Παράδειγμα #3
ΕΠΙΛΥΣΗ ΓΡΑΜΜΙΚΩΝ ΑΛΓΕΒΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΑΠΕΥΘΕΙΑΣ ΜΕΘΟΔΟΥΣ
Επιμέλεια: Ν. Βασιλειάδης

Άσκηση 1

Τα ισοζύγια μάζας του συστήματος διανομής ατμού σε μονάδα διυλιστηρίου δίνονται από τις παρακάτω εξισώσεις:

$$181.60 - x_3 - 132.57 - x_4 - x_5 = 5.1$$
$$1.17x_3 - x_6 = 0$$
$$132.57 - 0.745x_7 = 61.2$$
$$x_5 + x_7 - x_8 - x_9 - x_{10} + x_2 = 99.1$$
$$x_8 + x_9 + x_{10} + x_{11} - x_{12} - x_{13} = -8.4$$
$$x_6 - x_2 = 24.2$$
$$-1.15(181.60) + x_3 - x_6 + x_{12} + x_1 = -19.7$$
$$181.60 - 4.594x_{12} - 0.11x_1 = 35.05$$
$$-0.0423(181.60) + x_{11} = 2.88$$
$$-0.016(181.60) + x_4 = 0$$
$$x_8 - 0.0147x_1 = 0$$
$$x_5 - 0.07x_{14} = 0$$
$$-0.0805(181.60) + x_9 = 0$$
$$x_{12} - x_{14} + x_1 = -97.9$$

Τα x έχουν μονάδες σε 1000kg/hr. Οι εξισώσεις αντιστοιχούν σε ισοζύγια μάζας στις συσκευές: Header 680-psia, De-super heater, Alternator turbine, Header 170-psia, Header 37-psia, Header 215-psia, BFW balance, Condensate drum, Blow down flash drum, Boiler atomizing, Treated feed-water pump, Boiler feed-water pump, Boiler fan, De-aerator. Να προσδιοριστούν οι 14 άγνωστοι, λύνοντας το γραμμικό σύστημα με α) απαλοιφή Gauss με μερική οδήγηση και ολική οδήγηση β) παραγοντοποίηση LU.

Απάντηση:

Αν εφαρμοσθεί η παραγοντοποίηση LU απευθείας στο παραπάνω σύστημα θα αποτύχει, αφού για το πρώτο στοιχείο του κάτω τριγωνικού πίνακα θα ισχύει $l_{11} = a_{11} = 0$, το οποίο για τον υπολογισμό του $u_{12} = a_{12} / l_{11} = 0/0$ οδηγεί σε απροσδιοριστία. Για την εφαρμογή της παραγοντοποίησης LU είναι αναγκαίο να αναδιαταχθούν οι εξισώσεις ώστε τα διαγώνια στοιχεία του πίνακα A να είναι μη μηδενικά. Εκτελώντας τις αναγκαίες πράξεις και αναδιατάσσοντας τις εξισώσεις προκύπτει το ακόλουθο σύστημα:

$$x_1 + x_3 - x_6 + x_{12} = 189.14$$
$$-x_2 + x_6 = 24.2$$
$$x_3 + x_4 + x_5 = 43.93$$
$$x_4 = 2.9056$$
$$x_5 - 0.07 x_{14} = 0$$
$$1.17 x_3 - x_6 = 0$$
$$0.745 x_7 = 71.37$$
$$-0.0147 x_1 + x_8 = 0$$
$$x_9 = 14.6188$$
$$x_2 + x_5 + x_7 - x_8 - x_9 - x_{10} = 99.1$$
$$x_{11} = 10.5617$$
$$0.11 x_1 + 4.594 x_{12} = 146.55$$
$$x_8 + x_9 + x_{10} + x_{11} - x_{12} - x_{13} = -8.4$$
$$x_1 + x_{12} - x_{14} = -97.9$$

Το γραμμικό σύστημα μπορεί να γραφεί στην μορφή $\boldsymbol{Ax = b}$, όπου:

$\boldsymbol{A}$ =

1	0	1	0	0	-1	0	0	0	0	0	1	0	0
0	-1	0	0	0	1	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	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	-0.07
0	0	1.17	0	0	-1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0
-0.0147	0	0	0	0	0	0	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	1	0	1	-1	-1	-1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0
0.11	0	0	0	0	0	0	0	0	0	0	4.594	0	0
0	0	0	0	0	0	0	1	1	1	1	-1	-1	0
1	0	0	0	0	0	0	0	0	0	0	1	0	-1

$\boldsymbol{b}$ =

189.14
24.2
43.93
2.9056
0
0
71.37
0
14.6188
99.1
10.5617
146.55
-8.4
-97.9

Αρχικά το γραμμικό σύστημα επιλύεται σε Mathematica:

```
A={{1.,0.,1.,0.,0.,-1.,0.,0.,0.,0.,0.,1.,0.,0.}, \
{0.,-1.,0.,0.,0.,1.,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.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.}, \
{0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,-0.07}, \
{0.,0.,1.17,0.,0.,-1.,0.,0.,0.,0.,0.,0.,0.,0.}, \
{0.,0.,0.,0.,0.,0.,0.745,0.,0.,0.,0.,0.,0.,0.}, \
{-0.0147,0.,0.,0.,0.,0.,0.,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.,1.,0.,1.,-1.,-1.,-1.,0.,0.,0.,0.}, \
{0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.}, \
```

```
{0.11,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,4.594,0.,0.}, \
{0.,0.,0.,0.,0.,0.,0.,1.,1.,1.,1.,-1.,-1.,0.}, \
{1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,-1.}};

b={189.14,24.2,43.93,2.9056,0.,0.,71.37,0., \
14.6188,99.1,10.5617,146.55,-8.4,-97.9};

LinearSolve[A, b]
```

Με το παραπάνω πρόγραμμα η λύση του γραμμικού συστήματος προκύπτει:

```
{164.7,0.00196857,20.6854,2.9056,20.339,24.202,95.7987,
2.42109,14.6188,-0.000304565,10.5617,27.9567,8.0446,290.557}
```

