとって、プロットし直すと Fig.6 となる。セルペクレ数は 0.5 以下で、MJ が精しくなる(セルペクレ数が小さくなる)程、 u_{max} , u_{min} は小さくなる。

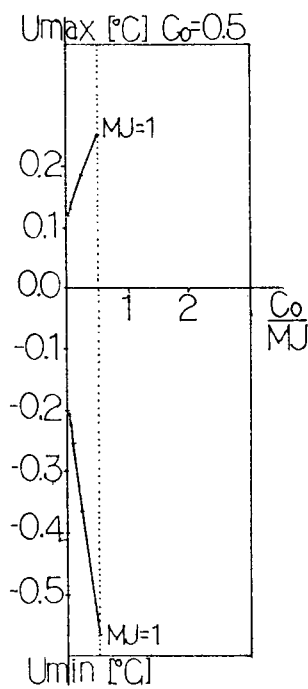


Fig.6 $\frac{C_0}{MJ}$ dependence of u_{min} , u_{max}

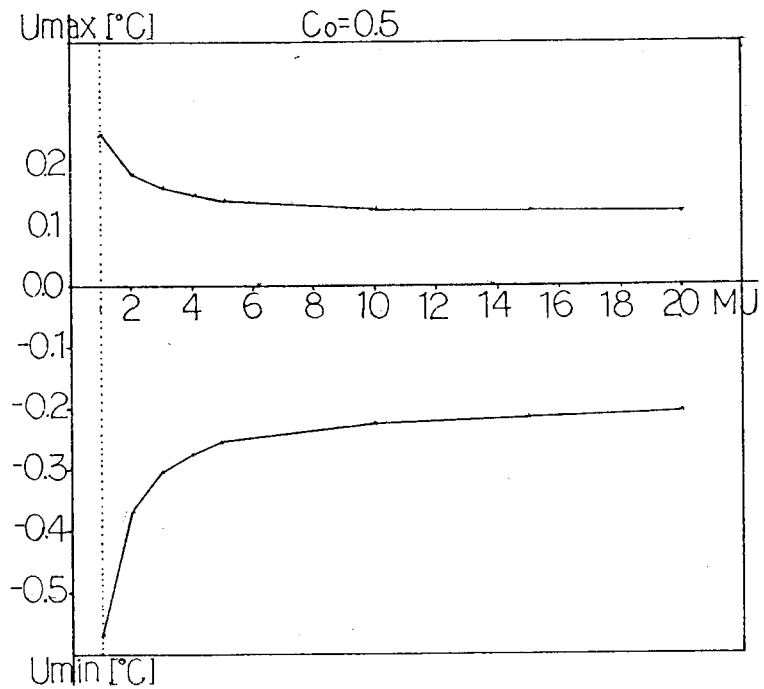


Fig.5 MJ dependence of u_{min} , u_{max}

先のBCGSTB(1,2)の場合、Aに近い A_0 として、「Aに、i行要素が $(i, i-m+2)$, $(i, i+m-2)$ の所だけ加わった行列: $A_0 = L \cdot U$ を使う。この時、 $R = A_0 - A = L \cdot U - A$, ただしLの対角要素は1。そして、Rは、 $(i, i-m+2)$, $(i, i+m-2)$ の要素だけを持つようにする。プログラム上では、上三角行列 $U(u_{ij})$ 、下三角行列 $L(l_{ij})$ において、

$$u_{ii} = \frac{1}{DI(I)}, \quad u_{i,i+1} = UB(I), \quad u_{i,i+m-1} = AE(I), \quad u_{i,i+m}(AC(I) \text{ のまま})$$

$$l_{ii}(=1.0), \quad l_{i,i-1} = UB1(I), \quad l_{i,i-m+1} = AE1(I), \quad l_{i,i-m}(AC1(I) \text{ のまま})$$

と対応しており、L,Uは3重対角部の内側にそれぞれ1本ずつ $(i, i+m-1): AE(i)$, $(i, i-m+1): AE1(i)$ を持つ。

後(うしろ)のパラメータUについて、捕捉する。 A_0X を考えると、AXに加えて、

$UB(i-m+1)DI(i-m+1)AE1(i)X(i-m+2) + UB1(i)DI(i-1)AE(i-1)X(i+m-2)$ という項ができる。ここで、 $X(i-m+2)$, $X(i+m-2)$ が、精メッシュならば $X(i)$ に近い値を取ると考えて、これらの係数を $X(i)$ の係数(対角部)に繰り入れるのである。こうすると、繰り入れた A_0X は、前の A_0X より、よりAXに近くなっていよう。よって、 A_0 は、 A_0 より、よりAに近い事と同じような効果が期待できる。Uは、通常 $U = 0.95$ とするが、0.98がベストで、0.95の場合のCPU時間は、0.98に比べ2割程度増える。この項($U=0.98$)を加えたものは、付け加えない: $U=0.0$ としたものに比べ、5割程度確かに速く収束する。理由に納得がいくかどうかよりも、ここでは実利を優先する。

また、ILUCM2のように、 $X(i-m+2)$, $X(i+m-2)$ の項を、 $X(i)$ に繰り入れるのではなく、プログラム上ではDIVの項の右辺のAA(I)に、 $UTOP*AA(I)$ として、直接DIVに少し変更を加えるやり方もある。この時UTOPは、 $UTOP = 1.01$ とする。このUTOPも後(うしろ)のパラメータの1種であり、不完全LU分解を安全に、安定に進行させるまじないである。UTOPを組み込んだサブルーチン: UPILU2は、次のようになる。

```

SUBROUTINE UPILU2(N,M,AA,AB,AC,AE,UB,DI,U,AB1,AC1,AE1,UB1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AA(N),AB(-M:N),AC(-M:N),DI(-M:N),
+ AE(-M:N),UB(-M:N),
+ AB1(-M:N+M),AC1(-M:N+M),AE1(-M:N+M),UB1(-M:N+M)
***
DO 100 I = 1, N
***
CC W= ABS(AC1(I))+ABS(AB1(I))+ABS(AB(I))+ABS(AC(I))
CC UUI = MAX(W,AA(I))
***
AE1(I)= -AC1(I)*UB(I-M)*DI(I-M)
UB1(I)= AB1(I)-AC1(I)*AE(I-M)*DI(I-M)
*
C/ U: UTOP
DIV=U*AA(I)-UB1(I)*UB(I-1)*DI(I-1)-AC1(I)*AC(I-M)*DI(I-M)
+ -AE1(I)*AE(I-M+1)*DI(I-M+1)
C DIV= AA(I)-UB1(I)*UB(I-1)*DI(I-1)-AC1(I)*AC(I-M)*DI(I-M)
C + -AE1(I)*AE(I-M+1)*DI(I-M+1)
DI(I)=1.0D0/DIV
*
UB(I)= AB(I)-AE1(I)*AC(I-M+1)*DI(I-M+1)
AE(I)= -UB1(I)*AC(I-1)*DI(I-1)
*
100 CONTINUE
RETURN
END

```


UPILU2におけるUTOP=1.0と、ILUCM2におけるU=0.0は、同じ計算(DIVに繰り返し込みをしないオリジナルの不完全LU分解)になる。UTOP=1.01はUTOP=1.0に比べ、元数nが大きい時に効果が出て、1割程度速くなる。ソルバと不完全LU分解サブルーチンとの関係を整理すれば、次のようになる。

ICCG12(対称行列用) + ICDCMP(パラメータUが入った不完全LU分解)
 BCGSTB2(非対称行列用) + ILUCM2(パラメータUが入った不完全LU分解)
 BCGSTB2(非対称行列用) + UPILU2(パラメータUTOPが入った不完全LU分解)
 後で、実測により、移流拡散系(Aが非対称行列になる)のシミュレーションでは、ILUCM2より、UPILU2の方がより収束を安定にさせる結果例を見る。実際に $C_0 = 0.5$, $DF = 1.0$ の下で、ILUCM2でU=0.98, 0.95とした場合、UPILU2でUTOP=1.01, 1.0とした場合について、CPU時間を比較計測する([]内は収束反復回数: kcount)と、Table 2aとなる。実測は、日立FLORA330 DK2(Pentium3) 933MHz メモリ256MBで行なった。ソルバは、BCGSTB2を使う。

Table 2a Parameter U,UTOP CPU time performance for BCGSTB2

$C_0=0.5$, $DF=1.0$; []:kcount

MJ	U=0.98(s)	U=0.95(s)	UTOP=1.01(s)	UTOP=1.0(s)
20	4 [47]	4 [59]	7 [105]	7 [100]
30	11 [72]	11 [72]	22 [148]	22 [148]
40	25 [93]	32 [126]	47 [186]	54 [207]

なお念のため、これら計測が、対称行列用ソルバICCG12を使った場合にどうなるかにも関心があるので、純拡散系(ドリフト項がない $C_0=0.0$ の場合に対応)の場合もシミュレーションしてみた。結果はTable 2bとなる。ソルバは、ICCG12を使う。

Table 2b Parameter U,UTOP CPU time performance for ICCG12

$C_0=0.0$, $DF=1.0$; []:kcount

MJ	U=0.98(s)	U=0.95(s)	UTOP=1.01(s)	UTOP=1.0(s)
20	2 [56]	3 [69]	5 [139]	5 [134]
30	6 [84]	7 [101]	14 [199]	14 [191]
40	14 [110]	17 [134]	36 [201+93]	33 [201+74]

パラメータU, UTOPによる結果の傾向は、BCGSTB2とICCG12のソルバで同じ傾向を示す。非対称行列を解くBCGSTB2の場合には、計算の負担はICCG12のおよそ2倍となる。これらの表により、UがかかるDIVに繰り返し込む項が、いかに収束に効くか実感できる。その効果は、ICCG12でもBCGSTB2においても同様に効いていることが分かる。ただし、後で示すようにUTOPを使った方が、Uを使った時に収束しない場合でも、安全に収束することが分かっている。したがって、非対称行列を扱う場合、実務上ではUTOP版を使う方が良い。

さらに、収束を速くして、計算時間を減らしたい場合には、不完全LU分解するAに近い A_0 として「Aにi行要素が(i,i-m+3)(i,i+m-3)の所だけ加わった行列((1,2)より1つ

内側)： $\mathbf{A}_0 = \mathbf{L} \cdot \mathbf{U}$ を使う。この時(1,2)と同様に、 $\mathbf{R} = \mathbf{A}_0 - \mathbf{A} = \mathbf{L}\mathbf{U} - \mathbf{A}$ ただし対角要素は1 とする。プログラムレベルでは、 $\mathbf{U}(u_{ij})$, $\mathbf{L}(l_{ij})$ において、

$u_{ii} = \frac{1}{DI(I)}$, $u_{i,i+1} = UB(I)$, $u_{i,i+m-2} = AF(I)$, $u_{i,i+m-1} = AE(I)$, $u_{i,i+m}(AC(I) \text{ のまま})$
 $l_{ii} (=1.0)$, $l_{i,i-1} = UB1(I)$, $l_{i,i-m+2} = AF1(I)$, $l_{i,i-m+1} = AE1(I)$, $l_{i,i-m}(AC1(I) \text{ のまま})$
と対応しており、 \mathbf{L}, \mathbf{U} は3重対角部の内側にそれぞれ2本ずつ $(i, i+m-2) : AF(i)$, $(i, i-m+2) : AF1(i)$ と $(i, i+m-1) : AE(i)$, $(i, i-m+1) : AE1(i)$ を持つ((1,2)より1本ずつ増える)。配列の大きさは、 $UB1(-M:N+M)$, $AE1(-M:N+M)$, $AF1(-M:N+M)$, $AC1(-M:N+M)$, $AF(-M:N)$ となる。 $\mathbf{A}_0\mathbf{X}$ から出る、 x_{i+m-3} , x_{i-m+3} の項の係数を(1,2)同様に、 \mathbf{U} を掛けて対角項に加え繰り返す。この不完全LU分解を使ったBCGSTab法を、BCGSTB(1,3)(:サブルーチンBCGSTB3)と呼び、(1,2)に比べて反復回数が減るので、ふつう2割程度CPU時間が短くなる。また、BCGSTB(1,2)は、原算法BCG(1,2)に比べて2割程度速くなる。MJ=40の時、BCGSTB(1,2)はCPU時間:25秒、反復回数kcount:[93回]、BCG(1,2)では31秒[126回]となる。パラメータUおよびUTOPを変えた時のTable 2ab((1,2)の時)に対応する(1,3)の結果は次のようになる(Table 3ab)。

Table 3a Parameter U,UTOP CPU time performance for BCGSTB3
 $C_0=0.5$, $DF=1.0$; []:kcount

MJ	U=0.98(s)	U=0.95(s)	UTOP=1.01(s)	UTOP=1.0(s)
20	4 [39]	4 [39]	6 [84]	6 [82]
30	13 [69]	11 [63]	21 [123]	18 [107]
40	22 [75]	24 [80]	43 [147]	44 [151]

Table 3b Parameter U,UTOP CPU time performance for ICCG13
 $C_0=0.0$, $DF=1.0$; []:kcount

MJ	U=0.98(s)	U=0.95(s)	UTOP=1.01(s)	UTOP=1.0(s)
20	2 [47]	3 [54]	4 [108]	4 [106]
30	5 [66]	6 [81]	12 [162]	12 [152]
40	12 [87]	14 [107]	26 [201]	26 [201]

不完全LU分解(1,2)より配列AF(-M:N),AF1(-M:N+M)が増えて、BCGSTB3では、1次元配列(n個程度)が20本から22本に増えるが、得られるCPU時間短縮効果の方が大きい。ICCG(1,3)とBCGSTB(1,3)に対する、U,UTOPの影響は、(1,2)の場合と同じ傾向を示す。

また、ソルバのくせを見るために次の実験を試みる。Table 2bのICCG12部は、対称行列用なので $C_0=0.0$ に対応している。比較のために、Table 2aのドリフト項の $C_0=0.5$ を $C_0=0.0$ として、U=0.98とUTOP=1.01の下でBCGSTB(1,2)をソルバにして解いてみる。その計算結果を、Table 4に示す。当然、結果は同じ(8桁まで)になる。MJ=20,30,40の時の u_{min} , u_{max} 値は、次のようになる。

MJ=20: $u_{min}=-0.1557$ (C), $u_{max}=0.1087$ (C);

MJ=30: $u_{min}=-0.1526$ (C), $u_{max}=0.1068$ (C);

MJ=40: $u_{min}=-0.1511$ (C), $u_{max}=0.1058$ (C);

U=0.98 の時 (ILUCM2 使用)、BCGSTB2 は、ICCG12 に比べて 2 倍 CPU 時間がかかる。これは、 $C_0=0.5$ の時も同様。U=1.01 の BCGSTB2 は、ICCG12 の 3 倍かかることが分かる。

Table 4 Parameter U,UTOP CPU time performance for ICCG12, BCGSTB2
 $C_0=0.0$, $DF=1.0$; []:kcount

MJ	U=0.98 (s)	U=0.0 (s)	UTOP=1.0 (s)	UTOP=1.01 (s)	U=0.98 (s)
	ICCG12	ICCG12	BCGSTB2	BCGSTB2	BCGSTB2
20	2 [56]	5 [134]	7 [94]	6 [94]	4 [48]
30	6 [84]	14 [191]	19 [128]	21 [142]	11 [69]
40	14 [110]	33 [201+74]	42 [166]	43 [168]	23 [88]

