Solving a Nonlinear Equation:
Newton's Method

Mike Bailey
mjb@cs.oregonstate.edu

Scenario: You have an equation \( y(x) = 0 \), but it is too messy to solve directly. You do have an initial guess of the correct value of \( x \). It is close, but it is wrong.

For example, solve this equation for \( x \):

\[
y(x) = \cos^3 x + \log_{10} x = 0
\]

Starting with an initial guess of \( x = 6 \)

You can take the \( x \) you have, \( x_{\text{have}} \), and plug it into the equation to produce \( y_{\text{have}} \) and thus see how close you are to \( y = 0 \). But now what?

From calculus, we know that:

\[
\frac{dy}{dx} = \Delta y = \Delta x
\]

We will use that to find the next value of \( x \) to try, and then repeat the process:

\[
\Delta x = y_{\text{have}} - y(\Delta x) = 0
\]

which gives us:

\[
\Delta x = -\frac{y_{\text{have}}}{dy/dx}
\]

We will use that to find the next value of \( x \) to try, and then repeat the process:

\[
x_{\text{next}} = x_{\text{have}} + \Delta x = x_{\text{have}} + \frac{-y_{\text{have}}}{dy/dx}
\]

We can keep repeating this process to get closer and closer to the actual value of \( x \).
A Collision Detection Problem Example

Let's say we have a nonlinear surface. How close is the point (3,1) to that surface?

Using our friend, the dot product:

\[(P_y - Q_y, P_x - Q_x) \cdot \text{slope} = 0\]

where the vector slope is:

\[\text{slope} = (dx, dy) = \left(1, \frac{dy}{dx}\right) = (1, \sin(x)) \cdot (1, \cos(x))\]

Substituting for Qx, Qy, and the slope:

\[f(x) = (P_y - x, P_x - \sin(x)) \cdot (1, \cos(x)) = 0\]

and expanding:

\[f(x) = (P_x - x) + \cos(x) \cdot (P_y - \sin(x)) = 0\]

Note that in this case, we are solving \( f(x)=0 \), not \( y(x)=0 \)!
A Collision Detection Problem Example

\[
f(x) = (P_x - x) + \cos x \cdot (P_y - \sin x) = 0
\]

\[
\cos x = \frac{x - x_{\text{next}}}{\text{dist}}
\]

\[
f(x) = \cos x \cdot \text{dist} - x + x_{\text{next}} = 0
\]

\[
\text{dist} = \sqrt{(3 - 2.59234)^2 + (1 - 0.52205)^2}
\]

\[
\text{dist} = 0.62819
\]