TO: Professor Andreas Linninger / Grant Hartung

FROM: Kehinde Abioye

DATE: 9/19/2017

SUBJECT: Sparse Matrix Notation

The purpose of this assignment is to use knowledge of the matrix equation to solve for unknown variables of biological networks, such as blood flow, and to compare the solutions using sparse notation, or matrices without zeroes, to the dense matrices using the “sparse” function in MATLAB.

Introduction

Part#1: Generation of Mass Conservation Balances Using Network Analysis

The flow and pressures of the Circle of Willis was analyzed. At any given point of pressure, the flow going in will be positive and any flow going out will be negative (Eq. 2). The Hagen-Poiseuille equation (eq. 1) described the resistance and flow between two points of pressures. The corresponding resistance will be matched with the flow between that point. The pressures at the outermost part of the Circle of Willis will have boundary conditions.

$$\Delta P = \alpha F, \text{ Eq. 1}$$

Resistance vector and Boundary Conditions, Eq. 1a

$$\alpha = [22423,\ 12457, 143729, 28391, 17744330, 2299666, 2299665, 17744339, 2299668,\ 2299667,\ 2299668,\ 2299667,\ 69,\ 69,\ 96, 173]\ (\text{mmHg}\cdot\text{min/mL})$$

$$Fin - Fout = 0, \text{ Eq. 2}$$

Boundary Conditions for the system pictured in Figure 1, Eq. 2a

P1 = 100 (mmHg);

P13 = 5 (mmHg)

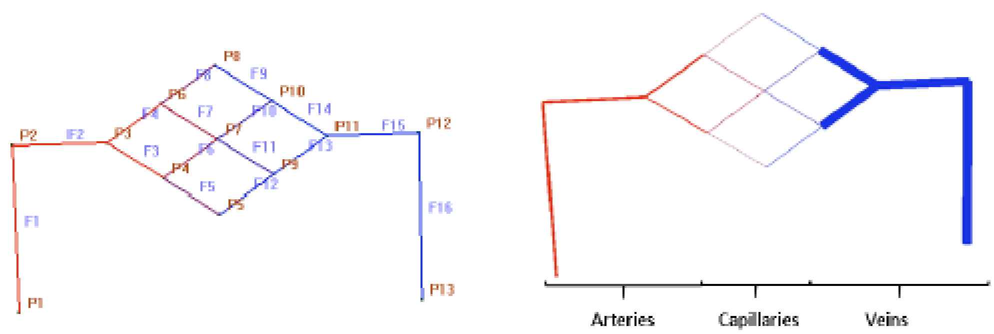

Fig1. Simplified Cerebral Angioarchitecture

Part#2: Writing the Network Problem in Matrix Format

Now that we have a set of equations from part 1, we can use the equation Ax=b (eq. 4) to find the flows and pressures of the Circle of Willis in figure 1. The equations will be assembled in matrix form in the format presented in equation 3a, 3b, and 3c to find the solution x. The pressures will represent C1, the resistance will represent –R, the flows will be the C2 matrix and the boundary conditions will be the C3 matrix. The flows and pressures will be solved using the "A\b" command. The solutions will be checked using the residual error equation r = Ax-b. If r equals a null vector, then the solution is accurate. Three nodes will be selected at random to test the solutions. This will be done by plugging in the corresponding solution to the corresponding equation.

$$C_1\, p = R\, f, \text{ Eq. 3a}$$

$$C_2\, f = 0, \text{ Eq. 3b}$$

$$C_3 p = \bar{p}, \text{ Eq. 3c}$$

$$\begin{bmatrix} C1 & -R \\ 0 & C2 \\ C3 & 0 \end{bmatrix} \begin{bmatrix} p \\ f \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \bar{p} \end{bmatrix}$$

$$A \quad * \quad x \;=\; b, \text{ Eq. 4}$$

Part#3: Sparse Notation

Using the "spalloc" function, a sparse matrix was created and the matrix was formed using the matrix notation of the non-zero values in matrix A of part 2. The system was solved and compared to the solution of part 2.

