Optimisation Methods I

Matteo Fasiolo

Optimisation Problems

Many problems in statistics can be characterised as solving

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\, f(\boldsymbol{\theta}), \tag{1}$$

where f is a real scalar valued function of vector $\boldsymbol{\theta}$, usually called the **objective function**. For example, f might be:

- the negative log-likelihood of a statistical model;
- the (negative) Bayesian posterior density of $\boldsymbol{\theta}$;
- a dissimilarity measure for the alignment of two DNA sequences, where the alignment is controlled by $\boldsymbol{\theta}$;
- the total distance travelled, and $\boldsymbol{\theta}$ defines the order of delivery drop off points.

Note that

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} -f(\boldsymbol{\theta}),$$

so we are not losing anything by concentrating on minimisation.

In the optimization literature, the convention is to minimise the objective, which is interpreted as a **cost** or **penalty** function.

Our setup and assumptions:

- $\boldsymbol{\theta} \in \Theta$, the set of all possible values
- we can evaluate $f(\boldsymbol{\theta})$ for all elements of $\boldsymbol{\theta} \in \Theta$
- f is "well-behaved'', e.g. $\boldsymbol{\theta}_1 \approx \boldsymbol{\theta}_2$ implies $f(\boldsymbol{\theta}_1) \approx f(\boldsymbol{\theta}_2)$

To further restrict our focus, we will assume Θ is continuous.

So excluding discrete optimisation problems, why?

There is a big divide between continuous and discrete problems.

We can not cover both in two lectures.

Continuous problem are:

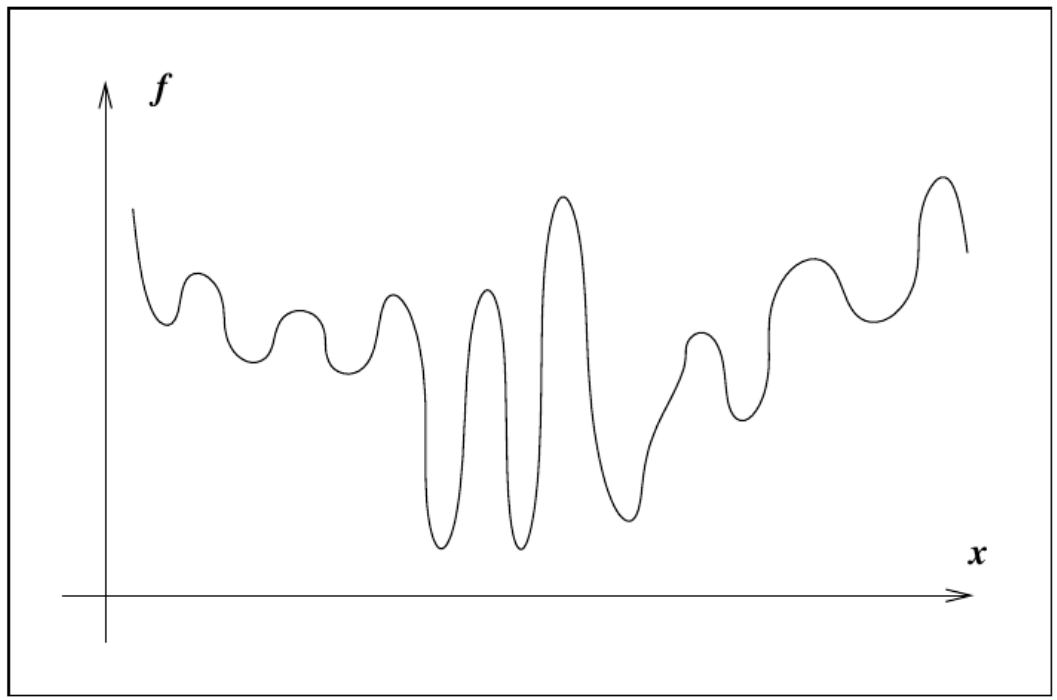
- generally easier, discrete problems can cost $> O(n^k)$ for any k
- more commonly encountered in Statistics
- often solvable via widely-used optimisation methods

We will consider unconstrained problems where $\Theta = \mathbb{R}^p$.

If $\boldsymbol{\theta}$ constrained:

- write $\boldsymbol{\theta} = g(\boldsymbol{\theta}_u)$ (e.g., $\theta = \exp(\theta_u)$);
- $\hat{\boldsymbol{\theta}}_u = \text{argmin}_{\boldsymbol{\theta}_u} f(g(\boldsymbol{\theta}_u))$

We will consider **local** optimisation methods.



Local = global if $f(\boldsymbol{\theta})$ convex.

A Simple Model Leading to Troubles

Consider the logistic map, that is

$$n_{t+1} = rn_t(1 - n_t/K), \qquad t = 0, 1, 2, \ldots$$

where

- n_t is the population at time t (assume n_0 is known)
- r the growth rate parameter
- K is the carrying capacity (if $n_t = K$ then $n_{t+1} = 0$)

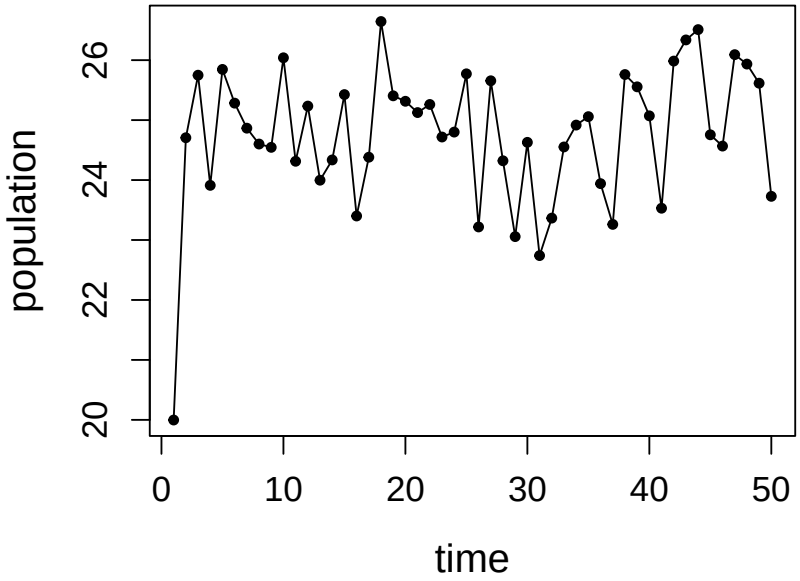
Suppose we observe $y_t = n_t + \epsilon_t$ where $\epsilon_t \sim N(0, \sigma^2)$.

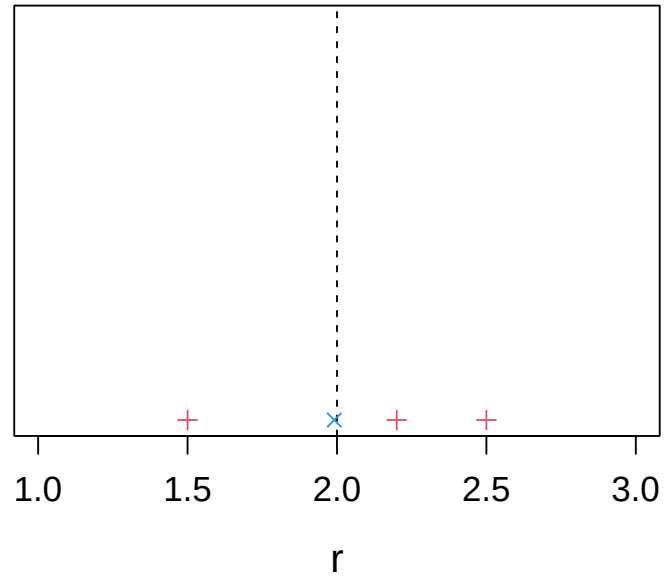
We want to estimate r and K by minimising

$$f(r, K) = \sum_{t=1}^{T} (y_t - n_t)^2$$

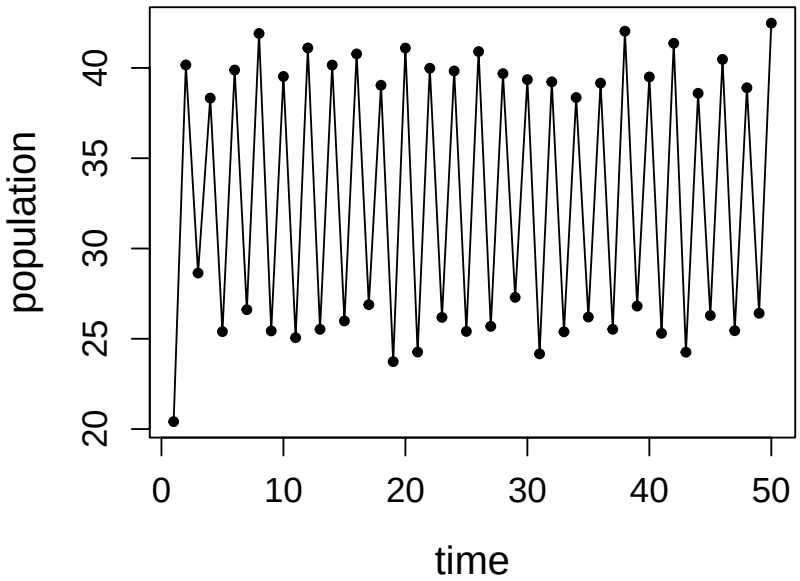
where $n_t = n_t(r, K, n_{t-1})$.

