Statistical Theory and Modeling (ST2601)
Lecture 10 - Numerical Maximum Likelihood

Mattias Villani

Department of Statistics
Stockholm University

Overview

- **Numerical maximum likelihood**

Numerical optimization

- Find the maximum of a function $f(x)$

$$x_{\max} = \arg\max_{x \in \mathcal{X}} f(x)$$

  involves **solving for x in $f'(x) = 0$**.
- **Gradient ascent**: find $x_{\max}$ by iterating until convergence:

$$x_{k+1} = x_k + \underbrace{\gamma}_{\text{learning rate}} \cdot \underbrace{f'(x_k)}_{\text{gradient}}$$

- **Newton-Raphson**: find $x_{\max}$ by iterating until convergence:

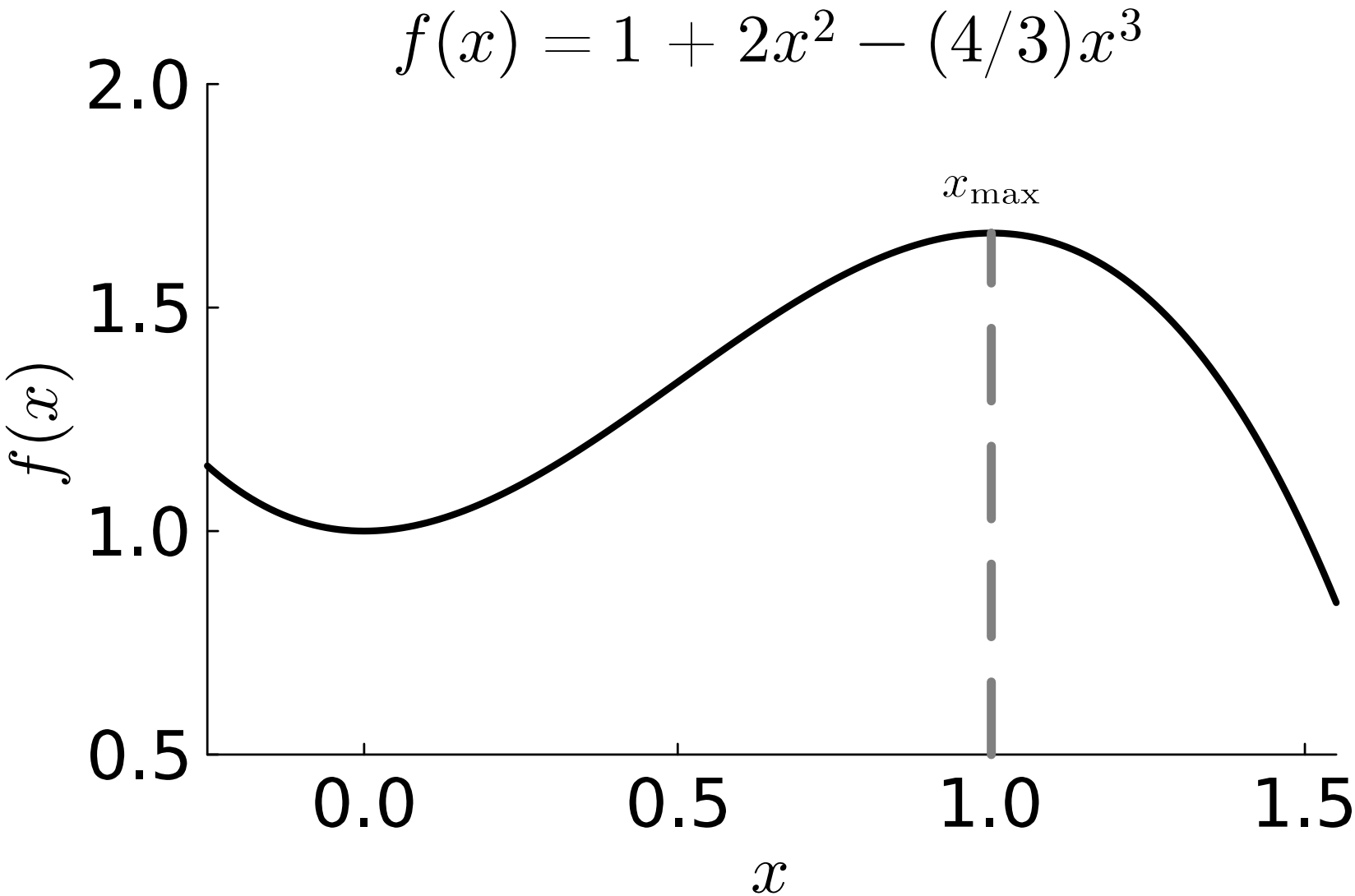
$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

