KME272 - Assesment 1.4

Baley Eccles - 652137

[2024-08-24 Sat 12:04]

Contents

1 KME272 - Assesment 1.4

1.1 Q3

1. Make an initial guess (x_0, y_0, z_0, t_0)
2. Construct the matrix $J^{(m)}$
3. Solve $J^{(m)}c^{(m)} = -f^{(m)}$ for $c^{(m)}$
4. Update the guess $x^{(m+1)} = x^{(m)} + c^{(m)}$
5. Check if it has converged $||f^{(m)}||_2 < 10^{-6}$
6. If not converged repeat steps 2 to 5

1.2 Q4

```
c = 299792.458;
pos = [ -15093, -519, -13414;
        -5681, 9216, -17053;
        -6228, 16581, -9711;
        -16728, 9532, -6110];
t = [0.069121, 0.071234, 0.070942, 0.070537];
                                % Initial Guess
x = [0; 0; 0; 0];
tol = 10^-6;
i = 0;
maxit = 1000;
while true

                                % Calculate f
  f = zeros(4, 1);
  for k = 1:length(f)
    f(k) = (x(1)-pos(k,1))^2 + ...
           (x(2)-pos(k,2))^2 + ...
           (x(3)-pos(k,3))^2 - ...
           c^2*(x(4)-t(k))^2;
  end
```

```
                                % Check for convergence
  if norm(f, 2) < tol
    fprintf('Converged after %i iterations.\n', i);
    break;
  end
                                % Calculate J(m)
  J = zeros(4, 4);
  for k = 1:length(J)
    J(k, :) = [2*(x(1)-pos(k,1)), ...
               2*(x(2)-pos(k,2)), ...
               2*(x(3)-pos(k,3)), ...
               -2*c^2*(x(4)-t(k))];
  end
                                % Solve for c(m)
  c_m = mldivide(J,-f);
                                % Update guess
  x = x + c_m;
  i = i + 1;
  if i > maxit
    fprintf("Max iterations reached (%i)\n", i)
    break;
  end
end
                                % Print results
fprintf("Calculated position and time correction (x, y, z, t): (%.2f,%.2f,%.2f,%.2f)\n",
↪  x(1),x(2),x(3),x(4))
```

```
Converged after 5 iterations.
Calculated position and time correction (x, y, z, t): (3438.33,-3491.41,4071.92,-0.02)
```

The code was tried with different inital guesses, the number of itterations required for convergence was typically more than with the inital guess being 0. $x^{(0)} = 0$ is a good inital guess becaues if the location of the reciver is close to $(x, y, z) = (0, 0, 0)$, then the Newton Raphson method will converge quickly.

1.3 Q5

```
x = [3438.332915,-3491.409159,4071.923288];
d = norm(x);
h = d - 6371;
fprintf("Recivers distance from the surface of the earth (km): %.4f\n", h)
```

```
Recivers distance from the surface of the earth (km): 0.2346
```