体験するプログラミング

その1：飽和定常流れは何次元でも簡単に解ける！

登坂博行*

Get the taste of practical programming: No. 1 A simple and short program for solving the multidimensional saturated, steady-state flow of groundwater

Hiroyuki TOSAKA *

1. はじめに

我々は概して他分野のことに疎いし、自分分野の本質を体験していないことがある。地下水に関わっている技術者・研究者の方で、実際に地下水流れの2次元、3次元プログラムを自分で作り解いたことがある人はどのくらい居られるだろうか？水文、地質、計測などを主体とする仕事では定性的に理論を理解していればそれで問題ないし、データの整理も表計算ソフトなどで行うから自作の必要がない。解析技術者であっても借り物のソフトを使う場合（使わざるを得ない場合）が多く、自作の時間が与えられなかった方も居られるかもしれない。

多くの方の業務や研究にとって、プログラミングや計算ができることは「本質」ではないのだが、逆に「流れの本質」を理解するためには計算体験は実に有益だと思われる。そこで、本稿は、①今まで地下水流れを定性的に理解しているつもりだが、解いたことがない方、②解析屋なのに基礎体験に不安の

ある方、③様々な理由から、数値解析を始めてみたい方、などに向けて書いてみたものである。

数式というのは、ある種の壁を持っている。さらに数値解析技術自体にも高い壁があることは確かであり、なんとなく敬遠してしまうことが最大の壁でもある。体験により「壁」がくずれ新たな地平が見えることを信じ、しばしお読み頂ければ幸いである。

参考としてつけてあるプログラムを見るとおわかりのように、数十行で2次元や3次元の式が解けるのである！本講座では細かな数理的な話は書かないが、“なるほど簡単なものだ”と思って頂ければうれしいところである。尚、今後順次、色々な問題が意外と簡単に解けることを紹介してゆきたい。

2. 計算対象となる地層のイメージ

まず、何を解くかをはっきりさせておこう。図1の様に、水平な広がりを持ち上下が難透水層によ

* 東京大学工学系研究科システム創成学専攻（
Graduate School of Engineering, The University of Tokyo

り被覆されている飽和帯水層があるものとし、その透水係数が k 、間隙率が ϕ 、厚さが T とわかっているものとする。中央に1点或いは複数点の井戸を配置し、揚水 Q を継続すると仮定し、境界位置の水頭は変化しない条件を考える。

この時、井戸付近でどの程度水頭が下がるか、周辺の水頭はどの程度下がるか、どのような分布となるかを計算・図化することが目標である。

図2は図1の地層を適当に分割した模式図である。地層内部は $(1 \sim NX)$ 、 $(1 \sim NY)$ に分け、境界線を含め $(NX+2) \times (NY+2)$ 個に分割したものである。空間の分割は非常に細かくすることもできるし、分割の仕方を変幻自在に行うこともできるが、ここでは一様な分割($NX = NY = 100$ 程度)を考えよう。

地層の境界線部分($i=0, i=NX+1, j=0, j=NY+1$ の線上)は一定値であり、解くのは内部の点、($i=1 \sim NX, j=1 \sim NY$)の未知水頭である。

図1 計算対象の地下水帯水層の形状

図2 離散化した地層と未知量の配置

3. どのような式を解くか？

3.1 定常飽和流れの式

このような地層内の任意の点において次の式が成立する。

$$\frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2} + \frac{\partial^2 H}{\partial z^2} = \frac{q}{k} \quad (1)$$

ここで、 $k[\text{m/s}]$ は透水係数、 $H[\text{m}]$ は全水頭、 $q[\text{m}^3/(\text{m}^3 \cdot \text{s})]$ は単位体積・単位時間の揚水量(揚水が正)である。このような空間の2階微分形は、ラプラス・ポアソン方程式(Laplace-Poisson's equation)などと呼ばれる(右辺が0の場合をラプラス方程式、0でない場合をポアソン方程式と呼ぶ)。