1α. Απαλοιφή Gauss με μερική και πλήρη οδήγηση

Πρόγραμμα Fortran επίλυσης του παραπάνω γραμμικού συστήματος με απαλοιφή Gauss:

```
Program Gauss_Elimination
implicit none
integer::i,j,k,n
real*8::t,pivot,s,tstart,tend
real*8,allocatable::A(:,:),X(:)
integer,allocatable::TX(:)

!Find program start time
call cpu_time(tstart)

!Open output file
open(100,File="GaussElimination_Results.dat")

!Definition of the linear system of equation
n=14
allocate(A(n,n+1),X(n),TX(n))
a(1,:)=(/1.,0.,1.,0.,0.,-1.,0.,0.,0.,0.,0.,1.,0.,0.,189.14/)
a(2,:)=(/0.,-1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,24.2/)
a(3,:)=(/0.,0.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,43.93/)
a(4,:)=(/0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,2.9056/)
a(5,:)=(/0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,-0.07,0./)
a(6,:)=(/0.,0.,1.17,0.,0.,-1.,0.,0.,0.,0.,0.,0.,0.,0.,0./)
a(7,:)=(/0.,0.,0.,0.,0.,0.,0.745,0.,0.,0.,0.,0.,0.,0.,71.37/)
a(8,:)=(/-0.0147,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0./)
a(9,:)=(/0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,14.6188/)
a(10,:)=(/0.,1.,0.,0.,1.,0.,1.,-1.,-1.,-1.,0.,0.,0.,0.,99.1/)
a(11,:)=(/0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,10.5617/)
a(12,:)=(/0.11,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,4.594,0.,0.,146.55/)
a(13,:)=(/0.,0.,0.,0.,0.,0.,0.,1.,1.,1.,1.,-1.,-1.,0.,-8.4/)
a(14,:)=(/1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,-1.,-97.9/)

!Matrix TX remembers column changes when full pivoting is used
do i=1,n
    TX(i)=i
end do

k=1
```

```
write(100,"(A)") "------------------------------------------------------"
write(100,"(A)") "            Initial linear system table               "
write(100,"(A)") "------------------------------------------------------"
call write_table
do while (k<=n)

    !Select pivoting method
    pivot=find_pivot(3,k)     ! 3=Full pivoting

      if (pivot==0) then
         write(*,*) "Pivot = 0. Gauss elimination cannot continue"
         stop
      endif
      do j=k,n+1
            a(k,j)=a(k,j)/pivot
      end do
      do i=k+1,n
            t=a(i,k)
            do j=k,n+1
                  a(i,j)=a(i,j)-a(k,j)*t
            end do
      end do

    write(100,"(A)") "------------------------------------------------------"
    call write_table

    k=k+1

end do

!Back sabstitution
x(n)=a(n,n+1)

do i=n-1,1,-1
    s=0
      do j=i+1,n
            s=s+a(i,j)*x(j)
    end do
      x(i)=a(i,n+1)-s
end do

!Write solution to screen and ouput file
write(*,"(A)") "For tables in each iteration open file
GaussElimination_Results.dat..."

write(*,*)
write(*,"(A)") "Solution of linear system given below:"
write(100,*)
write(100,"(A)") "----------------------------------"
write(100,"(A)") "----- Linear system solution -----"
write(100,"(A)") "----------------------------------"
do i=1,n
    do j=1,n
        if (tx(j)==i) then
```

```
        write(*,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(j)
        write(100,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(j)
        end if
    end do
end do

!Find program end time
call cpu_time(tend)
write(*,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."
write(100,"(A)") "----------------------------------------------------"
write(100,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."

!Close output file
Close(100)

Contains

subroutine write_table
integer::i
do i=1,n
    write(100,"(10000ES15.5)") a(i,:)
end do
end subroutine write_table

!Finding pivot of row k

!1=No pivoting
!2=Partial pivoting
!3=Full pivoting

real*8 function find_pivot(s,k)
integer,intent(IN)::s,k
integer::i,maxi,maxj,t
real*8::max,temp

if (s==1) then       ! No pivoting
    max=a(k,k)
elseif (s==2) then   ! Partial Pivoting
      max=a(k,k)
      maxi=k
    do i=k,n
        if (abs(a(i,k))>abs(max)) then
              max=a(i,k)
                maxi=i
         endif

      end do
    write(100,*)
    write(100,"(A)") "-------------------------------------------------------
---------"
    write(100,"(15X,A,1X,I0)") "Processing line",k
    write(100,"(A)") "-------------------------------------------------------
---------"
```

```
    write(100,"(A,ES15.5,A,I0,A,I0,A)") "Pivot is",a(maxi,k)," in
(",maxi,",",k,") of previous table"
    write(100,"(A)") "-----------------------------------------------------------
---------"
      if (maxi/=k) then
         do j=1,n+1
           temp=a(k,j)
             a(k,j)=a(maxi,j)
             a(maxi,j)=temp
       end do
      endif
    call write_table

elseif (s==3) then   ! Full Pivoting
    max=a(k,k)
      maxi=k
      maxj=k
      do i=k,n
          do j=k,n
              if (abs(a(i,j))>abs(max)) then
                    max=a(i,j)
                      maxi=i
                      maxj=j
                endif
        end do
    end do
    write(100,*)
    write(100,"(15X,A,1X,I0)") "Processing line",k

write(100,"(A)") "-----------------------------------------------------------
-----"
    write(100,"(A,ES15.5,A,I0,A,I0,A)") "Pivot is",a(maxi,maxj)," in
(",maxi,",",maxj,") of previous table"
    write(100,"(A)") "-----------------------------------------------------------
---------"
    if (maxi/=k) then
        do j=1,n+1
              temp=a(k,j)
                a(k,j)=a(maxi,j)
                a(maxi,j)=temp
        end do
    endif
    if (maxj/=k) then
          do i=1,n
              temp=a(i,k)
                a(i,k)=a(i,maxj)
                a(i,maxj)=temp

        end do
    t=TX(k)
    TX(k)=TX(maxj)
    TX(maxj)=t
    endif
    call write_table

endif
```

```
find_pivot=max
 end function find_pivot

end program
```

Ο παραπάνω κώδικας μπορεί να εκτελέσει την απαλοιφή Gauss χωρίς οδήγηση, με μερική οδήγηση και με πλήρη οδήγηση αλλάζοντας την γραμμή 45 σε pivot=find_pivot(1,k), pivot=find_pivot(2,k) και pivot=find_pivot(3,k) αντίστοιχα. Εκτελώντας το πρόγραμμα με μερική οδήγηση (pivot=find_pivot(2,k)) προκύπτουν τα εξής:

Αρχικό σύστημα

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	-1.000	0	0	0	1.000	0	0	0	0	0	0	0	0	24.200
0	0	1.000	1.000	1.000	0	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	1.170	0	0	-1.000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
-0.015	0	0	0	0	0	0	1.000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	1.000	0	0	1.000	0	1.000	-1.000	-1.000	-1.000	0	0	0	0	99.100
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0.110	0	0	0	0	0	0	0	0	0	0	4.594	0	0	146.550
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
1.000	0	0	0	0	0	0	0	0	0	0	1.000	0	-1.000	-97.900