- Multi-dimensional MLE solves $\ell'(\boldsymbol{\theta}) = 0$. Newton-Raphson:

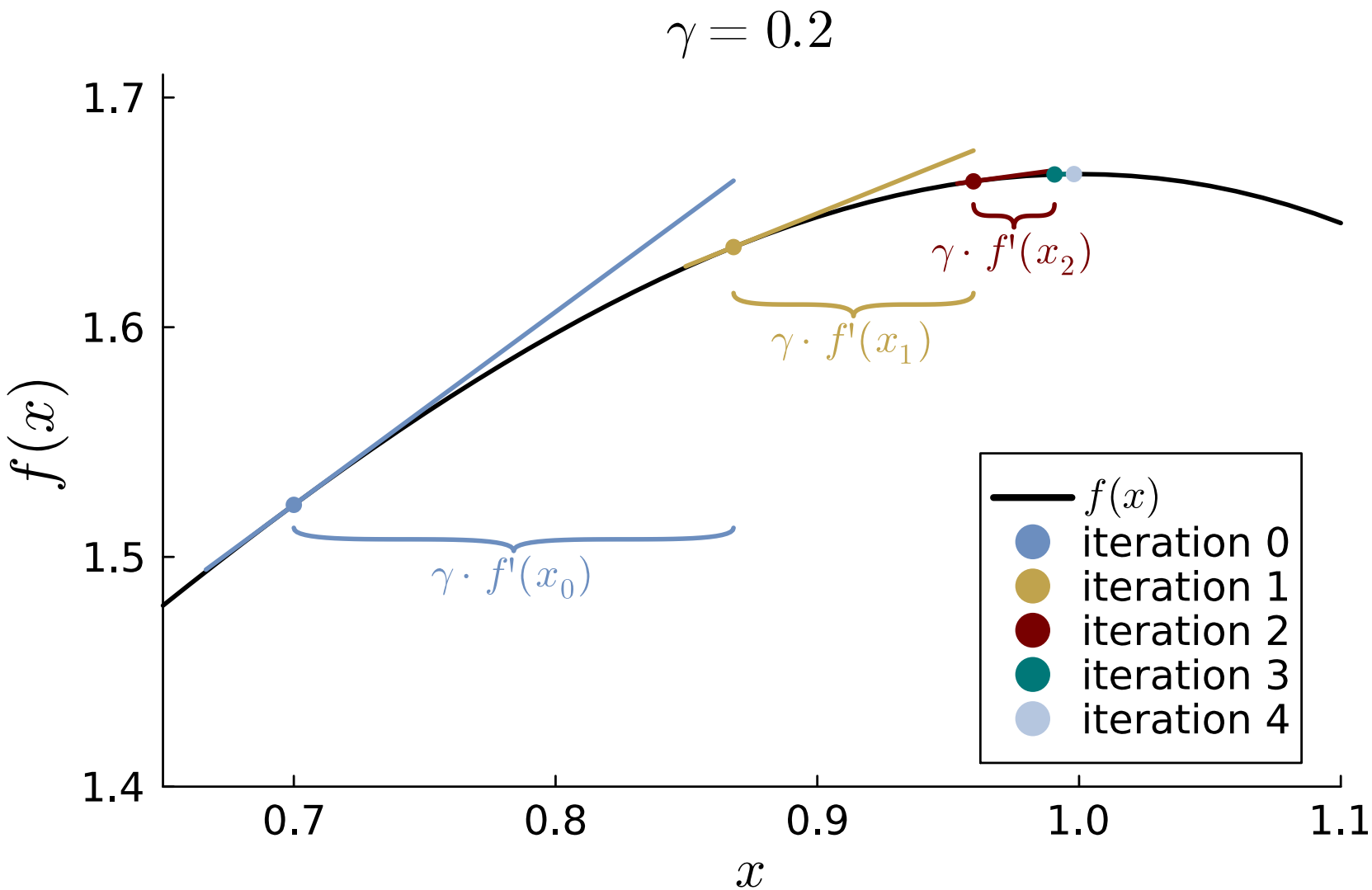
$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \underbrace{\mathcal{J}^{-1}(\boldsymbol{\theta}_k)}_{\text{inverse hessian}} \cdot \underbrace{f'(\boldsymbol{\theta}_k)}_{\text{gradient}}$$

- Inverse Hessian cheaply computed from gradients (BFGS).

