MTH 9821: Homework 2

Due on Sep 18, 2014

Weiyi Chen, Xia Hua, Sam Pfeiffer, Xiaoyu Zhang

Weiyi Chen: 6.10, 6.11, 6.12, 7.19, 8.3, 8.7, 8.8
Xia Hua: 6.3, 6.6, 6.7, 6.8, 6.9, 6.13, 7.19
Sam Pfeiffer: 6.3, 6.6, 6.7, 6.8, 6.9, 6.13, 7.19
Xiaoyu Zhang: 6.10, 6.11, 6.12, 7.19, 8.3, 8.7, 8.8

Problem 6.3

(i)

$R_1 = 1 \leq 1$, $R_2 = 0.5 \leq 2$ and $R_3 = 1.1 \leq 1.1$. By Gershgorin's Circle Theorem, we know that the eigenvalues λ of A are in $disc(1,1) \cup disc(2,0.5) \cup disc(1.1,1.1)$ which lies completely in the non-negative region of the real line. Since all the eigenvalues of A are non-negative, A is symmetric positive semidefinite.

(ii)

We compute the principle minors of A.

$$det\begin{pmatrix}1\end{pmatrix} = 1 > 0 \tag{1}$$

$$det\begin{pmatrix}1 & -0.2\\ -0.2 & 2\end{pmatrix} = 1.96 > 0 \tag{2}$$

$$det\begin{pmatrix}1 & -0.2 & 0.8\\ -0.2 & 2 & 0.3\\ 0.8 & 0.3 & 1.1\end{pmatrix} = 0.69 > 0 \tag{3}$$

The leading principle minors of A is greater than 0, hence we can conclude that A is symmetric positive definite.

Problem 6.6

Answer

We know the Algorithm for Cholesky decomposition of the tridiagonal matrix is:
We can prove the statement by induction. When $i = 1$, it is not hard to see

Algorithm 1 Cholesky_tridiagonal

1: **for** $i = 1 : (N-1)$ **do**
2: $U(i,i) = \sqrt{A(i,i)}$
3: $U(i,i+1) = A(i,i+1)/U(i,i)$
4: $A(i+1,i+1) = A(i+1,i+1) - (U(i,i+1))^2$
5: **end for**
6: $U(N,N) = \sqrt{A(N,N)}$
7: **return** U

$$U(1,1) = \sqrt{2} = \sqrt{\frac{1+1}{1}}, U(1,2) = -\sqrt{\frac{1}{2}} = -\sqrt{\frac{1}{1+1}} \tag{4}$$

Suppose when $i = k$, $U(k,k) = \sqrt{\frac{k+1}{k}}$, and $U(k,k+1) = -\sqrt{\frac{k}{k+1}}$. At this moment, the updated value of $A(k+1,k+1)$ is

$$A'(k+1,k+1) = A(k+1,k+1) - (U(k,k+1))^2 = 2 - \frac{k}{k+1} = \frac{(k+1)+1}{k+1} \tag{5}$$

which means

$$U(k+1,k+1) = \sqrt{\frac{(k+1)+1}{k+1}}, U(k+1,k+1+1) = -\sqrt{\frac{k+1}{(k+1)+1}} \tag{6}$$

The induction holds. So the proof complete. The operation count for the algorithm is $4n + O(1)$.

Problem 6.7

(i)

We know $U_N^t U_N x = b$. Let $U_N x = y$, we have $U_N^t y = b$. Then we can use forward substitution algorithm to compute $y(i)$.

$$y(1) = \frac{b(1)}{U(1,1)} = \frac{b(1)}{\sqrt{2}}; \tag{7}$$

and we know $U(i,i) = \sqrt{\frac{i+1}{i}}$, and $U(i-1,i) = -\sqrt{\frac{i-1}{i-1+1}}$, so for $i = 2 : N$

$$y(i) = \frac{b(i) - y(i-1)U(i-1,i)}{U(i,i)} = \frac{b(i) + y(i-1)\sqrt{i-1/i}}{\sqrt{i+1/i}} \tag{8}$$

After y is computed, then we can use backward substitution to compute x. It follows that

$$x(N) = \frac{y(N)}{U(N,N)} = \frac{y(N)\sqrt{N}}{\sqrt{N+1}}; \tag{9}$$

and for $i = (N-1) : 1$

$$x(i) = \frac{y(i) - x(i+1)U(i,i+1)}{U(i,i)} = \frac{y(i) + x(i+1)\sqrt{i/(i+1)}}{\sqrt{i+1/i}}. \tag{10}$$

The algorithm is established.