不完全 LU 分解 ILUCM2 に対応する (1,3) の ILUCM3 と、BCGSTB3 のリストを示しておく。

```

SUBROUTINE ILUCM3 (N,M,AA,AB,AC,AE,AF,UB,DI,U,AB1,AC1,AE1,AF1,UB1)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AA(N),AB(-M:N),AC(-M:N),DI(-M:N),
+ AE(-M:N),AF(-M:N),UB(-M:N),
+ AB1(-M:N+M),AC1(-M:N+M),AE1(-M:N+M),AF1(-M:N+M),UB1(-M:N+M)
***
DO 100 I = 1, N
***
CC W= ABS(AC1(I))+ABS(AB1(I))+ABS(AB(I))+ABS(AC(I))
CC UUI = MAX(W,AA(I))
***
AE1(I) = -AC1(I)*UB(I-M)*DI(I-M)
AF1(I) = -AE1(I)*UB(I-M+1)*DI(I-M+1)
UB1(I) = AB1(I)-AC1(I)*AE(I-M)*DI(I-M)-AE1(I)*AF(I-M+1)*DI(I-M+1)
C/
U:
DIV=AA(I)-UB1(I)*UB(I-1)*DI(I-1)-AC1(I)*AC(I-M)*DI(I-M)
+ -AE1(I)*AE(I-M+1)*DI(I-M+1)-AF1(I)*AF(I-M+2)*DI(I-M+2)
+ -U*(AF1(I)*AC(I-M+2)*DI(I-M+2)
+ +AF(I-M)*AC1(I)*DI(I-M)
+ +UB1(I)*AF(I-1)*DI(I-1)+UB(I-M+2)*AF1(I)*DI(I-M+2))
C DIV= AA(I)-UB1(I)*UB(I-1)*DI(I-1)-AC1(I)*AC(I-M)*DI(I-M)
C + -AE1(I)*AE(I-M+1)*DI(I-M+1)-AF1(I)*AF(I-M+2)*DI(I-M+2)
DI(I)=1.0D0/DIV
*
UB(I) = AB(I)-AE1(I)*AC(I-M+1)*DI(I-M+1)
+ -AF1(I)*AE(I-M+2)*DI(I-M+2)
AF(I) = -UB1(I)*AE(I-1)*DI(I-1)
AE(I) = -UB1(I)*AC(I-1)*DI(I-1)
*
100 CONTINUE
RETURN
END
CC
SUBROUTINE BCGSTB3 (AA,AB,AC,AE,AF,UB,DI,AB1,AC1,AE1,AF1,UB1,P,V,W
+ ,B0,R0,X,AP,EPSLON,ERR,M1,N,KCOUNT,BNRM,AEE,EE,RR)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION AA(N),AB(-M1:N),AC(-M1:N),DI(-M1:N)
+, P(-M1:N+M1),W(-M1:N+M1),B0(N),X(-M1:N+M1)
+, AP(N),V(-M1:N+M1),UB(-M1:N),AE(-M1:N),AF(-M1:N)
+, AB1(-M1:N+M1),AC1(-M1:N+M1),UB1(-M1:N+M1),AE1(-M1:N+M1)
+, AF1(-M1:N+M1),AEE(-M1:N+M1),EE(-M1:N+M1),RR(-M1:N+M1),R0(N)

```

```

DO 10 I = 1, N
10  V(I)=B0(I)-(AA(I)*X(I)+AB(I)*X(I+1)+AC(I)*X(I+M1)
+
+AB1(I)*X(I-1)+AC1(I)*X(I-M1))
***  KOKO DE WA  V = B -AX DE ARU  ***
*
DO 30 I = 1, N
30  W(I)=(V(I)-AC1(I)*W(I-M1)-UB1(I)*W(I-1)-AE1(I)*W(I-M1+1)
+
+AF1(I)*W(I-M1+2))*DI(I)
*
DO 40 I = N, 1, -1
40  W(I)=W(I)-DI(I)*(UB(I)*W(I+1)+AC(I)*W(I+M1)+AE(I)*W(I+M1-1)
+
+AF(I)*W(I+M1-2))
***  KOKO DE WA  W = INV(LU)*(B-AX) DE ARU  *****
CR= 0.0D0
DO 60 I = 1, N
P(I) = W(I)
RR(I)= W(I)
R0(I)= W(I)
60  CR = CR + W(I)*W(I)
***  ITERATION;  KONO ATO, V O V, W O Q TO SHITE TSUKAU  ***
DO 1000 K = 1, 400
DO 100 I = 1, N
AP(I)=AA(I)*P(I)+AB(I)*P(I+1)+AC(I)*P(I+M1)
+
+AB1(I)*P(I-1)+AC1(I)*P(I-M1)
100  CONTINUE
***
DO 110 I = 1, N
110  W(I)=(AP(I)-AC1(I)*W(I-M1)-UB1(I)*W(I-1)-AE1(I)*W(I-M1+1)
+
+AF1(I)*W(I-M1+2))*DI(I)
*
DO 120 I = N, 1, -1
120  W(I)=W(I)-DI(I)*(UB(I)*W(I+1)+AC(I)*W(I+M1)+AE(I)*W(I+M1-1)
+
+AF(I)*W(I+M1-2))
***  W = INV(LU)*AP  DE ARU  ***
SIG= 0.0D0
DO 130 I=1, N
130  SIG = SIG+R0(I)*W(I)
ALFA = CR/SIG
*
DO 140 I=1,N
140  EE(I)= RR(I)-ALFA*W(I)
*
DO 150 I = 1, N
150  AEE(I)= AA(I)*EE(I)+AB(I)*EE(I+1)+AC(I)*EE(I+M1)
+
+AB1(I)*EE(I-1)+AC1(I)*EE(I-M1)
***
DO 160 I = 1, N
160  V(I)=(AEE(I)-AC1(I)*V(I-M1)-UB1(I)*V(I-1)-AE1(I)*V(I-M1+1)
+
+AF1(I)*V(I-M1+2))*DI(I)
*
DO 170 I = N, 1, -1
170  V(I)=V(I)-DI(I)*(UB(I)*V(I+1)+AC(I)*V(I+M1)+AE(I)*V(I+M1-1)
+
+AF(I)*V(I+M1-2))
***  V = INV(LU)*A*E DE ARU  ***
EEV= 0.0D0
VV= 0.0D0
DO 180 I= 1,N
EEV= EEV+EE(I)*V(I)
180  VV = VV +V(I)*V(I)
OMEG =EEV/VV
*
DO 190 I= 1, N
X(I)= X(I)+ALFA*P(I)+OMEG*EE(I)
190  RR(I)=EE(I)-OMEG*V(I)
*
CR=0.0D0

```

```

                ERR = 0.0D0
DO 200 I = 1, N
    CR = CR+ R0(I)*RR(I)
200  ERR = ERR+RR(I)*RR(I)
    ERR = DSQRT(ERR)/BNRM
    IF( ERR.LT.EPSLON ) GO TO 2000
*
    BETA = CR/(OMEG*SIG)
DO 300 I = 1, N
300  P(I)= RR(I)+BETA*(P(I)-OMEG*W(I))
*
1000 CONTINUE
2000 KCOUNT = K
    RETURN
    END

```

実務上は、非対称用の BCGSTB の方が、対称用 ICCG より、およそ倍の CPU 時間がかかるので、ICCG は (1,2) を使い、BCGSTB には (1,3) を使うのが手頃である。(1,2)/(1,3) の CPU 比は、ICCG, BCGSTB ともおよそ 1.2 と見て良い。

前にふれた、不完全 LU 分解のパラメータ U(ILUCM) では収束しないが、UTOP による不完全 LU 分解ならば収束する場合を実際に示しておく。 $C_0=10.0$, $DF=1.0$ で、 $MJ=20$ の場合に、ソルバとして BCGSTB(1,3) と BCGSTB(1,2), BGLU を使って、BCGSTB のソルバでは $UTOP=1.01, 1.0$, $U=0.98, 0.95$ と変えて、CPU 時間と収束回数 (kcount) を見た (Table 5 上段)。この時、BGLU、BCGSTB(1,3) の $UTOP=1.01, 1.0$ の時には解けるが、 $U=0.98, 0.95$ では収束しない。一方ソルバが BCGSTB(1,2) の場合には、 $UTOP=1.01, 1.0$ は収束し、同 BCGSTB(1,2) の時の $U=0.98, 0.95$ は、BCGSTB の外側のループを数回繰り返して ($EPSN=1.0D-5$)、かろうじて収束する (Table 5 下段)。これらにより、パラメータ U より UTOP の方が、非対称行列の時には不完全 LU 分解を安全に進行させ、解を収束させることが分かる。また、この例によれば、MJ が小さく、あみ目が粗い場合には BCGSTB(1,2) の方が、かえって (1,3) より速い場合があることも分かる。この傾向は、 $MJ=40$ で同じシミュレーションをしても同様に起こる (Table 5; BGLU は主メモリ OVER で解けていない)。 $MJ=40$ の時には、BCGSTB(1,2) の $U=0.98, 0.95$ でも収束しなくなる。

Table 5 Parameter U, UTOP CPU time performance for BCGSTB3(2), BGLU
 $C_0=10.0$, $DF=1.0$; []:kcount

MJ	UTOP=1.01 (s) BCGSTB3	UTOP=1.0 (s) BCGSTB3	U=0.98 (s) BCGSTB3	U=0.95 (s) BCGSTB3	BGLU (s)
20	10 [120]	8 [92]	Not Conv.	Not Conv.	103
40	65 [222]	46 [161]	Not Conv.	Not Conv.	Mem. Over

MJ	UTOP=1.01 (s) BCGSTB2	UTOP=1.0 (s) BCGSTB2	U=0.98 (s) BCGSTB2	U=0.95 (s) BCGSTB2	BGLU (s)
20	7 [89]	7 [98]	163 [401×6]	244 [401×9]	103
40	71 [277]	64 [249]	Not Conv.	Not Conv.	Mem. Over

$C_0=10.0$, $DF=1.0$ で、 $MJ=20, 40$ の時の u_{min}, u_{max} 値は、次のようになる。

$MJ=20$: $u_{min}=-0.6803$ (C), $u_{max}=0.0309$ (C);

$MJ=40$: $u_{min}=-0.6577$ (C), $u_{max}=0.0302$ (C);

シミュレーションの結果出力は、 $C_0=0.5$, $DF=1.0$, $MJ=1$ の時に、ソルバとして UP-ILU2(UTOP=1.01)+BCGSTB(1,2) を使って、次のようになる。

```
% DF= 1.0000000000000000
% M= 10 M2= 10 N= 100 : ISTB= 2
% C0= 0.5000000000000000
% 1 0.25324 0.25324 -0.31872 -0.31872 0.133436d-06 6
% MJ= 1
% CPUT= 0.000000000e+00
% UMIN, (x,y)= -0.5690027045614141 (6 ,10 )
% UMAX, (x,y)= 0.2532408061978241 (6 ,3 )
z=[
0.01 0.01 0.02 0.04 0.06 0.06 0.04 0.02 0.01 0.01 ;
0.01 0.03 0.05 0.10 0.19 0.19 0.10 0.05 0.03 0.01 ;
0.02 0.04 0.08 0.14 0.25 0.25 0.14 0.08 0.04 0.02 ;
0.02 0.04 0.08 0.14 0.25 0.25 0.14 0.08 0.04 0.02 ;
0.01 0.03 0.05 0.08 0.11 0.11 0.08 0.05 0.03 0.01 ;
0.00 0.01 0.00 -0.01 -0.08 -0.08 -0.01 0.00 0.01 0.00 ;
-0.01 -0.03 -0.06 -0.11 -0.22 -0.22 -0.11 -0.06 -0.03 -0.01 ;
-0.04 -0.08 -0.13 -0.20 -0.32 -0.32 -0.20 -0.13 -0.08 -0.04 ;
-0.07 -0.14 -0.22 -0.30 -0.38 -0.38 -0.30 -0.22 -0.14 -0.07 ;
-0.12 -0.24 -0.36 -0.48 -0.57 -0.57 -0.48 -0.36 -0.24 -0.12 ;
];
contour(0:1:9, 0:1:9, z,10);
```

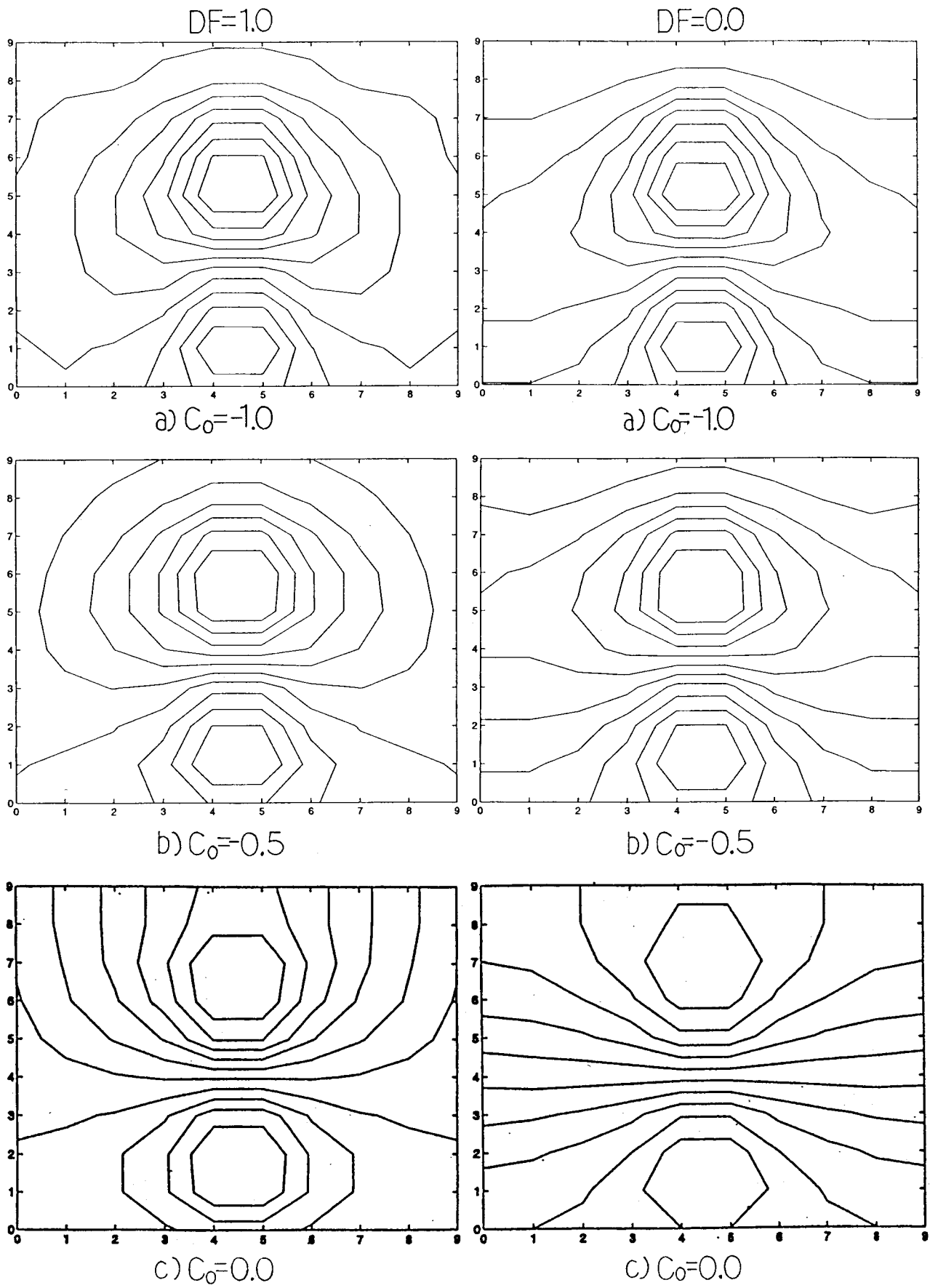
5 シミュレーション結果

$MJ=1$ ($n=100$ 元) で、 $DF=1.0$ と $DF=0.0$ の時に、 $C_0=-1.0,-0.5,0.0,0.5,1.0$ と変えて、それぞれの場合で温度の最小値 (u_{min})、最大値 (u_{max}) がどうなるか見る。この時、セルペクレ数は $\frac{C_0}{MJ}$ なので、1 以下である。結果を Table 6 に示す。Table 6 において、 $[j,k]$ は、格子点: $i=m \times (j-1)+k$, $m=MJ \times 10$ の $[x$ 方向 (j), y 方向 (k)] の位置を表す。 $k=m$ ならば、Fig.1 の上側のノイマン境界上 (CD) の格子点になる。

