

## Examples of application of iterative methods - Steady-state groundwater flow system (Laplace's equation)

Robert F. Beilicci\* Erika Beilicci\*

**Abstract** – The Paper present examples to solve a steady-state groundwater problem in which the aim is to calculate steady-state H distribution along a two-dimensional hillslope. This example presents usually methods in EXCEL or PASCAL to calculate H distribution. Several examples - easy and moderately difficult - illustrate the use of numerical methods.

**Keywords:** groundwater, hillslope, flow, boundary conditions

### I. INTRODUCTION

The purpose is to solve numerically a steady-state groundwater problem in which the aim is to calculate steady-state H distribution along a two-dimensional hillslope. Mathematically the hillslope is considered to be rectangular as shown in Fig. 1.

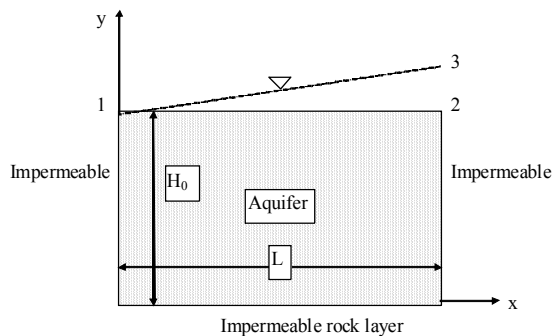


Fig. 1 Illustration of the steady state groundwater flow system

Toth (1962) has analyzed this type of groundwater flow system and Wang and Anderson (1982) have solved the problem using FORTRAN.

Fig. 1 represents a so called deep hillslope profile bounded on the right by water divide (zero-flow boundary) and on the left by stream in the valley bottom. The left boundary is also a zero-flow boundary since there is no exchange of water between the hillslopes located on the left and right side of the stream. Aquifer bottom is assumed impermeable and therefore, it is a zero-flow boundary. Left, right and bottom boundary conditions are of Neumann type (flux specified). Water table of the aquifer is assumed to vary linearly according to line 1..3 in Fig. 1. The upper boundary of the model follows line 1..2 indicating that we need to define a

Dirichleht boundary condition as an upper boundary condition.

The aquifer is assumed to be homogenous, isotropic and in steady-state condition. The steady-state assumption is valid when average value of water table position is used as boundary condition. It can be assumed that water table position at the beginning of the year is the same as position at the end of the year, i.e. There is no net accumulation or loss of water from the system over long time periods. Therefore, the two-dimensional Laplace's equation (1) is the required governing equation.

$$\frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2} = 0 \quad (1)$$

If hydraulic conductivity is not zero, zero-flow boundary is possible only if hydraulic gradient is zero: flux is calculated using Darcy's law,  $q_x = -K dH/dx = 0$  if  $dH/dx = 0$ . Boundary condition at the top of the aquifer is assumed to vary linearly between point 1 and 2 in such a way that at point 1 H equals the thickness of the aquifer,  $H_0$ , and increases with slope  $s_0$  and reaches its maximum at point 2 where  $x=L$ . The mathematical model of the groundwater flow system can now be summarized as:

Governing equation:

Laplace  $\frac{\partial^2 H}{\partial x^2} + \frac{\partial^2 H}{\partial y^2} = 0$

Boundary conditions:

Top  $0 \leq x \leq L$   $H(x, y_0) = H_0 + s_0 x$

Bottom  $0 \leq x \leq L$   $\frac{\partial H}{\partial y} \Big|_{y=0} = 0$

Left  $0 \leq y \leq H_0$   $\frac{\partial H}{\partial x} \Big|_{x=0} = 0$

Right  $0 \leq y \leq H_0$   $\frac{\partial H}{\partial x} \Big|_{x=L} = 0$

Frame 1. Summary of the mathematical model of the steady-state groundwater flow system

\*"Politehnica" University of Timisoara, HISGA Department, G. Enescu Str. No. 1/A, Zip Code 300022, Timisoara,

## II. EXAMPLE 1

The purpose is now to solve numerically the steady-state groundwater problem shown in Fig. 1 and summarized in Frame 1. The solution will be obtained using EXCEL. The numerical values of the constants of the problem are as follows:  $L=100$  m,  $H_0=50$  m and  $s_0=5/100=0.05$ . The number nodes in x-direction is  $NX=11$ , and the corresponding value for  $NY=6$ . This implies that  $\Delta x=\Delta y=10$  m. The grid is mesh-centered indicating that nodal points are located along the boundaries (in block-centered grid the nodal points are at the center of the grid).