(ii)

The first for loop runs $N-1$ times. Each iteration of the for loop has 9 operations. Similarly, the second for loop runs $N-1$ times with 9 operations in each iteration. Calculating $y(1)$ requires 2 operations. Calculating $x(N)$ requires 5 operations. Altogether, we have $18N + 5 = 18N + O(1)$ operations.

Problem 6.8

Answer

LU decomposition for matrix A is the same as LU decomposition for tridiagonal matrix without pivoting. (See Algorithm 2).

Then we can prove (6.91) and (6.92) by induction. It is easy to see when $i = 1$, the induction holds.

Algorithm 2 LU_tridiagonal_no_pivoting

L is a identity matrix; U is a zero matrix;
$U(1,1) = A(1,1) = 2$, $U(1,2) = A(1,2) = -1$
$L(2,1) = A(2,1)/U(1,1) = -1/2$
for $i = 1 : (N-1)$ **do**
    $U(i,i) = A(i,i)$
    $U(i,i+1) = A(i,i+1)$
    $L(i+1,1) = A(i+1,i)/U(i,i)$
    $A(i+1,i+1) = A(i+1,i+1) - L(i+1,1)U(i,i+1)$
end for
$U(N,N) = A(N,N)$
return U

Now suppose when $i = k$, $U(k,k) = (k+1)/k$, $U(k,k+1) = -1$, $L(k+1,k) = -k/(k+1)$. Then the next update of $A(k+1,k+1)$ is

$$A'(k+1,k+1) = A(k+1,k+1) - L(k+1,k)U(k,k+1) = 2 - k/(k+1) = (k+2)/(k+1) \tag{11}$$

So

$$U(k+1,k+1) = A(k+1,k+1) = ((k+1)+1)/(k+1) \tag{12}$$
$$U(k+1,k+2) = A(k+1,k+2) = -1 \tag{13}$$
$$L(k+2,k+1) = A(k+1,k)/U(k,k) = -(k+1)/((k+1)+1) \tag{14}$$

The induction is done. The operation count for this algorithm is $3n + O(1)$.

Problem 6.9

(i)

Similar to problem 6.7, we know $LUx = b$. Let $Ux = y$, we have $Ly = b$. Then we can use forward substitution algorithm to compute $y(i)$.

$$y(1) = \frac{b(1)}{L(1,1)} = b(1); \tag{15}$$

for $i = 2 : N$

$$y(i) = \frac{b(i) - y(i-1)L(i,i-1)}{L(i,i)} = b(i) + y(i-1)(i-1)/i \tag{16}$$

After y is computed, then we can use backward substitution to compute x. It follows that

$$x(N) = \frac{y(N)}{U(N,N)} = \frac{y(N)N}{N+1}; \tag{17}$$

and for $i = (N-1) : 1$

$$x(i) = \frac{y(i) - x(i+1)U(i,i+1)}{U(i,i)} = \frac{y(i) + x(i+1)}{(i+1)/i}. \tag{18}$$

The algorithm is established.

(ii)

The operation count for LU decomposition is $3N + O(1)$. The operation count for forward substitution is $2N + O(1)$; the operation count for backward substitution is $3N + O(1)$. So the total operation count is $8N + O(1)$, which is optimal.