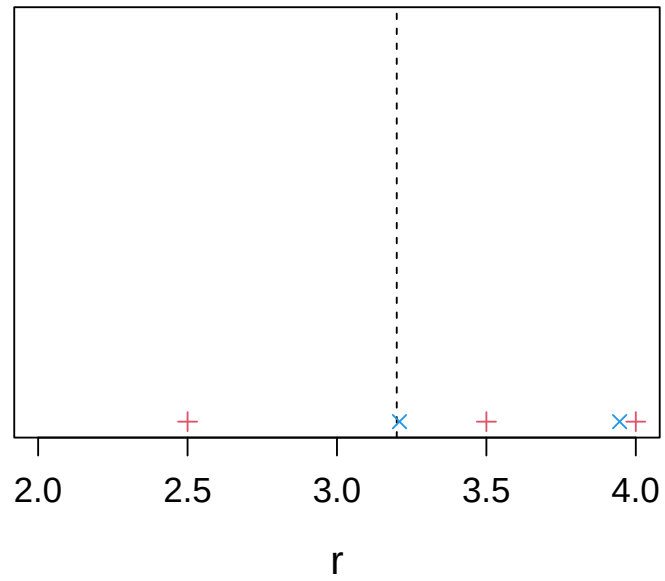
Simulation with $r = 2$ and $k = 50$



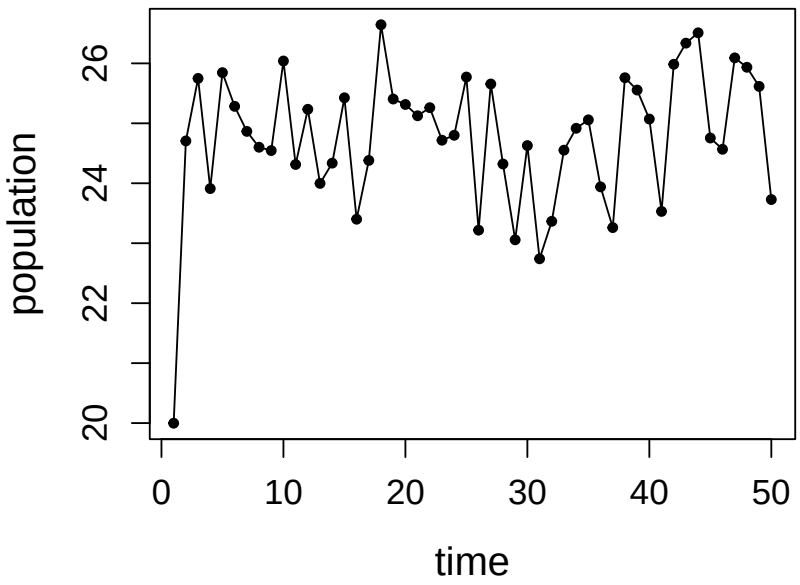


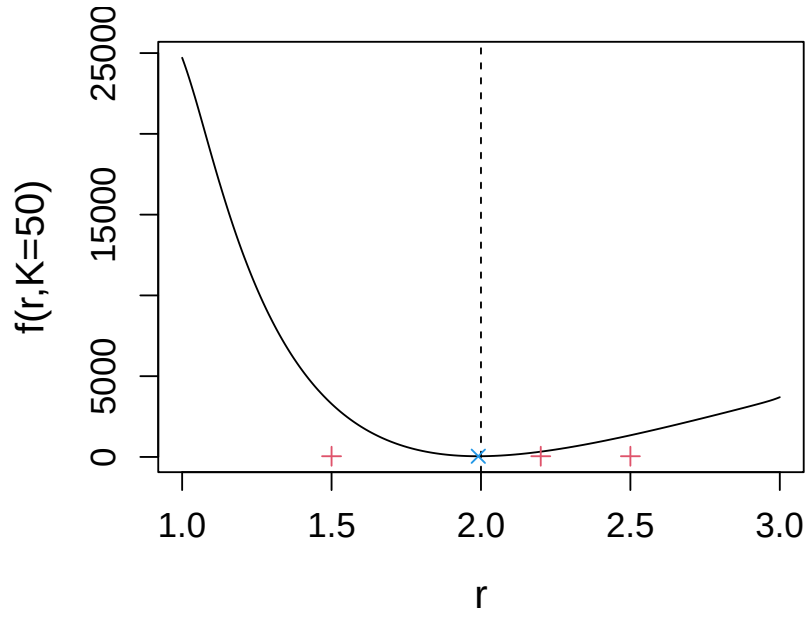
Simulation with $r = 3.2$ and $k = 50$



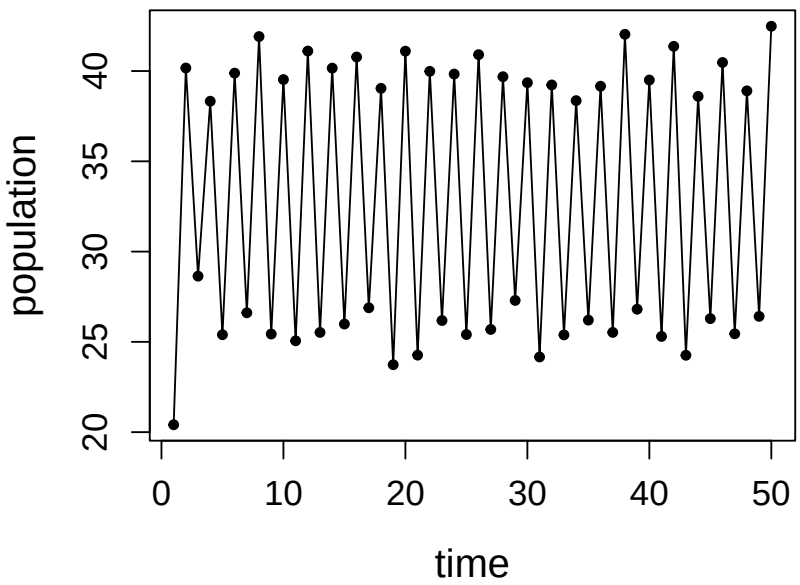


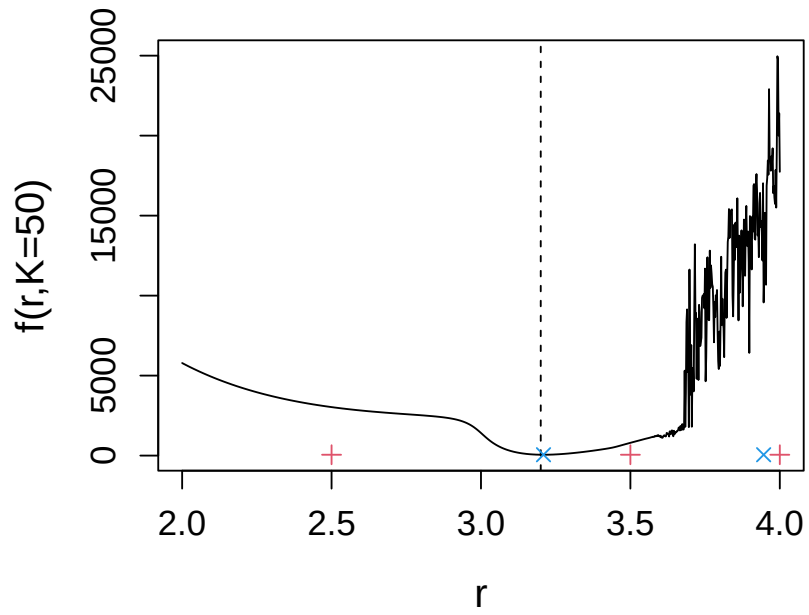
Simulation with $r = 2$ and $k = 50$



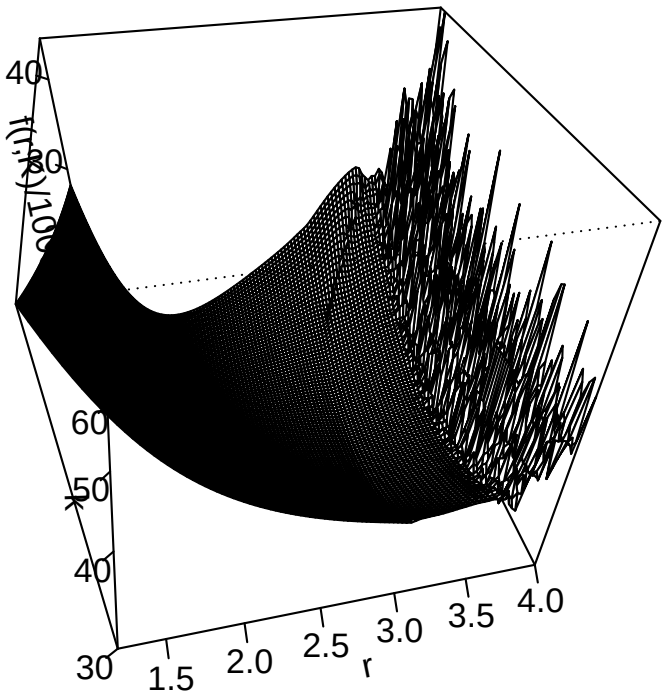


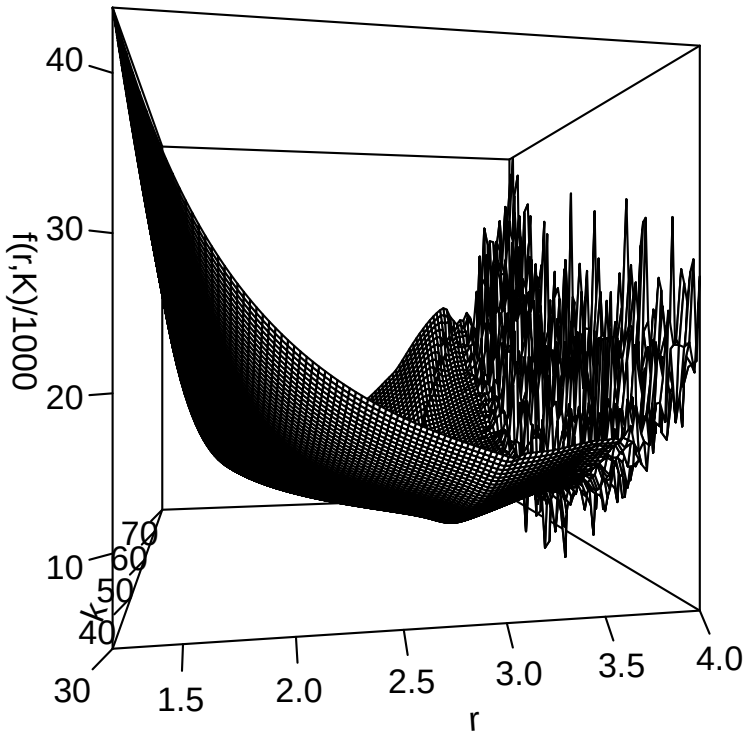
Simulation with $r = 3.2$ and $k = 50$





Likelihood with simulations from $r = 3.2$ and $k = 50$:

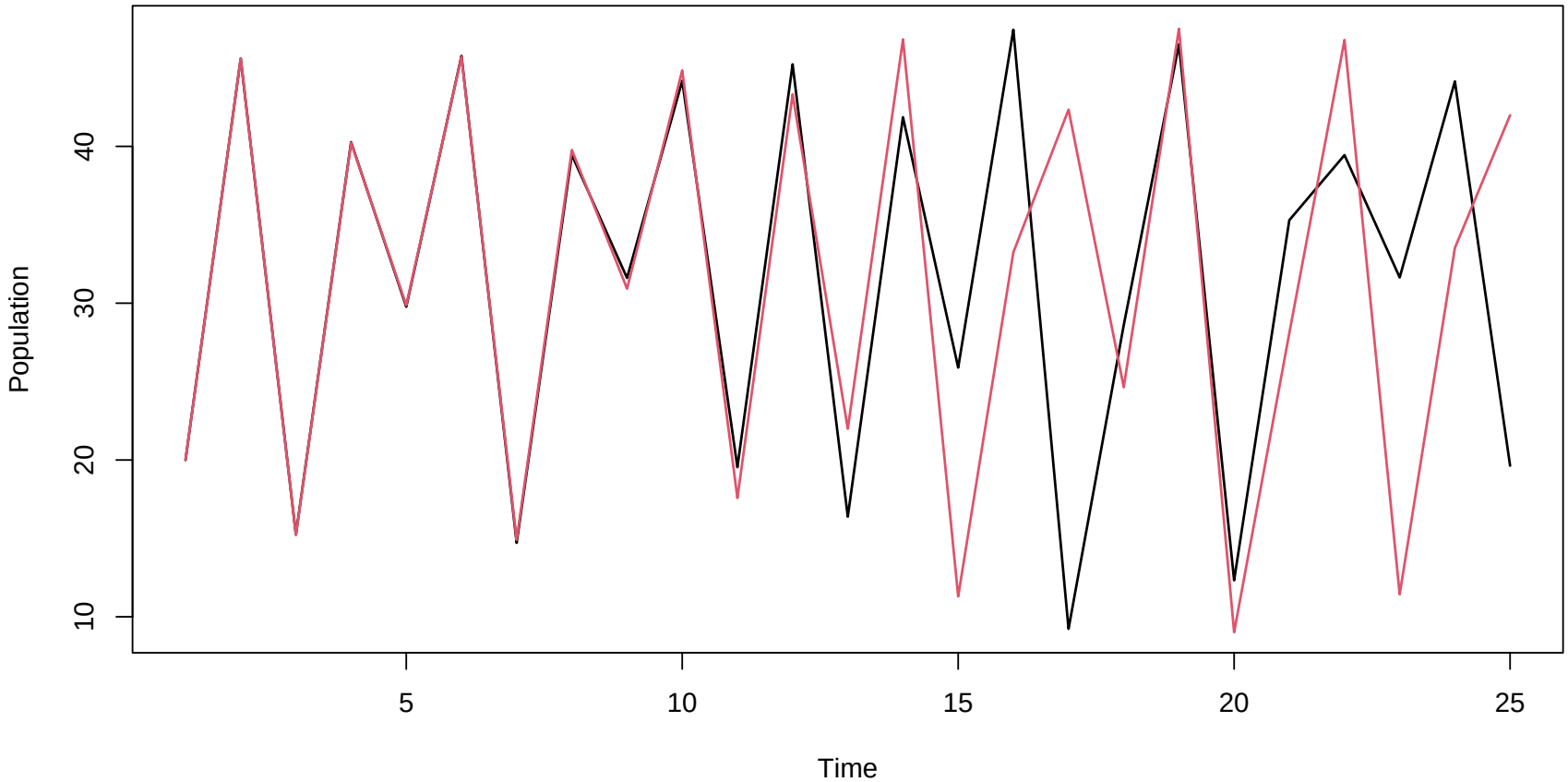




So:

- simple models can lead to hard optimisation problems;
- when the optimiser does not converge, look at the objective;
- often possible to do this even when d > 2 (see tomorrow).

Digression: What's wrong with this model?



Population n_t when $r = 3.8$ (black) and when $r = 3.8001$ (red).

Likelihood might not be a good objective when fitting chaotic models, see Fasiolo et al (2016).

Back to Local Optimisation

Given a guess $\boldsymbol{\theta}^{[0]}$ we consider methods that, at k-th iteration:

1. evaluate $f(\boldsymbol{\theta}^{[k]})$, and possibly $\nabla f(\boldsymbol{\theta})$ and $\nabla^2 f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^{[k]}$.
2. Use the information from step 1 to:
    a. Find a **search direction**, $\boldsymbol{\Delta}$, s.t. $f(\boldsymbol{\theta}^{[k]} + \alpha\boldsymbol{\Delta}) < f(\boldsymbol{\theta}^{[k]})$ for some $\alpha > 0$;
    b. Find α s.t. $f(\boldsymbol{\theta}^{[k]} + \alpha\boldsymbol{\Delta})$ is sufficiently lower than $f(\boldsymbol{\theta}^{[k]})$ (**sufficient decrease condition**);
    c. Set $\boldsymbol{\theta}^{[k+1]} = \boldsymbol{\theta}^{[k]} + \alpha\boldsymbol{\Delta}$.
3. If a minimum has yet been reached, stop, otherwise back to 1.

