```
function n = how_many_nodes(fun, a, b, max_error)
%  Find how many nddes needed to approximate the integral of a
%  function within a max error
%
%  Input:
%   fun       - function to approximate integral of
%   a, b      - end points of integral
%   max_error - maximum error bound
%
%  Output:
%    n - number of nodes needed

% min number of nodes is 4.  set to 2 so the loop will set it to 4
n = 2;

% (b - a) / 180
k = (b - a) / 180.0;

% set current error to max error at start of loop
error = max_error;

% check to see if we're under max error
while error >= max_error

    % update n
   n = n + 2;

    % calculate given the new value of n
   h = (b - a) / n;

   % create a linespace in whic values are h numbers apart
   x = a:h:b;

   % dx
   d = diff(x);
   diff_x = d(1);

   % find the 4 derivative of the function
   f = fun(x);
   df = diff(f) ./ diff_x;
   d2f = diff(df) ./ diff_x;
   d3f = diff(d2f) ./ diff_x;
   d4f = diff(d3f) ./ diff_x;

   % get the error bound
   error = k * h^4 * max(abs(d4f));

end

end
```

Not enough input arguments.

```
Error in how_many_nodes (line 17)
k = (b - a) / 180.0;
```

Published with MATLAB® R2018b