Table 6 u_{min}, u_{max} for C_0 at $DF=1.0, 0.0$

C_0	DF=1.0	DF=1.0	DF=0.0	DF=0.0
	u_{min} (C) [j,k]	u_{max} (C) [j,k]	u_{min} (C) [j,k]	u_{max} (C) [j,k]
-1.0	-0.2503 [6,6]	0.1174 [6,2]	-0.2591 [6,6]	0.0993 [6,2]
-0.5	-0.2954 [6,7]	0.1550 [5,2]	-0.3387 [6,6]	0.1089 [5,2]
0.0	-0.3525 [6,8]	0.2137 [6,3]	-0.6595 [6,8]	0.1133 [6,2]
0.5	-0.5690 [6,10]	0.2532 [5,3]	-6.1998 [5,10]	0.1146 [5,2]
1.0	-0.7268 [5,10]	0.2429 [5,4]	-62.526 [6,10]	0.1524 [5,3]

$DF=1.0$ については、 y 方向のドリフト (C_0) があるので、 $C_0=0.5,1.0$ では、温度の最小値 (u_{min})、最大値 (u_{max}) の位置は上方に移動して、負の温度領域が減ることになる。同じ $DF=1.0$ で、 C_0 を負にすると、逆に u_{min}, u_{max} は下方に移動して、負の温度領域が増える。 $DF=1.0$ で、上側だけノイマン境界だと、 u_{min} の位置は $C_0 > 0$ で C_0 値を増やすと上方へ移動し、じょじょに冷えていき u_{max} も下がる。 $C_0 < 0$ では、下側が固定境界 ($u=0$) なので、 u_{min}, u_{max} の位置は下方に移動し、その値も小さくなっていく。等高線表示で比較すると、Fig.7 となる。左が $DF=1.0$ で、右が $DF=0.0$ の場合。上から順に、 $C_0=-1.0,-0.5,0.0,0.5,1.0$ とする。



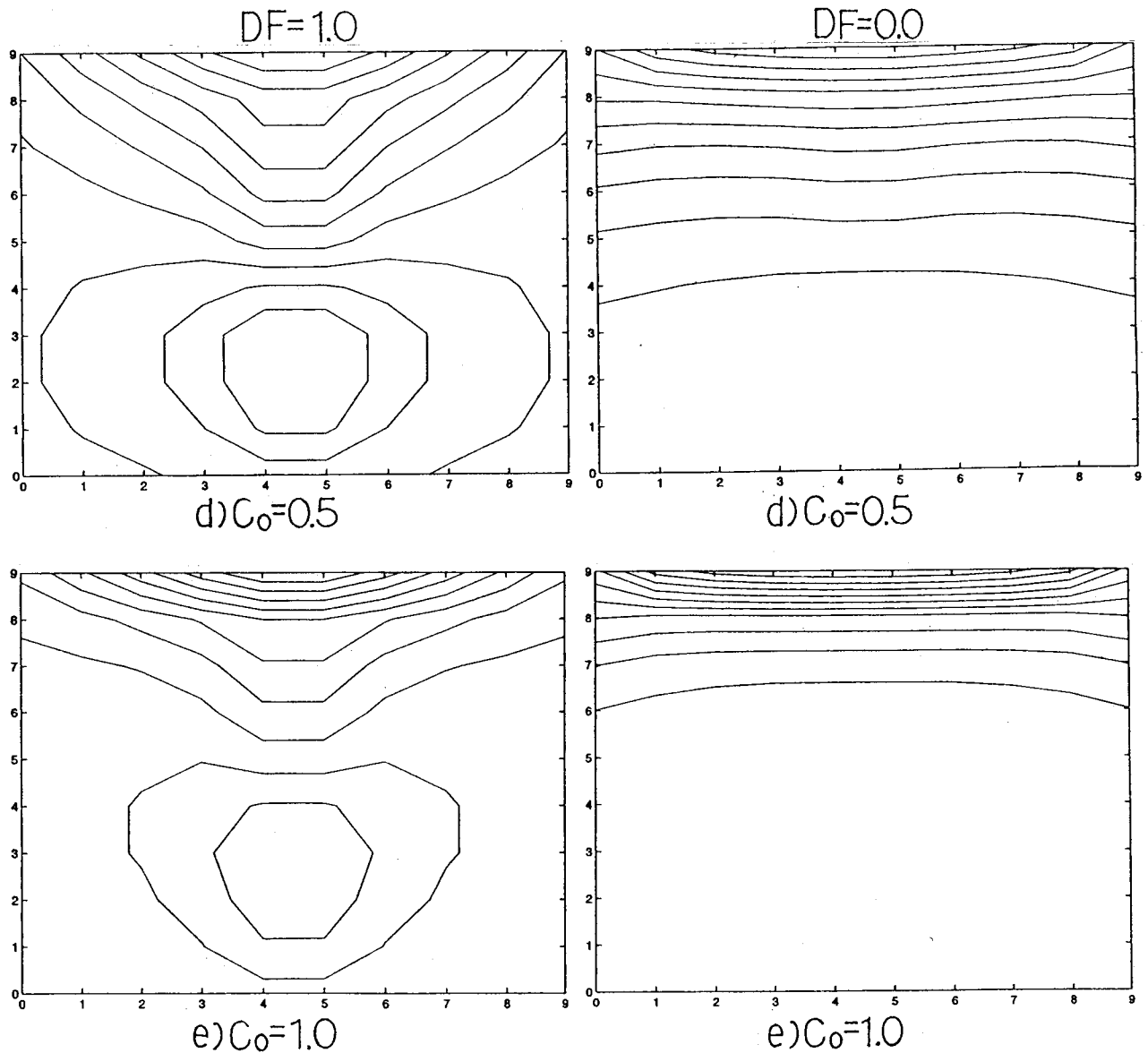


Fig.7 u_i distribution (DF=1.0, 0.0, MJ=1; for $C_0 = -1.0, -0.5, 0.0, 0.5, 1.0$)

DF=0.0(三方ノイマン境界、下側だけ固定境界 $u=0$; 場は共通で中央上が負の熱源、中央下が正の熱源) では、 $C_0 > 0$ だと熱は上方へ流れ u_{min} の位置は上方へ移動し、温度は上側のノイマン境界上中央で最小値をとり、中央部から周辺部にかけて、 $C_0 \geq 1$ において急冷する ($C_0=1.0$ で $u_{min}=-62.5(C)$)。DF=1.0 の場合には、 $C_0=10.0$ でも急冷は起こらない。DF=0.0 では三方(上左右)がノイマン境界となり熱の逃げ場がないために上部中央の負の熱源によって上部がどんどん冷やされる。DF=0.0 で $C_0 > 0$ の時に u_{max} は、その位置が上がりながら、値はじょじょに高くなっていく。一方DF=0.0 で $C_0 < 0$ の時には、 u_{min} の位置はじょじょに下方へ移動する。ただ下側は固定境界 $u=0$ なので、 u_{min}, u_{max} とも C_0 が小さくなるにつれて値は小さくなる。

さらに、DF=1.0 で、 C_0 を 1.0 から上げてみる。 $C_0=1.0, 2.0, 4.0, 6.0, 10.0$ としてみる。その時、セルペクレ数は $\frac{C_0}{MJ}$ で、MJ=1 のあみ目で、 $\frac{C_0}{MJ}=1.0, 2.0, 4.0, 6.0, 10.0$ となり、セルペ

クレ数は2.0を超える。その時のMJ=1の結果と、MJ=10として、 $C_0=1.0, 2.0, 4.0, 6.0, 10.0$ の時に、セルペクレ数： $\frac{C_0}{MJ}=0.1, 0.2, 0.4, 0.6, 1.0$ が正常範囲(< 2)になるようにしてシミュレーションした結果を比較する。両者は、あみ目の精しさが違うだけなので、物理的な温度分布の傾向は一致するはずである。比較した結果の u_{min}, u_{max} をTable 7に示す。その表中の数値 [j,k] ([x,y]) は、 u_{min}, u_{max} の格子点位置を表わす。右上の最終格子点は、MJ=1では[10,10]、MJ=10では[109,100]となり、一般には[MJ*11-1, MJ*10]である。

Table 7 u_{min}, u_{max} for C_0 at MJ=1, 10 (DF=1.0)

C_0	$\frac{C_0}{MJ}$	u_{min} (C) [j,k]	u_{max} (C) [j,k]
1.0	1.0	-0.7268 [5,10]	0.2429 [5,4]
1.0	0.1	-0.2936 [55,100]	0.1159 [55,36]
2.0	2.0	-0.8706 [6,10]	0.1790 [6,4]
2.0	0.2	-0.3660 [55,100]	0.08679 [55,38]
4.0	4.0	-1.3442 [6,10]	0.3786 [6,9]
4.0	0.4	-0.4861 [55,100]	0.05918 [55,39]
6.0	6.0	-1.6755 [6,10]	0.7632 [6,9]
6.0	0.6	-0.5811 [55,100]	0.04588 [55,40]
10.0	10.0	-0.7099 [5,10]	0.4177 [5,9]
10.0	1.0	-0.7261 [55,100]	0.03246 [55,40]

あみ目MJを精しくすれば、 u_{min}, u_{max} は小さくなるので(例えば $C_0=1.0$ で、MJ=1→10の時、 $u_{max}=0.2429 \rightarrow 0.1159$)、 $C_0=6.0$ で単純に、MJ=1とMJ=10の場合(セルペクレ数:6と0.6)を比較できないが、 u_{max} の位置がMJ=10では正の熱源近く[55,40]であるのに対し、MJ=1では上側のノイマン境界近く[6,9]になっているので、明らかに違う。

また、MJ=10の u_{max} 値は、 $C_0=1 \rightarrow 6$ につれ単調に下がるが、MJ=1の u_{max} 値は、 $C_0=2$ (セルペクレ数:2)までは位置が正の熱源近くで、MJ=10の場合と比べ妥当に見えるが、 $C_0=4$ になっていき値が大きくなり、 u_{max} 位置は上側のノイマン境界近くに移動しMJ=10の場合と違った動きをする(Fig.8)。

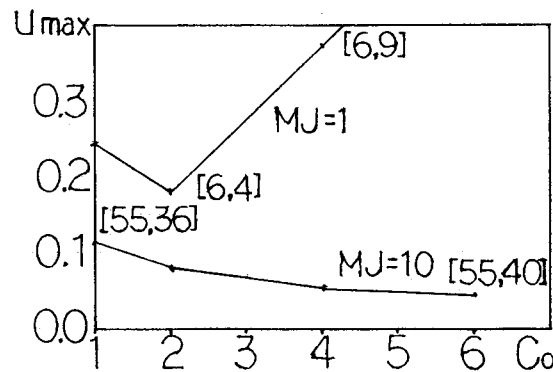
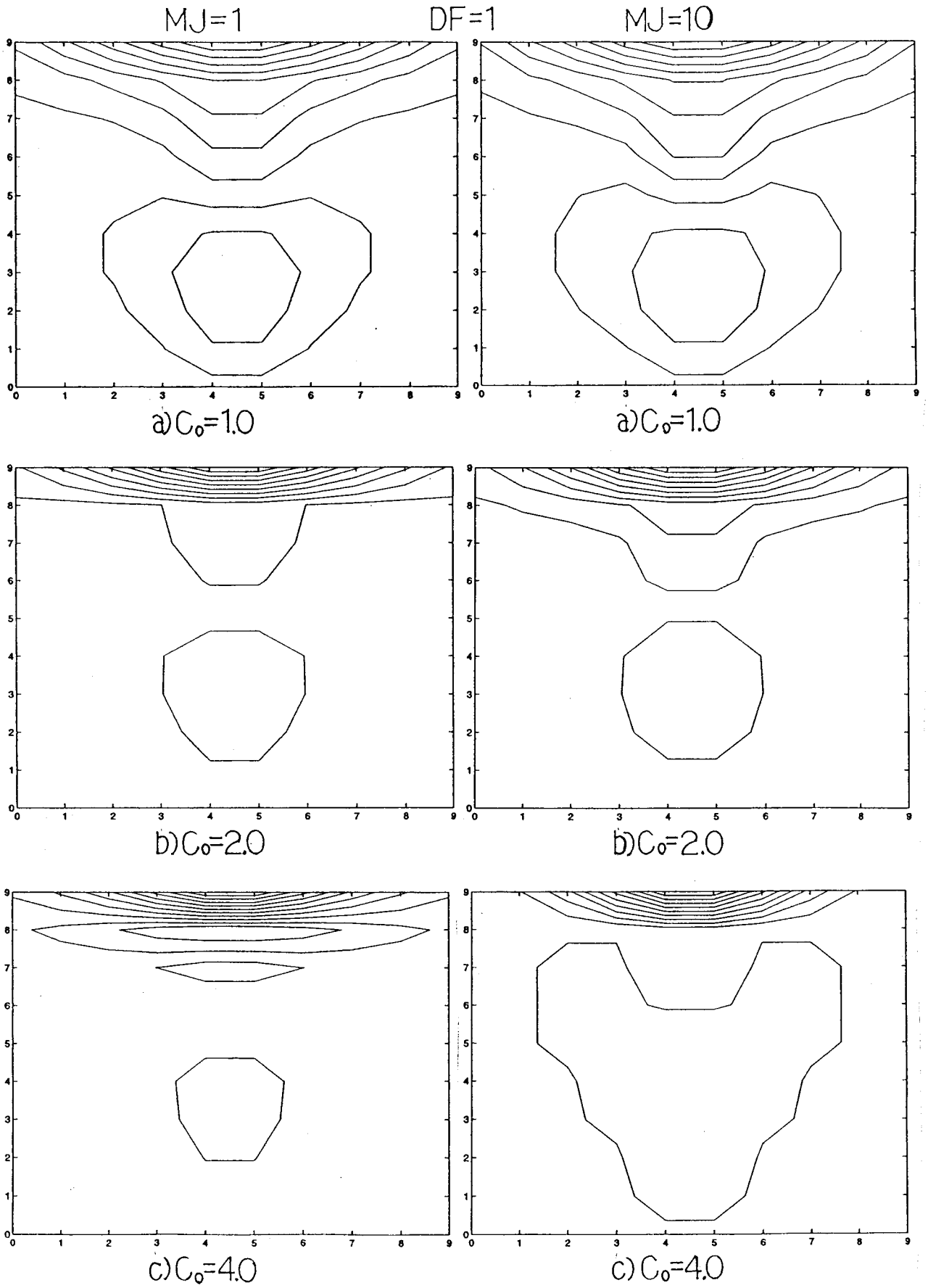


Fig.8 C_0 dependence of u_{max} for (MJ=1, 10)