1η γραμμή: οδηγός 1.000 στην θέση (1,1) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	-1.000	0	0	0	1.000	0	0	0	0	0	0	0	0	24.200
0	0	1.000	1.000	1.000	0	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	1.170	0	0	-1.000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0.015	0	0	-0.015	0	1.000	0	0	0	0.015	0	0	2.780
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	1.000	0	0	1.000	0	1.000	-1.000	-1.000	-1.000	0	0	0	0	99.100
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	-0.110	0	0	0.110	0	0	0	0	0	4.484	0	0	125.745
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	-1.000	0	0	1.000	0	0	0	0	0	0	0	-1.000	-287.040

2η γραμμή: οδηγός -1.000 στην θέση (2,2) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	1.000	1.000	0	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	1.170	0	0	-1.000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0.015	0	0	-0.015	0	1.000	0	0	0	0.015	0	0	2.780
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	1.000	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0	123.300
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	-0.110	0	0	0.110	0	0	0	0	0	4.484	0	0	125.745
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	-1.000	0	0	1.000	0	0	0	0	0	0	0	-1.000	-287.040

3η γραμμή: οδηγός 1.170 στην θέση (6,3) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	1.000	1.000	0.855	0	0	0	0	0	0	0	0	43.930
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	-0.002	0	1.000	0	0	0	0.015	0	0	2.780
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	1.000	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0	123.300
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0.016	0	0	0	0	0	4.484	0	0	125.745
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	0	0	0	0.145	0	0	0	0	0	0	0	-1.000	-287.040

4η γραμμή: οδηγός 1.000 στην θέση (4,4) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	1.000	0.855	0	0	0	0	0	0	0	0	41.024
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	-0.002	0	1.000	0	0	0	0.015	0	0	2.780
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	1.000	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0	123.300
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0.016	0	0	0	0	0	4.484	0	0	125.745
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	0	0	0	0.145	0	0	0	0	0	0	0	-1.000	-287.040

5η γραμμή: οδηγός 1.000 στην θέση (5,5) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	0.855	0	0	0	0	0	0	0	0.070	41.024
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	-0.002	0	1.000	0	0	0	0.015	0	0	2.780
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0.016	0	0	0	0	0	4.484	0	0	125.745
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	0	0	0	0.145	0	0	0	0	0	0	0	-1.000	-287.040

6η γραμμή: οδηγός 1.000 στην θέση (10,6) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0.002	0.998	-0.002	-0.002	0	0.015	0	0	3.044
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	-0.855	0.855	0.855	0.855	0	0	0	0.010	-64.360
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	-0.016	0.016	0.016	0.016	0	4.484	0	-0.001	123.774
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	0	0	0	0	-0.145	0.145	0.145	0.145	0	0	0	-1.010	-304.955

7η γραμμή: οδηγός -0.855 στην θέση (10,7) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0.745	0.745	0.745	0	0	0	0.009	15.270
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	4.484	0	-0.001	124.977
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

8η γραμμή: οδηγός 1.000 στην θέση (8,8) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0.745	0.745	0	-0.011	0	0.009	13.123
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	4.484	0	-0.001	124.977
0	0	0	0	0	0	0	0	1.000	1.000	1.000	-1.015	-1.000	0	-11.283
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

9η γραμμή: οδηγός 1.000 στην θέση (9,9) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0	0.745	0	-0.011	0	0.009	2.232
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	4.484	0	-0.001	124.977
0	0	0	0	0	0	0	0	0	1.000	1.000	-1.015	-1.000	0	-25.902
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

10η γραμμή: οδηγός 0.745 στην θέση (13,10) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0	1.000	1.000	-1.015	-1.000	0	-25.902
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	4.484	0	-0.001	124.977
0	0	0	0	0	0	0	0	0	0	-0.745	0.745	0.745	0.009	21.528
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

11 η γραμμή: οδηγός 1.000 στην θέση (11,11) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0	1.000	1.000	-1.015	-1.000	0	-25.902
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	4.484	0	-0.001	124.977
0	0	0	0	0	0	0	0	0	0	0	0.745	0.745	0.009	29.397
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

12 η γραμμή: 4.484 στην θέση (12,12) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0	1.000	1.000	-1.015	-1.000	0	-25.902
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	1.000	0	0	27.872
0	0	0	0	0	0	0	0	0	0	0	0	0.745	0.009	8.632
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

13 η γραμμή: οδηγός 0.745 στην θέση (13,13) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0	1.000	1.000	-1.015	-1.000	0	-25.902
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	1.000	0	0	27.872
0	0	0	0	0	0	0	0	0	0	0	0	1.000	0.012	11.587
0	0	0	0	0	0	0	0	0	0	0	0	0	-1.012	-294.014

14 η γραμμή: οδηγός -1.012 στην θέση (14,14) του προηγούμενου πίνακα.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	1.000	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	0	0	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0.070	123.300
0	0	0	0	0	0	1.000	-1.000	-1.000	-1.000	0	0	0	-0.012	75.302
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0	2.883
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	0	0	0	1.000	1.000	-1.015	-1.000	0	-25.902
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	0	1.000	0	0	27.872
0	0	0	0	0	0	0	0	0	0	0	0	1.000	0.012	11.587
0	0	0	0	0	0	0	0	0	0	0	0	0	1.000	290.557

Λύση του γραμμικού συστήματος με απαλοιφή Gauss και μερική οδήγηση.

x =

164.6998
0.001967092
20.68544
2.905600
20.33896
24.20197
95.79866
2.421088
14.61880
-0.0003015042
10.56170
27.95669
8.044598
290.5565

Στην συνέχεια εκτελώντας το πρόγραμμα με ολική οδήγηση (pivot=find_pivot(2,k)) προκύπτει:

Αρχικό σύστημα

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0	189.140
0	-1.000	0	0	0	1.000	0	0	0	0	0	0	0	0	24.200
0	0	1.000	1.000	1.000	0	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	1.170	0	0	-1.000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
-0.015	0	0	0	0	0	0	1.000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	1.000	0	0	1.000	0	1.000	-1.000	-1.000	-1.000	0	0	0	0	99.100
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0.110	0	0	0	0	0	0	0	0	0	0	4.594	0	0	146.550
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0	-8.400
1.000	0	0	0	0	0	0	0	0	0	0	1.000	0	-1.000	-97.900

1η Γραμμή: οδηγός 4.594 στην θέση (12,12) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	-1.000	0	0	0	1.000	0	0	0	0	0	0	0	0	24.200
0	0	1.000	1.000	1.000	0	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	1.170	0	0	-1.000	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0	1.000	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	1.000	0	0	1.000	0	1.000	-1.000	-1.000	-1.000	0	0	0	0	99.100
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	1.000	0	0	-1.000	0	0	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	0.024	-1.000	0	23.500
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

