In a nutshell: Step-by-step optimization

Given a continuous and differentiable real-valued function f of a real variable with an initial approximation of an extremum x_0. This algorithm uses iteration and interpolating polynomials.

Parameters:

$\varepsilon_{\text{step}}$	The maximum error in the value of the minimum cannot exceed this value.
ε_{abs}	The difference in the value of the function after successive steps cannot exceed this value.
h	An initial step size.
N	The maximum number of iterations.

1. Let $k \leftarrow 0$.
2. If $k > N$, we have iterated N times, so stop and return signalling a failure to converge.
3. Let and letting j take the values from 1 to n do the following:
   a. If $f\left(x_k + h\right) < f\left(x_k\right), f\left(x_k - h\right)$, continue calculating $f\left(x_k + nh\right)$ for positive integer values of n until $f\left(x_k + \left(n+1\right)h\right) > f\left(x_k + nh\right)$ and then set $x_{k+1} \leftarrow x_k + nh$,
   b. otherwise, if $f\left(x_k - h\right) < f\left(x_k\right), f\left(x_k + h\right)$, continue calculating $f\left(x_k - nh\right)$ for positive integer values of n until $f\left(x_k - \left(n+1\right)h\right) > f\left(x_k - nh\right)$ and then set $x_{k+1} \leftarrow x_k - nh$,
   c. otherwise, set $x_{k+1} \leftarrow x_k$.
4. If $|h| < \varepsilon_{\text{step}}$ and $|f(x_{k+1}) - \min\{f(x_k - h), f(x_k + h)\}| < \varepsilon_{\text{step}}$, we are done and return x_{k+1}.
5. Increment k and return to Step 2.

Acknowledgement: Jakob Koblinsky noted that I mistakenly copied $x_{k+1} \leftarrow x_k + nh$ in step 3b, which should be subtracting nh. This has been corrected.