Problem 6.10

```
Function Call:
  x = linear_solve_LU_tridiag_spd_systems(A,b)
Input:
  A = tridiagonal symmetric positive definite matrix of size n
  b(i) = column vectors of size n, i = 1:p
Output:
  x(i) = solution to Ax = b(i)
```

Problem 6.10 continued on next page...

```
for $i = 1:(n-1)$
  L(i,i) = 1; L(i+1,i) = A(i+1,i)/A(i,i);
  U(i,i) = A(i,i); U(i,i+1) = A(i,i+1);
  A(i+1,i+1) = A(i+1,i+1) - L(i+1,i)U(i,i+1);
end
L(n,n) = 1; U(n,n) = A(n,n); //LU decomposition of A
for i = 1:n
  y(1) = b(i)(1);
  for j = 2:n
    y(j) = b_{i}(j) - L(j,j-1)y(j-1); //forward substitution for Ly = b(i)
  end
  x(i)(n) = y(n) / U(n,n);
  for j = (n-1):1
    x(i)(j) = (y(j)-U(j,j+1)x(j+1)) / U(j,j) //backward substition for Ux = y
  end
end
```

The operation count is: $5n^2 - n - 3$

Problem 6.11

```
Function Call:
  U = cholesky(A)
Input:
  A = symmetric positive definite banded matrix of size n of band m
Output:
  U = upper triangular matrix such that U^tU = A
for i = 1:(n-1)
  U(i,i) = sqrt(A(i,i))
  for k = (i+1):min{n,i+m}
    U(i,k) = A(i,k)/U(i,i); //compute row i of U
  end
  for j = (i+1):n
    for k = j:min{n,j+m}
      A(j,k) = A(j,k) - U(i,j)U(i,k);
    end
  end
end
U(n,n) = sqrtA(n,n);
```

The operation count is

Problem 6.13

(i)

Let *disc* be the discount factor, T is the maturity, we know

$$r = -\frac{\ln(disc)}{T}, \tag{19}$$

Problem 6.13 [(i)] continued on next page...

so,

$$r(0, 2/12) = 0.012012016; \tag{20}$$
$$r(0, 5/12) = 0.015650921; \tag{21}$$
$$r(0, 11/12) = 0.019815241; \tag{22}$$
$$r(0, 15/12) = 0.01820559; \tag{23}$$

(ii)

We can construct the linear system based on part (i). According to the efficient cubic spline algorithm, the tridiagonal system is:

$$\begin{pmatrix} 5/6 & 1/4 & 0 \\ 1/4 & 3/2 & 1/2 \\ 0 & 1/2 & 5/3 \end{pmatrix} w = \begin{pmatrix} -0.075098863 \\ -0.037361875 \\ -0.078945557 \end{pmatrix} \tag{24}$$

(iii)

Then we have

$$w = \begin{pmatrix} 0 \\ -0.09220133 \\ 0.006942314 \\ -0.049450028 \\ 0 \end{pmatrix} \tag{25}$$

Then we get the four equations for the cubic spline interpolation for the zero rate curve.

$$\begin{cases} 0.075 + 0.029633244x - 0.09220133x^3, \text{ when } 0 \geq x \leq 1/6 \\ 0.006767143 + 0.042824669x - 0.079148546x^2 + 0.066095763x^3, \text{ when } 1/6 < x \leq 5/12 \\ 0.012908145 - 0.001390545x + 0.026967967x^2 - 0.018797448x^3, \text{ when } 5/12 < x \leq 11/12 \\ -0.020615233 + 0.108322327x - 0.092718803x^2 + 0.024725014x^3, \text{ when } 11/12 < x \leq 5/4 \end{cases} \tag{26}$$

(iv)

The coupon payment date for the 14 month quarterly bond is 2 month. 5 month, 8 month, 11 month, 14 month. The respective zero rates at these five points are:

$$r(0, 2/12) = 0.012012016; \tag{27}$$
$$r(0, 5/12) = 0.015650921; \tag{28}$$
$$r(0, 8/12) = 0.018397264; \tag{29}$$
$$r(0, 11/12) = 0.019815241; \tag{30}$$
$$r(0, 14/12) = 0.018822629; \tag{31}$$

The corresponding cash flow payment is \$0.625, \$0.625, \$0.625, \$0.625, \$100.625 at these five points. The discount factor is 0.998, 0.9935, 0.987810064, 0.982, and 0.978279625 at these five points. So we can calculate the current value of the bond: \$100.9152061.