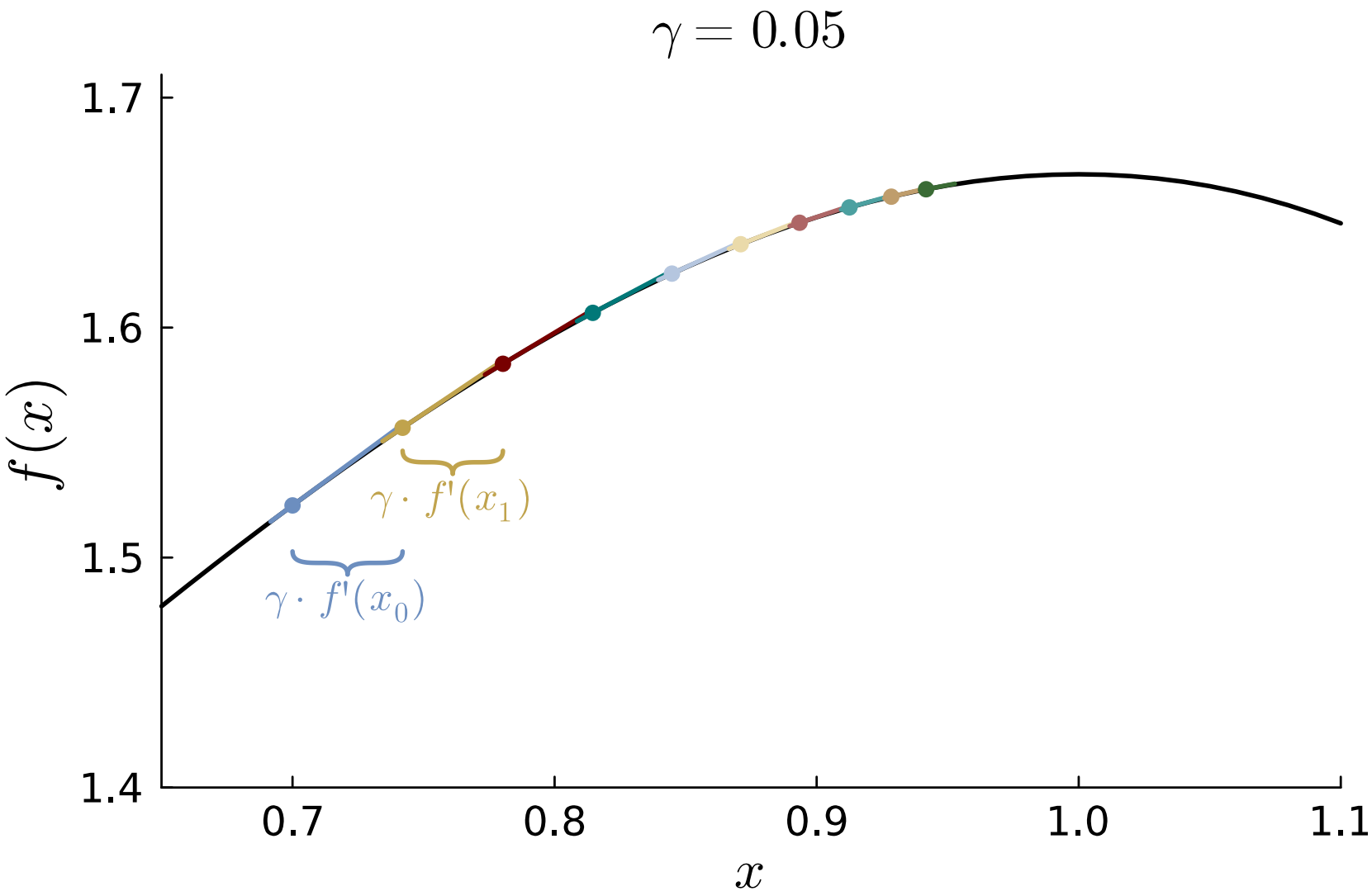
Finding the maximum