Note: in 2.a. we are trying to find an "optimal" direction, not any descent direction.

How to find $\boldsymbol{\Delta}$?

Assuming that f is twice differentiable, we can use Taylor expansion

$$f(\boldsymbol{\theta}+\boldsymbol{\Delta}) = f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\nabla^2 f(\boldsymbol{\theta})\boldsymbol{\Delta} + R^2_{\boldsymbol{\theta}}(\boldsymbol{\Delta}),$$

where $R^2_{\boldsymbol{\theta}}(\boldsymbol{\Delta})$ is the remainder of the expansion and

$$\nabla f(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} \\ \cdot \\ \cdot \end{pmatrix} \quad \text{and} \quad \nabla^2 f(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial^2 f}{\partial \theta_1^2} & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_2} & \cdot & \cdot \\ \frac{\partial^2 f}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 f}{\partial \theta_2^2} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

Note that

$$f(\boldsymbol{\theta}+\boldsymbol{\Delta}) = f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\nabla^2 f(\boldsymbol{\theta})\boldsymbol{\Delta} + R^2_{\boldsymbol{\theta}}(\boldsymbol{\Delta}),$$

can help us identify a minimum when we find it.

Consider candidate $\hat{\boldsymbol{\theta}}$, then for $||\boldsymbol{\Delta}|| \ll 1$ we have

$$f(\hat{\boldsymbol{\theta}}+\boldsymbol{\Delta}) \simeq f(\hat{\boldsymbol{\theta}}) + \nabla f(\hat{\boldsymbol{\theta}})^{\mathsf{T}}\boldsymbol{\Delta} + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\nabla^2 f(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta},$$

and if $\nabla f(\hat{\boldsymbol{\theta}}) = \mathbf{0}$ and $\nabla^2 f(\hat{\boldsymbol{\theta}})$ is positive definite we have

$$f(\hat{\boldsymbol{\theta}}+\boldsymbol{\Delta}) \simeq f(\hat{\boldsymbol{\theta}}) + \frac{1}{2}\boldsymbol{\Delta}^{\mathsf{T}}\nabla^2 f(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta} > f(\hat{\boldsymbol{\theta}}) \quad \text{for any small} \quad \boldsymbol{\Delta}.$$

Now back to search for $\boldsymbol{\Delta}$ based on Taylor expansion.

Steepest Descent (aka Gradient Descent)

SD uses a first order Taylor expansion around current $\boldsymbol{\theta}$

$$f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta}.$$

Under local model, along which direction $\boldsymbol{\Delta}$ should we move?

Note that local model does not provide any info about $||\boldsymbol{\Delta}||$.

Let $p = \dim(\boldsymbol{\theta})$ then if:

- $p = 1$ if local model is a line
- $p = 2$ a plane
- $p \geq 3$ a hyper-plane

Let's separate the magnitude and the length of the step.

So, let's assume $||\boldsymbol{\Delta}|| = 1$ and that the local model is

$$f(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \alpha\nabla f(\boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\Delta},$$

for some step-length $\alpha > 0$.

Then

$$f(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \alpha||\nabla f(\boldsymbol{\theta})||\cos(\phi),$$

where ϕ is the angle between $\nabla f(\boldsymbol{\theta})$ and $\boldsymbol{\Delta}$.

For any fixed α, $f(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta})$ is minimal when $\phi = \pi$, so

$$\boldsymbol{\Delta} = -\frac{\nabla f(\boldsymbol{\theta})}{||\nabla f(\boldsymbol{\theta})||}.$$

That's the **steepest descent** direction.

But, given $\boldsymbol{\Delta}$, how to choose α?

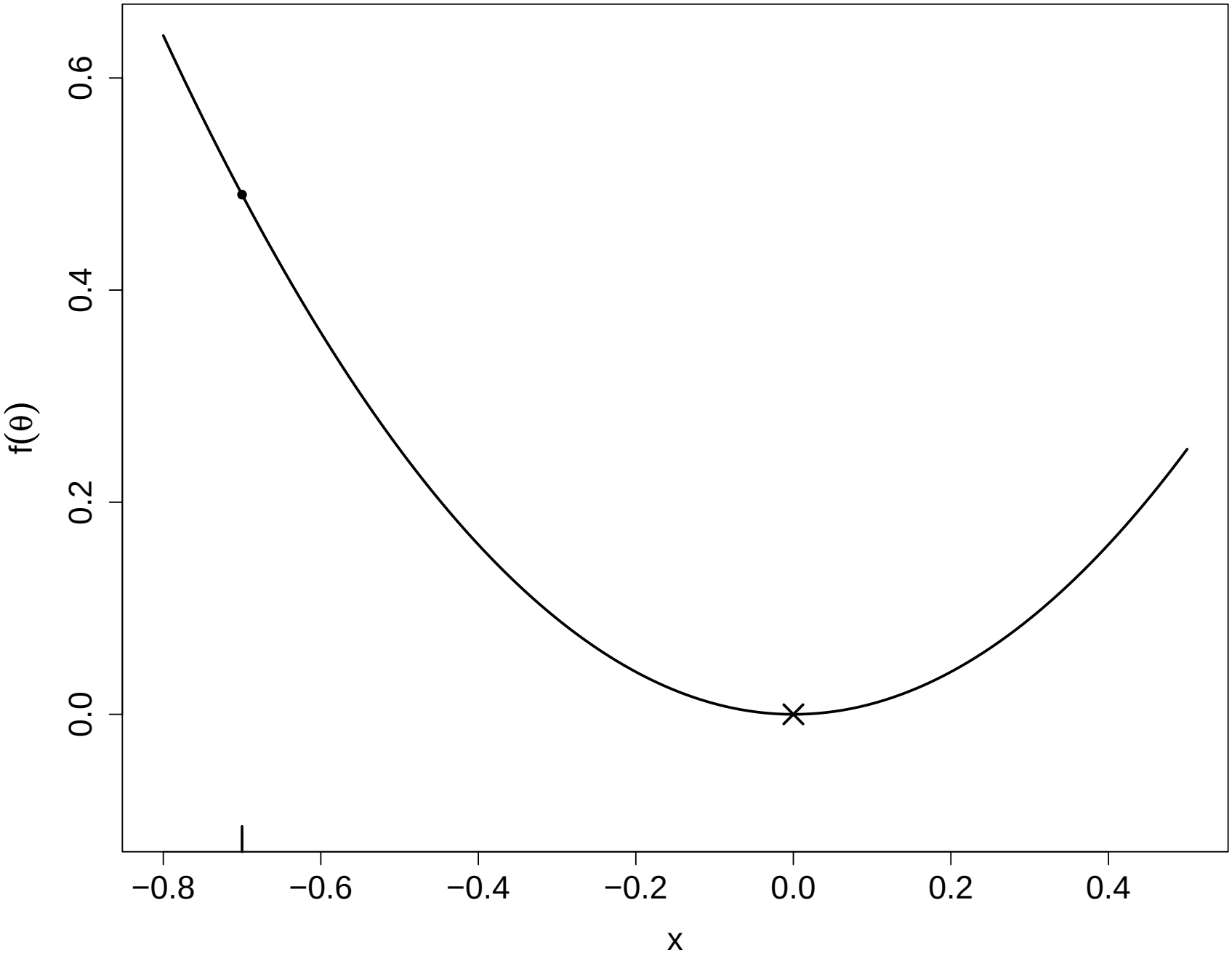
This is a **line search problem**, α moves us along $\boldsymbol{\theta} + \alpha\boldsymbol{\Delta}$.

Local linear model does not give any information to choose α.