Problem 7.19

We know

$$\Sigma_x = \begin{pmatrix} 1 & 1 & 0.5 \\ 1 & 4 & -2 \\ 0.5 & -2 & 9 \end{pmatrix} \tag{32}$$

After Cholesky decomposition, we find that

$$U = \begin{pmatrix} 1 & 1 & 0.5 \\ 0 & 1.7321 & -1.4434 \\ 0 & 0 & 2.5820 \end{pmatrix} \tag{33}$$

So $U^t \Sigma_Z U = \Sigma_x$. That means

$$\begin{pmatrix} 1 & 1 & 0.5 \\ 0 & 1.7321 & -1.4434 \\ 0 & 0 & 2.5820 \end{pmatrix} \begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \end{pmatrix} \tag{34}$$

is out desired distribution. Write it explicitly,

$$\begin{cases} X_1 = Z_1 \\ X_2 = Z_1 + 1.7321 Z_2 \\ X_3 = 0.5 Z_1 - 1.4434 Z_2 + 2.5820 Z_3 \end{cases} \tag{35}$$

Problem 8.3

(i)

The problem is same as solving

$$\overline{C} - \overline{P} = PVF - K \cdot disc \tag{36}$$

It follows that the values of PVF and $disc$ can be obtained by solving a least squares problem $y \approx Ax$ with $x = (PVF, disc)^t$ and with 18×1 matrix A and the 16×1 column vector y corresponding to $\overline{C} - \overline{P}$ for each strike.
The solution is $x = (A^t A)^{-1} A^t y$ to this least squares problem, computed as $x = least\_squares(A, y)$, we obtain

$$x = \begin{pmatrix} PVF \\ disc \end{pmatrix} = \begin{pmatrix} 1869.403121 \\ 0.997227746 \end{pmatrix} \tag{37}$$

(ii)

This is using Newton's method recursion for solving

$$f_C(x) = PVF \cdot N(d_1(x)) - K \cdot N(d_2(x)) - C_m \tag{38}$$

$$f_P(x) = -PVF \cdot N(-d_1(x)) + K \cdot N(-d_2(x)) - P_m \tag{39}$$

with $d_1(x)$ and $d_2(x)$ given as

$$d_1(x) = \frac{\ln(PVF/(K \cdot disc))}{x\sqrt{T}} + \frac{x\sqrt{T}}{2}, d_2(x) = \frac{\ln(PVF/(K \cdot disc))}{x\sqrt{T}} - \frac{x\sqrt{T}}{2} \tag{40}$$

Problem 8.3 [(ii)] continued on next page...

The newton's method recursion is

$$x_{k+1} = x_k - \frac{f_C(x_k)}{f'_C(x_k)} \tag{41}$$

$$x_{k+1} = x_k - \frac{f_C(x_k)}{f'_P(x_k)} \tag{42}$$

After calculations, the implied volatilities are as follows (note that the T above is set as 240/365):

Strike	Imp_Vol(call)	Imp_Vol(put)
1450	0.217612	0.218549
1500	0.207823	0.208539
1550	0.198264	0.198997
1600	0.189059	0.189635
1675	0.175484	0.175236
1700	0.170474	0.170174
1750	0.160755	0.160567
1775	0.15649	0.155881
1800	0.151343	0.151331
1825	0.146385	0.146148
1850	0.141347	0.141149
1875	0.136558	0.136063
1900	0.13159	0.131458
1925	0.126686	0.126247
1975	0.117725	0.11731
2000	0.113451	0.114061
2050	0.106157	0.1072
2100	0.100717	0.102236

The implied volatilities from the output corresponding to calls and puts with the same strike are nearly identical, but if more precisely, the implied volatilities corresponding to puts are always a bit larger than the implied volatilities corresponding to calls.

Problem 8.7

(i)

This can be written in least squares form as $y \approx Ax$, with

$$x = \begin{pmatrix} a \\ b_1 \\ b_2 \\ b_3 \end{pmatrix}; y = T_3; A = (1, T_2, T_5, T_{10}) \tag{43}$$

Recall from the textbook the solution to the least squares problem is

$$x \approx (A^t A)^{-1} A^t y \tag{44}$$

By using the Cholesky solver, we compute $x = linear\_solver\_choleasky(A^t A, A^t y)$ with A and y given above, we obtain that

$$x = \begin{pmatrix} a \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0.012302 \\ 0.127208 \\ 0.334045 \\ 0.529777 \end{pmatrix} \tag{45}$$

Problem 8.7 [(i)] continued on next page...