その状況を等高線で比較すると次のFig.9のようになる。DF=1の下、左はMJ=1の場合で、右はMJ=10の場合。



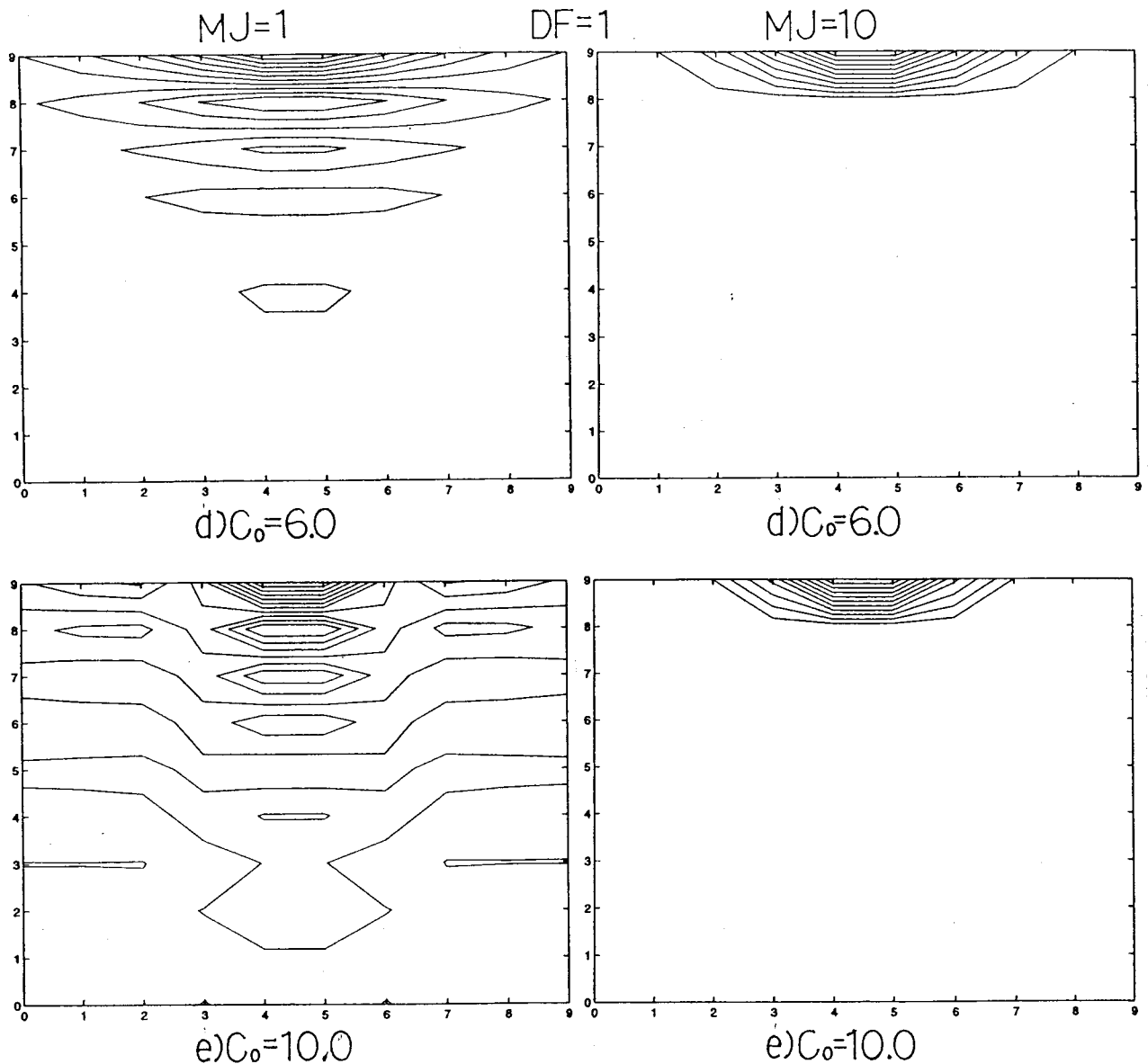


Fig.9 u_i distribution (DF=1.0, MJ=1, 10; for $C_0=1.0, 2.0, 4.0, 6.0, 10.0$)

$C_0=1.0, 2.0, 4.0, 6.0, 10.0$ と変えた等高線は、 $C_0=1.0, 2.0$ までは MJ=1, MJ=10 と両者の比較ではほぼ正常に見えるが、先の Fig.8 でも見たように、等高線でも明らかに $C_0=4.0$ 以降の MJ=1 の時の結果はおかしい。その時、 $C_0=2.0, 4.0$ の時の MATLAB 用出力 (出力下側が実際の上側のノイマン境界) は次のようになっている。

1 $C_0=2.0, MJ=1$

```

% UMIN, (x,y) = -0.8700349445612432 (6 ,10 )
% UMAX, (x,y) = 0.1790468998187159 (6 ,4 )
z = [
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.01 0.02 0.07 0.07 0.02 0.01 0.00 0.00 ;
0.00 0.00 0.02 0.05 0.13 0.13 0.05 0.02 0.00 0.00 ;
0.00 0.01 0.03 0.08 0.18 0.18 0.08 0.03 0.01 0.00 ;
0.01 0.02 0.04 0.08 0.15 0.15 0.08 0.04 0.02 0.01 ;
0.01 0.02 0.04 0.07 0.05 0.05 0.07 0.04 0.02 0.01 ;
0.01 0.02 0.04 0.04 -0.02 -0.02 0.04 0.04 0.02 0.01 ;
0.01 0.02 0.02 0.01 -0.08 -0.08 0.01 0.02 0.02 0.01 ;
0.01 0.02 0.01 -0.01 -0.05 -0.05 -0.01 0.01 0.02 0.01 ;
-0.09 -0.22 -0.41 -0.65 -0.87 -0.87 -0.65 -0.41 -0.22 -0.09 ;
];

```

$C_0=2.0, MJ=10$

```
% UMIN, (x,y)= -0.3660869575816891 (55 ,100 )
% UMAX, (x,y)= 8.679520497027761e-02 (55 ,38 )
z=[
  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.01  0.02  0.02  0.01  0.00  0.00  0.00 ;
  0.00  0.00  0.01  0.02  0.06  0.06  0.02  0.01  0.00  0.00 ;
  0.00  0.00  0.01  0.03  0.07  0.07  0.03  0.01  0.00  0.00 ;
  0.00  0.01  0.01  0.03  0.05  0.05  0.03  0.01  0.01  0.00 ;
  0.00  0.01  0.02  0.03  0.03  0.03  0.03  0.02  0.01  0.00 ;
  0.00  0.01  0.01  0.01 -0.02 -0.02  0.01  0.01  0.01  0.00 ;
  0.00  0.01  0.01  0.00 -0.04 -0.04  0.00  0.01  0.01  0.00 ;
  0.00 -0.01 -0.02 -0.04 -0.06 -0.06 -0.04 -0.02 -0.01  0.00 ;
 -0.03 -0.08 -0.16 -0.27 -0.35 -0.35 -0.27 -0.16 -0.08 -0.03 ;
];
```

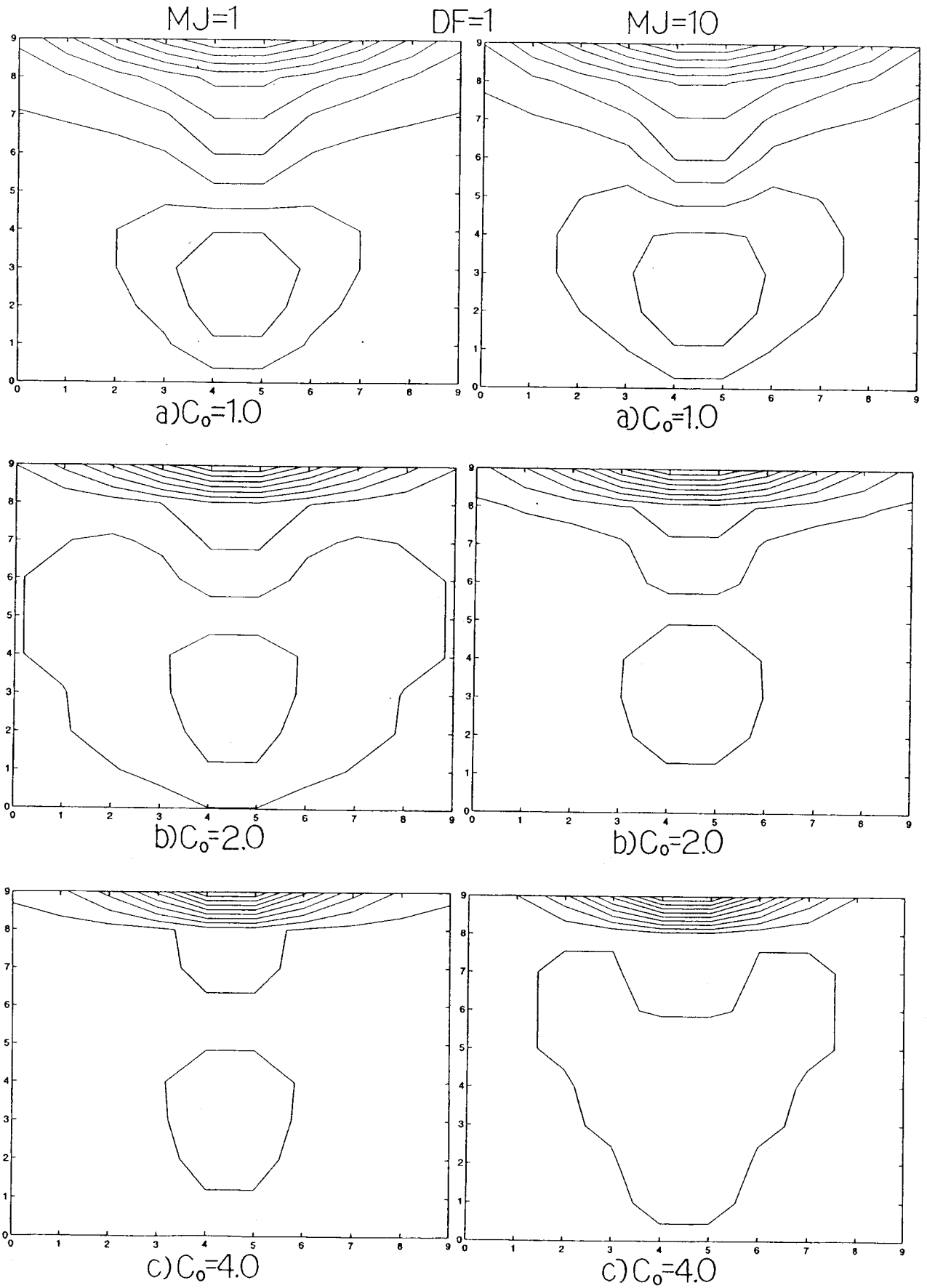
2 $C_0=4.0, MJ=1$

```
% UMIN, (x,y)= -1.344242588926947 (6 ,10 )
% UMAX, (x,y)= 0.3786458762520429 (6 ,9 )
z=[
  0.00  0.00  0.00  0.00 -0.01 -0.01  0.00  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.00  0.03  0.03  0.00  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.01  0.07  0.07  0.01  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.03  0.11  0.11  0.03  0.00  0.00  0.00 ;
  0.00  0.00  0.01  0.04  0.11  0.11  0.04  0.01  0.00  0.00 ;
  0.00  0.00  0.01  0.03  0.04  0.04  0.03  0.01  0.00  0.00 ;
  0.01  0.02  0.04  0.06  0.05  0.05  0.06  0.04  0.02  0.01 ;
 -0.01 -0.03 -0.05 -0.09 -0.17 -0.17 -0.09 -0.05 -0.03 -0.01 ;
  0.04  0.11  0.20  0.31  0.38  0.38  0.31  0.20  0.11  0.04 ;
 -0.11 -0.27 -0.53 -0.93 -1.34 -1.34 -0.93 -0.53 -0.27 -0.11 ;
];
```

$C_0=4.0, MJ=10$

```
% UMIN, (x,y)= -0.4861549640081257 (55 ,100 )
% UMAX, (x,y)= 5.918694119699087e-02 (55 ,39 )
z=[
  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.00  0.01  0.01  0.00  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.00  0.03  0.03  0.00  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.01  0.05  0.05  0.01  0.00  0.00  0.00 ;
  0.00  0.00  0.00  0.02  0.04  0.04  0.02  0.00  0.00  0.00 ;
  0.00  0.00  0.01  0.02  0.03  0.03  0.02  0.01  0.00  0.00 ;
  0.00  0.00  0.01  0.01  0.00  0.00  0.01  0.01  0.00  0.00 ;
  0.00  0.00  0.01  0.01 -0.02 -0.02  0.01  0.01  0.00  0.00 ;
  0.00  0.00  0.00  0.00 -0.02 -0.02  0.00  0.00  0.00  0.00 ;
 -0.01 -0.04 -0.12 -0.29 -0.46 -0.46 -0.29 -0.12 -0.04 -0.01 ;
];
```

$C_0=4.0$ の時の $MJ=1$ の数値を見ると、上側のノイマン境界から下へ向かって正負交互の温度 u 分布が現われている事が分かる。これは、同じ $C_0(=4.0)$ の $MJ=10$ の数値を見てもおかしく、セルペクレ数が $4(\geq 2)$ である時に中心差分を使ったことに起因するトラブルである。おかしい $MJ=1$ の結果では、 u_{min} のすぐ下に、 u_{max} がきている。このような、セルペクレ数が 2 を越えた時のトラブルを避けるための、いわば決定版として現れたのが指数法である(次節で解説)。この節の最後に、 C_0 を $C_0=1.0, 2.0, 4.0, 6.0, 10.0$ と変え、 $MJ=1$ と $MJ=10$ とした場合の内、セルペクレ数が $4.0, 6.0, 10.0$ に対応する $MJ=1$ の場合だけが、 u_{min} のすぐ下に u_{max} がくる等の、おかしい結果になることを見た。その対策として、実際に指数差分で解いた結果を等高線で見ると、次の Fig.10 のようになる。



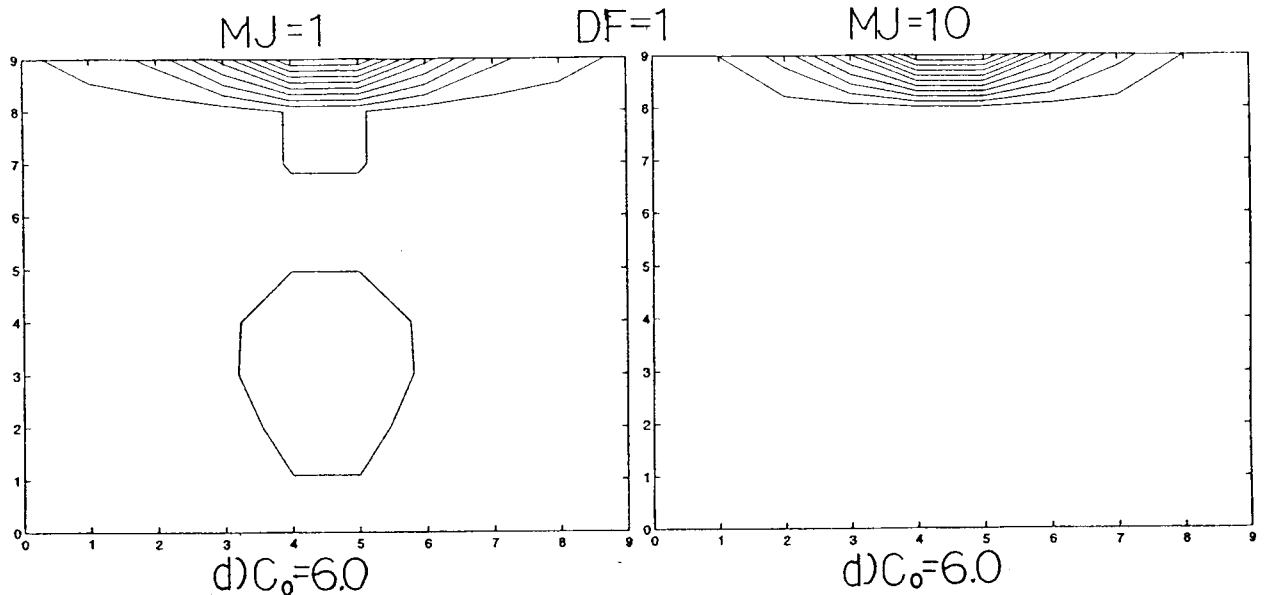


Fig.10 u_i distribution by Exp.method(DF=1.0, MJ=1, 10; for $C_0=1.0, 2.0, 4.0, 6.0$)
 この Fig.10 を見ると、中心差分では MJ=1 の $C_0=2.0$:セルペクレ数 2 以上で、MJ=10 の結果と比べおかしな結果となっていたが (Fig.9)、指数法では明らかに改善されており、精しくした MJ=10 (これも指数法) と相当な等高線になっている。 $C_0=4.0$ の時の MJ=1 と 10 の数値を見てみると (格子点上だけ)、次になる。