3.2 復習：なぜこの形がでてくるか。

簡単に前式がなぜ出てくるかについて説明しておきたい。不必要な方は読み飛ばしていただければよい。

(1) 構成則が勾配則であること

地下水の飽和流れは次のダルシー則に支配されていると考えられる。

$$v = -k \frac{\partial H}{\partial x} \quad (2)$$

ここで、 $v[\text{m}^3/(\text{m}^2 \cdot \text{s})]$ は単位断面積を流れる単位時間当たりの体積流量、即ち流速(軸方向を正)、 $k[\text{m/s}]$ は透水係数、 $H[\text{m}]$ は全水頭である。

この式は、地下水関係では常識ともいえるものである。このような法則は、物理現象にはいくつも知られており、物理量の局地的勾配が大きな役割を果たしている。例えば、濃度拡散の法則、熱伝導の法則、オームの法則、電界の法則、などがある(例えば、登坂、2006参照)。

(2) 保存則が満たされること

マクロな自然現象では、質量保存則(連続の式)、エネルギー保存則、運動量保存則などが満たされる。例えば、地下水流れでは、空間点 (x, y, z) を中心に微小領域(図3)を考えると、微小時間に各面から流入出する質量と内部の質量の間に次の保存関係が成立する。

$$M^{(t)} + (M_{x-\Delta x/2} - M_{x+\Delta x/2} + M_{y-\Delta y/2} - M_{y+\Delta y/2} + M_{z-\Delta z/2} - M_{z+\Delta z/2}) \Delta t - M_s \Delta t = M^{(t+\Delta t)} \quad (3)$$

ここで、 $M^{(t)}$ [kg]は微小領域内の水の質量、 $M_{x+\Delta x/2}$ [kg/s]はx方向に垂直な面を通じた単位時間あたりの流入水の質量（軸方向の流れを正とする、y、z方向についても同様）、 M_s [kg/s]は単位時間あたりの揚水質量（圧入の場合は負値）である。

時間的に変化しない状態（定常状態）を考えると、 $M^{(t)} = M^{(t+\Delta t)}$ であるから

$$M_{x-\Delta x/2} - M_{x+\Delta x/2} + M_{y-\Delta y/2} - M_{y+\Delta y/2} + M_{z-\Delta z/2} - M_{z+\Delta z/2} - M_s = 0 \quad (4)$$

となる。

(3) 離散化した構成則を保存則に代入する

図2の様に離散化した場合は、ダルシーの式は次のように近似できることは直感的に良く理解できるものであろう。

$$v_{x-\Delta x/2} = -k \frac{H_x - H_{x-\Delta x}}{\Delta x} = -k \frac{H_{ij} - H_{i-1,j}}{\Delta x} \quad (5)$$

水は非圧縮性と仮定し、密度 ρ [kg/m³] を一定とすると、質量の流入出や貯留量は次の様に書くことができる。

$$M_{x-\Delta x/2} = \rho \cdot S_x \cdot v_{x-\Delta x/2} = \rho S_x k \frac{H_{ij} - H_{i-1,j}}{\Delta x}$$

$$M_{y-\Delta y/2} = \rho \cdot S_y \cdot v_{y-\Delta y/2} = \rho S_y k \frac{H_{ij} - H_{i,j-1}}{\Delta y} \quad (6)$$

図3 微小領域の流入出

$$M_s = \rho Q_{s,ij} \quad (6)$$

ここで、 S_x 、 S_y [m²]はそれぞれx、y方向の流動断面積、 $Q_{s,ij}$ [m³/s]は格子点 (i, j) における実際の揚水量(体積)である。よって、式 (4) に (6) 式などを代入し整理すると、二次元では次の形となる。

$$\frac{k S_y}{\Delta y} H_{i,j-1} + \frac{k S_x}{\Delta x} H_{i-1,j} - \left(\frac{2k S_x}{\Delta x} + \frac{2k S_y}{\Delta y} \right) H_{ij} + \frac{k S_x}{\Delta x} H_{i+1,j} + \frac{k S_y}{\Delta y} H_{i,j+1} = Q_{s,ij} \quad (7)$$