We conclude that the ordinary least square linear regression for the yield of the 3-year bond in terms of the yields of the 2-year, 5-year, and 10-year bonds is

$$T_3 \approx 0.012302 \cdot \mathbf{1} + 0.127208T_2 + 0.334045T_5 + 0.529777T_{10} \tag{46}$$

The approximation error is

$$\text{error}_{linear\_interp} = ||T_3 - T_{3,LR}|| = 0.043013 \tag{47}$$

(ii)

Given the linear interpolation formula

$$T_3 \approx T_{3,linear\_interp} = \frac{2}{3}T_2 + \frac{1}{3}T_5 \tag{48}$$

The approximation error is

$$\text{error}_{linear\_interp} = ||T_3 - T_{3,linear\_interp}|| = 0.206613 \tag{49}$$

(iii)

Similar to problem 13 and problem 14, we need to apply cubic spline interpolation. We are looking for a function $f(x)$ of the form

$$f(x) = f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3, \forall x_{i-1} \leq x \leq x_i, \forall i = 1 : n \tag{50}$$

such that

$$f_i(x_{i-1}) = v_{i-1}, \forall i = 1 : n \tag{51}$$
$$f_i(x_i) = v_i, \forall i = 1 : n \tag{52}$$
$$f_i'(x_i) = f_{i+1}'(x_i), \forall i = 1 : (n-1) \tag{53}$$
$$f_i''(x_i) = f_{i+1}''(x_i), \forall i = 1 : (n-1) \tag{54}$$

Two more constraints is to require that $f_1''(x_0) = 0$ and $f_n''(x_n) = 0$, i.e.,

$$2c_1 + 6d_1 x_0 = 0 \tag{55}$$
$$2c_n + 6d_3 x_n = 0 \tag{56}$$

Let $\overline{x}$ be the $4n \times 1$ vector of the unknowns $a_i, b_i, c_i, d_i, i = 1 : n$, given by

$$\overline{x}(4i-3) = a_i, \overline{x}(4i-2) = b_i, \overline{x}(4i-1) = c_i, \overline{x}(4i) = d_i; \forall i = 1 : (n-1) \tag{57}$$

Above is a linear system with $4n$ equations and $4n$ unknowns which can be expressed in matrix notation as

$$\overline{M}\overline{x} = \overline{b} \tag{58}$$

where $\overline{b}$ is an $4n \times 1$ vector given by

$$\overline{b}(1) = 0; \overline{b}(4n) = 0; \tag{59}$$
$$\overline{b}(4i-2) = v_{i-1}, \overline{b}(4i-1) = v_i, \forall i = 1 : n; \tag{60}$$
$$\overline{b}(4i) = 0, \overline{b}(4i+1) = 0, \forall i = 1 : (n-1) \tag{61}$$

Problem 8.7 [(iii)] continued on next page...

and $\overline{M}$ is the $4n \times 4n$ matrix given by (2.86) - (2.96) on textbook.
In this problem,

$$x_{0:2} = 2, 5, 10; v_{0:2} = T_2[i], T_5[i], T_{10}[i]; n = 2 \tag{62}$$

for $i = 1 : 15$.
We are able to solve $\overline{x} = \overline{M}^{-1}\overline{b}$. As for the first example, $\overline{x}_1$ is

```
[[ 4.85328700e+00]
 [ -2.18580000e-02]
 [ -3.83900000e-03]
 [  6.40000000e-04]
 [  4.98125000e+00]
 [ -9.86360000e-02]
 [  1.15170000e-02]
 [ -3.84000000e-04]]
```

that is

$$f(x) = \begin{cases} 4.853287 - 0.021858x - 0.003839x^2 + 0.000640x^3, & \text{if } 2 \le x \le 5 \\ 4.981250 - 0.098636x + 0.011517x^2 - 0.000384x^3, & \text{if } 5 \le x \le 10 \end{cases} \tag{63}$$

Therefore after doing the same stuffs the other 14 datasets, the approximation error is

$$\text{error}_{cubic\_interp} = ||T_3 - T_{3,cubic\_interp}|| = 0.186435 \tag{64}$$

(iv)

Compare: the ordinary least squares method gives the best answer with lowest error, better than the other two methods using interpolation. For the other two, cubic interpolation generates lower error than linear interpolation, which is expected since using more parameters.