Methods

Part#1: Generation of Mass Conservation Balances Using Network Analysis

Equations were created for the flows and pressures of a Circle of Willis using the Hagen-Poiseuille and conservation balances equations. A vector α was given and inserted into the Hagen-Poiseuille equation accordingly. The boundaries of the Circle of Willis had no flow, therefore, a conservation balances equation could not be made for those nodes. Boundary conditions were provided.

Part#2: Writing the Network Problem in Matrix Format

Using the equations found in part 1, matrices were created. The Hagen-Poiseuille equations were assigned to a 18x18 Matrix C_1. This matrix represented the pressures. Any instance where a pressure was not used was assigned a value of zero. A 18x18 Matrix -R represented the α vector multiplied by one. A diagonal matrix of α values was created. The conservation balances were assigned to a 9x18 matrix C_2. Any place in the matrix missing value was given the value 0. This matrix represented the flows. Matrix C_3 is a 9x18 matrix that is the boundary conditions. The 36x36 matrix A was assembled by combing the matrices accordingly, inserting zero matrices when necessary. The vector x is a 36x1 matric and is the unknown variables p(pressures) and f (flows). The "\" symbol in matlab was used to find the values of the pressure and flows by multiplying the b vector with the inverse matrix of A. The values were checked using the residual equation r=Ax-b, as shown in appendix B. When r equals 0 then the solution satisfies the equations. The larger r is less accurate. For loops were used to obtain the r values.

Part#3: Sparse Notation

Using the "spalloc" function, a 29 by 29 matrix of all zeroes was created. This allows non-zero values to be assigned to the matrix without repeated storage. The non-zero values of the matrix were assigned using matrix notation, where the row and column of the non-zero values were assigned to the row and column of the sparse matrix. To check that the matrices were the same, matrix A from part 2 was subtracted from the sparse matrix S. If any value except zero is found then one of the matrices was incorrectly formed. The solution x was solved using "\" to multiply the inverse of matrix S by matrix b and compared to the solution of the dense matrix in part 2.

Results

Part#1: Generation of Mass Conservation Balances Using Network Analysis

The Hagan-Poiseuille equations, boundary condition equations, and conservation balance equations were formed as in equation 1. There are a total of 16 Hagen-Poiseuille equations, 11 conservation balance, and 2 boundary condition equations with a total of 29 equations.

Hagen-Poiseuille

1: P_1-P_2=$\alpha_1 F_1$

2: P_2-P_3=$\alpha_2 F_2$

3: P_3-P_4=$\alpha_3 F_3$

4: P_3-P_6=$\alpha_4 F_4$

5: P_4-P_7=$\alpha_6 F_6$

6: P_6-P_7=$\alpha_7 F_7$

7: P_7-$P1_{10}$=$\alpha_{10} F_{10}$

8: P_8-P_{10}=$\alpha_9 F_9$

9: P_6-P_8=$\alpha_8 F_8$

10: P_7-P_9=$\alpha_{11} F_{11}$

11: P_5-P_9=$\alpha_{12} F_{12}$

12: P_9-P_{11}=$\alpha_{13} F_{13}$

13: P_4-P_5=$\alpha_5 F_5$

14: P_{11}-P_{12}=$\alpha_{15} F_{15}$

15: P_{12}-P_{13}=$\alpha_{16} F_{16}$

16: P_{10}-P_{11}=$\alpha_{14} F_{14}$

Conservation Balances

1: F_1-F_2=0

2: F_2-F_3-F_4=0

3: F_3-F_6-F_5=0

4: F_5-F_{12}=0

5: F_4-F_8-F_7=0

6: F_7+F_6-F_{10}-F_{11}=0

7: F_8-F_9=0

8: F_{12}-F_{11}-F_{13}=0

9: F_9+F_{10}-F_{14}=0

10: F_{14}+F_{13}-F_{15}=0

11: F_{15}-F_{16}=0

Boundary Conditions

1: P_1=100

2: P_{13}=5

Part#2: Writing the Network Problem in Matrix Format

C_1, -R, C_2 and C_3 were created and concatenated in matrix A. The solution x was found using “A\b” in matlab which multiplies the inverse of A with b. The solution was tested for Flow 1, 2, and 3 and pressure 1, 2. And 3, as shown in appendix B.

