Fixed point iteration

designing fixed point iterations

very general class (but not including bisection) of iterations defined by

$$x_{n+1} = F(x_n)$$

necessary condition: exact solution satisfies fixed point equation

$$x^* = F(x^*)$$

simple iteration method for solution of $f(x) = 0$ where $\alpha \neq 0$:

$$F(x) = x - \alpha f(x)$$

```python
# simple fixed point iteration

f = lambda x : x**4 - 3*x**3 + 1.0
F = lambda x, alpha=1.0, f=f : x - alpha*f(x)

n = 100
x = np.zeros(n)
for k in range(1,n):
    x[k] = F(x[k-1], alpha=-0.02) # play with alpha!
```

```
pl.plot(x); pl.grid('on'); pl.ylabel('xk'); pl.xlabel('k');
```

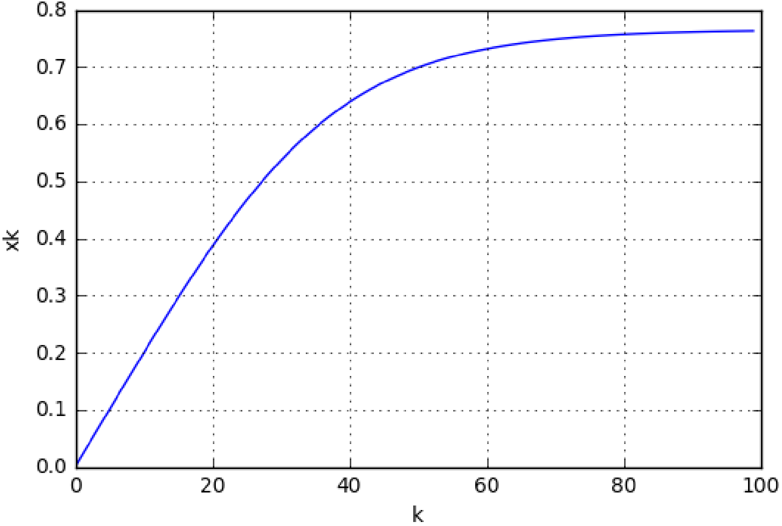

Figure 1: png

```
pl.semilogy(f(x)); pl.grid('on'); pl.ylabel('f(xk)'); pl.xl
```

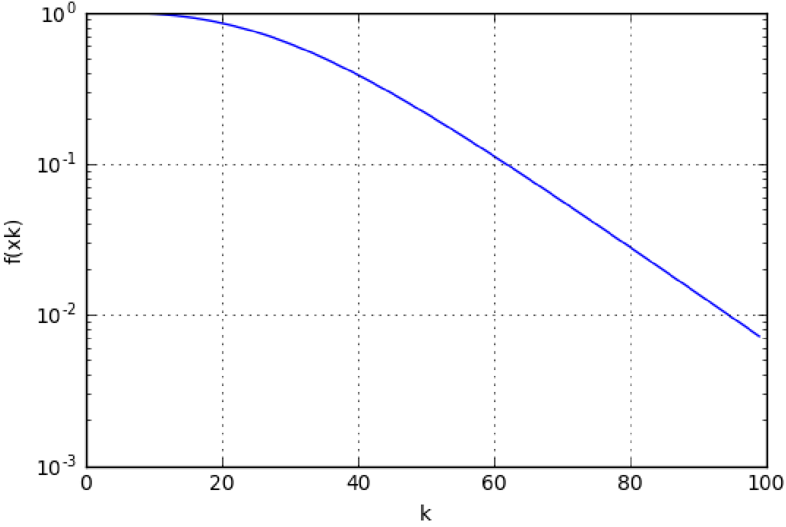

Figure 2: png

motivation of simple iteration:

- correct the approximation $x^{(k)}$ with the error:

$$x^* = x^{(k)} - e^{(k)}$$

- use $f(x^{(k)})$ as measure for error $e^{(k)}$ (approximately proportional)
- approximation

$$e^{(k)} \approx \alpha f(x^{(k)}$$

  gives

$$x^* \approx x^{(k)} - \alpha f(x^{(k)})$$

- use this approximation to be the next iterate

$$x^{(k+1)} = x^{(k)} - \alpha f(x^{(k)})$$

Contractive mapping theorem (calculus)

