**Algorithm for Jacobi’s Method**

The argumented matrix A of order n\*(n+1) is stored using a two dimensional array named a size n\*(n+1).The solution vectors in two successive iterations are stored in one dimensional arrays named old\_x and new\_x each of size n.The variable epsilon represents the prescribed tolerance and variable maxiter represents the number of permitted iterations.

1. Start
2. read : n
3. for i=1 to n by 1 do

for j=1 to (n+1) by 1 do

read : aij

endfor

endfor

1. read maxiter,epsilon
2. for i=1 to n by 1 do

set old\_xi = 0

endfor

1. for k=1 to maxiter by 1 do

set big\_error = 0

for i=1 to n by 1 do

set sum = 0

for j=1 to n by 1 do

if(i!=j) then

set sum = sum +

aij\*old\_xi

endif

endfor

set new\_xi = (aj(n+1)-sum)/aij

set error = |((new\_xj-old\_xj)/

new\_xj)|

if(error>big\_error) then

set big\_error = error

endif

endfor

if(big\_error<=epsilon) then

write : “Solution conver-

ges in”,k,

“iterations”

for i=1 to n by 1 do

write : xi

endfor

exit

endif

for i=1 to n by 1 do

set old\_xi=new\_xi

endfor

endfor

1. write : “Solution does not converges in”,maxiter,”iterations”
2. Stop