2η Γραμμή: οδηγός 1.170 στην θέση (6,3) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	0	1.000	1.000	0.855	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0	2.906
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	-1.000	0	0	1.000	0	0	0	0	0	0	0	0	24.200
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0	1.000	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	1.000	0	1.000	0	1.000	-1.000	-1.000	-1.000	0	0	0	0	99.100
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	-0.145	0	0	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	0.024	-1.000	0	23.500
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

3η Γραμμή: οδηγός 1.000 στην θέση (3,4) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	1.000	0	1.000	0.855	0	0	0	0	0	0	0	0	43.930
0	0	0	0	-1.000	-0.855	0	0	0	0	0	0	0	0	-41.024
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070	0
0	0	0	-1.000	0	1.000	0	0	0	0	0	0	0	0	24.200
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0	1.000	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	1.000	1.000	0	1.000	-1.000	-1.000	-1.000	0	0	0	0	99.100
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	-0.145	0	0	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	0.024	-1.000	0	23.500
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

4η Γραμμή: οδηγός -1.000 στην θέση (4,5) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	1.000	1.000	0	0.855	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0.855	0	0	0	0	0	0	0	0	41.024
0	0	0	0	0	-0.855	0	0	0	0	0	0	0	-0.070	-41.024
0	0	0	0	-1.000	1.000	0	0	0	0	0	0	0	0	24.200
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0	1.000	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	1.000	-0.855	1.000	-1.000	-1.000	-1.000	0	0	0	0	58.076
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	-0.145	0	0	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	0.024	-1.000	0	23.500
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

5η Γραμμή: οδηγός -1.000 στην θέση (6,5) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	-0.855	0	0	0	0	0	0	0	0	0
0	0	1.000	1.000	0	0.855	0	0	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0.855	0	0	0	0	0	0	0	0	41.024
0	0	0	0	1.000	-1.000	0	0	0	0	0	0	0	0	-24.200
0	0	0	0	0	-0.855	0	0	0	0	0	0	0	-0.070	-41.024
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0	1.000	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0.145	1.000	-1.000	-1.000	-1.000	0	0	0	0	82.276
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	-0.145	0	0	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	0.024	-1.000	0	23.500
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

6η Γραμμή: οδηγός 1.000 στην θέση (8,8) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	-0.855	0	0	0	0	0	0	0
0	0	1.000	1.000	0	0	0	0.855	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0.855	0	0	0	0	0	0	41.024
0	0	0	0	1.000	0	0	-1.000	0	0	0	0	0	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0	71.370
0	0	0	0	0	0	0	-0.855	0	0	0	0	0	-0.070	-41.024
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	14.619
0	0	0	0	0	0	1.000	0.145	-1.000	-1.000	0	-0.015	0	0	82.276
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	-0.145	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	0	1.000	1.000	1.000	0.039	-1.000	0	23.500
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

7η Γραμμή: οδηγός 1.000 στην θέση (9,9) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	-0.855	0	0	0	0	0	0	0
0	0	1.000	1.000	0	0	0	0.855	0	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0.855	0	0	0	0	0	0	41.024
0	0	0	0	1.000	0	0	-1.000	0	0	0	0	0	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	-0.855	0	0	0	0	0	-0.070	-41.024
0	0	0	0	0	0	0	0	0.745	0	0	0	0	0	71.370
0	0	0	0	0	0	0	0.145	1.000	-1.000	0	-0.015	0	0	96.894
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	-0.145	0	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	0	0	1.000	1.000	0.039	-1.000	0	8.882
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

8η Γραμμή: οδηγός 1.000 στην θέση (10,9) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	-0.855	0	0	0	0	0	0
0	0	1.000	1.000	0	0	0	0	0.855	0	0	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0.855	0	0	0	0	0	41.024
0	0	0	0	1.000	0	0	0	-1.000	0	0	0	0	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0.145	-1.000	0	-0.015	0	0	96.894
0	0	0	0	0	0	0	0	-0.108	0.745	0	0.011	0	0	-0.816
0	0	0	0	0	0	0	0	-0.855	0	0	0	0	-0.070	-41.024
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0	10.562
0	0	0	0	0	0	0	0	-0.145	0	0	0.976	0	0	157.240
0	0	0	0	0	0	0	0	0	1.000	1.000	0.039	-1.000	0	8.882
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

9η Γραμμή: οδηγός 1.000 στην θέση (11,11) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	0	0	-0.855	0	0	0	0
0	0	1.000	1.000	0	0	0	0	0	0	0.855	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0.855	0	0	0	41.024
0	0	0	0	1.000	0	0	0	0	0	-1.000	0	0	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0	-1.000	0.145	-0.015	0	0	96.894
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	10.562
0	0	0	0	0	0	0	0	0	0	-0.855	0	0	-0.070	-41.024
0	0	0	0	0	0	0	0	0	0.745	-0.108	0.011	0	0	-0.816
0	0	0	0	0	0	0	0	0	0	-0.145	0.976	0	0	157.240
0	0	0	0	0	0	0	0	0	1.000	0	0.039	-1.000	0	-1.680
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

10η Γραμμή: οδηγός 1.000 στην θέση (13,10) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	0	0	-0.855	0	0	0	0
0	0	1.000	1.000	0	0	0	0	0	0	0.855	0	0	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0.855	0	0	0	41.024
0	0	0	0	1.000	0	0	0	0	0	-1.000	0	0	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0	-1.000	0.145	-0.015	0	0	96.894
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	10.562
0	0	0	0	0	0	0	0	0	1.000	0	0.039	-1.000	0	-1.680
0	0	0	0	0	0	0	0	0	0	-0.108	-0.018	0.745	0	0.435
0	0	0	0	0	0	0	0	0	0	-0.145	0.976	0	0	157.240
0	0	0	0	0	0	0	0	0	0	-0.855	0	0	-0.070	-41.024
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-1.000	-129.800

11η Γραμμή: οδηγός -1.000 στην θέση (14,14) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	0	0	0	0	0	-0.855	0
0	0	1.000	1.000	0	0	0	0	0	0	0	0	0	0.855	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0.855	41.024
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-1.000	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0	-1.000	0	-0.015	0	0.145	96.894
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	10.562
0	0	0	0	0	0	0	0	0	1.000	0	0.039	-1.000	0	-1.680
0	0	0	0	0	0	0	0	0	0	1.000	-0.976	0	0	129.800
0	0	0	0	0	0	0	0	0	0	0	0.976	0	-0.145	157.240
0	0	0	0	0	0	0	0	0	0	0	-0.068	0	-0.855	-31.938
0	0	0	0	0	0	0	0	0	0	0	-0.018	0.745	-0.108	0.435

