In addition to solving nonlinear optimization problems with a single variable, we require an algorithm to do "line searches" as part of a multi-dimensional nonlinear optimization problem:

\[
\minimize_t f(x^k + t d^k)
\]

where \( x^k = \text{the } k^{th} \text{ iterate}, \ x^k \in \mathbb{R}^n \)

\( d^k = \text{(feasible) direction of descent} \)

\( t = \text{step size} \)

\( x^{k+1} = x^k + t^* d^k \text{ for optimal stepsize } t^* \)
- an analytic expression for \( f(x) \) might be unknown... \( f(x) \) might be "evaluated" by performing a laboratory or simulation experiment, for example
- it is assumed that the function \( f \) is \textit{unimodal}, i.e., a local optimum will be globally optimal.
- the result of minimization will be a "sufficiently small" \textit{interval of uncertainty} containing the optimum.
- the derivative of \( f \) need not be computed in many of these methods.

©D.L. Bricker, U. of IA, 1999

Three-Point Equi-Interval Search

Golden-Section Search

Fibonacci Search

Polynomial Interpolation

©D.L. Bricker, U. of IA, 1999
Three-Point Equi-Interval Search

Simple, but inefficient.... not recommended!

Assume that at the \( n \)th iteration we have the interval of uncertainty \([a^n, b^n]\) and its midpoint

\[
c^n = \frac{a^n + b^n}{2},
\]

along with the function values \( f(a^n), f(b^n), \) & \( f(c^n) \)

Find the midpoints of the two subintervals \([a^n, c^n]\) and \([c^n, b^n]\):

\[
d^n = \frac{3a^n + b^n}{4}, \quad e^n = \frac{a^n + 3b^n}{4}
\]

and evaluate \( f(d^n) \) and \( f(e^n) \):
Consider the relative magnitudes of the function at these five points:

There are several cases to consider:

For example, suppose that \( f(c) \) is lower than \( f(d) \) and \( f(e) \).

Assuming that the function is unimodal, the minimum cannot be in the interval \([e, b]\)!
And so, using our assumption of unimodality, this allows us to eliminate portions of the interval \([a, b]\) as possible locations for the optimum.

\(x^*\) cannot be in \([a, d]\) or \([e, b]\):

We therefore choose 
\[a^{n+1} = d^n, \quad c^{n+1} = c^n, \quad b^{n+1} = e^n\]
to begin iteration \(n+1\).

In cases I, II, and III, 50% of the interval is eliminated.

In cases IV and V, 75% of the interval is eliminated!
In the event that the smallest of $f(a)$, $f(b)$, $f(c)$, $f(d)$, & $f(e)$ is not unique, less than 25% of the interval can be eliminated.

This event will generally be very rare, especially given round-off errors, etc.
Golden-Section Search

As in 3-point equi-interval search, at the $n$ iteration we have an interval of uncertainty $[a^n, b^n]$ and an interior point $c^n \in (a^n, b^n)$, but $c^n$ is not the midpoint!

We insert a single additional point $d^n$ so that $c^n$ and $d^n$ are symmetric about the midpoint of the interval $[a^n, b^n]$ and compare the values of $f(a^n)$, $f(b^n)$, $f(c^n)$, & $f(d^n)$.

The new point $d^n$ is selected so as to be symmetric to $c^n$ in the interval.

$$b^n - d^n = c^n - a^n$$

$$d^n = a^n + b^n - c^n$$
Assuming unimodality, we can eliminate a segment from the interval of uncertainty:

The shaded segments (-----) can be eliminated from the interval of uncertainty:
Once the location of $c^0$ has been determined within the original interval of uncertainty $[a^0, b^0]$, the location of subsequent points is determined (by symmetry).

How should $c^0$ be located within $[a^0, b^0]$?