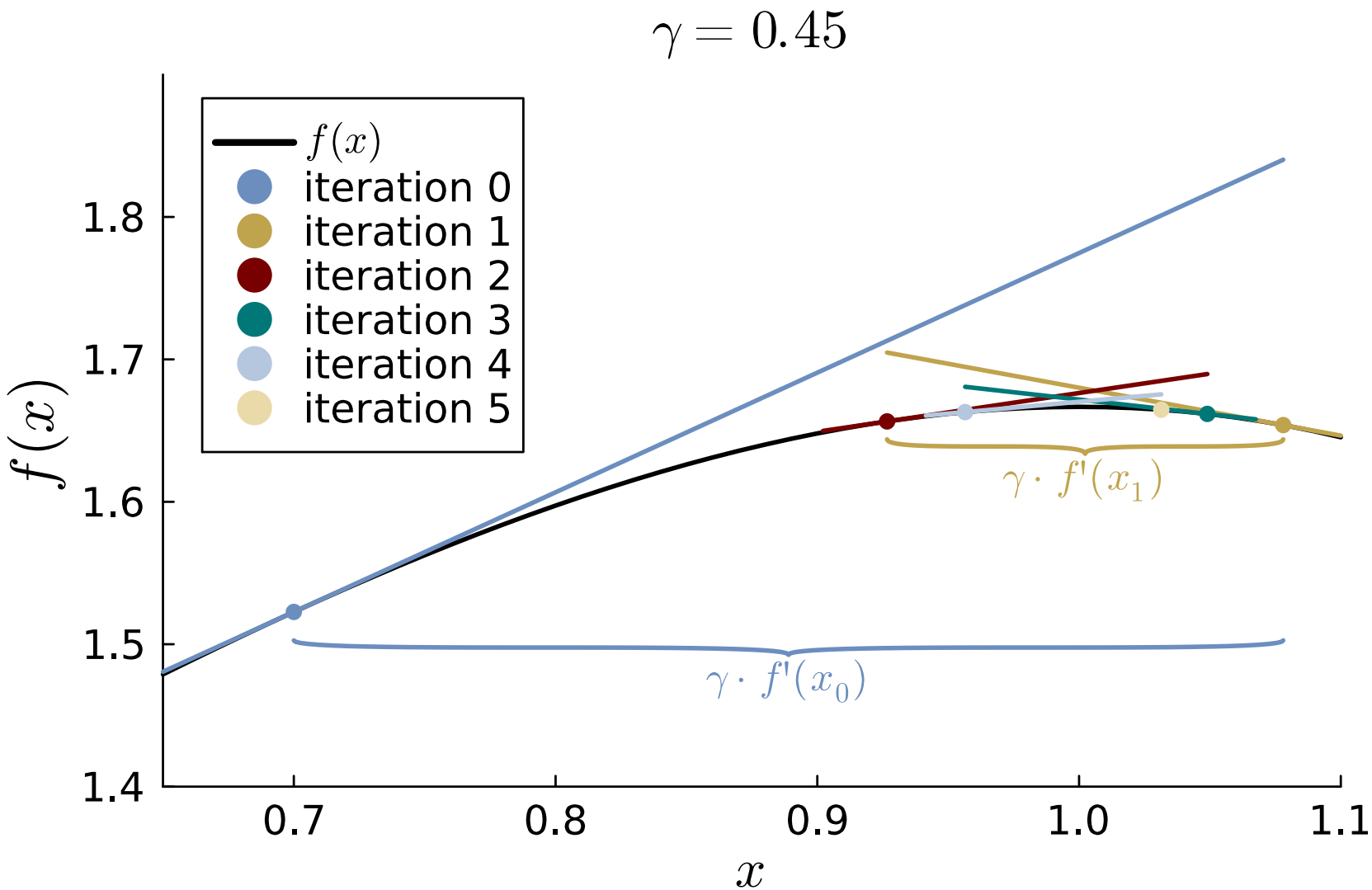
Gradient ascent - good learning rate



