Order and Rates of Convergence

Numerical Analysis

Math 465/565

“Speed of convergence”

We now have two algorithms which we can compare - bisection and the fixed-point method. Which is faster?

Hard to answer : Depends on what interval we start with, how close to a root we start with, etc. So for any two particular instances one method might converge in fewer iterations than the other.

How can we qualify more generally which method might be better in some sense?

Review of bisection

We showed that bisection has the property that

$$|x_k - \bar{x}| \le \frac{|b-a|}{2^k}$$

If we define the error as $e_k = |x_k - \bar{x}|$, we can write

$$e_k \le |b-a|\, 2^{-k}$$

Or, bisection "behaves" like the sequence

$$\beta_k = 2^{-k}$$

How fast does an algorithm converge?

Suppose that an algorithm produces iterates that converge as

$$\lim_{k \to \infty} x_k = \bar{x}$$

If there exists a sequence β_k that converges to zero and a positive constant C, such that

$$e_k = |x_k - \bar{x}| \leq C|\beta_k|$$

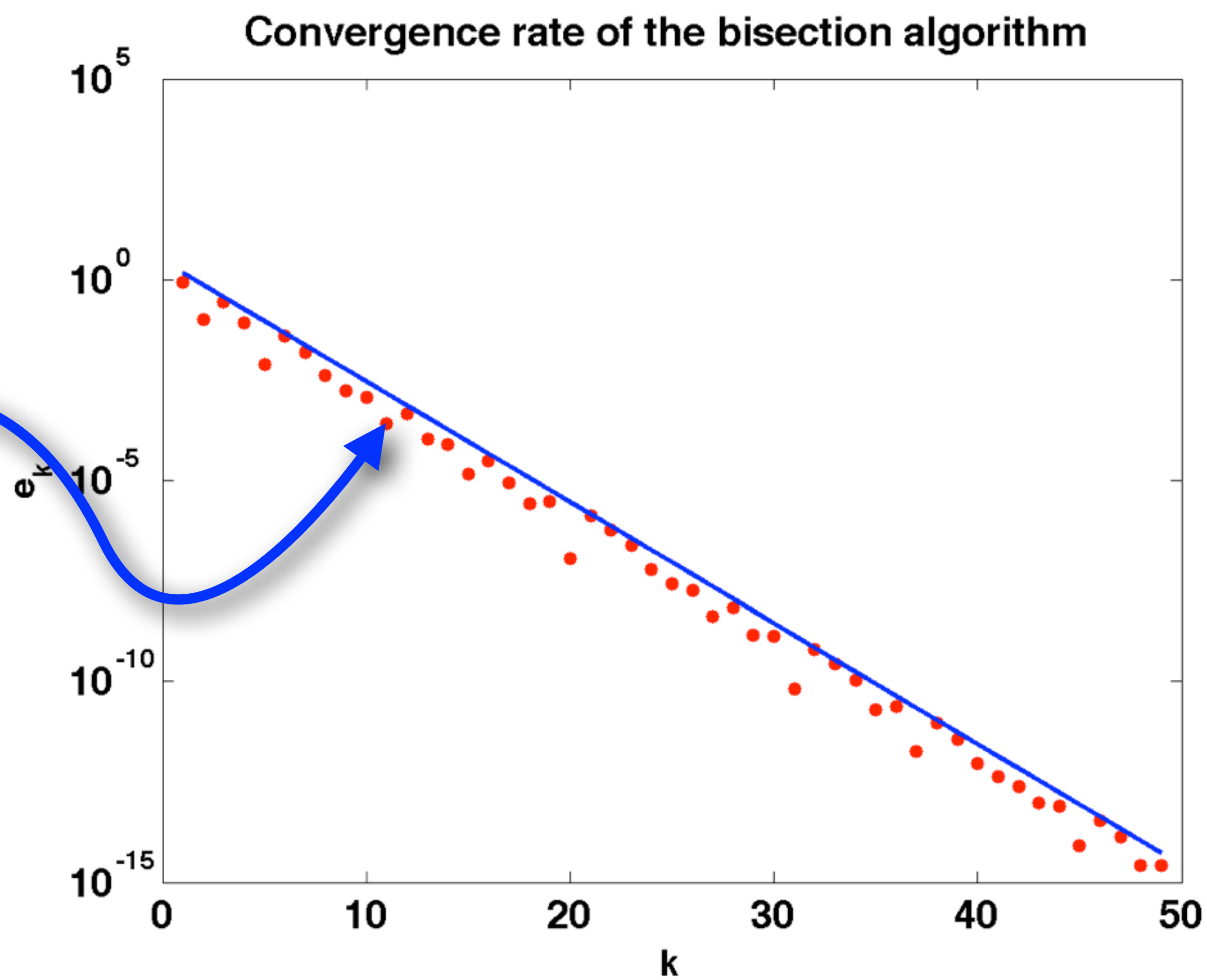
then x_k is said to converge with rate β_k.

