
```
function [a, b, c, d] = natural_cubic_spline(n, x, a)
% Form a natural cubic spline
% Input: n - the number of data points used to form spline
%         x - array of size n representing x-values for the data points
%         a - array of size n representing a = f(x) for all x
%
% Output: coefficients for the spline

% initialize h
h = zeros(1, n-1);

% calculate h
for i = 1:n-1
    h(i) = x(i+1) - x(i);
end

% initialize alpha
al = zeros(1, n-1);

% calculate alpha
for i= 2:n-1
    al(i) = (3.0/h(i)) * (a(i+1) - a(i)) - (3.0/h(i-1)) * (a(i) -
    a(i-1));
end

% initialize l, mu, and z
l = zeros(1, n);
m = zeros(1, n);
z = zeros(1, n);

% set l(1) to 1
l(1) = 1;

% calculate l, mu, z
for i=2:n-1
    l(i) = 2*(x(i+1) - x(i-1)) - h(i-1) * m(i-1);
    m(i) = h(i) / l(i);
    z(i) = (al(i) - h(i-1) * z(i-1)) / l(i);
end

l(n) = 1;

% initialize b, c, d
b = zeros(1, n-1);
c = zeros(1, n);
d = zeros(1, n-1);

% calculate b, c, d
j = n - 1;
while j >= 1
```

```
    c(j) = z(j) - m(j)*c(j+1);
    b(j) = (a(j+1) - a(j)) / h(j) - h(j) * (c(j+1) + 2 * c(j)) / 3.0;
    d(j) = (c(j+1) - c(j)) / (3.0 * h(j));

    j = j - 1;
end

end

Not enough input arguments.

Error in natural_cubic_spline (line 11)
h = zeros(1, n-1);
```

Published with MATLAB® R2018b