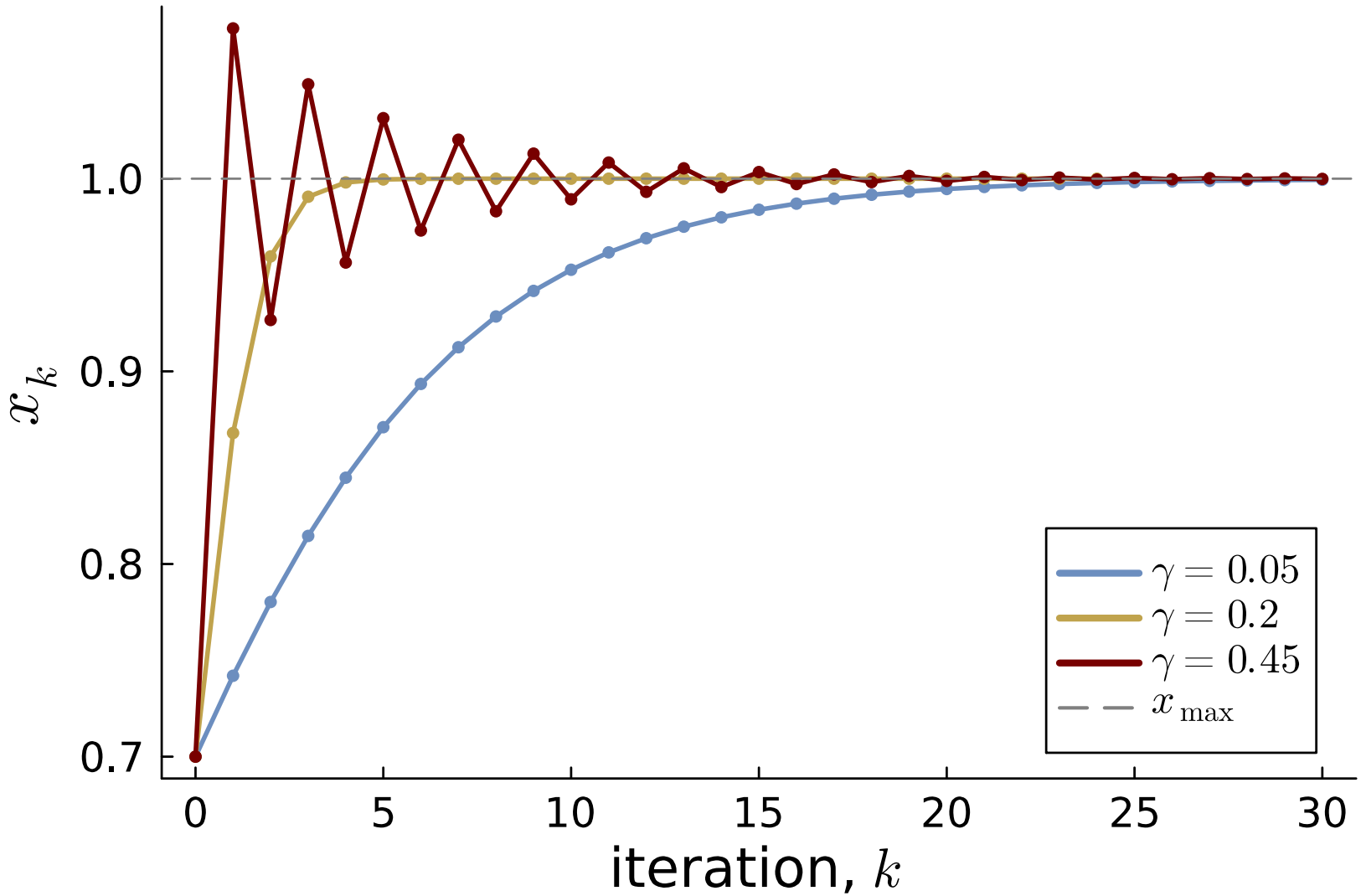
Gradient ascent - too small learning rate