Convergence rate - Bisection

$$|x_k - \bar{x}| \leq |b - a| \frac{1}{2^k}$$

So the bisection method has a convergence rate of $\frac{1}{2^k}$ with $|b - a|$ as the asymptotic convergence constant.

rate 2^{-k}



Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **polyfit**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
```

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
>> ab = polyfit(k,log(e),1)
```

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
>> ab = polyfit(k,log(e),1)

ab =
```

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
>> ab = polyfit(k,log(e),1)

ab =

  -0.692013993991203   0.046941635188405
```

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
>> ab = polyfit(k,log(e),1)

ab =

  -0.692013993991203   0.046941635188405

>> log(2)
```

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
>> ab = polyfit(k,log(e),1)

ab =

  -0.692013993991203   0.046941635188405

>> log(2)

ans =
```

Approximate rate of convergence

$$\log(e_k) = -\log(2)k + \log(C)$$

Using the Matlab command **`polyfit`**, we can get a best fit line to the data

$$ak + b = \log(e_k)$$

```
>> k = 1:49;
>> ab = polyfit(k,log(e),1)

ab =

  -0.692013993991203   0.046941635188405

>> log(2)

ans =

   0.693147180559945
```

Fixed point method

Suppose over the interval $[x_0 - \rho, x_0 + \rho]$, a function $g(x)$ satisfies a Lipschitz condition, and furthermore, x_0 is sufficiently close so that a sequence

$$x_{k+1} = g(x_k), \quad k = 0, 1, 2, \ldots$$

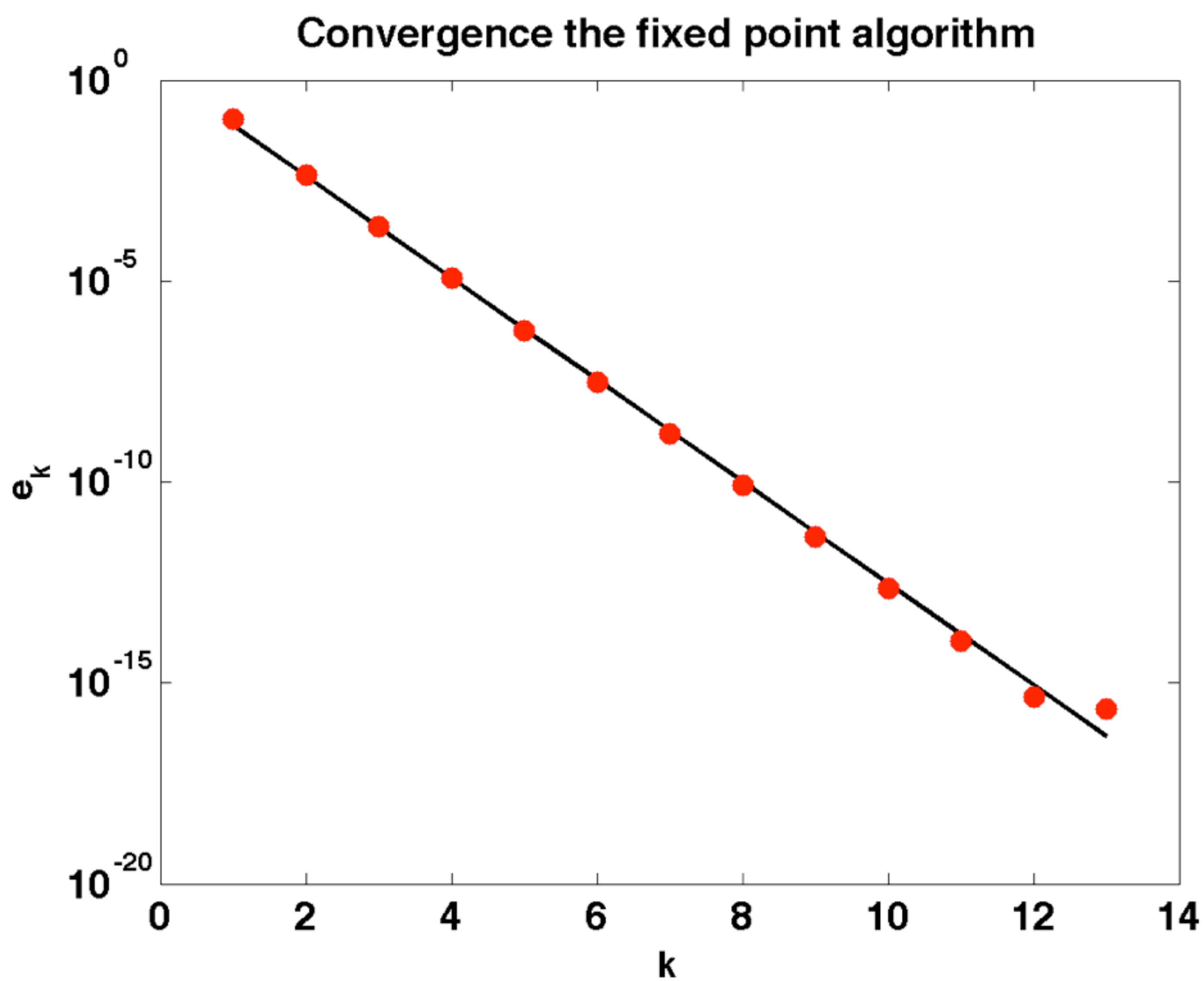
converges to a root $\bar{x}$ in $[x_0 - \rho, x_0 + \rho]$. Then