The iteration equation for inner nodes ( $i=2, \dots, NX-1; j=2, \dots, NY-1$ ) is shown in (2).

$$H_{i,j} = (H_{i+1,j} + H_{i-1,j} + H_{i,j+1} + H_{i,j-1})/4 \quad (2)$$

Mathematical treatment of boundary conditions need to be examined more closely. Dirichlet boundary conditions are easy to apply: in those cells known H-values are directly given.

The mathematical treatment of boundary conditions is illustrated in one-dimensional case in Fig. 2. Consider that the left boundary where  $i=1$  is a no-flow boundary. We extend the region under consideration one node left, i.e. we define a so called fictitious node where  $i=0$ . The flow across the left boundary is zero if  $H/x=0$  at  $x=0$ , which implies that  $(H_2 - H_0)/(2\Delta x) = 0$  which is possible only if  $H_0 = H_2$ .

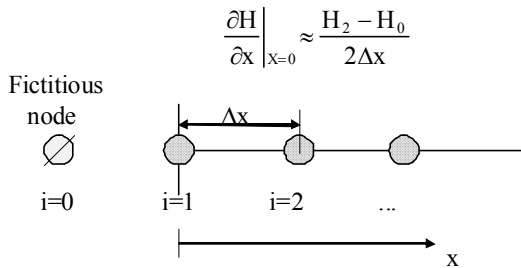


Fig. 2 Treatment of no-flow boundary conditions with fictitious nodes.

In numerical solution methods Neumann-type of boundary conditions can be treated in two different ways:

1. With direct use of the fictitious nodes.
2. The iteration equations are modified for Neumann-type boundaries.

In our problem we have a no-flow boundary along the boundary where  $i=1$ . By applying the iteration equation (2) for node (1,j) we get

$$H_{1,j} = (H_{2,j} + H_{0,j} + H_{1,j+1} + H_{1,j-1})/4 \quad (3)$$

and here we need the head  $H_{0,j}$  at the fictitious node.

It is straightforward to replace  $H_{0,j}$  with  $H_{2,j}$  which makes the solution possible. For left boundary this would lead to equation

$$H_{1,j} = (2H_{2,j} + H_{1,j+1} + H_{1,j-1})/4 \quad (4)$$

and this would be the solution using option 2. However, it is much more elegant to use option 1. and include the fictitious nodes in the nodal network. In EXCEL - solution this implies that for left boundary there is one extra column where a simple equation is given as shown below. Boundary nodes (1,j) are calculated in column D and the nodal value of the fictitious node C5 is replaced by value in E5 because  $H_{0,j} = H_{2,j}$ .

Row/Column	C	D	E	F
4	...		...	...
5	=E5	=(D4+D6+C5+E5)/4	...	...
6	...		...	...
7	...		...	...

The final results are shown in Table 1. It is worth mentioning that in EXCEL - solution it is useful to distinguish different type of cells with different background color. These include Dirichlet boundary cells, impermeable boundaries, fictitious nodes and ordinary nodes inside the grid (inner cells). Later on it is much easier to modify the grid if the background colors are used.

i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9	i=10	i=11
50.00	50.50	51.00	51.50	52.00	52.50	53.00	53.50	54.00	54.50	55.00
50.79	50.97	51.29	51.67	52.08	52.49	52.91	53.32	53.69	54.01	54.19
51.22	51.31	51.53	51.81	52.14	52.49	52.83	53.16	53.45	53.66	53.75
51.47	51.53	51.69	51.92	52.19	52.48	52.78	53.05	53.28	53.44	53.49
51.61	51.65	51.78	51.98	52.22	52.48	52.74	52.98	53.18	53.31	53.36
51.65	51.69	51.82	52.00	52.23	52.48	52.73	52.96	53.15	53.27	53.31

Table 1 Solution of the steady-state groundwater example of Fig. 1 using Gauss-Seidel iteration with EXCEL.

## III. EXAMPLE 2

In Example 2, the steady-state groundwater flow system of Fig. 1 is solved using SOR-method (Successive Over Relaxation). The changes to the program are very small, i.e. the formulas for the inner cells are replaced by the iterative equation (5) of the SOR-method:

$$H_{i,j}^{m+1} = (1-\omega)H_{i,j}^m + \omega(H_{i-1,j}^{m+1} + H_{i,j-1}^{m+1} + H_{i+1,j}^m + H_{i,j+1}^m)/4 \quad (5)$$

Row/Column	C	D	E	F
4	...		...	...
5	=E5	=(1-Omega)*D5+ Omega*(D4+D6+C5+E5)/4	...	...
6	...		...	...
7	...		...	...