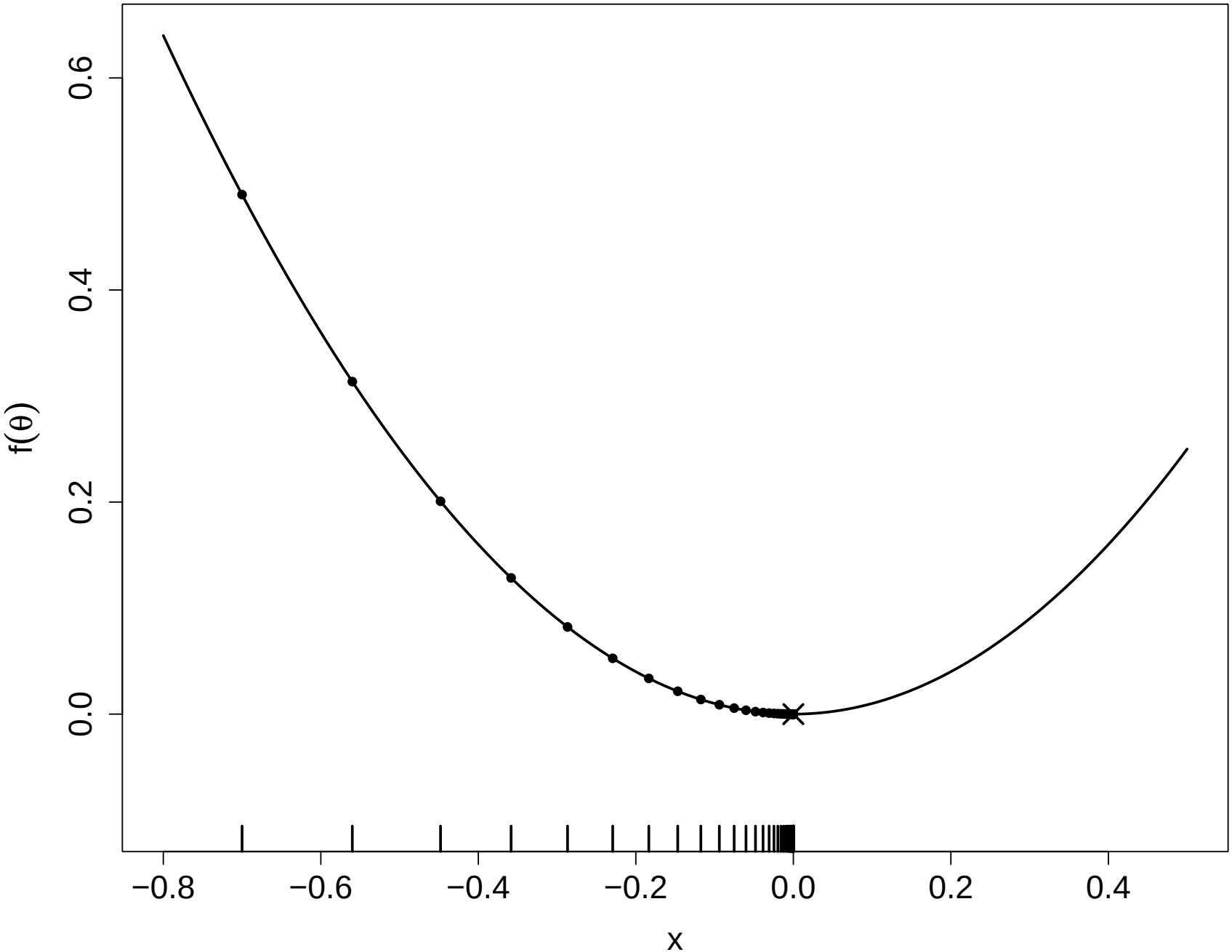
Possible approaches (omitting iteration index k):

1. Find $\alpha^* = \underset{\alpha}{\operatorname{argmin}}\, f(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta})$ (optimal but expensive);
2. Set $\alpha = \alpha_0$ and then **backtrack**:
   - if $f(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta}) > f(\boldsymbol{\theta})$ set $\alpha \leftarrow \alpha/2$ and retry;

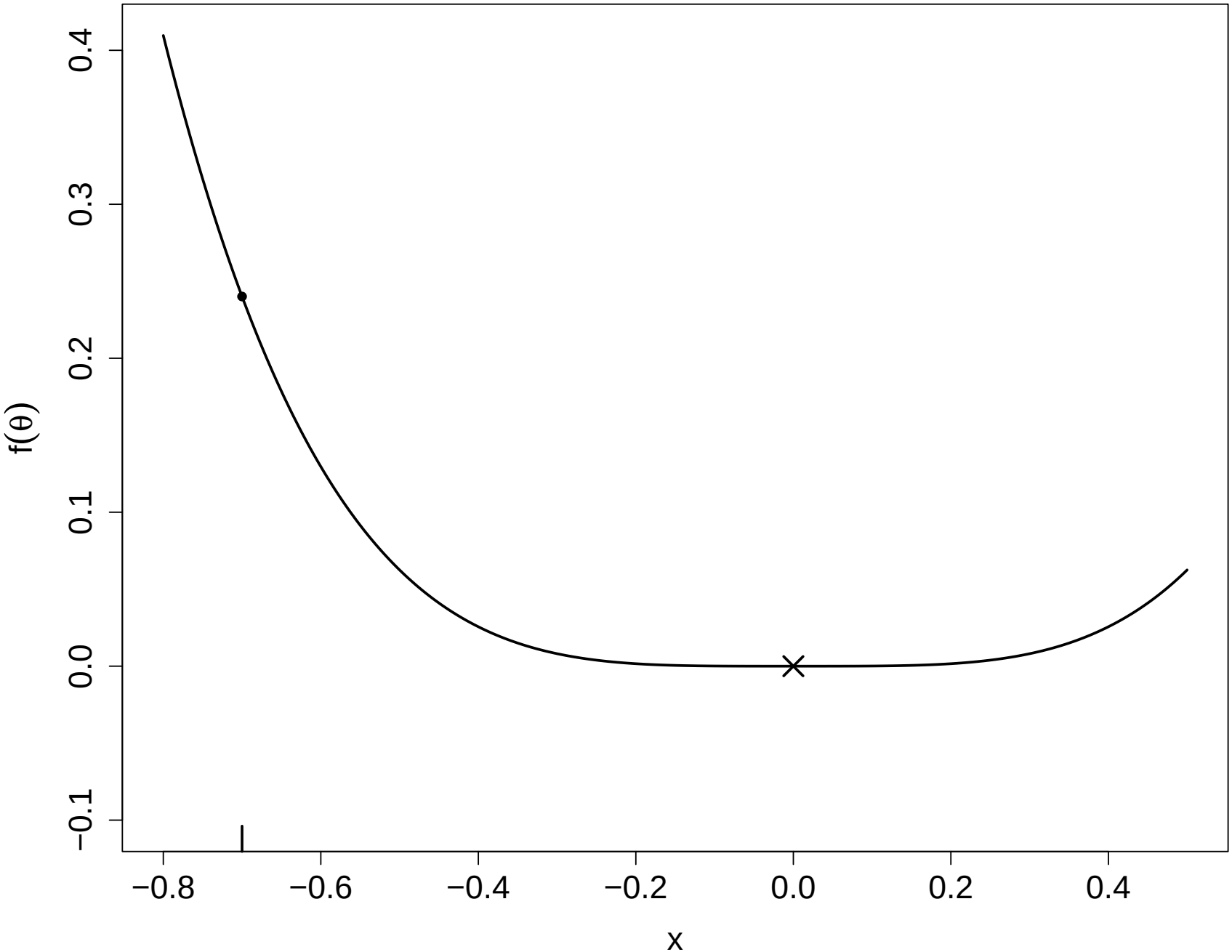
SD on $f(\theta) = \theta^2$, $f'(\theta) = 2\theta$, with 100 steps and $\alpha_0 = 0.1$.