この式は2階微分の形になっていないが、全体を $\Delta x \Delta y \Delta z$ で割ると

$$\frac{k}{\Delta y^2} H_{i,j-1} + \frac{k}{\Delta x^2} H_{i-1,j} - \left(\frac{2k}{\Delta x^2} + \frac{2k}{\Delta y^2} \right) H_{ij} + \frac{k}{\Delta x^2} H_{i+1,j} + \frac{k}{\Delta y^2} H_{i,j+1} = q_{s,ij} \quad (8)$$

となる。ここで、 q_s [m³/(m³·s)] = $Q_{s,ij}/(\Delta x \Delta y \Delta z)$ は地層単位体積・単位時間あたりの揚水体積速度である。式 (8) をさらに整理すると

$$\frac{1}{\Delta x} \left\{ \frac{H_{i+1,j} - H_{ij}}{\Delta x} - \frac{H_{ij} - H_{i-1,j}}{\Delta x} \right\} + \frac{1}{\Delta y} \left\{ \frac{H_{i,j+1} - H_{ij}}{\Delta y} - \frac{H_{ij} - H_{i,j-1}}{\Delta y} \right\} = \frac{q_{s,ij}}{k} \quad (9)$$

となり、この極限が式 (1) の2階微分の形となることは明らかである。即ち、式 (1) の微分方程式は、このように構成則と保存則から物理で（数学ではなく）考えれば、実に素直に出てくるものである。

4. 差分方程式を解く

4.1 連立一次方程式

さて、数値解析上は式 (7) か式 (8) の形にしたものが便利である。ここでは式 (7) を使って考えよう。左辺の係数を a 、 b 、 c 、 d 、 e 、右辺を s として式 (7) を次のように整理する。

$$a_{ij}H_{ij-1} + b_{ij}H_{i-1,j} + c_{ij}H_{ij} + d_{ij}H_{i+1,j} + e_{ij}H_{ij+1} = s_{ij}$$

$$a_{ij} = e_{ij} = \frac{kS_y}{\Delta y}, \quad b_{ij} = d_{ij} = \frac{kS_x}{\Delta x}, \quad s_{ij} = Q_{ij},$$

$$c_{ij} = -(a_{ij} + b_{ij} + d_{ij} + e_{ij}) \quad (10)$$

式 (10) は 5 個の未知量を含む一次式で、このような式は未知格子点の数 (NX×NY) だけできるから、(NX×NY) 元連立一次方程式となる。数学的に書けば、係数 a 、 b 、 c 、 d 、 e を並べた行列 A (五重対角行列)、解 H_{ij} のベクトル $\mathbf{h}$ 、右辺 s_{ij} のベクトル $\mathbf{s}$ を使い簡単に次のように表記される。

$$A\mathbf{h} = \mathbf{s} \quad (11)$$

$$\begin{bmatrix} c_{1,1} & d_{1,1} & e_{1,1} & & \\ b_{2,1} & c_{2,1} & d_{2,1} & \cdot & 0 \\ & \cdot & \cdot & \cdot & 0 \\ a_{ij} & b_{ij} & c_{ij} & d_{ij} & e_{ij} \\ & \cdot & 0 & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} H_{1,1} \\ H_{2,1} \\ \cdot \\ H_{ij} \\ \cdot \\ H_{NX-1, NY} \\ H_{NX, NY} \end{bmatrix} = \begin{bmatrix} s_{1,1} \\ s_{2,1} \\ \cdot \\ s_{ij} \\ \cdot \\ s_{NX-1, NY} \\ s_{NX, NY} \end{bmatrix} \quad (12)$$

あとはこの行列方程式さえ解ければ良いのであるが、この辺からさぞかし難しくなると思われる方が居るかも知れない。しかし、この式は数行のプログラムでいとも簡単に解くことができるのである。数学の威力を実感するところである。

