Bisection method for finding roots

- Root of function f: Value x such that f(x)=0.
- Many problems can be expressed as finding roots.
  - e.g. square root of w = root of $f(x) = x^2 - w$.
  - If $f(x) = 0$, then $x^2 - w = 0$, i.e. $x = \sqrt{w}$.
- Requirement for bisection method:
  - Need to be able to evaluate f.
  - f must be continuous.
  - We must be given points x_L and x_R such that $f(x_L)$ and $f(x_R)$ are not both positive or both negative.

Bisection method: basic iteration

- Basic iteration will bring x_L, x_R closer, while maintaining invariant:

  x_L, x_R have different signs or are 0.
- Invariant is true initially.
- Invariant + Continuity ⇒ root exists between x_L and x_R (both inclusive).
- We iterate till $x_R - x_L \leq \varepsilon$, our desired error bound
- We declare x_L as the root.
- Error in declared root is at most ε.

Iteration

Let $x_M = (x_L + x_R)/2$

midpoint of interval (x_L, x_R).

If $f(x_M)$ has same sign as $f(x_L)$,

then set $x_L = x_M$

Sign of x_L did not change

Sign of x_L continues to remain different from sign of x_R

else set $x_R = x_M$.

Sign of x_R does not change...

Invariant holds even here

Bisection method for finding square root of 2, i.e. root of $f(x)=x^2 - 2$.

```
double xL=0,     // f(xL) = 0 — 2 is negative
       xR=2,     // f(xR) = 4 — 2 is positive
       xM, epsilon = 0.00001;
while(xR — xL >= epsilon){
  xM = (xL+xR)/2;
  if((xL*xL — 2 > 0 && xM*xM — 2 > 0) ||
     (xL*xL — 2 < 0 && xM*xM — 2 < 0)) xL = xM;
  else xR = xM;
}
cout << xL << endl;
// How would you choose xL, xR for finding sqrt(3)?
// xL = 0, xR = 3 will work
```

Demo

- bisection.cpp

Remarks

- In each iteration, the interval (x_L, x_R) halves in size.
- The size of the interval gives the error in the root.
- Thus the error in the root halves in each iteration.
- Thus if you want the answer correct to k bits, you should use k iterations.
- The number of calculations in each iteration can be reduced.
  - See the book.

Exercise: Modify the program so that it calculates the cube root of any number w. Make sure you correctly initialize x_L, x_R.

What we discussed

- Bisection method is a very simple method for finding the root of a function f.
- Requirements:
  - Should be possible to evaluate f at any x.
  - f should be continuous.
  - Need x_L and x_R such that $f(x_L)$ and $f(x_R)$ dont have the same sign.
  - Method is simple but can be slow.

Next: Newton Raphson method for finding roots