1 $C_0=4.0, MJ=1$

```
% UMIN, (x,y) = -0.5677466701333758 (5 ,10 )
% UMAX, (x,y) = 0.1073625283559051 (6 ,4 )
z=[
  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
  0.00 0.00 0.00 0.01 0.04 0.04 0.01 0.00 0.00 0.00 ;
  0.00 0.00 0.00 0.02 0.08 0.08 0.02 0.00 0.00 0.00 ;
  0.00 0.00 0.01 0.03 0.11 0.11 0.03 0.01 0.00 0.00 ;
  0.00 0.00 0.01 0.04 0.09 0.09 0.04 0.01 0.00 0.00 ;
  0.00 0.01 0.02 0.04 0.04 0.04 0.04 0.02 0.01 0.00 ;
  0.00 0.01 0.02 0.03 0.00 0.00 0.03 0.02 0.01 0.00 ;
  0.00 0.01 0.01 0.01 -0.04 -0.04 0.01 0.01 0.01 0.00 ;
  0.00 0.01 0.01 0.00 -0.04 -0.04 0.00 0.01 0.01 0.00 ;
  -0.02 -0.06 -0.16 -0.34 -0.57 -0.57 -0.34 -0.16 -0.06 -0.02 ;
];
```

$C_0=4.0, MJ=10$

```
% UMIN, (x,y) = -0.4800142305675685 (55 ,100 )
% UMAX, (x,y) = 5.913429507527311e-02 (55 ,39 )
z=[
  0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
  0.00 0.00 0.00 0.00 0.01 0.01 0.00 0.00 0.00 0.00 ;
  0.00 0.00 0.00 0.00 0.03 0.03 0.00 0.00 0.00 0.00 ;
  0.00 0.00 0.00 0.01 0.05 0.05 0.01 0.00 0.00 0.00 ;
  0.00 0.00 0.00 0.02 0.04 0.04 0.02 0.00 0.00 0.00 ;
  0.00 0.00 0.01 0.02 0.03 0.03 0.02 0.01 0.00 0.00 ;
  0.00 0.00 0.01 0.01 0.00 0.00 0.01 0.01 0.00 0.00 ;
  0.00 0.00 0.01 0.01 -0.02 -0.02 0.01 0.01 0.00 0.00 ;
  0.00 0.00 0.00 0.00 -0.02 -0.02 0.00 0.00 0.00 0.00 ;
  -0.01 -0.04 -0.12 -0.28 -0.45 -0.45 -0.28 -0.12 -0.04 -0.01 ;
];
```

中心差分の時 (前掲) の $C_0=4.0, MJ=1$ と比べると、正負交互の温度値は無くなり、 u_{max} の位置も u_{min} の直下から中央になり、MJ=10 の解と類似のものになっている。MJ=10 の

時、 $C_0=4.0$ (MJ=10):セルペクレ数 0.4 で中心差分と指数差分による結果の差は、

$$\begin{array}{ll} \text{中心差分 } u_{min}: -0.486155 & \text{指数差分 } u_{min}: -0.480014 \quad (1.3\%) \\ u_{max}: 0.059187 & u_{max}: 0.059134 \quad (0.1\%) \end{array}$$

この時、中心差分と指数差分による解の相対誤差は、 u_{min} で 1.3%、 u_{max} で 0.1% である。一方、MJ=1 では、

$$C_0=2:\text{セルペクレ数 2 でも、中心と指数差分の相対誤差は、} u_{min}: 24.7\% \quad u_{max}: 3.8\%$$

$$C_0=1:\text{セルペクレ数 1 でも、中心と指数差分の相対誤差は、} u_{min}: 4.9\% \quad u_{max}: 3.3\%$$

となっている。このように、セルペクレ数が 2 以下の場合でも、指数法は必須であることが見てとれる。前掲の Fig.8: MJ=1, 10 における u_{max} の C_0 依存性 (中心差分) の結果に、指数法の結果を点線で加えると、Fig.11 になる。指数法では、MJ=1 でもセルペクレ数が 2 以上で妥当な結果となっていることが分かる。MJ=10 の時、この図で中心差分と指数差分の結果は重なる。

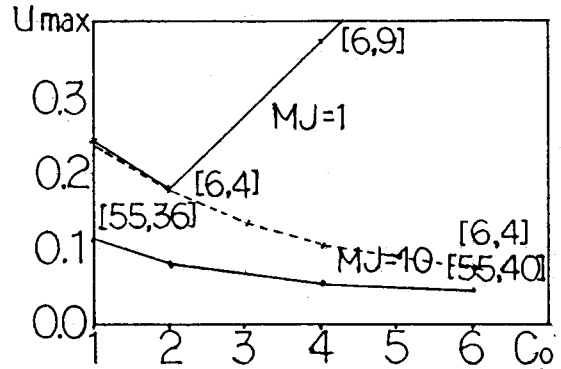


Fig.11 C_0 dependence of u_{max} for (MJ=1, 10) - -: Exp.method

6 指数法による離散化

前節に見た、セルペクレ数が 2 を超える時のトラブルを避けるための、いわば決定版として現れたのが指数法である (指数法による差分式は後述)。半導体デバイス解析の分野では、Sharfetter-Gummel 法と呼ばれる。

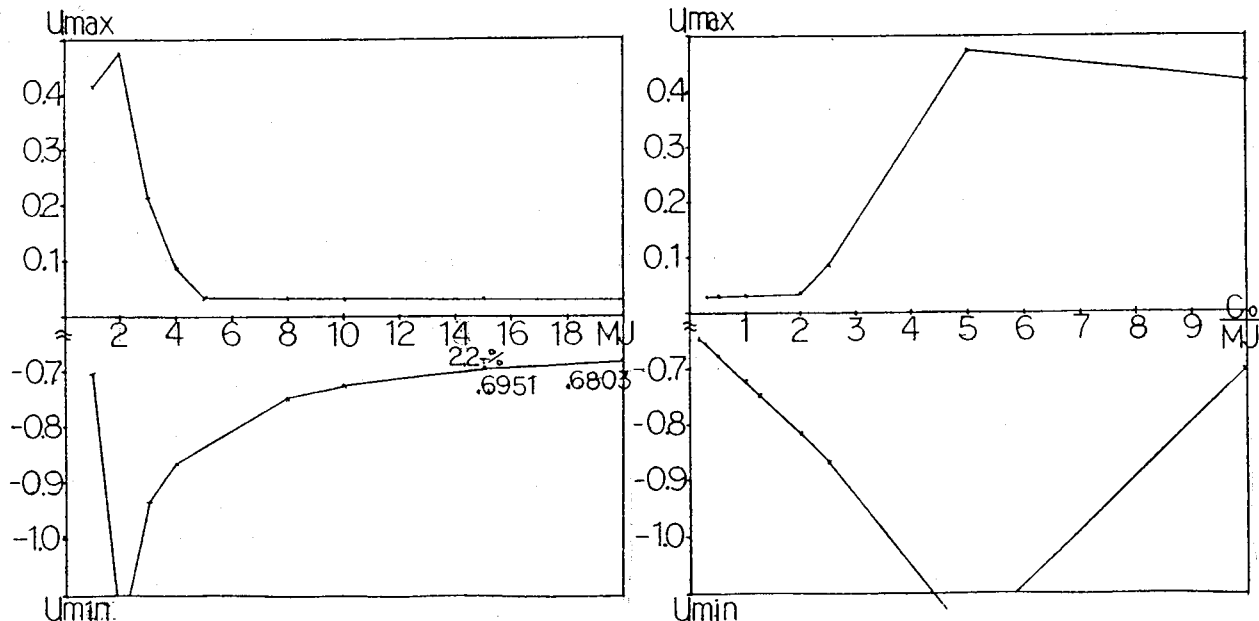


Fig.12a u_{max}, u_{min} distribution For MJ Fig.12b u_{max}, u_{min} dist. For $\frac{C_0}{MJ}$ (DF=1.0, $C_0=10$)
中心差分と指数差分との比較のために、今までの中心差分で $C_0=10$, DF=1 の場合に、

MJ を 1~20 まで変えて、あみ目依存性を見ておく。この時セルペクレ数は、 $\frac{C_0}{MJ}$ なので 10~0.5 まで変わる (2 以下にしなければならない)。数値は Table 8 に、グラフは Fig.12a に示す。この時、横軸にセルペクレ数 $\frac{C_0}{MJ}$ を取ったものが Fig.12b である。両者から、 $MJ < 5, \frac{C_0}{MJ} > 2$ の所から、おかしい振舞いをする事が分かる。

Table 8 u_{min}, u_{max} for MJ (Central: DF=1, $C_0=10$)

MJ	u_{min} (C) [j,k]	u_{max} (C) [j,k]
1	-0.7099 [5,10]	0.4177 [5,9]
2	-1.1812 [11,20]	0.4717 [11,19]
5	-0.8150 [27,50]	0.0352 [28,20]
10	-0.7261 [55,100]	0.0324 [55,40]
20	-0.6804 [110,200]	0.0309 [110,80]

その時、MJ=1, 2 の温度分布 u (MATLAB 用で上下が逆) は次のようになっている。

MJ=1 (中心差分), $C_0=10.0$, DF=1

```

z=[
0.00 0.00 0.00 0.01 0.00 0.00 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
-0.01 -0.01 0.00 0.01 0.05 0.05 0.01 0.00 -0.01 -0.01 ;
0.01 0.01 0.01 -0.01 0.01 0.01 -0.01 0.01 0.01 0.01 ;
-0.01 -0.02 -0.01 0.03 0.12 0.12 0.03 -0.01 -0.02 -0.01 ;
0.02 0.03 0.03 -0.01 -0.07 -0.07 -0.01 0.03 0.03 0.02 ;
-0.03 -0.05 -0.04 0.05 0.18 0.18 0.05 -0.04 -0.05 -0.03 ;
0.04 0.08 0.08 -0.04 -0.27 -0.27 -0.04 0.08 0.08 0.04 ;
-0.06 -0.12 -0.13 0.06 0.42 0.42 0.06 -0.13 -0.12 -0.06 ;
0.09 0.19 0.23 -0.04 -0.71 -0.71 -0.04 0.23 0.19 0.09 ;
];

```

MJ=2 (中心差分), $C_0=10.0$, DF=1

```

z=[
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.01 0.01 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.02 0.02 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.04 0.04 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.03 0.03 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.02 0.02 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 -0.01 -0.04 -0.04 -0.01 0.00 0.00 0.00 ;
0.00 -0.01 -0.03 -0.09 -0.19 -0.19 -0.09 -0.03 -0.01 0.00 ;
-0.02 -0.05 -0.14 -0.48 -1.06 -1.06 -0.48 -0.14 -0.05 -0.02 ;
];

```

本節の後半の指数法との比較で、温度分布の等高線を示すが、中心差分では、 C_0 が同じで MJ: あみ目の精しさが違うだけなのにもかかわらず、セルペクレ数が 10 の MJ=1 では、温度分布 u に正負交互の値が出て明らかにおかしい事が分かる。MJ=2 では、正負交互の値はなくなるが、 u_{max} の位置が、上のノイマン境界上の u_{min} と隣接しており、値もおかしい。MJ=5 (セルペクレ数:2) になって、 u_{max} の位置が+の熱源近くになり、妥当な結果となってくる。それは MJ=10 の結果とも似てくる。 u_{min} の位置は、上のノイマン上にくる。

そこで、セルペクレ数が 2 を超えるような問題では、指数法の出番となる。中心差分による、あみ目依存性 Fig.12a,b は、指数法による計算では、Fig13a,b のように変わる。指

数法の結果は、前掲の図に点線で書き加えてある。

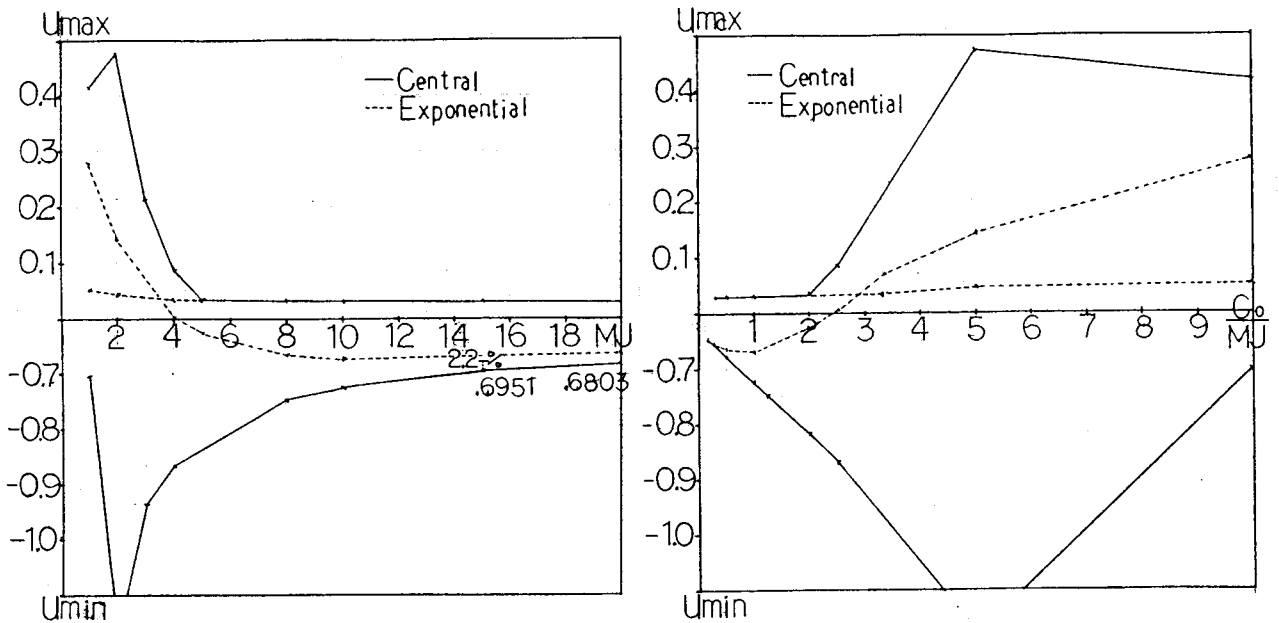


Fig.13a u_{max}, u_{min} distribution For MJ Fig.13b u_{max}, u_{min} dist. For $\frac{C_0}{MJ}$ (DF=1.0, $C_0=10$)
 - -: Exp.method