12η Γραμμή: 0.976 στην θέση (12,12) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	0	0	0	0	0	-0.855	0
0	0	1.000	1.000	0	0	0	0	0	0	0	0	0	0.855	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0.855	41.024
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-1.000	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0	-1.000	0	-0.015	0	0.145	96.894
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	10.562
0	0	0	0	0	0	0	0	0	1.000	0	0.039	-1.000	0	-1.680
0	0	0	0	0	0	0	0	0	0	1.000	-0.976	0	0	129.800
0	0	0	0	0	0	0	0	0	0	0	1.000	0	-0.149	161.097
0	0	0	0	0	0	0	0	0	0	0	0	0	-0.865	-20.932
0	0	0	0	0	0	0	0	0	0	0	0	0.745	-0.111	3.309

13η Γραμμή: οδηγός -0.865 στην θέση (13,14) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	0	0	0	0	-0.855	0	0
0	0	1.000	1.000	0	0	0	0	0	0	0	0	0.855	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0.855	0	41.024
0	0	0	0	1.000	0	0	0	0	0	0	0	-1.000	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0	-1.000	0	-0.015	0.145	0	96.894
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	10.562
0	0	0	0	0	0	0	0	0	1.000	0	0.039	0	-1.000	-1.680
0	0	0	0	0	0	0	0	0	0	1.000	-0.976	0	0	129.800
0	0	0	0	0	0	0	0	0	0	0	1.000	-0.149	0	161.097
0	0	0	0	0	0	0	0	0	0	0	0	1.000	0	24.202
0	0	0	0	0	0	0	0	0	0	0	0	0	0.745	5.993

14η Γραμμή: οδηγός 0.745 στην θέση (14,14) του προηγούμενου πίνακα.

1.000	0	0	0	0	0	0	0	0	0	0	0.024	0	0	31.900
0	1.000	0	0	0	0	0	0	0	0	0	0	-0.855	0	0
0	0	1.000	1.000	0	0	0	0	0	0	0	0	0.855	0	43.930
0	0	0	1.000	0	0	0	0	0	0	0	0	0.855	0	41.024
0	0	0	0	1.000	0	0	0	0	0	0	0	-1.000	0	-24.200
0	0	0	0	0	1.000	0	0	0	0	0	-0.015	0	0	0
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0	14.619
0	0	0	0	0	0	0	1.000	0	-1.000	0	-0.015	0.145	0	96.894
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0	10.562
0	0	0	0	0	0	0	0	0	1.000	0	0.039	0	-1.000	-1.680
0	0	0	0	0	0	0	0	0	0	1.000	-0.976	0	0	129.800
0	0	0	0	0	0	0	0	0	0	0	1.000	-0.149	0	161.097
0	0	0	0	0	0	0	0	0	0	0	0	1.000	0	24.202
0	0	0	0	0	0	0	0	0	0	0	0	0	1.000	8.045

Λύση του γραμμικού συστήματος με απαλοιφή Gauss και ολική οδήγηση.

	164.6998
	0.001967092
	20.68544
	2.905600
	20.33896
	24.20197
x **=**	95.79866
	2.421088
	14.61880
	-0.0003015042
	10.56170
	27.95669
	8.044598
	290.5565

1β. Παραγοντοποίηση LU

Πρόγραμμα Fortran επίλυσης του παραπάνω γραμμικού συστήματος με παραγοντοποίηση LU:

```
program LU_Decomposition
implicit none
integer::i,j,n,k
real*8::s,tstart,tend
real*8,allocatable::x(:),y(:)
real*8,allocatable::u(:,:),l(:,:),a(:,:)

!Find program start time
call cpu_time(tstart)

!Open output file
open(100,file="LUDecomposition_results.dat")

!Definition of the linear system of equations
n=14
allocate(a(n,n+1),l(n,n),u(n,n),y(n),x(n))
a(1,:)=(/1.,0.,1.,0.,0.,-1.,0.,0.,0.,0.,0.,1.,0.,0.,189.14/)
a(2,:)=(/0.,-1.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,24.2/)
```

```
a(3,:)=(/0.,0.,1.,1.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,43.93/)
a(4,:)=(/0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,2.9056/)
a(5,:)=(/0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0.,0.,-0.07,0./)
a(6,:)=(/0.,0.,1.17,0.,0.,-1.,0.,0.,0.,0.,0.,0.,0.,0.,0./)
a(7,:)=(/0.,0.,0.,0.,0.,0.,0.745,0.,0.,0.,0.,0.,0.,0.,71.37/)
a(8,:)=(/-0.0147,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,0.,0./)
a(9,:)=(/0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,0.,0.,14.6188/)
a(10,:)=(/0.,1.,0.,0.,1.,0.,1.,-1.,-1.,-1.,0.,0.,0.,0.,99.1/)
a(11,:)=(/0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,0.,0.,10.5617/)
a(12,:)=(/0.11,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,4.594,0.,0.,146.55/)
a(13,:)=(/0.,0.,0.,0.,0.,0.,0.,1.,1.,1.,1.,-1.,-1.,0.,-8.4/)
a(14,:)=(/1.,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,1.,0.,-1.,-97.9/)

write(100,"(A)") "-----------------------------------------------"

write(100,"(A)") "--------- Linear system of equations ----------"
write(100,"(A)") "-----------------------------------------------"
do i=1,n
    write(100,"(10000ES15.5)") a(i,:)
end do

!Find upper/lower diagonal matrices L and U
u=0.; l=0.
do i=1,n
    do j=i,n
        s=0.
        do k=1,i-1
        s=s+l(j,k)*u(k,i)
        end do
        l(j,i)=a(j,i)-s
    end do

    do j=i,n
        if (j==i) then
            u(i,j)=1.
        else
            s=0.
            do k=1,i-1
            s=s+l(i,k)*u(k,j)
            end do
            u(i,j)=(a(i,j)-s)/l(i,i)
        end if
    end do
end do

!Backsubstitution for L*y=b
y(1)=a(1,n+1)/l(1,1)
do i=2,n
    s=0.0
    do j=1,i-1
        s=s+l(i,j)*y(j)
    end do
    y(i)=(a(i,n+1)-s)/l(i,i)
end do
```

```
!Backsubstitution for U*x=y
x(n)=y(n)
do i=n-1,1,-1
    s=0.0
    do j=n,i+1,-1
        s=s+u(i,j)*x(j)
    end do
    x(i)=y(i)-s
end do

!Write results to screen and ouput file
write(*,"(A)") "For L and U tables open file LUDecomposition_results.dat..."
write(*,*)
write(*,"(A)") "Solution of linear system given below:"

do i=1,n
    write(*,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(i)
end do
write(100,"(A)") "------------------------------------------------"
write(100,"(A)") "----------- Lower diagonal matrix L -----------"
write(100,"(A)") "------------------------------------------------"
do i=1,n
    write(100,"(10000ES15.5)") l(i,:)
end do
write(100,"(A)") "------------------------------------------------"
write(100,"(A)") "----------- Upper diagonal matrix U -----------"
write(100,"(A)") "------------------------------------------------"
do i=1,n
    write(100,"(10000ES15.5)") u(i,:)
end do
write(100,"(A)") "-----------------------------------"
write(100,"(A)") "----- Linear system solution -----"
write(100,"(A)") "-----------------------------------"
do i=1,n
    write(100,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(i)
end do

!Find program end time
call cpu_time(tend)
write(*,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."
write(100,"(A)") "-----------------------------------------------------"
write(100,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."

!Close output file
close(100)

end program
```

Τρέχοντας τον παραπάνω κώδικα προκύπτουν τα εξής:

Κάτω τριγωνικός πίνακας L.