$$e_k = |x_k - \bar{x}| \leq \lambda^k \rho$$

where $|g'(x)| \leq \lambda < 1$.

Convergence of fixed point iteration

So the convergence rate is λ^k where $|g'(x)| \leq \lambda < 1$.



Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)
```

Convergence rate $\approx \dfrac{1}{20^k}$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)

ab =
```

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012
```

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012

>> exp(ab(1))
```

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012

>> exp(ab(1))

ans =
```

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012

>> exp(ab(1))

ans =

    0.0542
```

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012

>> exp(ab(1))

ans =

    0.0542

>> exp(ab(2))
```

Convergence rate

$$\approx \frac{1}{20^k}$$

Estimated rate for fixed point

$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

Convergence rate

$$\approx \frac{1}{20^k}$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012

>> exp(ab(1))

ans =

    0.0542

>> exp(ab(2))

ans =
```

Estimated rate for fixed point

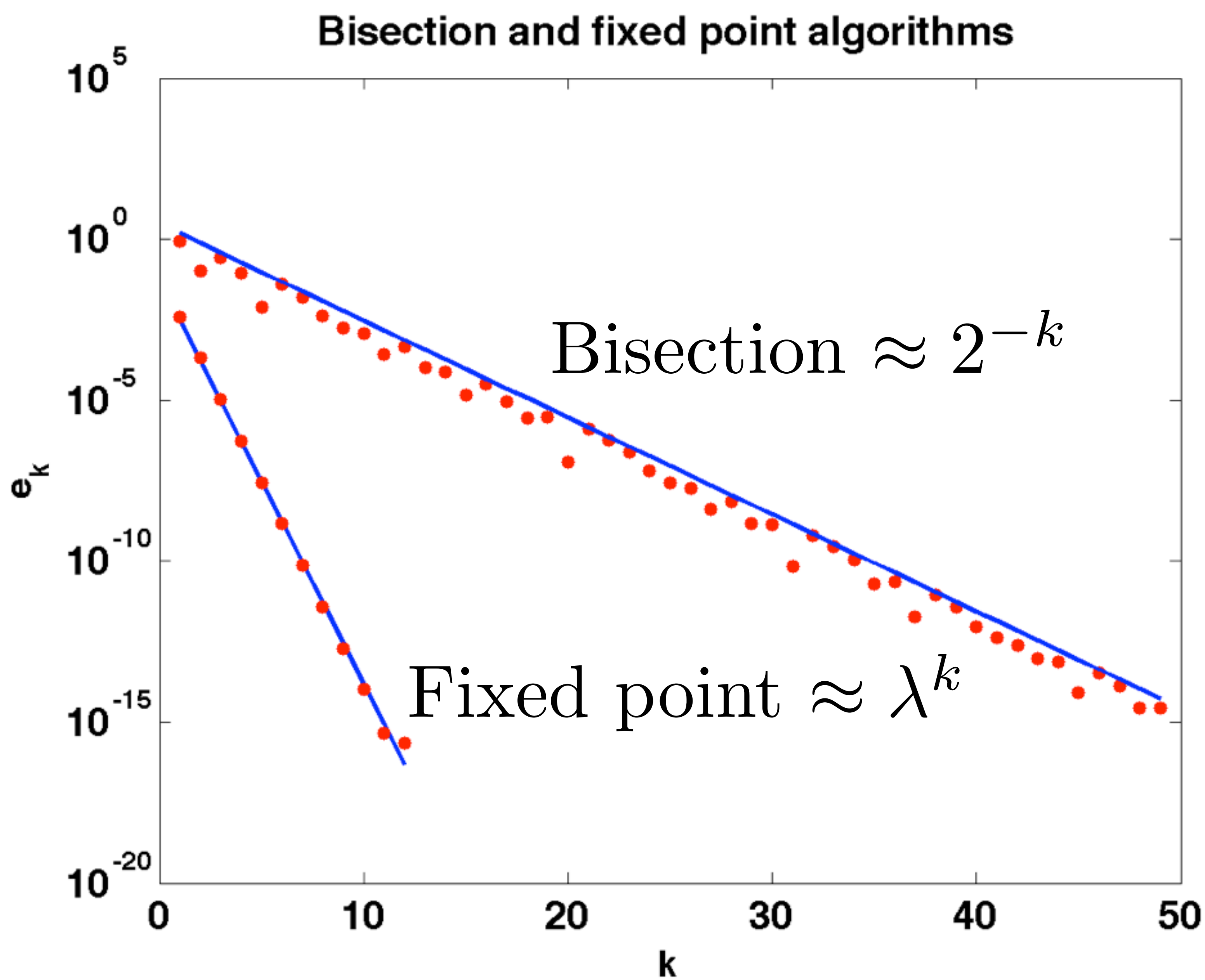
$$\log(e_k) = \log(\lambda)k + \log(\rho)$$

Convergence rate

$$\approx \frac{1}{20^k}$$

```
>> ab = polyfit(kvec,log(e),1)

ab =

   -2.9148    0.3012

>> exp(ab(1))

ans =

    0.0542

>> exp(ab(2))

ans =

    1.3514
```

Comparison of convergence rates



Which is better?

Order of convergence

Suppose we have that

$$\lim_{k\to\infty} \frac{|x_{k+1} - \bar{x}|}{|x_k - \bar{x}|^\alpha} = \lim_{k\to\infty} \frac{e_{k+1}}{e_k^\alpha} = \mu$$

Then the convergence of the sequence x_k to $\bar{x}$ is said to be of order α.