Problem 8.8

(i)

Date	JPM	GS	MS	BAC	RBS	CS	UBS	RY (RBC)	BCS (Barclays)
15-Oct-12	0.016215284	0.028452579	0.012709417	0.035087719	0.034722222	0.047258136	0.034892942	0.016215284	0.00879567
8-Oct-12	-0.012267848	0.007459559	-0.010857143	-0.021459227	0.021276596	-0.006202924	-0.01484375	-0.012267848	0.019310345
1-Oct-12	0.022295767	0.049524982	0.045400239	0.055492639	0.016826923	0.06713948	0.05090312	0.022295767	0.045421774
24-Sep-12	0	-0.026045236	-0.019906323	-0.030735456	-0.066217733	-0.075611888	-0.057275542	0	-0.03814147
17-Sep-12	-0.006575532	-0.038233355	-0.063596491	-0.046073298	-0.011098779	-0.009952402	-0.041543027	-0.006575532	-0.026333558
10-Sep-12	0.005568122	0.043239061	0.067915691	0.085227273	0.146310433	0.092155009	0.091497976	0.005568122	0.124525437
4-Sep-12	0.026066774	0.10035944	0.138666667	0.102756892	0.093184979	0.098650052	0.107623318	0.026066774	0.132416165
27-Aug-12	0.033013648	0.011674641	0.03021978	-0.020858896	0.014104372	0.002602811	-0.004464286	0.033013648	-0.019392917
20-Aug-12	-0.006231672	0.013087736	-0.002056203	0.020025031	-0.027434842	0.05260274	0.014492754	-0.006231672	-0.017398509
13-Aug-12	0.055727554	0.005654675	-0.001368925	0.033635188	0.035511364	0.03811149	0.012844037	0.055727554	0.047743056
6-Aug-12	0.003690037	0.020190969	0.060232221	0.041778976	0.033773862	0.022093023	0.012070566	0.003690037	0.085768143
30-Jul-12	-0.000194175	-0.006521739	0.021497405	0.016438356	-0.01447178	-0.036414566	-0.017335766	-0.000194175	0.01047619
23-Jul-12	0.006645817	0.079466667	0.059701493	0.033994334	0.081377152	0.054341406	0.079802956	0.006645817	0.065989848
16-Jul-12	0	-0.033505155	-0.090714286	-0.09603073	-0.01236476	-0.028686173	-0.0505145	0	-0.032416503
9-Jul-12	-0.002340094	0.020515518	-0.006387509	0.020915033	0.031897927	-0.021336328	-0.029945554	-0.002340094	-0.002938296
2-Jul-12	0.012438302	-0.004086337	-0.030282175	-0.063647491	-0.077941176	-0.026243849	-0.058923997	0.012438302	-0.002929687
25-Jun-12	0.01017152	0.023814632	0.031227821	0.030264817	-0.107611549	-0.024533333	-0.020083682	0.01017152	-0.183413078
18-Jun-12	0.010072522	-0.021209576	-0.010533708	0.005069708	-0.026819923	-0.00424854	-0.003336113	0.010072522	-0.007125891
11-Jun-12	0.024349979	0.011792202	0.042459736	0.045033113	0.121776504	-0.069664032	0.023911187	0.024349979	0.064924115
4-Jun-12	0.01999579	0.020600672	0.077287066	0.07703281	0.129449838	0.069202324	0.041814947	0.01999579	0.113615023
29-May-12	-0.016152413	-0.037466082	-0.039393939	-0.016830295	-0.052147239	-0.040060852	-0.027681661	-0.016152413	-0.063324538
21-May-12	-0.03555023	0.012682308	-0.007518797	0.018571429	0.039872408	0.007150153	0.026642984	-0.03555023	0.024324324