In "Golden Section" search, this is done so that the ratio $\frac{c^n - a^n}{b^n - a^n}$ is constant ($\alpha$) $\forall n$ (assuming points are labeled so that $c^n < d^n$)

This requirement uniquely determines

$$\alpha = \frac{c^n - a^n}{b^n - a^n} = \frac{3 - \sqrt{5}}{2} \forall n$$
$$= 0.381966$$
$$\beta = 1 - \alpha = 0.618034$$

known to early Greek mathematicians as the "Golden Section"

If a rectangle with ratio width:length = $\beta$ is cut to yield a square, the other rectangle also has width:length = $\beta$
As is the case with Golden Section Search, this method begins each iteration with an interval of uncertainty \([a, b]\) and one interior point \(c\), and then inserts another interior point \(d\) which is symmetric to \(c\).

In Fibonacci search, however, the ratio \(\frac{c^n - a^n}{b^n - a^n}\) is not constant, but converges to \(\frac{1}{2}\!\). 

\[f\]

\[a^n\hspace{1cm}c^n\hspace{1cm}b^n\]

Given: \([a^1, b^1]\) = initial interval of uncertainty

\[I_k = b^k - a^k\]

\[I_n = b^n - a^n = \text{desired length of interval of uncertainty}\]

\(\varepsilon = "\text{distinguishability constant}" \geq 0\)

(i.e., \(x \& y\) are indistinguishable if \(|x - y| < \varepsilon\)).

For ease of discussion, assume \(I_1 = b^1 - a^1 = 1\)
At the last iteration

The distance between \( c^n \) and \( d^n \) will be \( \varepsilon \)

\[
I_{n-1} = 2I_n - \varepsilon
\]

The final interval of uncertainty will be one of these two intervals

In general, we have

\[
I_{k-1} = I_k + I_{k+1}
\]
\[
\begin{align*}
I_{k-1} &= I_k + I_{k+1} \\
\Rightarrow I_{n-1} &= 2I_n - \varepsilon = F_2 I_n - F_0 \varepsilon \\
I_{n-2} &= 3I_n - \varepsilon = F_3 I_n - F_0 \varepsilon \\
I_{n-3} &= 5I_n - 2 \varepsilon = F_4 I_n - F_1 \varepsilon \\
I_{n-4} &= 8I_n - 3 \varepsilon = F_5 I_n - F_2 \varepsilon \\
I_{n-5} &= 13I_n - 5 \varepsilon = F_6 I_n - F_3 \varepsilon \\
&\vdots \\
I_{n-k} &= F_{k+1} I_n - F_k \varepsilon \\
&\vdots \\
1 &= I_1 = F_n I_n - F_{n-2} \varepsilon 
\end{align*}
\]

**Fibonacci Numbers**

Leonardo of Pisa, son of Bonacci ("Fibonacci") 1202 AD.

**Rule for Generating the Sequence:**

\[
\begin{align*}
F_0 &= F_1 \equiv 1 \\
F_n &= F_{n-1} + F_{n-2}, n \geq 2 \\
\Rightarrow F_2 &= F_1 + F_0 = 1 + 1 = 2 \\
F_3 &= F_2 + F_1 = 2 + 1 = 3 \\
F_4 &= F_3 + F_2 = 3 + 2 = 5 \\
F_5 &= F_4 + F_3 = 5 + 3 = 8 \\
&\vdots \\
\end{align*}
\]
\[ I_1 = F_n I_n - F_{n-2} \varepsilon \]

Solving for the "reduction ratio" \( \frac{I_1}{I_n} \)

\[ \frac{I_1}{I_n} = \frac{F_n}{1 + F_{n-2} \varepsilon} \]

Given a desired reduction ratio, we can find \( n \), the required number of iterations.

Example: Suppose we desire \( I_n \leq 0.01 I_1 \)

i.e., \( \frac{I_1}{I_n} \geq 100 \)

and suppose \( \varepsilon \approx 0 \)

Then

\[ \frac{I_1}{I_n} = \frac{F_n}{1 + F_{n-2} \varepsilon} \approx F_n \]

Choose \( n \) so that \( F_n \geq 100 \)

\[ \Rightarrow n = 11 \]
Once we have determined $n$ (the # of iterations), we can compute

$$I_2 = F_{n-1} \left[ \frac{1 + F_{n-2}}{F_n} \right] - F_{n-3} \varepsilon$$

$$= \frac{F_{n-1}}{F_n} + \left[ \frac{F_{n-1} F_{n-2}}{F_n} \right] \varepsilon$$

This will tell us where to put our initial interior point $c^1$ within $[a^1,b^1]$. 