C1=

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
1	-1	0	0	0	0	0	0	0	0	0	0	0
0	1	-1	0	0	0	0	0	0	0	0	0	0
0	0	1	-1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	-1	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	1	-1	0	0	0	0	0	0
0	0	0	0	0	1	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	-1	0	0	0
0	0	0	0	0	0	1	0	0	-1	0	0	0
0	0	0	0	0	0	1	0	-1	0	0	0	0
0	0	0	0	1	0	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	-1	0	0
0	0	0	0	0	0	0	0	0	1	-1	0	0
0	0	0	0	0	0	0	0	0	0	1	-1	0
0	0	0	0	0	0	0	0	0	0	0	1	-1

C2=

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	-1	-1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	-1	0	0	0	0
0	0	0	1	0	0	-1	-1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	-1	-1	0	0	0	0	0
0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	-1	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1

C3=

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1

-R=

R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
-22423	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-12457	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-143729	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	-28391	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-17744330	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-2299666	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	-2299665	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-17744339	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-2299668	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	-2299667	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-2299668	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	-2299667	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-69	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	-69	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-96	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-173

There are 29 solutions (table 1) for the set of equations presented from matrix A that contains matrices C_1, C_2, C_3, and R that represent the Hagan-Poiseuille, boundary condition, and conservation balance equation from part 1. There are 13 values for pressures 1 through 13 and 16 values for flows 1 through 16 from the Circle of Willis structure. The residual vector from the results in Table 1 was about zero (shown in Appendix B), which satisfies the equations.

P1	100	F1	0
P2	98.9064	F2	0
P3	98.2989	F3	0
P4	94.8881	F4	0
P5	15.326	F5	0
P6	97.588	F6	0
P7	50.6264	F7	0
P8	15.6358	F8	0
P9	5.0148	F9	0
P10	5.0148	F10	0
P11	5.0131	F11	0
P12	5.0084	F12	0
P13	5	F13	0
		F14	0
		F15	0
		F16	0

Table 1: Solution vector x for pressures and flows of figure 1

Part#3: Sparse Notation

The solution x (flows and pressures) are listed in table 2.

P1	100	F1	0
P2	98.9064	F2	0
P3	98.2989	F3	0
P4	94.8881	F4	0
P5	15.326	F5	0
P6	97.588	F6	0
P7	50.6264	F7	0
P8	15.6358	F8	0
P9	5.0148	F9	0
P10	5.0148	F10	0
P11	5.0131	F11	0
P12	5.0084	F12	0
P13	5	F13	0
		F14	0
		F15	0
		F16	0

Table 2: Solution vector x for pressures and flows of figure 1 using sparse matrices

The solution x for part 3 is the same solution for part 2 (see tables 1 and 2).

Conclusion

The matrix equation has various applications. When solving for a system of equations it proves to be beneficial. The residual error equation can be used to check the solution of the equation solved for using the matrix equation.

Discussion

I have learned that the matrix equation is useful when solving for unknowns in a biological network and sparse notation is more efficient where it uses less memory. Sparse notation can be quicker to use with a preexisting data set, but when writing the data by hand it can prove to be less efficient because it takes more time to write the non-zero values in matrix notation.