14-May-12	-0.060600375	-0.065019763	-0.106783076	-0.070385126	-0.144611187	-0.070716659	-0.075533662	-0.060600375	-0.134165367
7-May-12	-0.022914757	-0.062962963	-0.06587202	-0.024611399	-0.069796954	-0.027688048	-0.011363636	-0.022914757	-0.040419162
30-Apr-12	-0.050643926	-0.047367028	-0.055687204	-0.061968408	-0.008805031	-0.088346655	-0.021445592	-0.050643926	-0.07093185
23-Apr-12	0.018613721	0.017501346	-0.02764977	-0.013189448	0.039215686	-0.054870775	0.01614205	0.018613721	0.063609467
16-Apr-12	0.028253737	-0.022974395	0.011655012	-0.036951501	-0.026717557	0.013295729	0.001616815	0.028253737	0.004457652
9-Apr-12	-0.021231044	-0.024632227	-0.060755337	-0.059717698	-0.018726592	-0.034241245	-0.047729022	-0.021231044	-0.021802326
2-Apr-12	-0.013725145	-0.051282051	-0.06355715	-0.035602094	-0.093891403	-0.065454545	-0.065467626	-0.013725145	-0.082666667
26-Mar-12	-0.000351803	-0.014316564	-0.03368004	-0.027494908	-0.014492754	-0.023784168	-0.014883062	-0.000351803	-0.042145594
19-Mar-12	-0.008718396	0.026434611	0.040721649	0.005117707	-0.002224694	-0.0140007	-0.01397624	-0.008718396	-0.02125
12-Mar-12	0.021371327	0.048098434	0.063013699	0.216687422	0.091019417	0.116451739	0.071910112	0.021371327	0.066666667
5-Mar-12	0.005011634	-0.022293262	-0.026147279	-0.009864365	-0.066817667	-0.030314513	-0.028384279	0.005011634	-0.0625
27-Feb-12	0.044299065	0.038438019	0.020141535	0.033121019	-0.023230088	-0.013826607	-0.030345801	0.044299065	0.032258065
21-Feb-12	0.023531663	-0.000349314	-0.034682081	-0.017521902	0.021468927	0.031213873	-0.002112676	0.023531663	-0.000644745
13-Feb-12	-0.001337409	0.01569984	-0.025601639	-0.006218905	0.009122007	0.042168675	0.030478955	-0.001337409	0.070393375
6-Feb-12	-0.003427266	-0.029024201	-0.031730293	0.029449424	-0.03520352	-0.076751947	-0.052269601	-0.003427266	-0.018957346
30-Jan-12	0.024780488	0.05153052	0.094411286	0.075757576	0.043628014	0.044134727	0.04379038	0.024780488	0.071843251
23-Jan-12	-0.004854369	0.02792776	0.011525796	0.03125	0.013969732	0.028264331	0.025773196	-0.004854369	0.01026393
17-Jan-12	0.044201135	0.098813421	0.106253795	0.069908815	0.167119565	0.150710032	0.147928994	0.044201135	0.127272727

(ii)

$$r_{JPM} = 0.71916929 r_{GS} - 0.03296911 r_{MS} + 0.273018867 r_{BAC} + 0.28177779 r_{RBS} - 0.04069423 r_{CS} - 0.41320041 r_{UBS} + 0.118243051 r_{RY} - 0.0116432 r_{BCS} \tag{65}$$

$$error = 0.130901079 \tag{66}$$

(iii)

$$r_{JPM} = 0.637082 r_{GS} - 0.02102 r_{MS} + 0.252836 r_{BAC} \tag{67}$$

$$error = 0.165013 \tag{68}$$

(iv)

$$P_{JPM} = -0.00121 * r_{GS} - 0.06248 * P_{MS} + 1.311893 * P_{BAC} + 1.063319 * P_{RBS} + 0.853286 * P_{CS} - 1.72196 * P_{UBS} + 0.525196 * P_{RY} - 0.41481 * P_{BCS} \tag{69}$$

$$error = 6.639531029 \tag{70}$$