set 4, prob 3

```
thist = [0; 0.1000; 0.2000; 0.3000; 0.4000; 0.5000;
0.6000; 0.7000; 0.8000; 0.9000; 1.0000];

zhist = [7.7969; 1.4177; -3.0970; -7.6810; -9.8749; -6.1828;
-0.8212; 4.5074; 8.2259; 9.5369; 6.2827];

I = eye(11);
R = I + 0.5 * circshift(I, 1) + 0.5 * circshift(I, -1) ;
R(1, end) = 0; R(end, 1) = 0;

Ra = chol(R);
```

First guess

```
xg0 = [ 2 2 2 ];
xg0_OG = xg0;

% Normalized NL at guess
h = inv(Ra') * h_NL(xg0, thist);

% Normalized jacobian at guess
H = inv(Ra') * Hhist(xg0, thist);

% Normalized measurement
z = inv(Ra') * zhist;

% Gauss-Newton dx
dx = inv((H' * H)) * H' * (z - h);

% Cost function
Jg = norm(z - h);

% first a step
a = 1;
```

First guess + dx

```
xg = xg0 + a * dx';

% Normalized NL at guess
h = inv(Ra') * h_NL(xg, thist);

% Normalized jacobian at guess
H = inv(Ra') * Hhist(xg, thist);

% Normalized measurement
z = inv(Ra') * zhist;

% Cost function
```

```
Jgnew = norm(z - h);

% Gauss-Newton dx
dx = inv((H' * H)) * H' * (z - h);
```

The while loop: Jgnew > Jg

```
while norm(dx) > 0.00001
```

```
    while Jgnew >= Jg

        % Next a
        a = a/2;

        % First guess + dx
        xg = xg0 + a * dx';

        % Normalized NL at guess
        h = inv(Ra') * h_NL(xg, thist);

        % Normalized jacobian at guess
        H = inv(Ra') * Hhist(xg, thist);

        % Normalized measurement
        z = inv(Ra') * zhist;

        % Cost function
        Jgnew = norm(z - h);

    end
```

While loop: "New" first guess - saved from last iteration

```
    xg0 = xg;

    Jg = Jgnew;

    % Gauss-Newton dx (H, z, and h saved from last iteration)
    dx = inv((H' * H)) * H' * (z - h);

    % first a step
    a = 1;
```

While loop: "new" first guess + dx

```
    xg = xg0 + a * dx';

    % Normalized NL at guess
    h = inv(Ra') * h_NL(xg, thist);

    % Normalized jacobian at guess
    H = inv(Ra') * Hhist(xg, thist);
    % Normalized measurement
    z = inv(Ra') * zhist;

    % Cost function
    Jgnew = norm(z - h);

end
```

output

```
% original initial guess
xg0_OG

% solution to initial guess
xg0

% covariance
Pxx = inv(H' * H)
```

```
xg0_OG =

     2     2     2


xg0 =

       9.5107          6.5      0.67809


Pxx =

      0.30725   -0.0039402    0.0065448
   -0.0039402     0.028114    -0.014111
    0.0065448    -0.014111     0.010471
```

subfunctions

```
function h = h_NL(x, t)
% Nonlinear measurement h

    hrow = @(x, t) x(1) * cos( x(2) * t + x(3) );

    h = [ hrow(x, t(1)) ];

    for j = 2:11
        h = [ h ; hrow(x, t(j)) ];
    end

end
```

```
function H = Hhist(x, t)
% Jacobian of h

    Hrow = @(x, t) [ cos( x(2)*t +x(3) ) , ...
                    -x(1)*sin( x(2)*t + x(3) )*t , ...
                    -x(1) * sin( x(2)*t + x(3) ) ];

    H = [ Hrow(x, t(1)) ];

    for j = 2:11
        H = [ H ; Hrow(x, t(j)) ];
    end

end
```

Published with MATLAB® R2020b