Fig.13a から、 u_{max} は MJ=5 以上：セルペクレ数 2 以下で、中心差分と指数法（差分）を比べて 1 % 以下の差で良く一致するが、MJ が 5 以下：セルペクレ数が 2 以上では大きく差が開くことが分かる。 u_{min} は、MJ=10：セルペクレ数 1 でも、中心差分と指数法（差分）との差は 8 % もあり不安である。中心差分でも使えるとされる MJ=5：セルペクレ数 2 では、両差分の差が 30 % もあり、解に信頼できない。この場合には、セルペクレ数 2 以下でも、中心差分は信用できない。MJ ≤ 5：セルペクレ数 2 以上ならば、両差分の差は大きく開く。 u_{min} は MJ=20 の時、中心差分と指数法の差が 2 % の差となり、MJ=15 では両差分の差は 3.6 % となる。MJ=20 の時の指数法の結果 u_{min} に対する、MJ=15 の時の中心差分の結果は収束誤差 2.1 % の差で、この時の指数法の結果は収束誤差 0.6 % なので、中心差分でも指数法でも、2 % の精度が欲しければ、MJ=20 (セルペクレ数 0.5) は欲しい。

この状況を、横軸にセルペクレ数をとってプロットし直した図が、Fig.13b となる。指数法の結果は点線で示してある。この図から、セルペクレ数が 2 以上の場合には、中心差分は使えず、指数法に頼らざるを得ないことが分かる。たとえ、中心差分の u_{max} の結果がまともになる、セルペクレ数が 2 以下でも、 u_{min} 値の指数法との差は 30 % もあり、やっとセルペクレ数が 1 になって、指数法との差が 8 % に縮まる。あみ目をセルペクレ数が 1 以下にせよということだ。

指数法のシミュレーション結果は、中心差分の結果 Table 8 に対して、Table 9 のようになる。中心差分でおかしかった結果が、指数法の差分では正常になる。計算は、後のパラメータ UTOP=1.01(UFILU4) とし、BCGstab3(1,3) を使用した。

MJ=2 の時に、 u_{max} の位置が、上のノイマン境界上にある u_{min} の位置のすぐ下に隣接して正負交互の値を取るおかしな結果は、 u_{max} の位置が中央の少し下にくるので、妥当な結果に近づいている。両差分で、CPU 時間、BCGstab3 の収束回数は、ほぼ同じで、差分に

よる計算負担は、今のところ気にする必要はない。また、 $\nu = \frac{\mu}{d}$ が一定でない場合は、指数法の BER() の計算が2倍に増えるが、今のところまだ、BER() の計算負担を気にする必要はない。

Table 9 u_{min}, u_{max} for MJ (Exp.: DF=1, $C_0=10$)

MJ	u_{min} (C) [j,k]	u_{max} (C) [j,k]
1	-0.3232 [5,10]	0.0509 [6,4]
2	-0.4555 [11,20]	0.0428 [11,8]
5	-0.6243 [27,50]	0.0349 [28,20]
10	-0.6719 [55,100]	0.0324 [55,40]
20	-0.6667 [110,200]	0.0309 [110,80]

中心差分で温度分布 u に正負交互の値が現れたり、 u_{max} の位置が不自然 (u_{min} のすぐ下にくる) だった、MJ=1, 2 の数値は、指数法では次のように落ち着く。

MJ=1 (指数法), $C_0=10.0$, DF=1

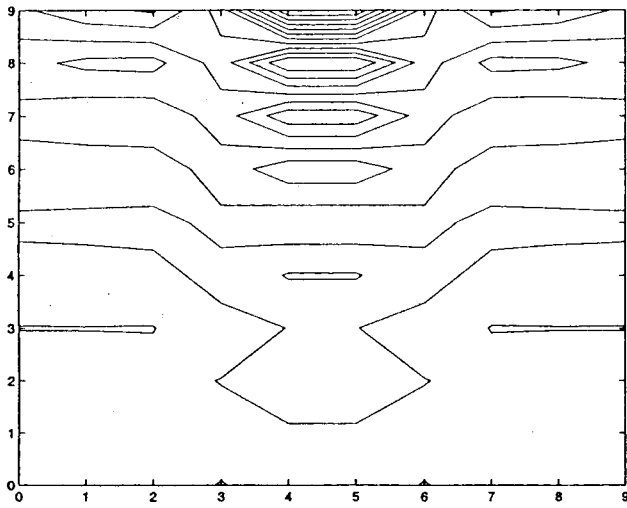
```
z=[
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.02 0.02 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.04 0.04 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.05 0.05 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.05 0.05 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.03 0.03 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.01 0.01 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 -0.01 -0.01 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 -0.01 -0.01 0.01 0.00 0.00 0.00 ;
0.00 -0.01 -0.03 -0.12 -0.32 -0.32 -0.12 -0.03 -0.01 0.00 ;
];
```

MJ=2 (指数法), $C_0=10.0$, DF=1

```
z=[
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.01 0.01 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.02 0.02 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 0.04 0.04 0.00 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.03 0.03 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.02 0.02 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 0.00 0.00 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.01 -0.01 -0.01 0.01 0.00 0.00 0.00 ;
0.00 0.00 0.00 0.00 -0.01 -0.01 0.00 0.00 0.00 0.00 ;
0.00 0.00 -0.03 -0.15 -0.40 -0.40 -0.15 -0.03 0.00 0.00 ;
];
```

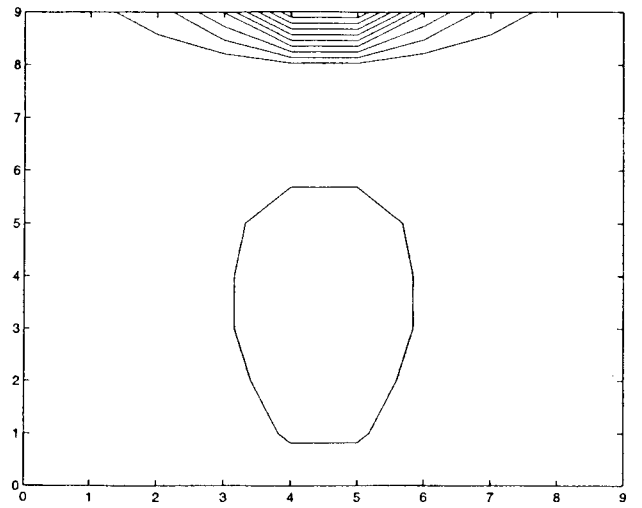
中心差分で結果がおかしかった MJ=1, 2 も含めて、MJ=1, 2, 5, 10 の時：セルペクレ数で 10, 5, 2, 1 ($C_0=10$) の場合に温度分布を等高線表示すると、Fig.14 となる。左側が中心差分の結果で、右側が指数法による結果を示す。指数法によれば、等高線からもセルペクレ数によらず、妥当な結果が得られていることが分かる。

中心差分

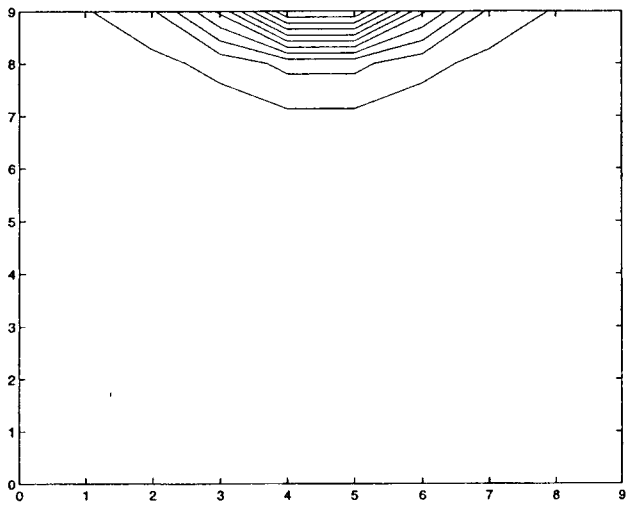


a) MJ=1

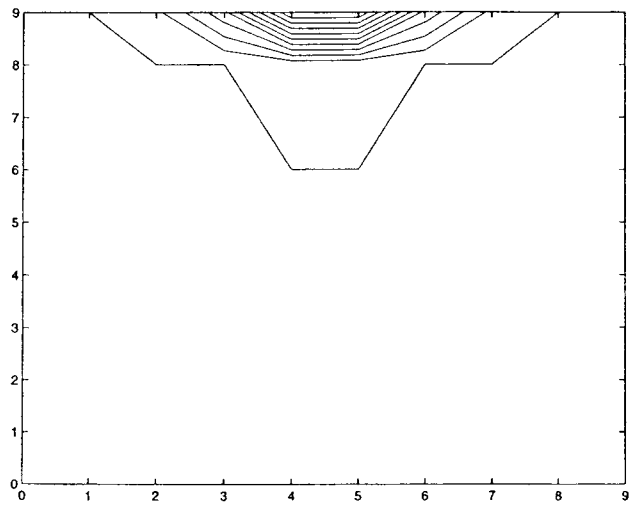
指数法



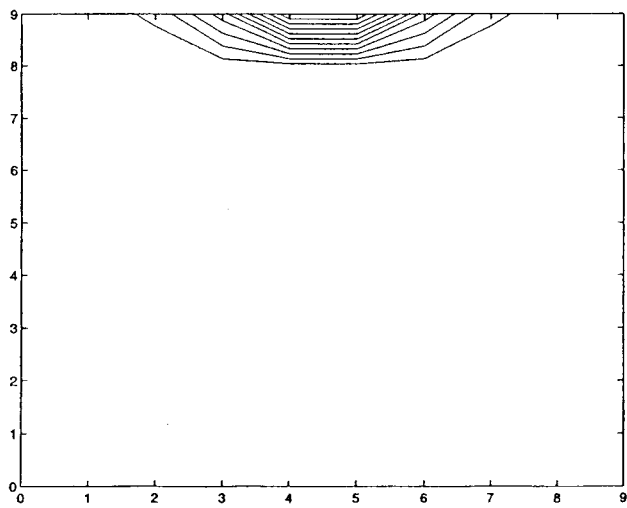
a) MJ=1



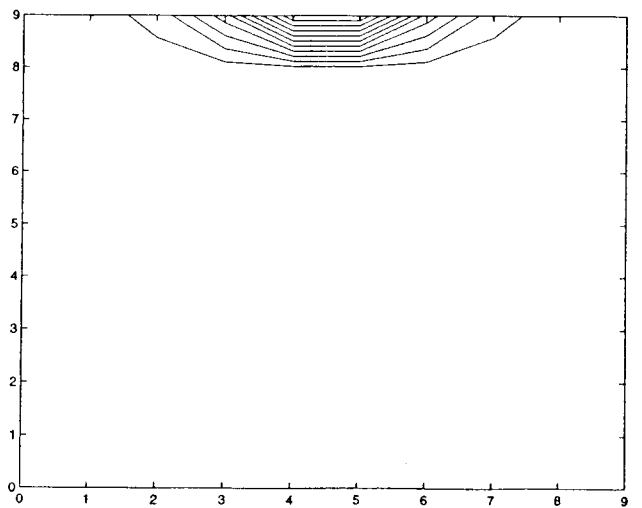
b) MJ=2



b) MJ=2



c) MJ=5



c) MJ=5

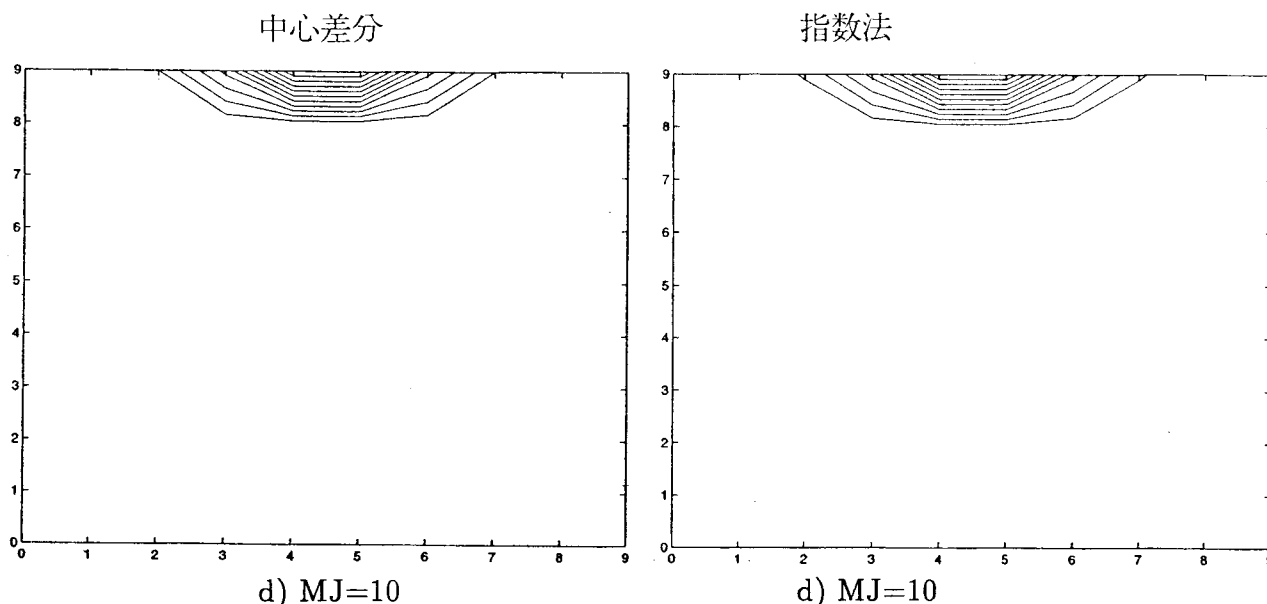


Fig.14 u_i distribution by Cent.(L) and Exp.(R) method (DF=1.0, $C_0=10.0$;
for MJ=1, 2, 5, 10)

これまで見たように、セルペクレ数が 2 を超えるような問題で必須となる、指数法に基づく離散化について、次に説明する。

$div(-d\nabla u + \mu b u) = f$ を相手にする。

この式の x 成分を考え、 $\frac{d}{dx}(-d\frac{du}{dx} + \mu b u) = 0$, $u(0) = u_0$, $u(h) = u_h$ とすると、 $\omega = \frac{\mu b}{d}$ とおく時、

$$(6.1) \quad u(x) = u_0 + (u_h - u_0) \frac{\exp(\omega x) - 1}{\exp(\omega h) - 1}$$

の解析解がある。この式から、flux を計算すると、

$$(6.2) \quad -d\frac{du}{dx} + \mu b u = \mu b \left[u_0 - \frac{u_h - u_0}{\exp(\omega h) - 1} \right] = \mu b \left[u_0 - \frac{(u_h - u_0)B(\omega h)}{\omega h} \right]$$

ここで、 $B(z)$ はベルヌーイ関数で、 $B(z) = \frac{z}{e^z - 1}$, $1 + \frac{B(z)}{z} = \frac{B(-z)}{z}$
この flux を使って、前掲の Fig.2 の小領域 PQRS の周囲を γ に見立てて、移流拡散系の熱方程式 (積分系) :

$$(6.3) \quad \int_{\Gamma} (-d\nabla u + \mu b u) \cdot \mathbf{n} \, ds = \int_{\Omega} f \, dv$$

の CV 法による離散式を作る。左辺の線積分は次式のように細分化できる。

$$(6.4) \quad \int_{\Gamma} (-d\nabla u + \mu b u) \cdot \mathbf{n} \, ds = \int_{PQ} + \int_{QR} + \int_{RS} + \int_{SP}$$

右辺の面積分は純拡散の場合と同様に次式として始末し、節点 i での f_i を求める。

$$(6.5) \quad f_i = \int_{\Omega} f \, dv = f_P \frac{hx_-}{2} \frac{hy_-}{2} + f_Q \frac{hx_+}{2} \frac{hy_-}{2} + f_R \frac{hx_+}{2} \frac{hy_+}{2} + f_S \frac{hx_-}{2} \frac{hy_+}{2} \quad \left[\frac{cal}{sec} \right]$$