1.000	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1.000	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1.000	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1.000	0	0	0	0	0	0	0	0	0
0	0	1.170	-1.170	-1.170	-1.000	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.745	0	0	0	0	0	0	0
-0.015	0	0.015	-0.015	-0.015	-0.015	0	1.000	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0
0	1.000	0	0	1.000	1.000	1.000	-1.000	-1.000	-1.000	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0
0.110	0	-0.110	0.110	0.110	0.110	0	0	0	0	0	4.484	0	0
0	0	0	0	0	0	0	1.000	1.000	1.000	1.000	-1.000	-1.000	0
1.000	0	-1.000	1.000	1.000	1.000	0	0	0	0	0	0	0	-1.012

Άνω τριγωνικός πίνακας U.

1.000	0	1.000	0	0	-1.000	0	0	0	0	0	1.000	0	0
0	1.000	0	0	0	-1.000	0	0	0	0	0	0	0	0
0	0	1.000	1.000	1.000	0	0	0	0	0	0	0	0	0
0	0	0	1.000	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1.000	0	0	0	0	0	0	0	0	-0.070
0	0	0	0	0	1.000	0	0	0	0	0	0	0	0.082
0	0	0	0	0	0	1.000	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1.000	0	0	0	0.015	0	0
0	0	0	0	0	0	0	0	1.000	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1.000	0	-0.015	0	0.012
0	0	0	0	0	0	0	0	0	0	1.000	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1.000	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1.000	0.012
0	0	0	0	0	0	0	0	0	0	0	0	0	1.000

Λύση του γραμμικού συστήματος με παραγοντοποίηση LU.

***x* =**

164.6998
0.001967092
20.68544
2.9056
20.33896
24.20197
95.79866
2.421088
14.6188
-0.000301504
10.5617
27.95669
8.044598
290.5565

Άσκηση 2

Το παρακάτω αλγεβρικό τρι-διαγώνιο σύστημα προκύπτει από την διακριτοποίηση της συνήθους διαφορικής εξίσωσης μεταφοράς θερμότητας με αγωγή σε μια ράβδο με αδιάστατο μήκος $0 < x < 1$. Οι άγνωστοι του συστήματος T_i, $i = 1, 2, ..., N$ αντιστοιχούν στις θερμοκρασίες της ράβδου στα εσωτερικά ισαπέχοντα σημεία της ράβδου x_i, $i = 1, 2, ..., N$, ενώ οι ποσότητες A και B είναι οι θερμοκρασίες στα δύο άκρα της ράβδου αντίστοιχα.

$$\begin{bmatrix} -2 & 1 & & & & & & \\ 1 & -2 & 1 & & & & & \\ & . & . & . & & & & \\ & & 1 & -2 & 1 & & & \\ & & & . & . & . & & \\ & & & & 1 & -2 & 1 & \\ & & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ . \\ T_i \\ . \\ T_{N-1} \\ T_N \end{bmatrix} = \begin{bmatrix} -A \\ 0 \\ . \\ 0 \\ . \\ 0 \\ -B \end{bmatrix}, \quad i = 1, 2, ..., N$$

Να λυθεί το παραπάνω τρι-διαγώνιο σύστημα για τρία εσωτερικά σημεία και θεωρώντας $A = 600$ και $B = 300$ με α) απαλοιφή Gauss χωρίς οδήγηση, με μερική και με ολική οδήγηση, β) παραγοντοποίηση LU, γ) παραγοντοποίηση Cholesky και δ) Αλγόριθμο Thomas.
Τέλος συγκριθούν οι χρόνοι που απαιτούνται από τις παραπάνω μεθόδους για την επίλυση του συστήματος με N=1001 εσωτερικά σημεία και θεωρώντας τα ίδια A και B.

Απάντηση:

2α. Απαλοιφή Gauss.

Για την απαλοιφή Gauss χρησιμοποιείται ο κώδικας που δόθηκε στο υποερώτημα α της 1^{ης} άσκησης. Η μόνη αλλαγή που απαιτείται είναι ο ορισμός του συστήματος που πρέπει να επιλυθεί ο οποίος δίνεται παρακάτω:

```
!Definition of the linear system of equations
n=3
allocate(A(n,n+1),X(n),TX(n))
a=0. ; a(1,1)=-2 ; a(2,1)=1
Do i=2,n-1
    a(i,i)=-2
    a(i-1,i)=1
    a(i+1,i)=1
Enddo
a(n,n)=-2; a(n-1,n)=1; a(1,n+1)=-600; a(n,n+1)=-300
```

N=3: Η απαλοιφή Gauss χωρίς οδήγηση (pivot=find_pivot(1,k)) δίνει τα εξής:

Αρχικός πίνακας:

-2.000	1.000	0	-600
1.000	-2.000	1.000	0
0	1.000	-2.000	-300

1^{η} γραμμή:

1.000	-0.500	0	300
0	-1.500	1.000	-300
0	1.000	-2.000	-300

2^{η} γραμμή:

1.000	-0.500	0	300
0	1.000	-0.667	200
0	0	-1.333	-500

3^{η} γραμμή:

1.000	-0.500	0	300
0	1.000	-0.667	200
0	0	1.000	375

Η απαλοιφή Gauss με μερική οδήγηση (pivot=find_pivot(2,k)) δίνει τα εξής:

Αρχικός πίνακας:

-2.000	1.000	0	-600
1.000	-2.000	1.000	0
0	1.000	-2.000	-300

1^{η} γραμμή: οδηγός -2 στην θέση (1,1)

1.000	-0.500	0	300
0	-1.500	1.000	-300
0	1.000	-2.000	-300

2^{η} γραμμή: οδηγός -1.5 στην θέση (2,2)

1.000	-0.500	0	300
0	1.000	-0.667	200
0	0	-1.333	-500

3^{η} γραμμή: οδηγός -1.333 στην θέση (3,3)