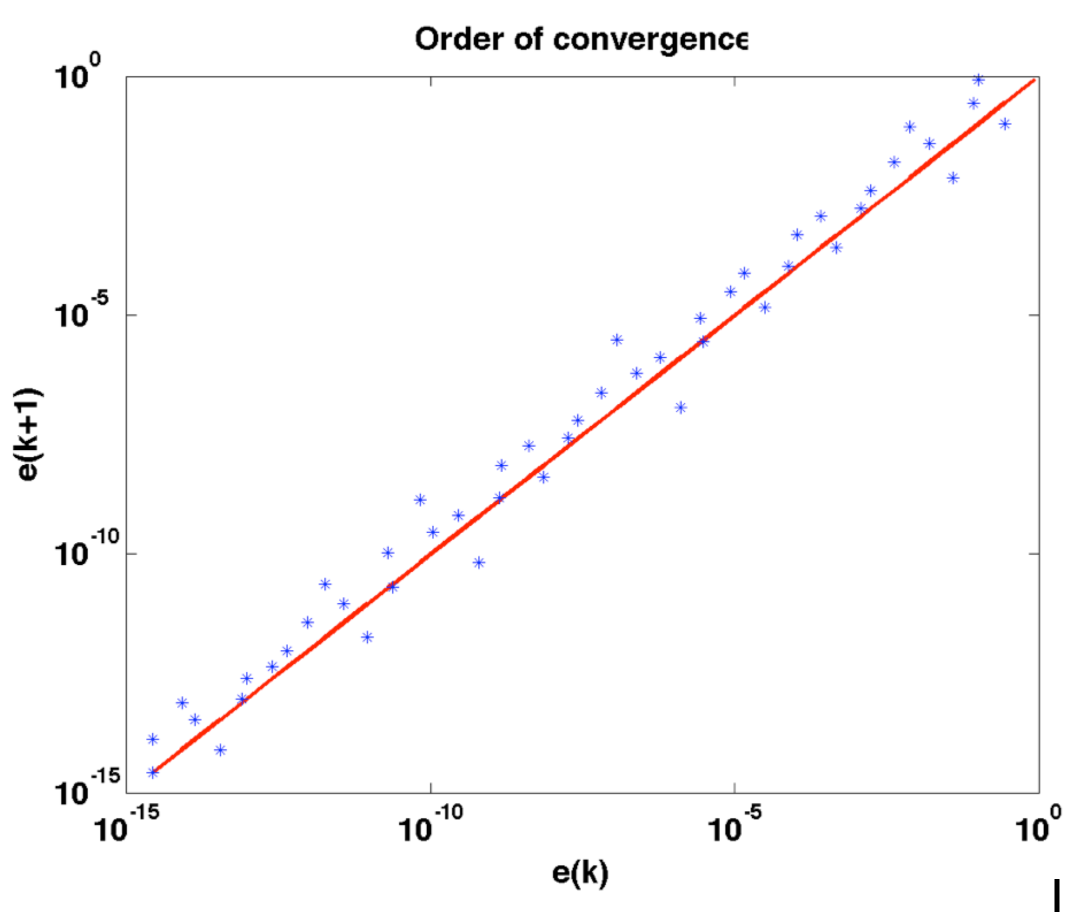
Estimating order of convergence

Again, we can write

$$\log(e_{k+1}) = \alpha \log(e_k) + \log(\mu)$$

and use a best-fit-line approach to finding α, given a sequence of errors e_k.