4.2 連立方程式を解く

ここでは、点過緩和法 (Point Successive Over Relaxation, PSOR) と呼ばれる行列解法を紹介しよう。まず、前記の式を次の残差形で書く。

$$r = a_{ij}H_{ij-1} + b_{ij}H_{i-1,j} + c_{ij}H_{ij} + d_{ij}H_{i+1,j} + e_{ij}H_{ij+1} - s_{ij} \quad (13)$$

この残差がどの点でも 0 となるように全未知数を決定することと、行列方程式が解けたことは同じことである。そのために、次の順序で計算を行う。

(1) 適当な初期推定値による残差計算

各点の未知量に、例えば判明している初期水頭値を代入して残差を計算する。

$$r^{(0)} = a_{ij}H_{ij-1}^{(0)} + b_{ij}H_{i-1,j}^{(0)} + c_{ij}H_{ij}^{(0)} + d_{ij}H_{i+1,j}^{(0)} + e_{ij}H_{ij+1}^{(0)} - s_{ij} \quad (14)$$

(2) 修正ステップ

前式は湧き出しのある点など変化が起こる所では当然 0 でない。そこで、ちょうど式の値が 0 になるような中心格子の値 $H_{ij}^{(*)}$ を考えると次式が成り立つ。

$$0 = a_{ij}H_{ij-1}^{(0)} + b_{ij}H_{i-1,j}^{(0)} + c_{ij}H_{ij}^{(*)} + d_{ij}H_{i+1,j}^{(0)} + e_{ij}H_{ij+1}^{(0)} - s_{ij} \quad (15)$$

前式と辺々引いて

$$-r^{(0)} = c_{ij}(H_{ij}^{(*)} - H_{ij}^{(0)})$$

$$dH = H_{ij}^{(*)} - H_{ij}^{(0)} = -\frac{r^{(0)}}{c_{ij}} \quad (16)$$

よって、初期推定値を dH だけ修正するとこの点の式は 0 にすることができる。しかし、この操作は局地的すぎて他の点の式を 0 にできるとは限らない。そこで、そのまま単純に修正することはせず、次のように係数をかけて修正しておく。

$$H_{ij}^{(*)} = H_{ij}^{(0)} + \omega \cdot dH \quad (17)$$

この修正値 $H_{ij}^{(*)}$ をそのまま新しい H_{ij} とみなして、次の格子点の計算式に現れたときは、これを使用する。 ω は 1～2 の間の値である。

(3) 反復収斂させる

このプロセスは 1 回計算しただけでは良い解には到達しない。上記計算プロセスを何回も繰り返すと、次第に全ての点での dH の絶対値の最大値が収束判定値 ε より小さくなるか、或いは、残差の絶対値の最大値が収束判定値 ε_r 以下になった時点で計算を終了する。収束判定値は問題により (単位系によっても) 多少異なるが、例えば、 $\varepsilon = 1.0e^{-6}$ などを与える。