1.000	-0.500	0	300.000
0	1.000	-0.667	200.000
0	0	1.000	375.000

N=3: Η απαλοιφή Gauss με ολική οδήγηση (pivot=find_pivot(3,k)) δίνει τα εξής:

Αρχικός πίνακας:

-2.000	1.000	0	-600
1.000	-2.000	1.000	0
0	1.000	-2.000	-300

1^{η} γραμμή: οδηγός -2 στην θέση (1,1)

1.000	-0.500	0	300
0	-1.500	1.000	-300
0	1.000	-2.000	-300

2^{η} γραμμή: οδηγός -2 στην θέση (3,3)

1.000	0	-0.500	300
0	1.000	-0.500	150
0	0	-1.000	-450

3^{η} γραμμή: οδηγός -1 στην θέση (3,3)

1.000	0	-0.500	300
0	1.000	-0.500	150
0	0	1.000	450

Χρησιμοποιώντας απαλοιφή Gauss η λύση του συστήματος προκύπτει να είναι:

x	T(x)		
	Χωρίς οδήγηση	Μερική οδήγηση	Ολική οδήγηση
0.25	525	525	525
0.50	450	450	450
0.75	375	375	375

Παρατηρείται ότι λόγω της ιδιαίτερης μορφής του πίνακα συντελεστών $\boldsymbol{A}$ του συστήματος η απαλοιφή Gauss χωρίς οδήγηση δεν αποτυγχάνει και μάλιστα δίνει αποτελέσματα σε τέλεια συμφωνία με την μερική και την ολική οδήγηση.

2β. Παραγοντοποίηση LU

Για την παραγοντοποίηση LU χρησιμοποιείται ο κώδικας που δόθηκε στο υποερώτημα β της 1ης άσκησης. Η μόνη αλλαγή που απαιτείται είναι ο ορισμός του συστήματος που πρέπει να επιλυθεί ο οποίος δίνεται παρακάτω:

```
!Definition of the linear system of equations
n=3
allocate(a(n,n+1),l(n,n),u(n,n),y(n),x(n))
a=0. ; a(1,1)=-2 ; a(2,1)=1
Do i=2,n-1
    a(i,i)=-2
    a(i-1,i)=1
    a(i+1,i)=1
Enddo
a(n,n)=-2; a(n-1,n)=1; a(1,n+1)=-600; a(n,n+1)=-300
```

N=3: Τρέχοντας τον παραπάνω κώδικα η παραγοντοποίηση LU δίνει:

Κάτω τριγωνικός πίνακας L

-2.000	0	0
1.000	-1.500	0
0	1.000	-1.333

Άνω τριγωνικός πίνακας U

1.000	-0.500	0
0	1.000	-0.667
0	0	1.000

Η λύση του συστήματος με παραγοντοποίηση LU προκύπτει να είναι:

x	T(x)
0.25	525
0.50	450
0.75	375

2γ. Παραγοντοποίηση Cholesky

Εφόσον ο πίνακας συντελεστών $\boldsymbol{A}$ του συστήματος είναι συμμετρικός ως προς την κύρια διαγώνιο του μπορεί να εφαρμοσθεί η παραγοντοποίηση Cholesky.

Ο κώδικας σε Fortran που επιλύει το εν λόγω σύστημα με Cholesky δίνεται παρακάτω:

```
program Cholesky_Decomposition
implicit none
integer::i,j,n,k
real*8::s,tstart,tend
real*8,allocatable::x(:),y(:)
real*8,allocatable::a(:,:),l(:,:)

!Find program start time
call cpu_time(tstart)

!Open output file
open(100,file="CholeskyDecomposition_results.dat")

!Definition of the linear system of equations
n=3
```

```
allocate(a(n,n+1),l(n,n),x(n),y(n))
a=0. ; a(1,1)=-2 ; a(2,1)=1
Do i=2,n-1
    a(i,i)=-2
    a(i-1,i)=1
    a(i+1,i)=1
Enddo
a(n,n)=-2; a(n-1,n)=1; a(1,n+1)=-600; a(n,n+1)=-300

!Transform to -A*x=-b so that all diagonal elements are positive
a=-a

!Write system of equations to be solved to output file
write(100,"(A)") "---------------------------------------------"
write(100,"(A)") "--------- Linear system of equations ----------"
write(100,"(A)") "---------------------------------------------"
do i=1,n
    write(100,"(10000ES15.5)") a(i,:)
end do

!Find lower diagonal matrice L
l=0.
l(1,1)=sqrt(a(1,1))
do i=2,n
    !Computations below the diagonal
    do j=1,i-1
        s=0.

        do k=1,j-1
            s=s+l(j,k)*l(i,k)
        end do
        l(i,j)=(a(i,j)-s)/l(j,j)
    end do
    !Computations for the diagonal
    s=0.
    do k=1,i-1
    s=s+l(i,k)**2
    end do
    l(i,i)=sqrt(a(i,i)-s)
end do

!Backsubstitution for L*y=b
y(1)=a(1,n+1)/l(1,1)
do i=2,n
    s=0.0
    do j=1,i-1
        s=s+l(i,j)*y(j)
    end do
    y(i)=(a(i,n+1)-s)/l(i,i)
end do

!Backsubstitution for Transpose(L)*x=y
x(n)=y(n)/l(n,n)
do i=n-1,1,-1
    s=0.0
    do j=n,i+1,-1
```

```
        s=s+l(j,i)*x(j)
    end do
    x(i)=(y(i)-s)/l(i,i)
end do

!Write results to screen and ouput file
write(*,"(A)") "For L table open file CholeskyDecomposition_results.dat..."
write(*,*)
write(*,"(A)") "Solution of linear system given below:"
do i=1,n
    write(*,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(i)
end do
write(100,"(A)") "-----------------------------------------------"
write(100,"(A)") "----------- Lower diagonal matrix L -----------"
write(100,"(A)") "-----------------------------------------------"
do i=1,n
    write(100,"(10000ES15.5)") l(i,:)
end do
write(100,"(A)") "----------------------------------"
write(100,"(A)") "----- Linear system solution -----"

write(100,"(A)") "----------------------------------"
do i=1,n
    write(100,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(i)
end do

!Find program end time
call cpu_time(tend)

write(*,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."
write(100,"(A)") "----------------------------------------------------"
write(100,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."

!Close output file
close(100)

end program
```

Παρατηρείται ότι για το πρώτο διαγώνιο στοιχείο ισχύει ότι $a_{11} = -2$ που οδηγεί σε σφάλμα στο υπολογισμό του στοιχείου $l_{11} = \sqrt{-2}$. Για την παράκαμψη του προβλήματος είναι αναγκαίος ο μετασχηματισμός του συστήματος από $\boldsymbol{Ax} = \boldsymbol{b}$ σε $-\boldsymbol{Ax} = -\boldsymbol{b}$, ο οποίος γίνεται στην γραμμή 26 του κώδικα. Το παραπάνω πρόγραμμα δίνει τα εξής (N=3):

