Incremental Search Method in MATLAB

Mike Renfro

September 14, 2004
Incremental Search Method in MATLAB

An Incremental Search Problem
  Problem Analysis
  Problem Restatement
  Program Strategy

Incremental Search Program Solution
  A Function That Calculates Stiffness of Fastened Plates
  A Program to Find the Bolt Diameter
  A Better Program to Find the Bolt Diameter

“Homework”
Part I

Review of Previous Lecture
Review of Previous Lecture

- Control structures
  - If/then/else
  - For loops
  - While loops
  - Vectorized loops

- Working with data and graphics
  - Loading and saving data with text files
  - Loading and saving data with MATLAB binary files
  - X-Y plots
Part II

Incremental Search Method in MATLAB
Recall the incremental search method from basic numerical methods.

- Starting at an initial value of $x_1$, the value of $x$ is incremented until a change in the sign of a given function $f(x)$ is observed.
- Since $f(x)$ changes sign between $x_i$ and $x_{i+1}$, it is assumed that it has a root on the interval $(x_i, x_{i+1})$. 
If two plates are fastened by a bolted joint, the stiffness of the fastened members or plates $k_m$ is given by:

$$ k_m = \frac{0.577\pi Ed}{2 \ln \left( \frac{5 \frac{0.577l+0.5d}{0.577l+2.5d}}{5 \frac{0.577l+0.5d}{0.577l+2.5d}} \right) } $$

where $E$ is Young’s modulus of the fastened members, $d$ is the diameter of the bolt, and $l$ is the thickness of the fastened members. Find the value of $d$ that corresponds to a value of $k_m = 7.5 \times 10^6$ psi when $E = 12 \times 10^6$ psi and $l = 1.5\text{ in.}$
Problem Analysis

$$k_m = \frac{0.577\pi Ed}{2 \ln \left( \frac{5 \frac{0.577l+0.5d}{0.577l+2.5d}} \right)}$$

What form does the equation have? $f(x) = 0$, $f(x) = g(x)$, or $f(x) = C$? $f(x) = C$ — we’re looking for a value where the stiffness equation yields a value of $7.5 \times 10^6$ psi. So, to do root-finding, we’ll reformat the equation as $f(x) - C = 0$.

What variables are held constant for this problem? $E$, $l$ — they’re both given specific values in the problem setup.

What variables are allowed to vary for this problem? $d$
Problem Restatement

Find a value of \( d \) such that

\[
\frac{0.577\pi Ed}{2 \ln \left( 5 \frac{0.577l+0.5d}{0.577l+2.5d} \right)} - k_m = 0
\]

when \( E = 12 \times 10^6 \) psi, \( l = 1.5 \) in, and \( k_m = 7.5 \times 10^6 \) psi.
Program Strategy

- Write a function that computes the stiffness of the fastened plates $k_m$ when given values for $E$, $l$, and $d$.
- Write a program that does the following:
  - Sets $E$ and $l$ to their given constant values.
  - Sets $k_m$ to its target value.
  - Sets an initial guess for $d$, and a stepsize $\Delta d$.
  - Calls the function to calculate the stiffness of the fastened plates using both $d$ and $d + \Delta d$ for the bolt diameter.
  - Compares the sign of the returned function values.
  - If the signs match (both positive or both negative), increase $d$ to $d + \Delta d$ and repeat.
  - If the signs don’t match (one function returns positive, and the other returns negative), the required value of $d$ is between the two bolt diameters used. Print that range of $d$ values and exit.
Write a function that computes the stiffness of the fastened plates $k_m$ when given values for $E$, $l$, and $d$.

Function header for computebolerplatestiffness.m:

```matlab
function k=computebolerplatestiffness(E,d,l)
```
A Function That Calculates Stiffness of Fastened Plates

Function header and help for the function:

```matlab
function k = computeboltedplatestiffness(E, d, l)
% computeboltedplatestiffness: Calculate stiffness
% of plates fastened by a bolted joint. (Rao,
% Problem 2.6)
%
% Usage: k = computeboltedplatestiffness(E, d, l)
% where: k = stiffness
%       E = modulus of elasticity
%       d = bolt diameter
%       l = thickness of fastened plates
```
A Function That Calculates Stiffness of Fastened Plates

Full function:

```matlab
function k = computeboltedplatestiffness(E, d, l)
% computeboltedplatestiffness: Calculate stiffness
% of plates fastened by a bolted joint. (Rao, % Problem 2.6)
%
% Usage: k = computeboltedplatestiffness(E, d, l)
%
% where: k = stiffness
% E = modulus of elasticity
% d = bolt diameter
% l = thickness of fastened plates
k = (0.577*pi*E*d) / ...
    (2*log(5*(0.577*l + 0.5*d)/(0.577*l + 2.5*d)));
```
Program documentation for prob_2_6.m:

% Rao Problem 2.6 -- If two plates are fastened by
% a bolted joint, the stiffness of the fastened
% members or plates $k_m$ is given by:
%
% $$k_m = \frac{(0.577 \cdot \pi \cdot E \cdot d)}{(2 \cdot \log(5 \cdot (0.577 \cdot l + 0.5 \cdot d)/(0.577 \cdot l + 2.5 \cdot d)))}$$
%
% where $E$ is Young's modulus of the fastened
% members, $d$ is the diameter of the bolt, and $l$
% is the thickness of the fastened members. Find
% the value of $d$ that corresponds to a value of
% $k_m = 7.5e6$ psi when $E = 12e6$ psi and
% $l = 1.5$ in.
A Program to Find the Bolt Diameter