For example, suppose $n = 11$ and $\varepsilon \approx 0$

$$I_2 \approx \frac{F_{10}}{F_{11}} \approx 0.6180555$$

Throughout the remainder of the iterations, the other interior points are located to retain the symmetry of $c^k$ and $d^k$. 
In quadratic & cubic interpolation methods, we use information about the function at two or more points to determine a polynomial in agreement with the known information about the function $f$.

A minimum point is then computed for the interpolating polynomial to obtain a new point interior to the interval of uncertainty $[a,b]$. 

<table>
<thead>
<tr>
<th>Polynomial</th>
<th>Given information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadratic</td>
<td>$a, f(a), b, f(b), c, (a,b), f(c)$</td>
</tr>
<tr>
<td>Cubic</td>
<td>$a, f(a), f'(a), b, f(b), f'(b)$</td>
</tr>
</tbody>
</table>
Polynomial Interpolation

Lagrange's Interpolating Polynomials

polynomials \( p(x) \) with \( p(a) = f(a) \), etc.

Quadratic Interpolation

Hermite Interpolating Polynomials

polynomials \( p(x) \) with \( p(a) = f(a) \), \( p'(a) = f'(a) \)

Cubic Interpolation

Assume that we are given the \( n+1 \) values

\[ \{ x_0, x_1, x_2, \ldots, x_n \} \]

and function values \( f(x_i) \).

What is the polynomial \( p(x) \) of degree \( n \) which agrees exactly with \( f(x) \) at the values \( x_i, i=0,1,2,\ldots,n \)?
Define
\[ L_j(x) = \prod_{k \neq j} \frac{x - x_k}{x_j - x_k} = \frac{x - x_0}{x_j - x_0} \times \frac{x - x_1}{x_j - x_1} \times \cdots \times \frac{x - x_n}{x_j - x_n} \]

Properties:
- \( L_j(x) \) is a polynomial of degree \( n \)
- \( L_j(x_j) = 1 \)
- \( L_j(x_k) = 0 \) for \( k \neq j \)

That is, for each \( x_j \) we define a polynomial of degree \( n \) which is 1 at \( x_j \) but zero for \( x_k, k \neq j \).

Lagrange's interpolating polynomial is
\[ p(x) = \sum_{j=0}^{n} f(x_j) \cdot L_j(x) \]

This polynomial agrees exactly with the function \( f \) at \( x_0, x_1, x_2, \ldots, x_n \)

i.e., \[ p(x_k) = \sum_{j=0}^{n} f(x_j) \cdot L_j(x_k) = f(x_k) \quad \forall \ k = 0, 1, 2, \ldots, n \]
Assume that we are given the \( n+1 \) triplets of values
\[
x_0, f(x_0), f'(x_0) \\
x_1, f(x_1), f'(x_1) \\
\vdots \\
x_n, f(x_n), f'(x_n)
\]
We want to find a polynomial \( p(x) \) such that
\[
p(x_i) = f(x_i) \quad \forall \ i = 0, 1, \ldots, n
\]
\[
p'(x_i) = f'(x_i)
\]

Notation
\[
f_i \equiv f(x_i)
\]
\[
f'_i \equiv f'(x_i) = \frac{df}{dx}(x_i)
\]
The polynomial \( p(x) \) will be of the form
\[
p(x) = \sum_{j=0}^{n} f_j h_j(x) + \sum_{j=0}^{n} f'_j \overline{h}_j(x)
\]
In order that \( p(x_i) = f(x_i) \)
h\(_j\) and \( \overline{h}_j \) will satisfy
\[
\begin{align*}
h_j(x_j) &= 1 \\
h_j(x_k) &= 1 \quad \forall \ k \neq j \\
\overline{h}_j(x_k) &= 0 \quad \forall \ k \neq j
\end{align*}
\]
Differentiating \( p(x) = \sum_{j=0}^{n} f_j h_j(x) + \sum_{j=0}^{n} f'_j \bar{h}_j(x) \)

yields \( p'(x) = \sum_{j=0}^{n} f_j h'_j(x) + \sum_{j=0}^{n} f'_j h'_j(x) \)