Both bisection and the fixed point method are *linear* methods



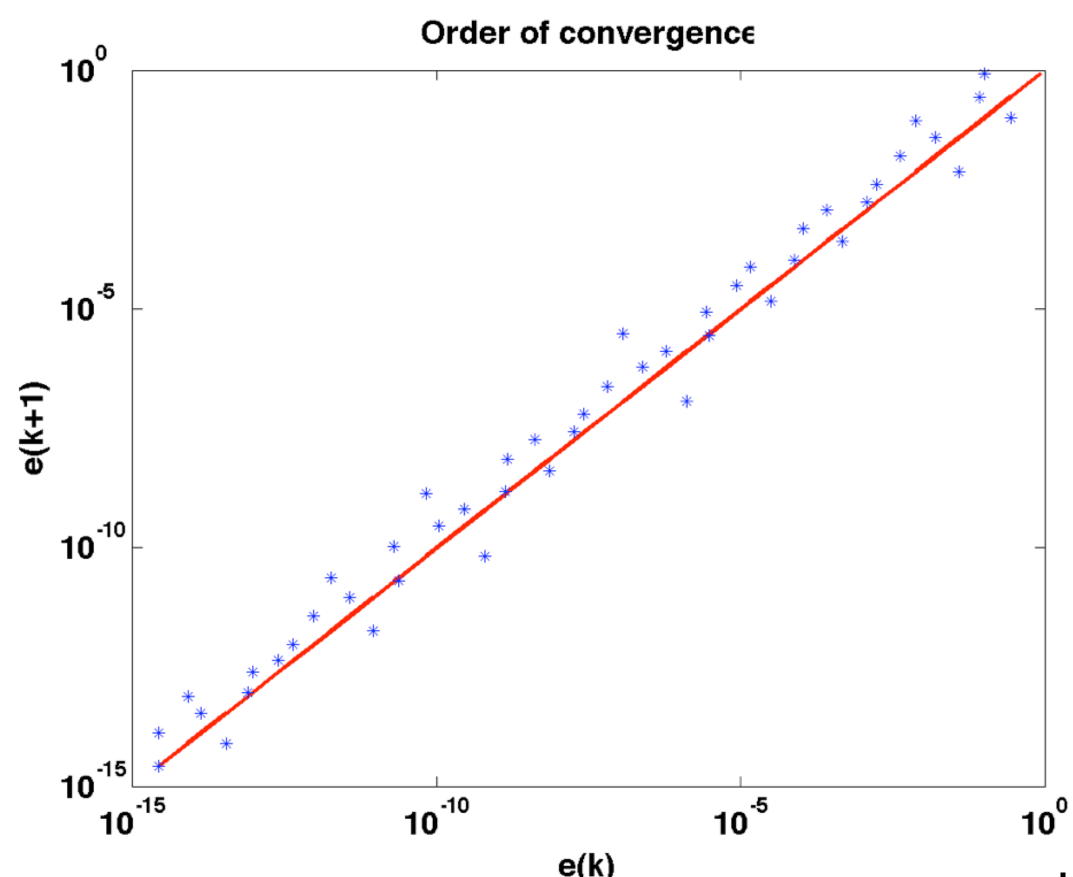
Estimating order of convergence

Again, we can write

$$\log(e_{k+1}) = \alpha \log(e_k) + \log(\mu)$$

and use a best-fit-line approach to finding α, given a sequence of errors e_k.

```
>> ab = polyfit(log(e(1:end-1)),log(e(2:end)),1)
```