すべての左辺の線積分は、 $\omega_P = \frac{\mu_P b_y^T}{d_P}$ などとして次のように書ける。 (d_P, μ_P) は、点 P を中点とする面口 $u_i, u_{i-m}, u_{i-m-1}, u_{i-1}$ の拡散係数 (d) と、移動度 (μ) を表わし、 b_y^T は、境界 PQ を流れる移流ベクトル: $\mathbf{b} = [b_x^T, b_y^T]^T$ の y 成分を表わす。

$$(6.6) \quad \int_{PQ} = \int_{PT} + \int_{TQ} = \mu_P b_y^T \left[u_{i-1} - \frac{u_i - u_{i-1}}{\exp(\omega_{Py} h y_-) - 1} \right] \cdot [0 \quad -1]^T \frac{h x_-}{2} + \\ \mu_Q b_y^T \left[u_{i-1} - \frac{u_i - u_{i-1}}{\exp(\omega_{Qy} h y_-) - 1} \right] \cdot [0 \quad -1]^T \frac{h x_+}{2} \\ = -\mu_P b_y^T \left[u_{i-1} - \frac{(u_i - u_{i-1}) B(\omega_{Py} h y_-)}{\omega_{Py} h y_-} \right] \cdot \frac{h x_-}{2} - \mu_Q b_y^T \left[u_{i-1} - \frac{(u_i - u_{i-1}) B(\omega_{Qy} h y_-)}{\omega_{Qy} h y_-} \right] \cdot \frac{h x_+}{2}$$

$$(6.7) \quad \int_{QR} = \int_{QU} + \int_{UR} = \mu_Q b_x^U \left[u_i - \frac{u_{i+m} - u_i}{\exp(\omega_{Qx} h x_+) - 1} \right] \cdot [1 \quad 0]^T \frac{h y_-}{2} + \\ \mu_R b_x^U \left[u_i - \frac{u_{i+m} - u_i}{\exp(\omega_{Rx} h x_+) - 1} \right] \cdot [1 \quad 0]^T \frac{h y_+}{2} \\ = \mu_Q b_x^U \left[u_i - \frac{(u_{i+m} - u_i) B(\omega_{Qx} h x_+)}{\omega_{Qx} h x_+} \right] \cdot \frac{h y_-}{2} + \mu_R b_x^U \left[u_i - \frac{(u_{i+m} - u_i) B(\omega_{Rx} h x_+)}{\omega_{Rx} h x_+} \right] \cdot \frac{h y_+}{2} \\ = [\mu_Q b_x^U u_i - d_Q \frac{(u_{i+m} - u_i)}{h x_+} B(\omega_{Qx} h x_+)] \cdot \frac{h y_-}{2} + [\mu_R b_x^U u_i - d_R \frac{(u_{i+m} - u_i)}{h x_+} B(\omega_{Rx} h x_+)] \cdot \frac{h y_+}{2}$$

ここで、 $\omega_{Qx} = \frac{\mu_Q b_x^U}{d_Q}$, $\omega_{Rx} = \frac{\mu_R b_x^U}{d_R}$, $\nu_Q = \frac{\mu_Q}{d_Q}$ など。

$$(6.8) \quad \int_{RS} = \int_{RV} + \int_{VS} = \mu_R b_y^V \left[u_i - \frac{u_{i+1} - u_i}{\exp(\omega_{Ry} h y_+) - 1} \right] \cdot [0 \quad 1]^T \frac{h x_+}{2} + \\ \mu_S b_y^V \left[u_i - \frac{u_{i+1} - u_i}{\exp(\omega_{Sy} h y_+) - 1} \right] \cdot [0 \quad 1]^T \frac{h x_-}{2} \\ = \mu_R b_y^V \left[u_i - \frac{(u_{i+1} - u_i) B(\omega_{Ry} h y_+)}{\omega_{Ry} h y_+} \right] \cdot \frac{h x_+}{2} + \mu_S b_y^V \left[u_i - \frac{(u_{i+1} - u_i) B(\omega_{Sy} h y_+)}{\omega_{Sy} h y_+} \right] \cdot \frac{h x_-}{2}$$

$$(6.9) \quad \int_{SP} = \int_{SW} + \int_{WP} = \mu_S b_x^W \left[u_{i-m} - \frac{u_i - u_{i-m}}{\exp(\omega_{Sx} h x_-) - 1} \right] \cdot [-1 \quad 0]^T \frac{h y_+}{2} + \\ \mu_P b_x^W \left[u_{i-m} - \frac{u_i - u_{i-m}}{\exp(\omega_{Px} h x_-) - 1} \right] \cdot [-1 \quad 0]^T \frac{h y_-}{2} \\ = -\mu_S b_x^W \left[u_{i-m} - \frac{(u_i - u_{i-m}) B(\omega_{Sx} h x_-)}{\omega_{Sx} h x_-} \right] \cdot \frac{h y_+}{2} - \mu_P b_x^W \left[u_{i-m} - \frac{(u_i - u_{i-m}) B(\omega_{Px} h x_-)}{\omega_{Px} h x_-} \right] \cdot \frac{h y_-}{2}$$

以上の式から、「3 移流拡散系の (中心差分による) 解法」のように、式 (6.4) の右辺 4 項全部を書き下だし、 $u_{i-m}, u_{i-1}, u_i, u_{i+1}, u_{i+m}$ 項毎に整理すれば、第 i 番 (格子点番号) 方程式:

$$(6.10) \quad a_{i,i-m} u_{i-m} + a_{i,i-1} u_{i-1} + a_{i,i} u_i + a_{i,i+1} u_{i+1} + a_{i,i+m} u_{i+m} = f_i$$

ができる。この時、式(6.10)の左辺の係数は、次式となる。

$$\begin{aligned}
a_{i,i} &= \frac{1}{2} \left[d_P \frac{hx_-}{hy_-} B(\omega_{P_y} hy_-) + d_Q \frac{hx_+}{hy_-} B(\omega_{Q_y} hy_-) \right] \\
&+ \frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} B(-\omega_{Q_x} hx_+) + d_R \frac{hy_+}{hx_+} B(-\omega_{R_x} hx_+) \right] \\
&+ \frac{1}{2} \left[d_R \frac{hx_+}{hy_+} B(-\omega_{R_y} hy_+) + d_S \frac{hx_-}{hy_+} B(-\omega_{S_y} hy_+) \right] \\
&+ \frac{1}{2} \left[d_S \frac{hy_+}{hx_-} B(\omega_{S_x} hx_-) + d_P \frac{hy_-}{hx_-} B(\omega_{P_x} hx_-) \right] \\
a_{i,i-m} &= -\frac{1}{2} \left[d_S \frac{hy_+}{hx_-} B(-\omega_{S_x} hx_-) + d_P \frac{hy_-}{hx_-} B(-\omega_{P_x} hx_-) \right] \\
a_{i,i-1} &= -\frac{1}{2} \left[d_P \frac{hx_-}{hy_-} B(-\omega_{P_y} hy_-) + d_Q \frac{hx_+}{hy_-} B(-\omega_{Q_y} hy_-) \right] \\
a_{i,i+1} &= -\frac{1}{2} \left[d_R \frac{hx_+}{hy_+} B(\omega_{R_y} hy_+) + d_S \frac{hx_-}{hy_+} B(\omega_{S_y} hy_+) \right] \\
(6.11) \quad a_{i,i+m} &= -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} B(\omega_{Q_x} hx_+) + d_R \frac{hy_+}{hx_+} B(\omega_{R_x} hx_+) \right]
\end{aligned}$$

$$\begin{aligned}
\omega_{R_y} &= \frac{\mu_R b_y^V}{d_R}, \quad \omega_{S_y} = \frac{\mu_S b_y^V}{d_S}, \quad \omega_{Q_x} = \frac{\mu_Q b_x^U}{d_Q}, \quad \omega_{R_x} = \frac{\mu_R b_x^U}{d_R}, \quad \omega_{P_y} = \frac{\mu_P b_y^T}{d_P}, \quad \omega_{Q_y} = \frac{\mu_Q b_y^T}{d_Q}, \\
\omega_{S_x} &= \frac{\mu_S b_x^W}{d_S}, \quad \omega_{P_x} = \frac{\mu_P b_x^W}{d_P}, \quad \nu_R = \frac{\mu_R}{d_R}, \quad \nu_S = \frac{\mu_S}{d_S}, \quad \nu_Q = \frac{\mu_Q}{d_Q}, \quad \nu_P = \frac{\mu_P}{d_P} \quad \text{など。}
\end{aligned}$$

(また、Fig.2 の境界 PQ を流れる移流ベクトル: $\mathbf{b} = [b_x^T \ b_y^T]^T$, 境界 RS を流れる $\mathbf{b} = [b_x^V \ b_y^V]^T$, 境界 SP を流れる $\mathbf{b} = [b_x^W \ b_y^W]^T$, 境界 QR を流れる $\mathbf{b} = [b_x^U \ b_y^U]^T$)

ここで、係数 $a_{i,t}$ は、行列 $\mathbf{A}(a_{i,t})$ の要素と対応しており、行 i は前掲 Fig.1 の場の格子点番号 $i (= m \times (j-1) + k; j = 1, m, k = 1, m)$ を表わす。

例えば、式(6.11)の係数 $a_{i,i+m}$ の式において、ベルヌーイ関数 $B(z)$ のベキ級数展開式:

$$(6.12) \quad B(z) = \frac{z}{e^z - 1} \simeq 1 - \frac{z}{2} + \frac{z^2}{12} - \dots$$

の内、第2項まで: $B(z) \simeq 1 - \frac{z}{2}$ を使って代入した式:

$$(6.13) \quad a_{i,i+m} = -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} \left(1 - \frac{\omega_{Q_x} hx_+}{2}\right) + d_R \frac{hy_+}{hx_+} \left(1 - \frac{\omega_{R_x} hx_+}{2}\right) \right]$$

は、中心差分式(3.7)の $a_{i,i+m}$ の式と同式となる。中心差分は、指数法の近似式になっている。中心差分で近似できる(セルペクレ数が2以下)範囲で、 $b_y^T = b_y^V$, $b_x^W = b_x^U$, $\nu = \frac{\mu}{d} = C_0$ 一定、 $d =$ 一定、等間隔($hx = hy$) ならば、

$$a_{i,i} = -(a_{i,i-1} + a_{i,i+1} + a_{i,i-m} + a_{i,i+m})$$

となり行対角優位となり得る。行対角優位ならば、 \mathbf{A} が正則で、軸選択なしのガウスで、

終りまでLU分解できるなどの恩恵が期待できる。 $b_y^T \neq b_y^V$, $b_x^W \neq b_x^U$ でも、メッシュを細かくするなどして、ズレ(差)が小さいならば、行対角優位性のズレも小さくなる。

ベルヌーイ関数 $B(z)$ のプログラム上における扱いを説明する。ベルヌーイ関数で、 $z=0$ は除去可能の特異点で、式(6.12)のべき級数展開式から、 $B(0) = 1$ と定義し直す。ベルヌーイ関数の計算は、特異点近辺の狭い範囲では、式(6.12)を使い、次のような Function BER4(Z)で行なう。

FUNCTION BER4(z)

if($z < -10^{-2}$) then BER4(z) = $\frac{z}{e^z - 1}$

if($-10^{-2} < z < 10^{-2}$) then BER4(z) = $1 - (0.5 - 0.0833333 \cdot z) \cdot z$

if($z > 10^{-2}$) then BER4(z) = $\frac{z \cdot e^{-z}}{1 - e^{-z}}$; 計算精度を上げるため

このBER4(z)の計算と、中心差分 $Bcent(z) = 1 - 0.5z$ による計算値の差は、Table 10 のようになる。

Table 10 $B(z)$:exp, $1 - 0.5z$:cent for z

z	B(z) :exp	1-0.5z :cent
-4.0	4.075	3.0
-2.0	2.313	0.0
-1.0	1.582	1.5
-0.5	1.271	1.25(1.6 %)
-0.25	1.130	1.125(0.44 %)
-10^{-2}	1.005	1.005
0.0	1.0	1.0
10^{-2}	0.995	0.995
0.25	0.880	0.875(0.56 %)
0.5	0.771	0.750(0.77 %)
1.0	0.582	0.5
2.0	0.313	0.0
4.0	0.075	-1

ここで、ベルヌーイ関数 $B(z)$ の引数 z (特に y 方向) は、 $hy = \frac{1}{MJ}$, $\mathbf{b} = (bx, by)^T = (0, 1)^T$ を使って、

$$z = \omega \cdot hy = \frac{\mu \cdot by}{d} \cdot hy = C_0 \cdot by \cdot hy = \frac{C_0}{MJ} \cdot by = \text{セルペクレ数} \cdot by$$

となり、 z はセルペクレ数と等しい ($by=1$)。Table 10 のように、ベルヌーイ関数 $B(z)$ と、中心差分に対応する $B(z) \simeq 1 - 0.5z$ のグラフは、Fig.15 となる。セルペクレ数(絶対値)が 2 を超える、すなわち $|z|$ が 2 を超える範囲では、ベルヌーイ関数値と中心差分による値の差は大きく開くことが分かる。

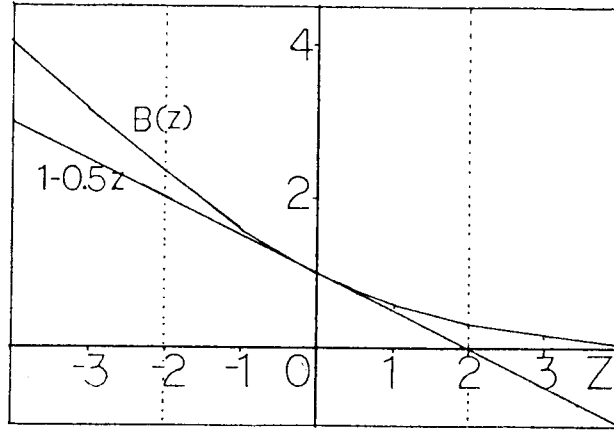


Fig.15 Plot of $B(z)$:exp and $1 - 0.5z$:cent

今、 $\nu = \frac{\mu}{d} = C_0 = \text{一定}$ 、等間隔あみ目： $hx_{\pm} = hy_{\pm} = h$ として、さらに $\mathbf{b} = (bx, by)^T = (0, 1)^T$ ($: b_y^T = b_y^V = 1$)、Fig.1 の両側以外は、拡散係数 d に段差なし ($d_P = d_Q = d_R = d_S = 1[\frac{\text{cal}}{C \cdot \text{sec} \cdot \text{cm}}]$) を使うと式 (6.11) の各係数は次のように簡単になる。

$$\begin{aligned} a_{i,i} &= \frac{1}{2} [B(\omega_{P_y}h) + B(\omega_{Q_y}h) + B(-\omega_{R_y}h) + B(-\omega_{S_y}h) + 4 \cdot B(0)] \\ &= \frac{1}{2} [2 \cdot B(C_0h) + 2 \cdot B(-C_0h) + 4] \quad \text{ここで } C_0h \text{ はセルペクレ数} \end{aligned}$$

$$a_{i,i-m} = -\frac{1}{2} [B(0) + B(0)] = -1$$

$$a_{i,i-1} = -\frac{1}{2} [B(-\omega_{P_y}h) + B(-\omega_{Q_y}h)] = -\frac{1}{2} [B(-C_0h) + B(-C_0h)] = -B(-C_0h)$$

$$a_{i,i+1} = -\frac{1}{2} [B(\omega_{R_y}h) + B(\omega_{S_y}h)] = -\frac{1}{2} [B(C_0h) + B(C_0h)] = -B(C_0h)$$

$$(6.14) \quad a_{i,i+m} = -\frac{1}{2} [B(0) + B(0)] = -1$$

また、Fig.1 の両側の拡散係数 d には段差をつけている ($d = DF$) ので、式 (6.14) には、その始末が必要となる。 $\frac{\mu}{DF} = C_0$ となる。その時、if ($DF = 0.0$) then $C_0h = 0.0$ として、左サイドの係数は、

$$\begin{aligned} a_{i,i} &= \frac{1}{2} [DF \cdot B(C_0h) + B(C_0h) + B(0) + B(0)] \\ &\quad + \frac{1}{2} [B(-C_0h) + DF \cdot B(-C_0h) + DF \cdot B(0) + DF \cdot B(0)] \\ &= \frac{1}{2} [2 + 2 \cdot DF + DF \cdot B(C_0h) + B(C_0h) + B(-C_0h) + DF \cdot B(-C_0h)] \end{aligned}$$

$$a_{i,i-m} = 0.0 \quad a_{i,i-1} = -\frac{1}{2} [DF \cdot B(-C_0h) + B(-C_0h)]$$

$$(6.15) \quad a_{i,i+1} = -\frac{1}{2} [B(C_0h) + DF \cdot B(C_0h)] \quad a_{i,i+m} = -\frac{1}{2} [B(0) + B(0)] = -1$$