We would therefore like \( h_j \) and \( \bar{h}_j \) to satisfy

\[
\begin{align*}
  h'_j(x_k) &= 0 \quad \forall \ k \neq j \\
  h'_j(x_k) &= \begin{cases} 
    1 & \text{if } k = j \\
    0 & \text{if } k \neq j 
  \end{cases}
\end{align*}
\]

The following functions have the desired properties:

\[
\begin{align*}
  h_j(x) &= \left[ 1 - 2(x - x_j) \ L'_j(x_j) \right] L_j^2(x) \\
  \bar{h}_j(x) &= (x - x_j) \ L_j^2(x)
\end{align*}
\]
Quadratic Interpolation

Given: \( a, b, c \) and \( f(a), f(b), f(c) \)

The interpolating quadratic polynomial is

\[
p(x) = f(a) \frac{x - b}{a - b} \times \frac{x - c}{a - c} + f(b) \frac{x - a}{b - a} \times \frac{x - c}{b - c} + f(c) \frac{x - a}{c - a} \times \frac{x - b}{c - b}
\]

We want to find the minimum of this polynomial, and so we will find \( x \) such that \( \frac{dp(x)}{dx} = 0 \)

\[
\frac{dp(x)}{dx} = \frac{f(a)}{(a-b)(a-c)} (2x-b-c) + \frac{f(b)}{(b-a)(b-c)} (2x-a-c) + \frac{f(c)}{(c-a)(c-b)} (2x-a-b) = 0
\]

\[
\Rightarrow \quad x^* = \frac{1}{2} \frac{f(a)(b^2 - c^2) + f(b)(c^2 - a^2) + f(c)(a^2 - b^2)}{f(a)(b - c) + f(b)(c - a) + f(c)(a - b)}
\]

Having located this 4th point (call it d), evaluate \( f(d) \) and proceed as in Golden Section or Fibonacci search, eliminating a portion of the interval \([a,b]\)
f(x*) is evaluated and a portion of the interval is eliminated, leaving 3 points with which to begin the next iteration.

Cubic Interpolation

Given a, f(a), f'(a), and b, f(b), f'(b), there is a unique cubic polynomial which passes through the points (a, f(a)), (b, f(b)) and is tangent to the graph at these points.
Cubic Interpolation, Using Hermite Polynomials

Given \( a, f(a), f'(a), \) and \( b, f(b), f'(b) \)

The interpolating CUBIC polynomial is

\[
p(x) = f(a)h_0(x) + f(b)h_1(x) + f'(a)\overline{h}_0(x) + f'(b)\overline{h}_1(x)
\]

where

\[
h_0(x) = \left[ 1 - 2 \frac{x-a}{a-b} \right] \left[ \frac{x-b}{a-b} \right]^2
\]

\[
h_1(x) = \left[ 1 - 2 \frac{x-b}{b-a} \right] \left[ \frac{x-a}{b-a} \right]^2
\]

\[
\overline{h}_0(x) = (x-a)\left[ \frac{x-b}{a-b} \right]^2
\]

\[
\overline{h}_1(x) = (x-b)\left[ \frac{x-a}{b-a} \right]^2
\]
Finding the stationary point of \( p(x) \) in \([a,b]\)

**Step 1:** 
\[
z = f'(a) + f'(b) + 3 \left[ \frac{f(a) - f(b)}{b - a} \right]
\]

**Step 2:** 
\[
w = \sqrt{\max \{ 0, z^2 - f'(a)x f'(b) \}}
\]

**Step 3:** 
\[
x^* = b - \frac{(b - a)(f'(b) + w - z)}{f'(b) + 2w - f'(a)}
\]