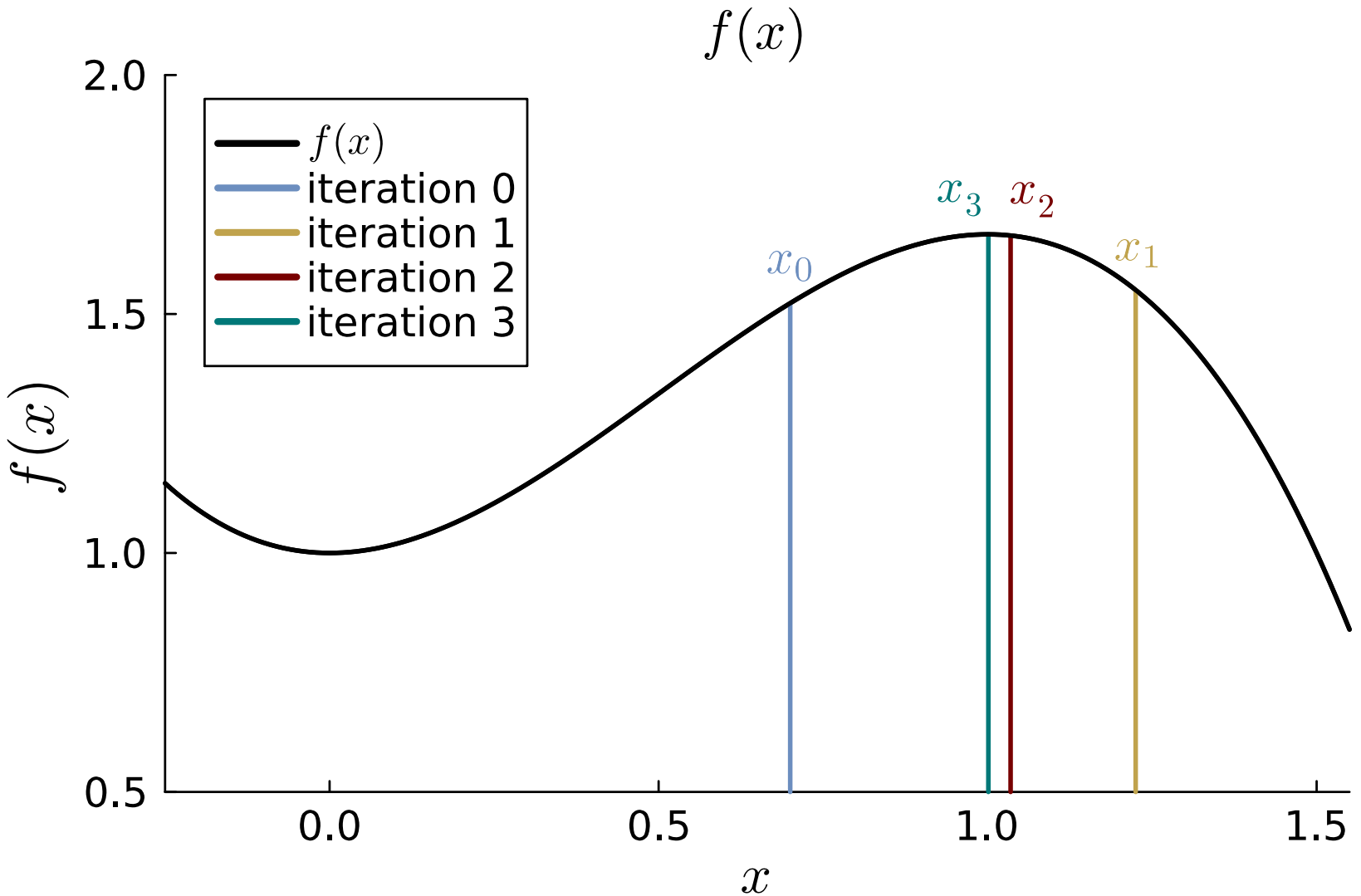
Gradient ascent - too large learning rate