となる。ここで、「3 移流拡散系の解法(中心差分)」の場合と同様に、節点*i*に隣あう点*i-m*には、固定境界があるので、違った扱いをする必要がある。例えば、左側に固定境界 $u = u_0 (= 0)$ を持つ *E* 点 (Fig.1) の場合には、*i* 番方程式が、

$$(6.16) \quad a_{i,i-m}u_0 + a_{i,i-1}u_{i-1} + a_{i,i}u_i + a_{i,i+1}u_{i+1} + a_{i,i+m}u_{i+m} = f_i$$

となる。この時、未知数項でない $a_{i,i-m}u_0$ は、右辺に移項して f_i の仲間に入れる。したがって、

$$(6.17) \quad a_{i,i-1}u_{i-1} + a_{i,i}u_i + a_{i,i+1}u_{i+1} + a_{i,i+m}u_{i+m} = f_i - a_{i,i-m}u_0$$

を解くことになるので、式 (6.15) において係数 $a_{i,i-m}$ は、

$$a_{i,i-m} = 0.0$$

と設定してある。 $u_0 = 0$ ならば、なおさら簡単で、右辺に手を加える必要はなくなる。

式 (6.15) と同様に、拡散係数 d に段差をつける右サイドの係数は、次式となる。

$$\begin{aligned} a_{i,i} &= \frac{1}{2} [B(C_0h) + DF \cdot B(C_0h) + DF \cdot B(0) + DF \cdot B(0)] \\ &\quad + \frac{1}{2} [DF \cdot B(-C_0h) + B(-C_0h) + B(0) + B(0)] \\ &= \frac{1}{2} [2 + 2 \cdot DF + B(C_0h) + DF \cdot B(C_0h) + DF \cdot B(-C_0h) + B(-C_0h)] \\ a_{i,i-m} &= -\frac{1}{2} [B(0) + B(0)] = -1 \quad a_{i,i-1} = -\frac{1}{2} [B(-C_0h) + DF \cdot B(-C_0h)] \\ (6.18) \quad a_{i,i+1} &= -\frac{1}{2} [DF \cdot B(C_0h) + B(C_0h)] \quad a_{i,i+m} = 0.0 \end{aligned}$$

これら係数も、上側のノイマン境界の節点 (Fig.1 CD 上) では、左辺の積分区間を Fig.2 の下半分だけにする必要があり、扱いが異なる。線積分は、

$$(6.19) \quad \int_{\Gamma_{CD}} (-\kappa \nabla u) \cdot \mathbf{n} \, ds = \int_{WP} + \int_{PQ} + \int_{QU}$$

となり、Fig.1 CD 上の節点では、係数は、式 (6.11) と同様の手順を経て次式となる。

$$\begin{aligned} a_{i,i} &= \frac{1}{2} \left[d_P \frac{hx_-}{hy_-} B(\omega_{P_y} hy_-) + d_Q \frac{hx_+}{hy_-} B(\omega_{Q_y} hy_-) \right] \\ &\quad + \frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} B(-\omega_{Q_x} hx_+) \right] + \frac{1}{2} \left[d_P \frac{hy_-}{hx_-} B(\omega_{P_x} hx_-) \right] \\ &= \frac{1}{2} [B(C_0h) + B(C_0h) + B(0) + B(0)] = \frac{1}{2} [2 + 2 \cdot B(C_0h)] \\ a_{i,i-m} &= -\frac{1}{2} \left[d_P \frac{hy_-}{hx_-} B(-\omega_{P_x} hx_-) \right] = -\frac{1}{2} [B(0)] = -0.5 \\ a_{i,i-1} &= -\frac{1}{2} \left[d_P \frac{hx_-}{hy_-} B(-\omega_{P_y} hy_-) + d_Q \frac{hx_+}{hy_-} B(-\omega_{Q_y} hy_-) \right] \\ &= -\frac{1}{2} [B(-C_0h) + B(-C_0h)] = -\frac{1}{2} [2 \cdot B(-C_0h)] \\ (6.20) \quad a_{i,i+1} &= 0.0 \quad a_{i,i+m} = -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} B(\omega_{Q_x} hx_+) \right] = -\frac{1}{2} [B(0)] = -0.5 \end{aligned}$$

$$= -\frac{1}{2} [B(-C_0h) + B(-C_0h)] = -\frac{1}{2} [2 \cdot B(-C_0h)]$$

$$(6.20) \quad a_{i,i+1} = 0.0 \quad a_{i,i+m} = -\frac{1}{2} \left[d_Q \frac{hy_-}{hx_+} B(\omega_{Qx} hx_+) \right] = -\frac{1}{2} [B(0)] = -0.5$$

このように、左辺の積分区間を Fig.2 の下半分だけにする、上側のノイマン境界の節点 (Fig.1 CD 上) の両端の 2 点については、両側の拡散係数 d に段差をつけているので、特別に扱う必要がある。Fig.1 CD 上の左サイドの点については、係数は次のようになる。

$$a_{i,i} = \frac{1}{2} [DF \cdot B(C_0h) + B(C_0h) + B(0) + DF \cdot B(0)]$$

$$= \frac{1}{2} [DF \cdot B(C_0h) + B(C_0h) + 1 + DF]$$

$$a_{i,i-m} = 0.0 \quad a_{i,i-1} = -\frac{1}{2} [DF \cdot B(-C_0h) + B(-C_0h)]$$

$$(6.21) \quad a_{i,i+1} = 0.0 \quad a_{i,i+m} = -\frac{1}{2} [B(0)] = -0.5$$

Fig.1 CD 上の右サイドの点の係数は、次式となる。

$$a_{i,i} = \frac{1}{2} [B(C_0h) + DF \cdot B(C_0h) + DF \cdot B(0) + B(0)]$$

$$= \frac{1}{2} [B(C_0h) + DF \cdot B(C_0h) + DF + 1]$$

$$a_{i,i-m} = -\frac{1}{2} [B(0)] = -0.5 \quad a_{i,i-1} = -\frac{1}{2} [B(-C_0h) + DF \cdot B(-C_0h)]$$

$$(6.22) \quad a_{i,i+1} = 0.0 \quad a_{i,i+m} = 0.0$$

これらの係数により、指数法の離散化主部とベルヌーイ関数 BER4(Z) のプログラムは、次のようになる。

```

CCCCC
  IF (DF .EQ. 0.0D0) THEN
    COH=0.0D0
  ENDIF
  DO 15 I=1,M-1
    AA(I) = 0.5D0*(2.0D0+2.0D0*DF +DF*BER4(COH) + BER4(COH) +
+          BER4(-COH) + DF*BER4(-COH))
    AB(I) = -0.5D0*(BER4(COH) + DF*BER4(COH))
    AC(I) = -0.5D0*2.0D0
    AB1(I) = -0.5D0*(DF*BER4(-COH) + BER4(-COH))
    AC1(I) = 0.0D0
  15 CONTINUE
  AB1(1) = 0.0D0
*
  AA(M) = 0.5D0*(DF*BER4(COH) + BER4(COH) + 1.0D0 + DF)
  AB(M) = 0.0D0
  AC(M) = -0.5D0*1.0D0
  AB1(M) = -0.5D0*(DF*BER4(-COH) + BER4(-COH))
  AC1(M) = 0.0D0

```

```

C
  COH=C0*DH
  DO 35 J=1,M2-2
    DO 25 K=1,M-1
      AA(M*J+K)=0.5D0*(2.0D0*BER4(C0H)+ 2.0D0*BER4(-C0H)+4.0D0)
      AB(M*J+K)=-0.5D0*(2.0D0*BER4(C0H))
      AC(M*J+K)=-0.5D0*2.0D0
      AB1(M*J+K)=-0.5D0*(2.0D0*BER4(-C0H))
      AC1(M*J+K)=-0.5D0*2.0D0
    25 CONTINUE
      AB1(M*J+1)=0.0D0
  *
      AA(M*J+M)= 0.5D0*(2.0D0 +2.0D0*BER4(C0H))
      AB(M*J+M)=0.0D0
      AC(M*J+M)=-0.5D0*1.0D0
      AB1(M*J+M)=-0.5D0*(2.0D0*BER4(-C0H))
      AC1(M*J+M)=-0.5D0*1.0D0
    35 CONTINUE
C
  IF(DF .EQ. 0.0D0) THEN
    COH=0.0D0
  ENDIF
  DO 45 I=N-M+1, N-1
    AA(I)= 0.5D0*(2.0D0+2.0D0*DF +BER4(C0H)+ DF*BER4(C0H) +
    + DF*BER4(-C0H)+ BER4(-C0H))
    AB(I)=-0.5D0*(DF*BER4(C0H)+ BER4(C0H))
    AC(I)=0.0D0
    AB1(I)=-0.5D0*(BER4(-C0H)+ DF*BER4(-C0H))
    AC1(I)=-0.5D0*2.0D0
  45 CONTINUE
    AB1(N-M+1)=0.0D0
  *
    AA(N)= 0.5D0*(BER4(C0H)+ DF*BER4(C0H)+ DF+ 1.0D0)
    AB(N)=0.0D0
    AC(N)=0.0D0
    AB1(N)=-0.5D0*(BER4(-C0H)+ DF*BER4(-C0H))
    AC1(N)=-0.5D0*1.0D0
CCCCC

C *** FUJITSU
  REAL*8 FUNCTION BER4(BEX)
    REAL*8 BEX, HX
C *** HITACHI
C REAL FUNCTION BER4*8(BEX)
C REAL*8 BER4, BEX, HX
C** HX=1.000D-4
    HX=1.000D-2
  *
    IF(BEX.LT.-HX) THEN
      BER4=BEX/(DEXP(BEX)-1.0D0)
    ELSE IF(BEX.GE.-HX .AND. BEX.LE.HX) THEN
      BER4=1.0D0-(0.5D0-0.0833333*BEX)*BEX
  *
CCCC BER4=1.0D0-0.5D0*BEX
    ELSE IF(BEX.GT.HX) THEN
      BER4=BEX*DEXP(-BEX)/(1.0D0-DEXP(-BEX))
    ENDIF
  *
  *
  END
  *

```

7 おわりに

これまで、移流拡散系において2解法（中心差分と指数法）を試し、 C_0 を固定して、MJを変えてセルペクレ数 $\frac{C_0}{MJ}$ による変化を見た。一方、MJ=1, 10とした時（粗と精しいあみ目）に、 C_0 を変えてセルペクレ数による変化も見た。

いずれの場合も、セルペクレ数が2を超えると、中心差分では解が振動（正負交互）して不安定で、指数法でないとは正解とならない。セルペクレ数が、たとえ2以下でも、中心差分と指数法の解の差は、セルペクレ数が2で、およそ u_{max} で25%余りあり（ u_{min} で30%）、セルペクレ数が1でも5%（ u_{min} で8%）の差がある。

したがって、中心差分でがまんできるのは、セルペクレ数が1以下の場合に限られる。セルペクレ数を下げるには、あみ目を精しくする必要があり、そのためメモリ容量を食う。結論として、移流拡散問題の場合には、指数法が必須ということになる。

なお、より具体的な例として、半導体デバイス解析のドリフト・拡散モデルがあるが、これは、連立非線形の移流拡散問題となる。

$$(7.1) \quad \text{div}[-\epsilon \nabla \psi] = e(p - n - N_a + N_d)$$

$$\text{div}[-D_n \nabla n + (\mu_n \nabla \psi)n] = GR$$

$$(7.2) \quad \text{div}[-D_p \nabla p - (\mu_p \nabla \psi)p] = GR$$

のポアソン方程式(7.1)、電子密度 n と正孔密度 P の移流拡散方程式(7.2)の3本の方程式を、ポアソン方程式の両辺の整合性を収束判定基準として、反復解法により解く。これをガンメル反復という。ポアソン方程式は、右辺の $e(p - n - N_a + N_d)$ の電荷 p や n に、 $e \frac{\psi}{kT}$ (T:温度、k:ボルツマン定数)の形の非線形性を持つために、(7.1)式をそのまま離散化しても収束しない。そこで、Newton反復の方法を取り入れて、式(7.1)を線形化した：

$$\text{div}[-\epsilon \nabla \psi^{(k+1)}] + e\nu(p^{(k)} + n^{(k)})\psi^{(k+1)} = e\nu(p^{(k)} + n^{(k)})\psi^{(k)} + e(p^{(k)} - n^{(k)} - N_a + N_d)$$

$$\text{ここで、}(k)\text{は反復回数を示し、Einsteinの関係から、}\frac{\mu}{d} = \frac{e}{kT} = \nu = 38.68$$

に基づいて離散化すれば、収束する。これは、天才Gummelが最初に発見した。この線形化したポアソン方程式をCV法（積分形式に基づく離散化法）により中心差分して、ICCG法ソルバで解く。 n と p の移流拡散方程式は、これまで解説したように、CV法により指数差分（指数法）して、BCGstab法ソルバで解く。ガンメル反復の手順を整理すると、次のようになる（EPSGは、収束判定値）。

初期値 $\psi^{(0)}, n^{(0)}, p^{(0)}$ を用意、 $k = 0$

do while $\| \text{div}(-\epsilon \nabla \psi^{(k)} - e(p^{(k)} - n^{(k)} - N_a + N_d)) \| \geq \text{EPSG} * \| e(p^{(k)} - n^{(k)} - N_a + N_d) \|$

$\text{div}[-\epsilon \nabla \psi^{(k+1)}] = e(p^{(k)} - n^{(k)} - N_a + N_d)$ ($\psi^{(k+1)}$ をICCG法で解く)

解く式は、 $\text{div}[-\epsilon \nabla \psi^{(k+1)}] + e\nu(p^{(k)} + n^{(k)})\psi^{(k+1)}$

$= e\nu(p^{(k)} + n^{(k)})\psi^{(k)} + e(p^{(k)} - n^{(k)} - N_a + N_d)$

$\text{div}[-D_n \nabla n^{(k+1)} + (\mu_n \nabla \psi^{(k+1)})n^{(k+1)}] = GR$ ($n^{(k+1)}$ を(ILU)BCGstab法で解く)

$\text{div}[-D_p \nabla p^{(k+1)} - (\mu_p \nabla \psi^{(k+1)})p^{(k+1)}] = GR$ ($p^{(k+1)}$ を(ILU)BCGstab法で解く)

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] スハス V. パタンカー原著, 水谷幸夫, 香月正司 共訳, コンピュータによる熱移動と流れの数値解析, 森北出版,Feb,1985