Κάτω τριγωνικός πίνακας L

1.414	0	0
-0.707	1.225	0
0	-0.816	1.155

Η λύση του συστήματος με παραγοντοποίηση Cholesky προκύπτει να είναι:

x	T(x)
0.25	525
0.50	450
0.75	375

2δ. Αλγόριθμος Thomas

Εφόσον το προς επίλυση σύστημα είναι τριδιαγώνιο εφαρμόζεται ο αλγόριθμος Thomas.
Πρόγραμμα Fortran επίλυσης του παραπάνω συστήματος με αλγόριθμο Thomas:

```
program Thomas
implicit none
integer::i,n
real*8::tstart,tend
real*8,allocatable::e(:),g(:),x(:)
real*8,allocatable::a(:,:)

!Find program start time
call cpu_time(tstart)

!Open output file
open(100,file="Thomas_results.dat")
!Definition of the linear system of equations
n=3
allocate(a(n,n+1),e(n),g(n),x(n))
a=0. ; a(1,1)=-2 ; a(2,1)=1
Do i=2,n-1
    a(i,i)=-2
    a(i-1,i)=1

    a(i+1,i)=1
Enddo
a(n,n)=-2; a(n-1,n)=1; a(1,n+1)=-600; a(n,n+1)=-300
write(100,"(A)") "-------------------------------------------------"
write(100,"(A)") "--------- Linear system of equations ----------"
write(100,"(A)") "-------------------------------------------------"
do i=1,n
    write(100,"(10000ES15.5)") a(i,:)
end do

!Find vectors e and g
e(1)=a(1,1) ; g(1)=a(1,n+1)/e(1)
do i=2,n
    e(i)=a(i,i)-a(i,i-1)*a(i-1,i)/e(i-1)
    g(i)=(a(i,n+1)-a(i,i-1)*g(i-1))/e(i)
end do

!Backsubstitution to find x
x(n)=g(n)
do i=n-1,1,-1
    x(i)=g(i)-a(i,i+1)*x(i+1)/e(i)
end do
```

```
!Write results to screen and ouput file
write(*,"(A)") "For e and g vectors open file Thomas_results.dat..."
write(*,*)
write(*,"(A)") "Solution of linear system given below:"
do i=1,n
    write(*,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(i)
end do
write(100,"(A)") "-----------------------------------------------"
write(100,"(A)") "--------------- Vectors e and g ---------------"
write(100,"(A)") "-----------------------------------------------"
do i=1,n
    write(100,"(2ES15.5)") e(i),g(i)
end do
write(100,"(A)") "---------------------------------"
write(100,"(A)") "----- Linear system solution -----"
write(100,"(A)") "---------------------------------"
do i=1,n
    write(100,"(A2,I0,,A2,ES20.10)")"x(",i,")=",x(i)
end do

!Find program end time
call cpu_time(tend)
write(*,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."
write(100,"(A)") "----------------------------------------------------"
write(100,"(A,ES14.4,2X,A)") "Program has used", tend-tstart,"seconds of CPU
time."

!Close output file
close(100)

end program
```

Εκτελώντας το παραπάνω πρόγραμμα ο αλγόριθμος Thomas δίνει τα εξής:

Διάνυσμα e	Διάνυσμα g	x	T(x)
-2.000	300	0.25	525
-1.500	200	0.50	450
-1.333	375	0.75	375

2ε. Σύγκριση χρόνων για την επίλυση του συστήματος για N=3003

Για την επίλυση του αλγεβρικού γραμμικού συστήματος με N=3003 χρησιμοποιούνται οι κώδικες που δόθηκαν παραπάνω. Η μόνη αλλαγή που απαιτείται είναι η αλλαγή του μεγέθους του συστήματος προς επίλυση ως:

```
!Definition of the linear system of equations
n=3003
```

Επίσης για την σωστή μέτρηση του απαιτούμενου χρόνου πρέπει όλες οι εντολές εξόδου (write, print) των προγραμμάτων να σβηστούν ή να μετατραπούν σε σχόλια, ώστε ο μετρούμενος χρόνος να αντιστοιχεί στον χρόνο που απαιτείται καθαρά για την επίλυση του συστήματος και όχι για το γράψιμο τον αποτελεσμάτων στην οθόνη ή στον σκληρό δίσκο (αρχείο).

Στον παρακάτω πίνακα δίνεται ο χρόνος που απαιτείται για την λύση του 3003×3003 συστήματος με τις διαφορετικές μεθόδους καθώς και η τιμή της λύσης που προκύπτει στα σημεία $x = 0.25$, $x = 0.5$ και $x = 0.75$.

Μέθοδος	Χρόνος CPU (sec)	T(x=0.25)	T(x=0.50)	T(x=0.75)
Απαλοιφή Gauss (χωρίς οδήγηση)	32.701	525	450	375
Απαλοιφή Gauss (μερική οδήγηση)	32.836	525	450	375
Απαλοιφή Gauss (ολική οδήγηση)	111.37	525	450	375
Παραγοντοποίηση LU	19.026	525	450	375
Παραγοντοποίηση Cholesky	13.427	525	450	375
Αλγόριθμος Thomas	0.0090	525	450	375

Παρατηρείται ότι η λύση για $N = 3003$ έχει τέλεια σύγκριση με τις διαφορετικές μεθόδους αλλά και με την λύση του συστήματος για $N = 3$ στα ίδια x.

Όσο αφορά τους απαιτούμενους χρόνους CPU η απαλοιφή Gauss απαιτεί τον μεγαλύτερο υπολογιστικό χρόνο. Η χρήση μερικής οδήγησης αυξάνει ελάχιστα τον απαιτούμενο χρόνο ενώ η χρήση ολικής οδήγησης αυξάνει τον χρόνο κατά ~3.5 φορές.

Παρατηρείται επίσης, η παραγοντοποίηση Cholesky απαιτεί υπολογιστικό χρόνο μειωμένο κατά ~30% σε σχέση με την παραγοντοποίηση LU.

Τέλος, η γρηγορότερη μέθοδος είναι ο αλγόριθμος Thomas ο οποίος μειώνει τον υπολογιστικό χρόνο κατά ~10^3-10^4 φορές.

Από τα παραπάνω γίνεται προφανές ότι η επιλογή της μεθόδου που θα εφαρμοστεί ανάλογα με το σύστημα προς επίλυση είναι μείζονος σημασίας και ότι σε τριδιαγώνια συστήματα επιλέγεται ο αλγόριθμος Thomas.