Both bisection and the fixed point method are *linear* methods

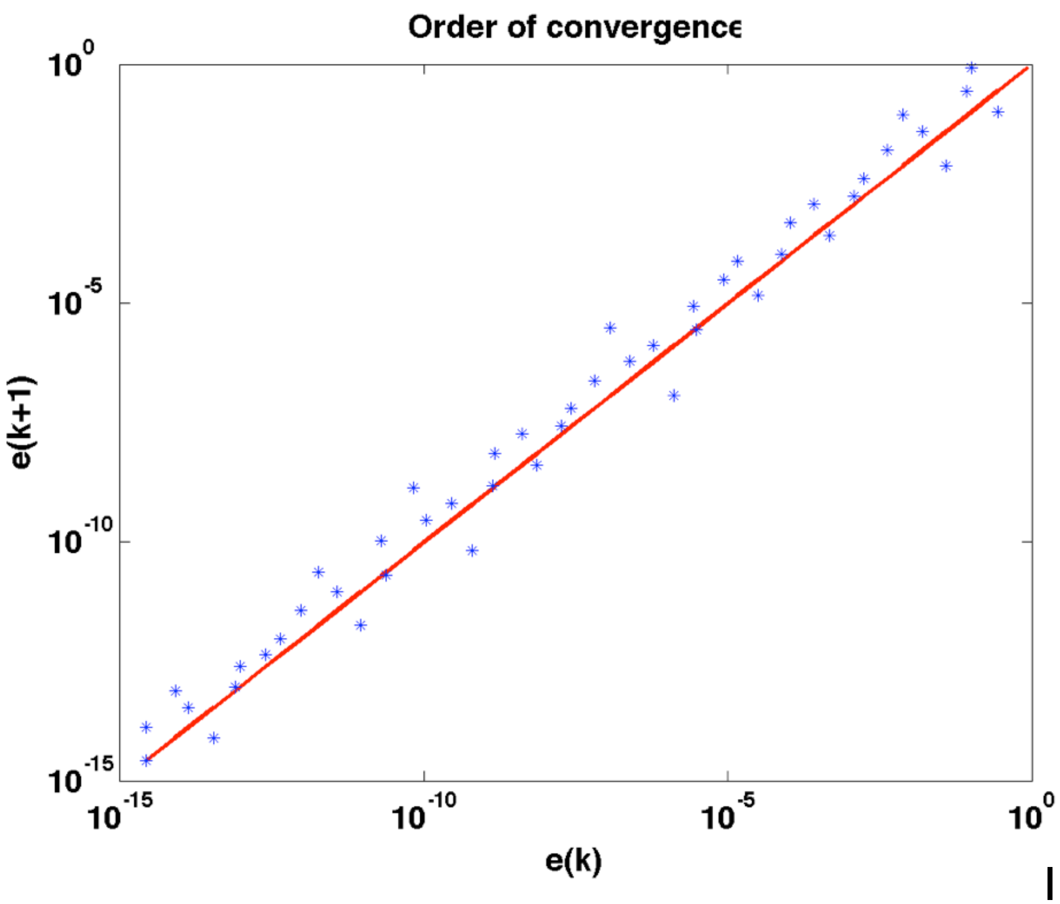
Estimating order of convergence

Again, we can write

$$\log(e_{k+1}) = \alpha \log(e_k) + \log(\mu)$$

and use a best-fit-line approach to finding α, given a sequence of errors e_k.

```
>> ab = polyfit(log(e(1:end-1)),log(e(2:end)),1)

ab =
```



Both bisection and the fixed point method are *linear* methods