Set $E$, $l$, $k_m$, $d$, and $\Delta d$ to their constant or initial values.

clear all;
stepSize = 0.1;

desiredStiffness = 7.5e6;
youngsMod=12e6;
boltDiam = 0.1;
plateThick = 1.5;
A Program to Find the Bolt Diameter

Call the function to calculate the stiffness of the fastened plates using both $d$ and $d + \Delta d$ for the bolt diameter, then subtract the target $k_m$ from each result.

```matlab
computeboltedplatestiffness(youngsMod,...
    boltDiam,plateThick) - desiredStiffness
```

```matlab
computeboltedplatestiffness(youngsMod,...
    boltDiam+stepSize,plateThick) - desiredStiffness
```
A Program to Find the Bolt Diameter

Compare the sign of the final results.

\[
\text{sign}(\text{computeboltedplatestiffness}(\text{youngsMod}, \ldots, \\
\text{boltDiam}, \text{plateThick}) - \text{desiredStiffness})
\]

\[
\text{sign}(\text{computeboltedplatestiffness}(\text{youngsMod}, \ldots, \\
\text{boltDiam} + \text{stepSize}, \text{plateThick}) - \text{desiredStiffness})
\]
If the signs match (both positive or both negative), increase $d$ to $d + \Delta d$ and repeat.

```matlab
while (sign(computeboltedplatestiffness(youngsMod,...
        boltDiam ,plateThick) - desiredStiffness) ... == ...
        sign(computeboltedplatestiffness(youngsMod,...
        boltDiam + stepSize ,plateThick) - ...
        desiredStiffness))
    boltDiam = boltDiam + stepSize;
end
```
A Program to Find the Bolt Diameter

When the signs don’t match, print out the two bolt diameter values with the opposite signs.

```matlab
fprintf('Found a root between %f and %f\n',... boltDiam,boltDiam+stepSize);
```
A Program to Find the Bolt Diameter

clear all;
stepSize = 0.1;
desiredStiffness = 7.5e6;
youngsMod=12e6;
boltDiam = 0.1;
plateThick = 1.5;
while (sign(computeboltedplatestiffness(youngsMod,...
boltDiam,plateThick)-desiredStiffness) == ...
sign(computeboltedplatestiffness(youngsMod,...
boltDiam+stepSize,plateThick)-... desiredStiffness))
    boltDiam = boltDiam + stepSize;
end
fprintf('Found a root between %f and %f\n',...
boltDiam,boltDiam+stepSize);
Drawbacks and Warnings for First Program

What if we make a bad choice for the initial value of $d$?

This program will run forever if our initial guess for $d$ is higher than the root value.
Set a maximum number of steps the program will run. If the program runs for too many steps, it can automatically stop itself so we can pick a more appropriate initial guess or step size for the next run.
Changes to First Program

Add a variables to set a limit on the number of steps, and to keep track of the current step number:

```matlab
stepMax = 10000;

nSteps=0;
```
Changes to First Program

Use the new variables as part of the decision-making process:

```matlab
while (sign(computeboltedplatestiffness(youngsMod,...
boltDiam,plateThick)-desiredStiffness) == ...
    sign(computeboltedplatestiffness(youngsMod,...
boltDiam+stepSize,plateThick)-desiredStiffness) ...
    & nSteps < stepMax)
```
Changes to First Program

Make sure you update the step number each time you go through the loop:

\[
boltDiam = boltDiam + \text{stepSize}; \\
nSteps = nSteps + 1;
\]
Changes to First Program

Change the program’s output to show whether or not we found a solution:

```matlab
if nSteps == stepMax
    fprintf('Gave up after %d steps',nSteps);
else
    fprintf('After %d steps, ',nSteps);
    fprintf('found a root between %f and %f\n',...
            boltDiam,boltDiam+stepSize);
end
```
Final Program (Part 1)

```matlab
clear all;
stepMax = 10000;
stepSize = 0.1;
desiredStiffness = 7.5e6;
youngsMod = 12e6;
boltDiam = 0.1;
plateThick = 1.5;
nSteps = 0;
while (sign(computeboltedplatestiffness(youngsMod,...
boltDiam, plateThick) - desiredStiffness) == ...
    sign(computeboltedplatestiffness(youngsMod,...
boltDiam + stepSize, plateThick) - ...
    desiredStiffness) ...
    & nSteps < stepMax)
    boltDiam = boltDiam + stepSize;
nSteps = nSteps + 1;
end
```
if nSteps == stepMax
    fprintf('Gave up after %d steps',nSteps);
else
    fprintf('After %d steps, ',nSteps);
    fprintf('found a root between %f and %f\n',...
            boltDiam,boltDiam+stepSize);
end
Use your incremental search program to solve the following problem:

Find a root of \( f(x) = x^3 - 3 \) starting at \( x_0 = 1 \) and \( \Delta x = 0.1 \). Then, using the first root estimate for a new \( x_0 \), use a step size of \( \Delta x = 0.01 \) to find the root more precisely. How does each estimate differ from the analytical solution?

Next, make sure your program will automatically terminate if it can’t find a solution in a reasonable number of steps. Test what happens if you try to find a root of \( f(x) = x^3 - 3 \) starting at \( x_0 = 1 \) and \( \Delta x = -0.1 \). Does the program stop correctly?