Jacobi Method:

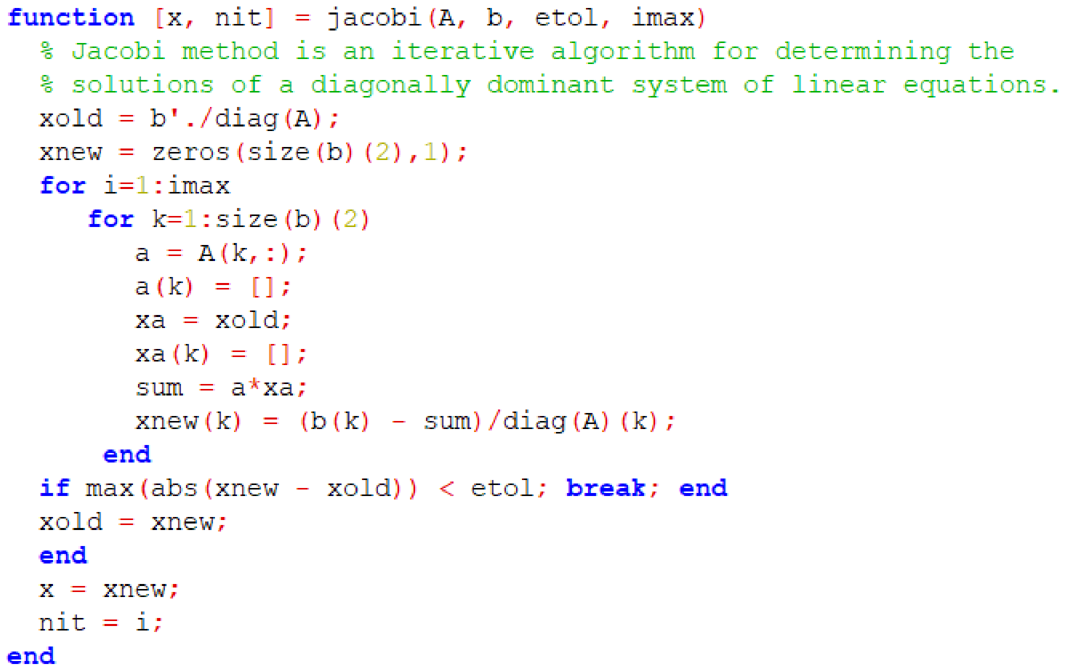
It is an iterative algorithm for determining the solutions of a diagonally dominant system of linear equations. Each diagonal element is solved for, and an approximate value is plugged in. The process is then iterated until it converges.

Pseudocode:

Input: Matrix (A), RHS Column Vector (b), Tolerance (etol), Max. no. of iterations (imax)

Output: Solution Vector (x)

```
Start
Construct initial guess vector xold from the element-wise division of b and the main diagonal of A
While the max number of iteration has not been reached or the required tolerance has not been met
    For k = 1 step until k = size of the vector b
        Let a be  the kth row of matrix A excluding the kth element
        Let xa be the guess vector excluding the kth element
        Let sum be the dot product of a and xa
        Construct a new guess vector xnew where xnew(k) = (b(k) - sum)/A(k, k)
        end for
End
```



```
function [x, nit] = jacobi(A, b, etol, imax)
  % Jacobi method is an iterative algorithm for determining the
  % solutions of a diagonally dominant system of linear equations.
  xold = b'./diag(A);
  xnew = zeros(size(b)(2),1);
  for i=1:imax
     for k=1:size(b)(2)
        a = A(k,:);
        a(k) = [];
        xa = xold;
        xa(k) = [];
        sum = a*xa;
        xnew(k) = (b(k) - sum)/diag(A)(k);
      end
  if max(abs(xnew - xold)) < etol; break; end
  xold = xnew;
  end
  x = xnew;
  nit = i;
end
```