$F(x)$ is *contractive* on interval I if for some $0 \leq \lambda < 1$ one has

$$|F(x) - F(y)| \leq \lambda |x - y|,$$

The Contractive Mapping Theorem *Let F be contractive for all x in a closed bounded interval $I = [a, b]$ with $F(x) \in I$ for all $x \in I$. Then F has a unique fixed point in that interval. Further, for any $x_0 \in I$, the iteration defined by will converge to this fixed point.*

Convergence of fixed point iteration

If F is contractive on a real interval $[a, b]$ and $F([a, b]) \subset [a, b]$ then the sequence x_n defined by fixed point iteration with F converges and the error satisfies

$$|e_n| \leq \lambda^n |e_0|.$$

- example $F(x) = x - \alpha f(x)$ needs to satisfy

$$|x - y - \alpha(f(x) - f(y))| \leq \lambda |x - y|$$

- if f is differentiable on some interval and $0 < \beta < f'(x)$ one can use the intermediate value theorem to get an idea how to choose α

Function Iteration Formulation

- Newton's method can be written in the form

$$x^{(k)} = F(x^{(k)})$$

  with $F(x) = x - f(x)/f'(x)$ (to come in a few days)
- fundamental strategy:
  1. start with some approximation
  2. use approximation and the function to estimate the error of the approximation
  3. subtract the approximate error to get updated approximation
  4. repeat

```python
# iteration for fixed point problem x = exp(-x)

F = lambda x : np.exp(-x)
f = lambda x : x - F(x)

n = 10
x = np.zeros(n)
for k in range(1,n):
    x[k] = F(x[k-1]) # play with alpha!
```

```
pl.plot(x); pl.grid('on'); pl.ylabel('xk'); pl.xlabel('k');
```

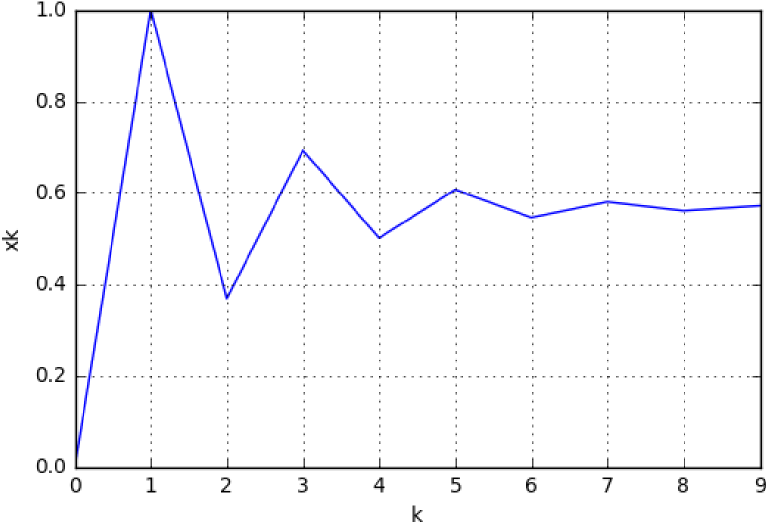

Figure 3: png

```
pl.plot(f(x)); pl.grid('on'); pl.ylabel('f(xk)'); pl.xlabel
```

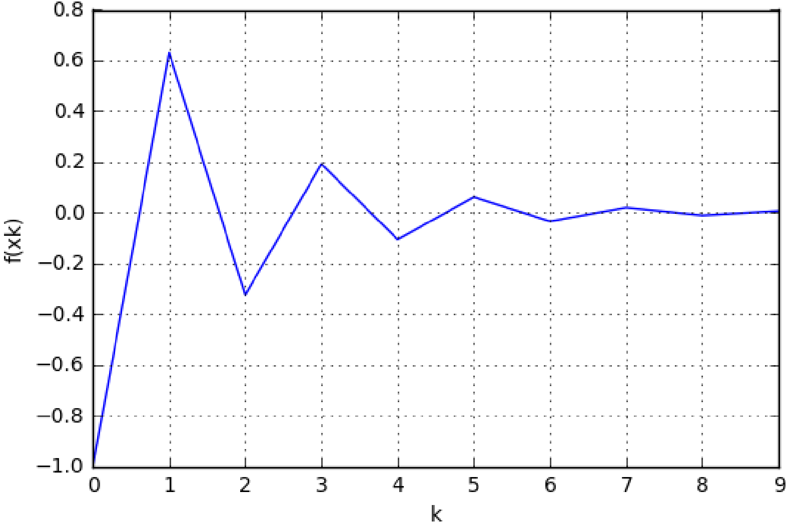

Figure 4: png

Show that the sequence $x^{(k)}$ converges ...