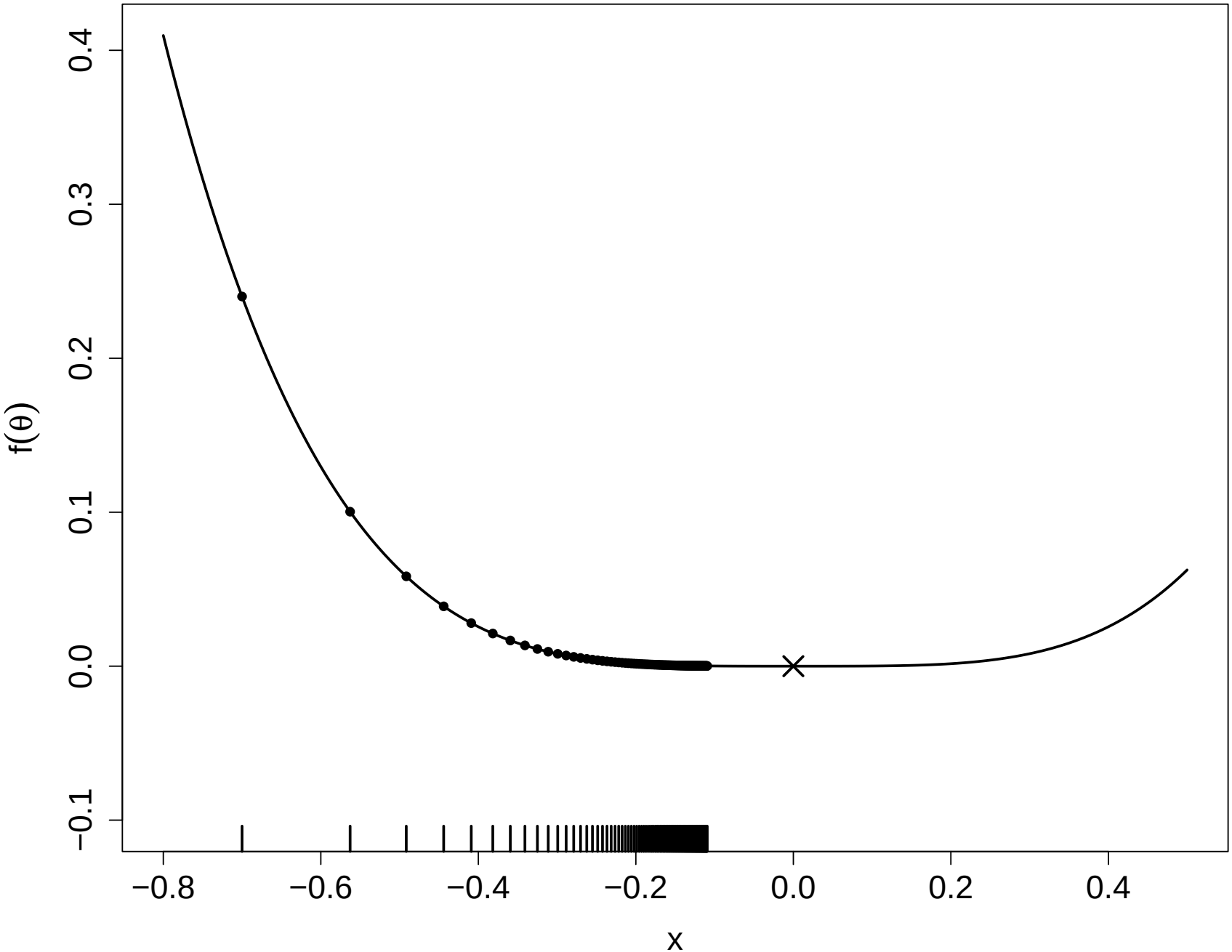
SD on $f(\theta) = \theta^2$, $f'(\theta) = 2\theta$, with 100 steps and $\alpha_0 = 0.1$.



SD on $f(\theta) = \theta^4$, $f'(\theta) = 4\theta^3$, with 100 steps and $\alpha_0 = 0.1$.



SD on $f(\theta) = \theta^4$, $f'(\theta) = 4\theta^3$, with 100 steps and $\alpha_0 = 0.1$.



So one might look for more complicated line searches.

E.g. we might look for **sufficient** decrease of $f(\boldsymbol{\theta} + \alpha\boldsymbol{\Delta})$.

But further problem is that $\boldsymbol{\Delta}$ from SD is itself not amazing.

It is based on a local linear approximation to the objective.

It can lead to zig-zagging behaviour.

See animated examples from Finn Lindgren:

```
library(devtools)
devtools::install_bitbucket("finnlindgren/FLtools")

library(FLtools)
FLtools::optimisation()
```

Conclusions on SD

Proposed step k-th is:

$$\boldsymbol{\theta}^{[k+1]} \leftarrow \boldsymbol{\theta}^{[k]} - \alpha \nabla f(\boldsymbol{\theta}^{[k]}).$$

It's prudent to couple it with a line search for α to avoid:

- very slow convergence (small steps)
- divergence (overly large steps)

See Nocedan and Wright (2006), Chapter 3.

It's simple, but not a great method when we actually want to reach the minimum.

In ML estimation, we want to reach the MLE exactly.

Then we use asymptotic theory to quantify uncertainty via Hessian of log-likelihood.

So SD not widely used in maximum likelihood estimation.

Exception is Big Data problems where **stochastic gradient descent** (SGD) is used.

At each iteration we compute gradient on a random subsample.

SGD is method of choice in Machine Learning, e.g. to fit NNs.

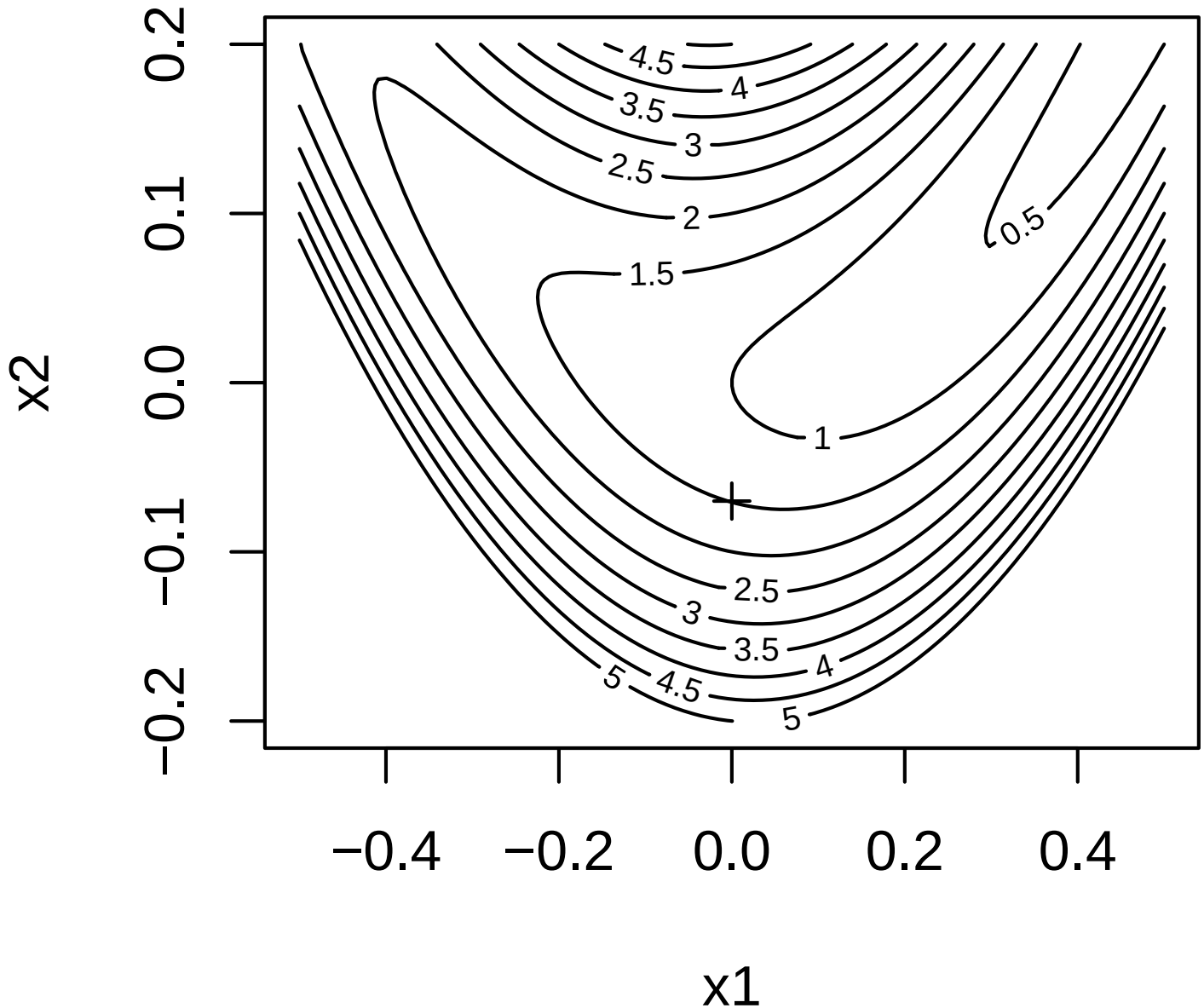
But aim is exactly minimising the objective on train data.

Noise of SGD and early stopping are used to **regularise** the fit.

So SGD is used not purely as an optimiser.

