

## Bisection Search

- half interval each iteration
- new guess is halfway in between
- ~~• to iterate, etc~~

### Code:

```
>>> cube = 27
>>> epsilon = 0.01
>>> num_guess = 0
>>> low = 0
>>> high = cube
>>> guess = (high + cube) / 2.0
>>> while abs(guess**3 - cube) >= epsilon:
    if -guess**3 < cube:
        low = guess
    else:
        high = guess
    guess = (high + low) / 2
num_guess += 1
print(num_guess, num_guess)
print(guess, 'is close to the cube root of', cube)
```

### Bisection Search convergence:

- search space
  - ↳ first guess:  $N/2$
  - ↳ 2<sup>nd</sup> guess:  $N/4$
  - ↳  $k^{\text{th}}$  guess:  $N/2^k$
- guess converges on the order of  $\log_2 N$  steps
- bisection search works when value of function varies monotonically in the input
- code as shown only works for positive cubes  $> 1$ , wh