
```
function x = jacobi_method(A,b)

sz= length(b);
Xo= zeros(sz,1);
Xl= zeros(sz,1);

iter=1e4;
op=0;

% Pivoting
for i=1:sz
    diagonal_max=A(i,:);
    max_row=i;
    for k= i+1 : sz
        if(abs(A(k,i)) > abs(diagonal_max(i))) % Condition
            diagonal_max=A(k,:);
            max_row=k;
        end
    end

    A(max_row,:)=A(i,:); % Updating pivot row to max in column
    A(i,:)=diagonal_max;
    temp=b(i);
    b(i)=b(max_row); % Updating pivot row for b vector
    b(max_row)=temp;
end

% Jacobi Method
while(iter)
    for i=1:sz
        temp_sum=0;
        for j=1:sz
            if (j==i)
                continue;
            end
            temp_sum = temp_sum + A(i,j).*Xo(j); % taking sum of old
values
        end
        op= op + 2.*(sz-1);
        Xl(i)= (b(i) - temp_sum)./A(i,i); % Jacobi formula
        op= op + 1;
    end
    max_err=0;
    for k=1:sz
        temp_err= abs((Xl(k)-Xo(k))./Xl(k));
        if(temp_err > max_err) % Checking max error
            max_err= temp_err;
        end
    end
    if(max_err < 1e-10) % Convergence condition
        x=Xl;
        fprintf('No. of operations: %i\n',op);
    end
end
end
```

```
        return;
    end
    Xo= Xl; % Replacing old values with new values
    iter=iter-1;
end
x= NaN;
disp('Did not converge');
fprintf('No. of operations: %i\n',op);
end
```

Published with MATLAB® R2021a