The user of the program in EXCEL file should see that the number of iterations required is reduced considerably as compared to Example 1 when relaxation parameter  $\omega$  is around 1.6..1.7. This can be seen e.g. by giving different values for  $H_0$  and comparing the rate of convergence of SOR- and Gauss-Seidel methods (SOR = Gauss-Seidel if  $\omega = 1.0$ ).

Toth (1962) has published an analytical solution for this problem:

$$H(x,y)=H_0 + \frac{cL}{2} - \frac{4cL}{\pi^2} \sum_{m=0}^{\infty} \frac{\cos[(2m+1)\pi x/L] \cosh[(2m+1)\pi y/L]}{(2m+1)^2 \cosh[(2m+1)\pi H_0/L]} \quad (6)$$

The analytical solution is also given in the EXCEL file and the results of the comparison between the analytical solution and numerical solution using the SOR-method is shown in Table 2 for some selected points.

x	y	H(x,y)	SOR	Error (m)	Error (%)
0	50	50.00625	50	0.006254	0.012507
0	40	50.85472	50.7996	0.055121	0.108388
0	30	51.26794	51.23561	0.03233	0.063062
0	20	51.51325	51.48997	0.023276	0.045185
0	10	51.64607	51.62626	0.019805	0.038348
0	0	51.68829	51.66941	0.018874	0.036515
100	0	53.31171	53.33016	-0.01845	-0.03461

Table 2 Comparison of analytical and numerical solution (SOR-method) of the steady-state groundwater flow system of Fig. 1.

#### IV. EXAMPLE 3

Solution of the groundwater flow system of Fig. 1 can be carried out using some programming language and in this case it is necessary to do the iteration in the program. The following example uses Pascal for solving the problem with the SOR-method. The Pascal - program is given in Fig. 3.

```

Const
  NX=11; NY=6; {Number of nodes in x- and y-directions}
  dx=10.0; SlopeGW=0.05; H0=50.0;
  Omega=1.7; MaxIter=1000; IterationStoppingCriteria=0.001;
Var
  H:array[0..NX+1,0..NY+1] of real; {fictitious nodes: i=0; i=NX+1; j=NY+1}
  i,j,iter:integer;
  OldH,Error,MaxErr:real;
begin
  {MAIN PROGRAM BEGINS..}
  for i:=0 to NX+1 do {As initial condition for all nodes H[i,j]=H0}
    for j:=0 to NY+1 do
      H[i,j]:=H0;
  for i:=1 to NX do {Boundary condition at the top of aquifer}
    H[i,1]:=H0 + SlopeGW * DX * (i-1);
  iter:=0; {iteration begins}
  repeat
    inc(iter);
    MaxErr:=0.0;
    for j:=1 to NY do {Left and right boundary condition at the beginning of each iteration}
      begin
        H[0,j]:=H[2,j];
        H[NX+1,j]:=H[NX-1,j];
      end;
    for i:=1 to NX do {Bottom boundary condition}
      H[i,NY+1]:=H[i,NY-1];
    for j:=2 to NY do {Sweeping node-by-node}
      begin
        for i:=1 to NX do
          begin
            OldH:=H[i,j]; {store temporarily old iteration}
            H[i,j]:=(H[i-1,j]+H[i+1,j]+H[i,j+1]+H[i,j-1])/4;
            H[i,j]:=Omega*H[i,j] + (1.0-Omega)*OldH;
            Error:=abs(H[i,j]-OldH);
            if(Error>MaxErr)then MaxErr:=Error;
          end; {end i}
        end; {end j}
      end; {end iter}
  until(Iter>MaxIter) or (MaxErr<IterationStoppingCriteria);
  {Continue iteration as long as MaxErr less than IterationStoppingCriteria
  or Max. number of iterations exceeded}
  writeln('Solution:');
  writeln('Number of iterations = ',Iter:7,' Omega=',Omega:6:2,' Max.error= ',MaxErr:12);
  for j:=1 to NY do
    begin
      for i:=1 to NX do
        write(H[i,j]:7:2);
      writeln;
    end;
  write('Press ENTER to continue. '); readln;
end.