Appendix A

```
clc
clear
filename = 'Memo3_Excel.xlsx';
C1 = xlsread(filename, 'C1');
C2 = xlsread(filename, 'C2');
C3 = xlsread(filename, 'C3');
R = xlsread(filename, 'R');
resistance =
[22423,12457,143729,28391,17744330,2299666,2299665,17744339,2299668,2299667,2
299668,2299667,69,69,96,173];

% Part 2
A = [C1 R;(zeros(11,13)) C2;C3 (zeros(2,16))];
b = [0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;100;5];
x=A\b
r = (A*x)-b;
for i = 1:29
r(i)=(A(i,:)*x)-b(i);
end

test1P=(x(1)-x(2))-(resistance(1)*x(14));
test2P=(x(2)-x(3))-(resistance(2)*x(15));
test3P=(x(3)-x(4))-(resistance(3)*x(16));

test1F=(x(14)-x(15));
test2F=(x(15)-x(16)-x(17));
test3F=(x(16)-x(19)-x(18));

% Part 3
%matrix notation
S=spalloc(29,29,1000);
S(1,1)=1;S(1,2)=-1;S(2,2)=1;S(2,3)=-1;S(3,3)=1;S(3,4)=-1;S(4,3)=1;
S(4,6)=-1;S(5,4)=1;S(5,5)=-1;S(6,4)=1;S(6,7)=-1;S(7,6)=1;S(7,7)=-1;
S(8,6)=1;S(8,8)=-1;S(9,8)=1;S(9,10)=-1;S(10,7)=1;S(10,10)=-1;S(11,7)=1;
S(11,9)=-1;S(12,5)=1;S(12,9)=-1;S(13,9)=1;S(13,11)=-1;S(14,10)=1;
S(14,11)=-1;S(15,11)=1;S(15,12)=-1;S(16,12)=1;S(16,13)=-1;S(17,14)=1;
S(17,15)=-1;S(18,15)=1;S(18,16)=-1;S(18,17)=-1;S(19,16)=1;S(19,18)=-1;
S(19,19)=-1;S(20,18)=1;S(20,25)=-1;S(21,17)=1;S(21,20)=-1;S(21,21)=-1;
S(22,19)=1;S(22,20)=1;S(22,23)=-1;S(22,24)=-1;S(23,21)=1;S(23,22)=-1;
S(24,24)=1;S(24,25)=1;S(24,26)=-1;S(25,22)=1;S(25,23)=1;S(25,27)=-1;
S(26,26)=1;S(26,27)=1;S(26,28)=-1;S(27,28)=1;S(27,29)=-1;S(28,1)=1;
S(29,13)=1;S(1,14)=-22423;S(2,15)=-12457;S(3,16)=-143729;S(4,17)=-28391;
S(5,18)=-17744330;S(6,19)=-2299666;S(7,20)=-2299665;S(8,21)=-17744339;
S(9,22)=-2299668;S(10,23)=-2299667;S(11,24)=-2299668;S(12,25)=-2299667;
S(13,26)=-69;S(14,27)=-69;S(15,28)=-96;S(16,29)=-173;
% S-A must equal 0
x=S\b
```

Appendix B

```
x =

  100.0000
   98.9064
   98.2989
   94.8881
   15.3260
   97.5880
   50.6264
   15.6358
    5.0148
    5.0148
    5.0131
    5.0084
    5.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000


r =

   1.0e-13 *

   -0.0400
    0.0222
   -0.0622
   -0.0588
         0
    0.0711
    0.0711
    0.1421
    0.0178
    0.0711
    0.0711
    0.1954
    0.0051
    0.0387
   -0.0000
    0.0038
   -0.0000
    0.0000
   -0.0000
         0
   -0.0000
   -0.0000
    0.0000
         0
    0.0000
         0
         0
    0.1421
   -0.0089


test1P =

  -3.9968e-15


test2P =

   2.2204e-15


test3P =

  -6.2172e-15


test1F =

  -6.5052e-19


test2F =

   7.0812e-19


test3F =

  -1.5247e-20
```

```
x =

  100.0000
   98.9064
   98.2989
   94.8881
   15.3260
   97.5880
   50.6264
   15.6358
    5.0148
    5.0148
    5.0131
    5.0084
    5.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
```