5. プログラムを読む、書く

5.1 プログラムを読む

実際にプログラムを見てみよう。参考プログラムとして図 4 に Fortran のプログラムを、図 5 に C のプログラムを示した。分かりやすくするた

```

C SOLUTION OF 2-D, Saturated, Static Groundwater Flow
implicit real*8(a-h,o-z)
parameter (nnx=100, nny=100)
dimension h(0:nnx,0:nny), a(nnx,nny), b(nnx,nny), c(nnx,nny)
*, d(nnx,nny), e(nnx,nny), s(nnx,nny)
data dx, dy, dz, perm/10.0, 10.0, 10.0, 1.0E-4/
data omega, eps/1.8, 1.0E-6/
data Q/1000.0/
open(1, file='Head-out.dat', form='formatted')

C SET PARAMETER VALUES
nx=99
ny=99
dv=dx*dy*dz
sx=dy*dz
sy=dx*dz

C INITIALIZE
do i=0, nx+1
do j=0, ny+1
h(i,j)=10.0
enddo
enddo
do i=1, nx
do j=1, ny
s(i,j)=0.0d0
enddo
enddo

C SET COEFFICIENTS
do j=1, ny
do i=1, nx
a(i,j)=sy*perm/dy
b(i,j)=sx*perm/dx
c(i,j)=-2.0d0*(sx*perm/dx+sy*perm/dy)
d(i,j)=sx*perm/dx
e(i,j)=sy*perm/dy
enddo
enddo

C Set SOURCE/SINK TERM at arbitrary points(m3/day to m3/s)
s(49,49)=Q/86400.

C SOLVE MATRIX EQUATION WITH PSOR
do n=1, 100000
rmax=0.0d0
do i=1, nx
do j=1, ny
r=a(i,j)*h(i,j-1)+b(i,j)*h(i-1,j)
+c(i,j)*h(i,j)+d(i,j)*h(i+1,j)
+e(i,j)*h(i,j+1)-s(i,j)
dh=-r/c(i,j)
h(i,j)=h(i,j)+omega*dh
if (rmax.lt.abs(dh)) then
rmax=abs(dh)
end if
enddo
enddo
write(*,*) n, rmax
if (rmax.lt.eps) go to 99
enddo
99 continue
C PRINT OUT RESULTS
do j=ny+1, 0, -1
write(1,100) (h(i,j), i=0, nx+1)
enddo
100 format(101(g12.5,1x))
stop
end

```

図4 Fortran プログラムの例

め、必ずしもよいプログラムとはしていない。フローとしては、①宣言文、②初期データの設定、③係数の計算、④井戸位置と揚水量の設定、⑤行列方程式の解、⑥出力、からなっていることがご理解頂けよう。主要な部分はせいぜい30行程度で書けている。出力では、101列101行のデータを書き出しており、excel等に読み込むことで図化することができる。

図6は、中央に揚水井1000m³/dayを置いた最も基本的なケースの水頭分布を示している。井戸位置では10m程度水頭が下がり、全体としてきれいな曲面が描かれている。但し、離散化が粗いため、井戸位置での水頭計算値は、実際の井戸半径

```

//Solution of 2-D, Saturated Groundwater Flow (by Yamazaki Yuichi)
#include <stdio.h>
#include <math.h>

#define nnx 100
#define nny 100
#define dx 10.0
#define dy 10.0
#define dz 10.0
#define perm 1.0e-4
#define omega 1.8
#define eps 1.0e-6
#define Q 1000.0

main()
{
FILE *fp;
double a[nnx][nny], b[nnx][nny], c[nnx][nny], d[nnx][nny];
double e[nnx][nny], s[nnx][nny], h[nnx+1][nny+1];
int i, j, n, nx=99, ny=99;
double dv=dx*dy*dz;
double sx=dy*dz;
double sy=dx*dz;
double r, rmax, dsol;

//initialize
for(i=0; i<=nx+1; i++){
for(j=0; j<=ny+1; j++){
h[i][j]=10.0;
s[i][j]=0.0;
}
}

//calculate coefficient
for(j=1; j<=ny; j++){
for(i=1; i<=nx; i++){
a[i][j] = perm*sy / dy;
b[i][j] = perm*sx / dx;
c[i][j] = - ( 2*perm*sx / dx + 2*perm*sy / dy );
d[i][j] = perm*sx / dx;
e[i][j] = perm*sy / dy;
}
}

//set source/sink
s[49][49] = Q/86400.;

//solve matrix equation
for(n=1; n<=100000; n++){
rmax = 0;
for(i=1; i<=nx; i++){
for(j=1; j<=ny; j++){
r = a[i][j]*h[i-1][j] + b[i][j]*h[i][j-1] + c[i][j]*h[i][j]
+ d[i][j]*h[i+1][j] + e[i][j]*h[i][j+1] - s[i][j];
dsol = - omega * r / c[i][j];
h[i][j] += dsol;
if (rmax < fabs(dsol)) {
rmax = fabs(dsol);
}
}
}
printf("%d,%le\n", n, rmax);
if (rmax < eps) {
break;
}
}

//output results
fp = fopen("A.txt", "w");
for(j=ny+1; j>=0; j--){
for(i=0; i<=nx+1; i++){
fprintf(fp, "%le, ", h[i][j]);
}
fprintf(fp, "\n");
}
fclose(fp);
printf("end\n");
}