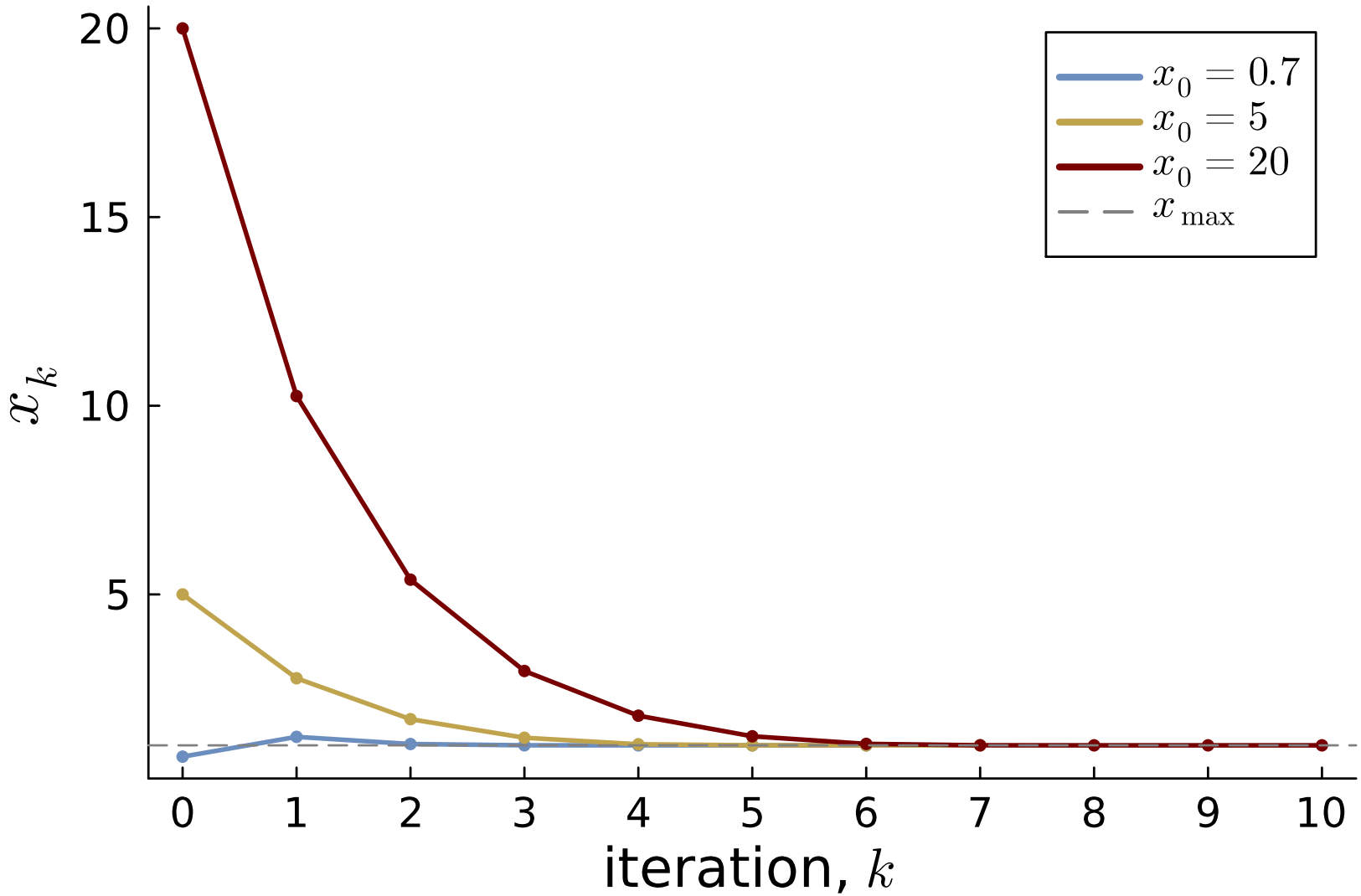
Gradient ascent convergence



Newton-Raphson



Newton-Raphson convergence



Optim in R

```r
# Function to maximized. Depends on parameter a
myFunc <- function(x, a){
  funcVal = 1 - (x/a-2)^2
  return (funcVal)
}

# Run optim
a = 1
initVal <- c(3)       # initial guess
OptimResults <- optim(initVal, myFunc, gr=NULL, a,
                      method=c("BFGS"),
                      control=list(fnscale=-1),
                      hessian=TRUE)

OptimResults$par      # maximizer
OptimResults$value    # maximum function value
OptimResults$hessian # second derivative
```