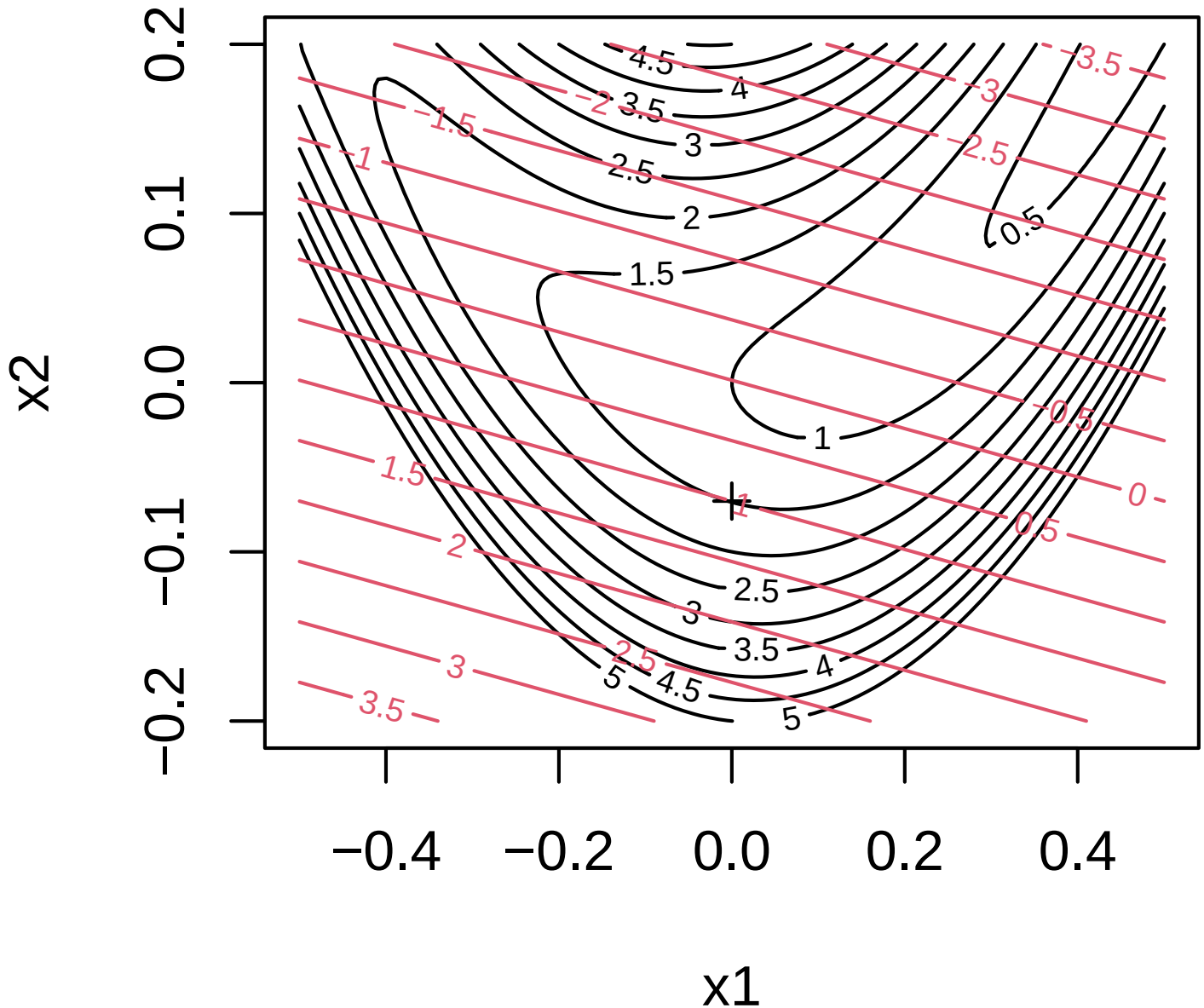
How to improve SD

Recall the local linear model is $f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta}$.



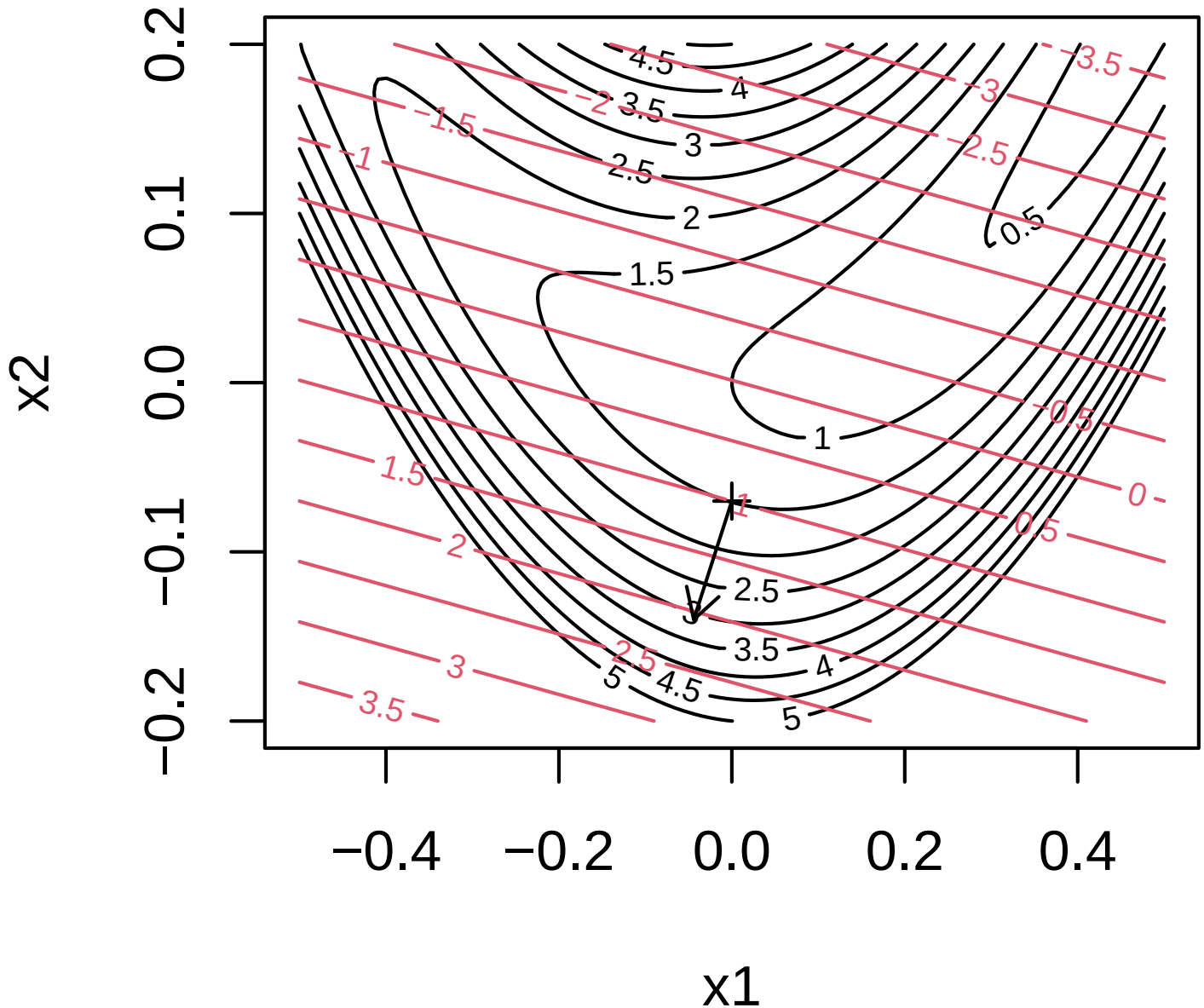
How to improve SD

Recall the local linear model is $f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta}$.



