

Fixed-point iteration:

The principle of fixed point iteration is that we convert the problem of finding root for $f(x)=0$ to an iterative method by manipulating the equation so that we can rewrite it as $x=g(x)$. Then we use the iterative procedure

$$x_{i+1}=g(x_i)$$

The condition for convergence of the fixed-point iteration is that the derivative of g is smaller than 1 in absolute magnitude. This shapes our creation of the equation.

Example: When I was an undergraduate, calculators had only the four basic operations and no square roots. So we used a fixed point iteration to solve for the square root of a number. Consider the equation $f(x)=x^2-a=0$ that will give us the square root of a . We can write it as $x=g(x)=a/x$. However $|g'(x)|=a/x^2$ and it will be larger than 1 whenever our guess x will be smaller than the square root.

So instead we modify this as

$$2x=a/x+x \text{ or } x=g(x)=0.5(a/x+x).$$

Now $g'(x)=0.5(-a/x^2+1)$. This will be small whenever x is not very different from \sqrt{a} . In fact starting with $x=1$ will work well for most cases of moderately large or small numbers. For example, for the square root of 10 five iterations would be enough to get 5 digits.

```
>> a=10; x=1
```

```
x =
```

```
1
```

```
>> x=0.5*(a/x+x)
```

```
x =
```

```
5.5000
```

```
>> x=0.5*(a/x+x)
```

```
x =
```

3.6591

>> x=0.5*(a/x+x)

x =

3.1960

>> x=0.5*(a/x+x)

x =

3.1625

>> x=0.5*(a/x+x)

x =

3.1623

>>

The Newton-Raphson method is usually faster to find a root, because it uses the derivative of the function

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

For example, applied to the cosine function we get

$$x_{i+1} = x_i + \frac{\cos(x_i)}{\sin(x_i)} = x_i + \cot(x_i)$$

See the fast convergence from x=1

>> x=1

x =

1

>> x=x+cot(x)

x =

1.6421

>> x=x+cot(x)

x =

1.5707

>> x=x+cot(x)

x =

1.5708

However, if we start too far from the solution there may be problems:

>> x=0.3

x =

0.3000

>> x=x+cot(x)

x =

3.5327

>> x=x+cot(x)

x =

5.9577

>> x=x+cot(x)

x =

2.9950

>> x=x+cot(x)

x =

-3.7774

>> x=x+cot(x)

x =

-5.1323

>> x=x+cot(x)

x =

-4.6858

>>

The good behavior of the fixed-point iteration for the square root function was because the formula we used is the same that we would obtain using Newton-Raphson. Indeed if $f(x)=a-x^2$ then

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i + \frac{a - x_i^2}{2x_i} = 0.5(x_i + a/x_i)$$

So also for this equation, we will have problems or slow convergence if we start far from the solution. For example, if we want to estimate the square root of 1,000,000 as 1, we will get

```
>> a=1000000
```

```
a =
```

```
1000000
```

```
>> x=1
```

```
x =
```

```
1
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
5.0000e+005
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
2.5000e+005
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
1.2500e+005
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
6.2505e+004
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
3.1261e+004
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
1.5646e+004
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
7.8551e+003
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
3.9912e+003
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
2.1209e+003
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
1.2962e+003
```

```
>> x=0.5*(x+a/x)
```

```
x =
```

```
1.0338e+003
```

```
>> x=0.5*(x+a/x)
```

x =

1.0006e+003