```

図5 C プログラムの例

付近の水位低下量とは直接は比較できないことに注意してほしい。このような計算は通常のPCなら即座に終わる。画面には次第に残差の最大値が小さくなる様子が表示される。

5.2 プログラムを書いてみる

このプログラムはいくらでも修正や工夫が可能である。例えば、

(1) 井戸を複数配置する

適当な位置に湧き出し項を与える。例えば、 $f(20, 20) = 100$ などと追加すれば良い。揚水量が

負値の場合は圧入となり、その点を中心に水頭は増加する。

(2) 異方性の有る2次元媒体

透水係数を k_x 、 k_y と別々に与え、参考プログラムで k と書いてある部分をどちらかに直す。入力を $k_x = 10^{-4}$ 、 $k_y = 10^{-5}$ などと与えると、描かれる

図6 等方均質地層からの揚水時の水頭変化 (揚水量 $1000\text{m}^3/\text{日}$)

図7 異方性地層からの揚水時の水頭変化 (揚水量 $1000\text{m}^3/\text{day}$)

水頭分布 (図7) が等方性の場合の結果 (図6) より扁平になるのがよくわかる。

(3) 3次元化

z 方向の効果を考慮するため、配列を $H(i, j, k)$ 、 $a(i, j, k)$ などのように三次元化し、係数行列の配列数を5個から7個に増やす ($a \sim g$)。プログラムの行数は10行程度しか増えないだろう。 $N_x = N_y = N_z = 30$ 程度として立方体の領域を考えて計算してみると良い。これは大きな飽和均質岩体中の球状流を考えていることになる。結果は適当な断面を選んで出力し、2次元と同様に excel 等で表示すると良い。

プログラムを書き実行するためにはテキストエディター (word や wordpad などテキストで保存するか、フリーのもの) と。コンパイラ (Fortran、C、VB など) が必要である。会社や研究機関ならその環境はそろっているはずであるが、自宅などで整っていない場合にはまず環境整備をしてしまいたい。HPを検索して、無料版などをダウンロードするか、有料版を購入するかすればよいし、WindowsPC上に簡単にLinux環境を作れるソフトも安価で市販されている。また、計算結果を図化するためには、excelなどの表計算ソフトがあればよいが、グラフィック用のソフトがあると時間変化なども見ることができる (試供版がダウンロードできるものもある)。

6. まとめ

本稿では、基礎的な2次元地下水飽和定常流が、非常に短いプログラムで書けることを紹介した。とりあえずプログラムを打ち込んで動かして見て頂くと、様々な美しい解が現れ、なるほどと納得できると思う。

細かな説明は省略したが、3次元問題や物性分布がより複雑な問題などでも、少々修正・追加すれば同じように解くことができる。なお、井戸の地点でどの程度水頭が下がるかをより詳しく見積もりたい場合には、デカルト座標系ではなく円筒座標系に変換して解くとよい。また、透水係数を

熱伝導率、水頭を温度と見れば、定常熱伝導問題になり、拡散係数と濃度と見れば濃度拡散問題が同じ様に解けることになる。

今後、一見難しそうに見える水文・水理解析が大変簡単なプログラムで解ける例をいくつか紹介してゆきたいと考えている。

謝 辞

本稿作成にあたり、C のプログラムは本学四年

生の山崎裕一君に書いてもらった。また、出力図面の作成では吉岡真弓さんに協力して頂いた。記して感謝する次第です。

参考文献

登坂博行 (2006) : 地圏水循環の数理、東大出版会、p130-133、p147-150.

(受付 : 2008年6月13日、受理 : 2008年7月14日)