Contents


```
function [x,nit] = SOR(x0,A,b,tol,verbose)
```

```
% HW 2 Problem 2(a)
```

Check the inputs

```
narginchk(3,5);
if nargin<4
    tol=1e-6;
end %if
if nargin<5
    verbose=false;
end %if
```

```
Error using SOR (line 6)
Not enough input arguments.
```

Setup iterations

```
maxit=100;    %max number of iterations
n=size(A,1);  %system size
residual=10*ones(n,1);
difftot=1e3+tol;   %max sure we enter iterations
x=x0;
omega = 1.1; % Relaxation parameter
```

Perform iterations

```
it=1;
while(difftot>tol && it<=maxit)
    difftotprev=difftot;
    resprev=residual;
    xprev=x;
    for i=1:n
        residual(i)=b(i);
        for j=1:n
            residual(i)=residual(i)-A(i,j)*xprev(j);
        end %for
        x(i) = xprev(i) + omega*residual(i)/A(i,i);
    end %for
    difftot=sum(abs(residual-resprev));

    if (verbose)
        fprintf('x= ');
        for i=1:n
```

```
            fprintf('%f   ',x(i));
        end %for
        fprintf('\n');
        fprintf('it=%d; difftot = %e\n',it,difftot);
    end %if

    if (difftot>difftotprev & it>2)
        error('Solution appears to be diverging, check diagonal dominance...')
    end %if
    it=it+1;
end %while

nit=it-1;
if (nit==maxit)
    warning('Solution may not have converged fully...')
end %if
```

```
end %function
```

Published with MATLAB® R2020b