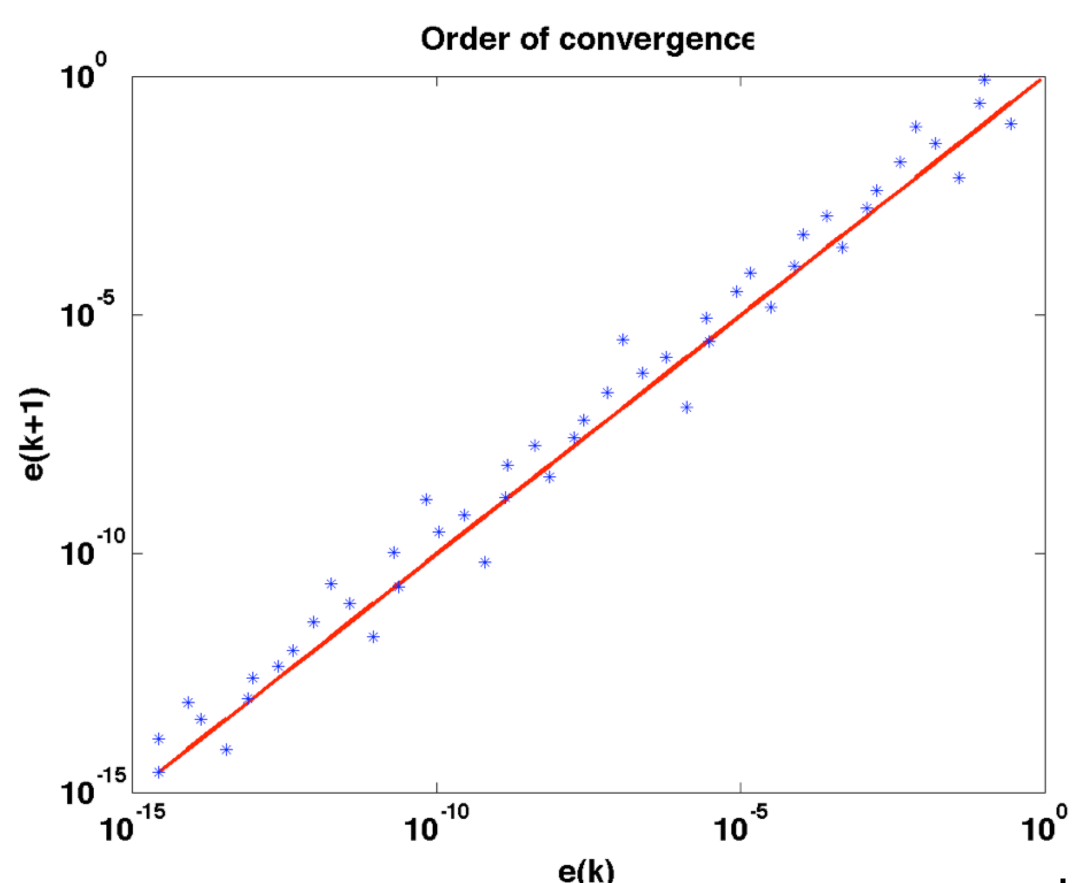
Estimating order of convergence

Again, we can write

$$\log(e_{k+1}) = \alpha \log(e_k) + \log(\mu)$$

and use a best-fit-line approach to finding α, given a sequence of errors e_k.

```
>> ab = polyfit(log(e(1:end-1)),log(e(2:end)),1)

ab =

    0.9885   -0.8910
```



Both bisection and the fixed point method are *linear* methods