```

Fig. 3 Pascal-program for solving the steady-state groundwater flow system of Fig. 1 using the SOR-method

The example program includes some comments, but it necessary to point out that the no-flow boundary conditions need to update at the beginning of each iteration. Moreover, in the sweeping cycle, the old iteration value from level m must be temporarily stored to variable OldH so that it is possible to calculate the change between successive iterations, i.e. c in Eq. (7). In the program the new iteration value is immediately updated and therefore it is necessary to store only one H-matrix in the computer program.

$$c = H_{i,j}^{m+1} - H_{i,j}^m \quad (7)$$

The results of the computer program are shown in Table 3 for two values of the relaxation parameter  $\omega$ : Gauss-Seidel results for  $\omega = 1.0$  and results for SOR-method when  $\omega = 1.7$ . The total number of iterations needed in the SOR-method is 37 compared with 114 needed in the Gauss-Seidel-method.

#### Gauss-Seidel method ( $\omega = 1.0$ )

Solution:  
 Number of iterations = 114 Omega= 1.00 Max.error= 9.87006E-04  
 50.00 50.50 51.00 51.50 52.00 52.50 53.00 53.50 54.00 54.50 55.00  
 50.79 50.97 51.29 51.67 52.08 52.49 52.91 53.32 53.69 54.01 54.19  
 51.22 51.31 51.53 51.81 52.14 52.49 52.83 53.16 53.45 53.66 53.75  
 51.47 51.53 51.69 51.92 52.19 52.48 52.78 53.05 53.28 53.43 53.49  
 51.61 51.65 51.78 51.98 52.22 52.48 52.74 52.98 53.18 53.31 53.35  
 51.65 51.69 51.81 52.00 52.23 52.48 52.73 52.96 53.14 53.27 53.31

#### SOR-method ( $\omega = 1.7$ )

Solution:  
 Number of iterations = 37 Omega= 1.70 Max.error= 9.82048E-04  
 50.00 50.50 51.00 51.50 52.00 52.50 53.00 53.50 54.00 54.50 55.00  
 50.80 50.98 51.30 51.68 52.08 52.50 52.92 53.32 53.70 54.02 54.20  
 51.23 51.32 51.54 51.82 52.15 52.50 52.84 53.17 53.46 53.67 53.76  
 51.49 51.55 51.70 51.93 52.20 52.50 52.79 53.06 53.29 53.45 53.51  
 51.62 51.67 51.80 52.00 52.24 52.50 52.76 53.00 53.19 53.32 53.37  
 51.66 51.71 51.83 52.02 52.25 52.50 52.75 52.97 53.16 53.28 53.33

Table 3 Output of the steady-state groundwater flow system shown if Fig. 1. Comparison of results obtained by using Gauss-Seidel-and and SOR-methods. Computations carried out using the Pascal program given in Fig. 1.

The total number of iterations needed in the SOR-method is heavily dependent on the relaxation parameter  $\omega$  as indicated by the results of Table 4. In this case there is an optimum value for  $\omega_{opt}$ . If too high value for relaxation parameter is used ( $\omega = 1.9$  in our example), the convergence is not attained at all, which implies that extrapolating too much can lead to instability.

$\omega$	Number of iterations
1.00	114
1.10	99
1.20	86
1.30	75
1.40	64
1.50	55
1.60	46
1.70	37
1.75	35
1.80	43
1.90	no convergence

Table 4 Total number of iterations needed as a function of relaxation parameter  $\omega$  in solving the steady-state groundwater flow system described in Fig. 1.

## REFERENCES

- [1] Nelder, J.A. and Mead, R. (1965). *Computer Journal*, vol 7., p. 308.
- [2] O'Neill, R. (1971). *Function minimization using the simplex procedure*.
- [3] Pinder, G.F. and Gray, W.G. (1977). *Finite element simulation in surface and subsurface hydrology*. Academic Press, New York.
- [4] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1988). *Numerical recipes in C. The art of scientific computing*. Cambridge University Press, Cambridge.
- [5] Remson, I., Homberger, G.M. and Molz, F.J. (1971). *Numerical methods in subsurface hydrology*. Wiley-Interscience, New York.
- [6] Rushton, K.R. and Redshaw, S.C. (1979). *Seepage and groundwater flow*. Wiley, New York.
- [7] Sun Ne-Zheng (1994). *Inverse problems in groundwater modeling*. Kluwer Academic Publishers. London.
- [8] Toth, J. (1962). A theory of groundwater motion in in small drainage basins in Central Alberta, Canada. *Journal of Geophysical Research*, 67(11): 4375-4387.
- [9] Wang, H.F. and Anderson, M.P. (1982). *Introduction to groundwater modeling: Finite difference and finite element methods*. W.H. Freeman and Company, San Fransisco.