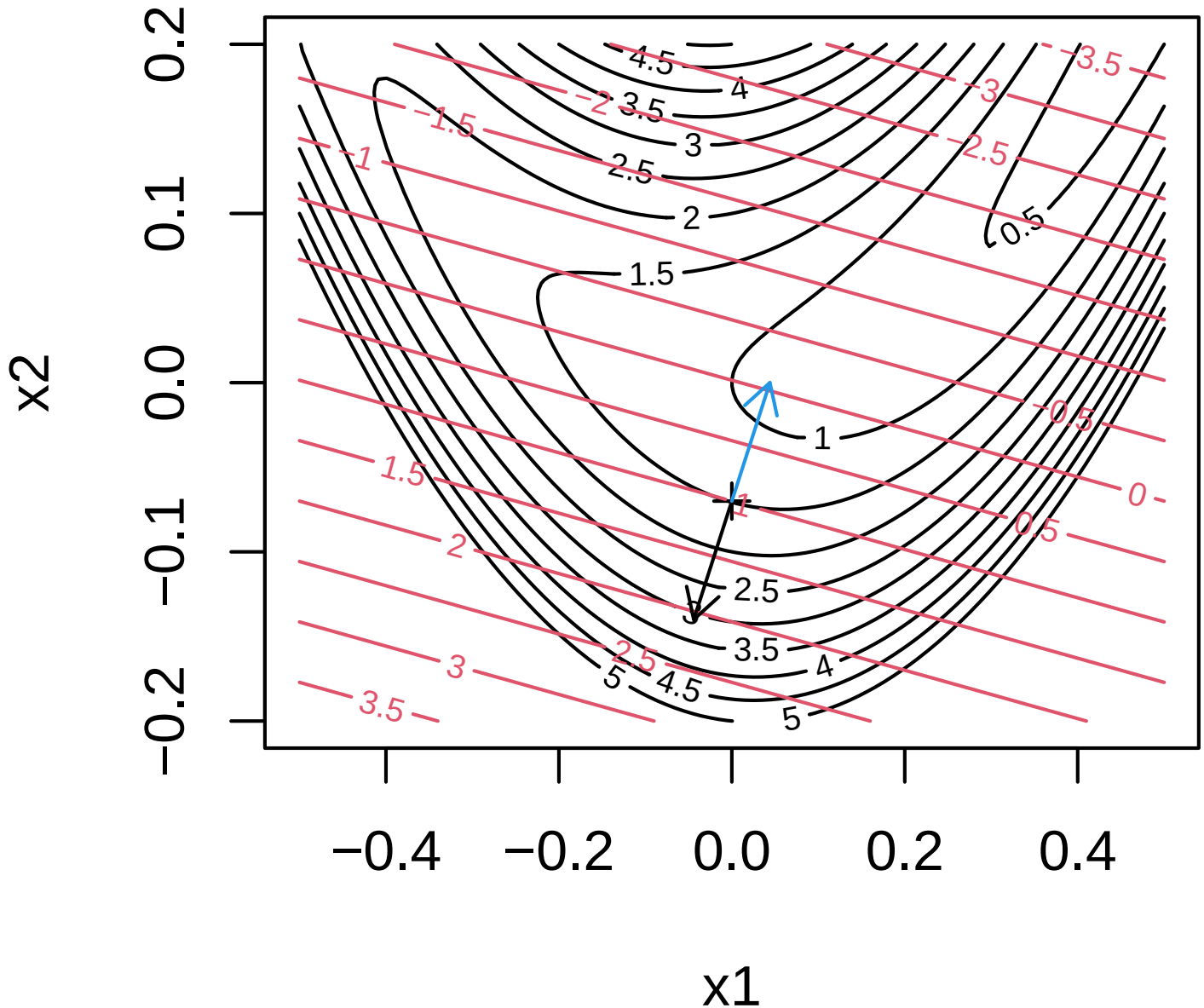
How to improve SD

Recall the local linear model is $f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta}$.



How to improve SD

Recall the local linear model is $f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta}$.



Note:

- Linear model is tangent to $f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}$
- $\nabla f(\boldsymbol{\theta})$ is $\perp$ to contour of f passing by $\boldsymbol{\theta}$.
- Any $\boldsymbol{\Delta}$ s.t. $\boldsymbol{\Delta}^T \nabla f(\boldsymbol{\theta}) < 0$ goes downhill on local model

Hence we can modify SD step via **preconditioning** matrix $\mathbf{A}^{[k]}$:

$$\boldsymbol{\theta}^{[k+1]} \leftarrow \boldsymbol{\theta}^{[k]} - \alpha \mathbf{A}^{[k]} \nabla f(\boldsymbol{\theta}^{[k]}).$$

Effect of $\mathbf{A}^{[k]}$ is to rotate and rescale SD step.

What are the conditions on $\mathbf{A}^{[k]}$ to ensure that we go downhill on local linear model?

$\mathbf{A}^{[k]}$ must be **positive definite**: $\mathbf{x}^\mathsf{T}\mathbf{A}^{[k]}\mathbf{x} \ \forall \ \mathbf{x}$.

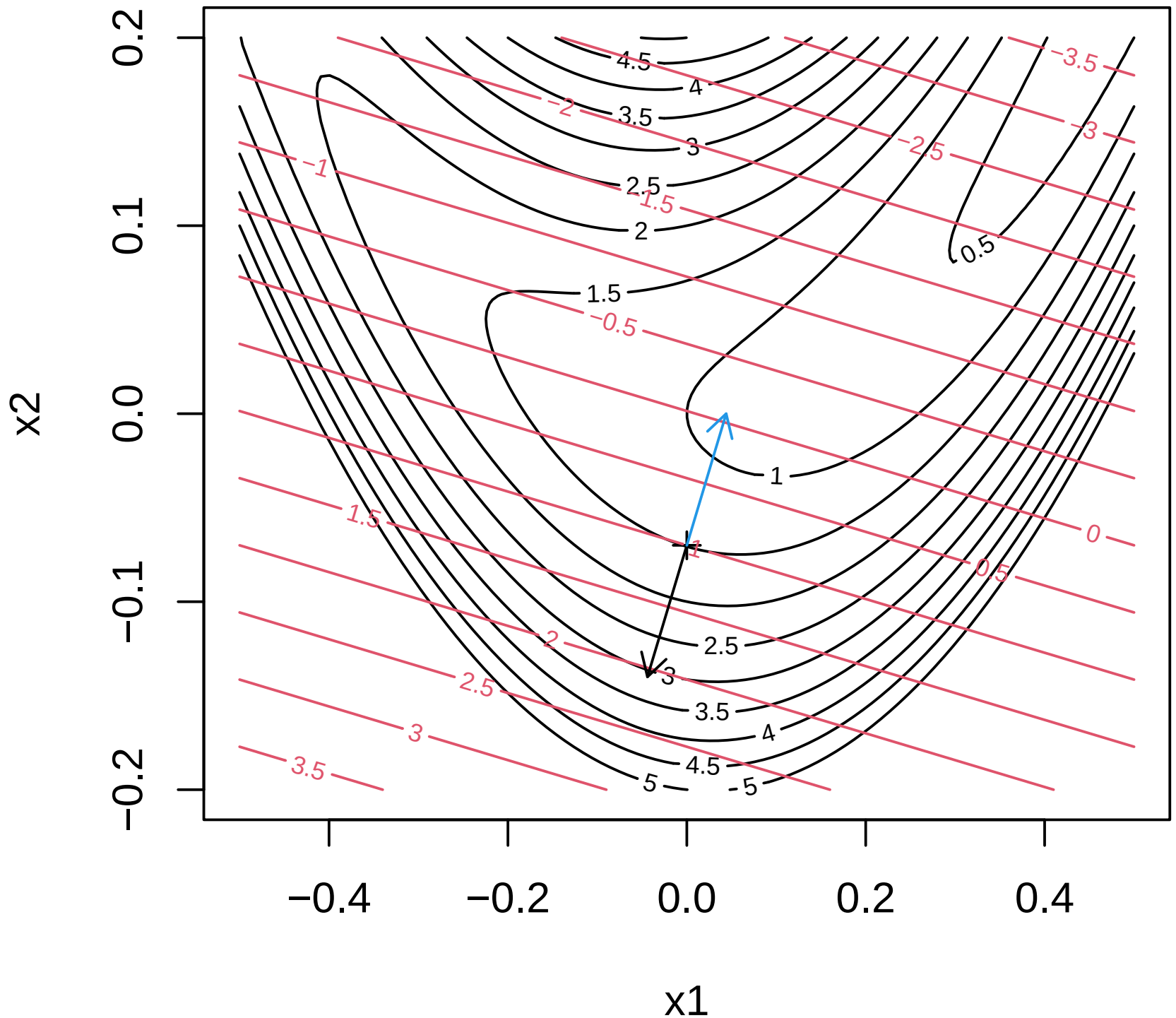
Why does it work? Drop index k and look at local linear model

$$f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^\mathsf{T}\boldsymbol{\Delta}.$$

plug in $\boldsymbol{\Delta} = -\mathbf{A}\nabla f(\boldsymbol{\theta})$

$$f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) - \nabla f(\boldsymbol{\theta})^\mathsf{T}\mathbf{A}\nabla f(\boldsymbol{\theta}) < f(\boldsymbol{\theta}).$$

P.d. $\mathbf{A}$ ensures that it does not rotate $-\nabla f(\boldsymbol{\theta})$ more than $\pi/2$.


x2
0.2
0.1
0.0
−0.1
−0.2
−0.4
−0.2
0.0
0.2
0.4
x1
4.5
4
3.5
3
2.5
2
1.5
1
0.5
5
-5
−3.5
−3
−2.5
−2
−1.5
−1
−0.5
0
0.5
1
1.5
2
2.5
3
3.5

How to find a good preconditioner **A**?

Newton's method chooses **A** via a better local model of $f(\boldsymbol{\theta})$.

Consider 2nd order Taylor approximation

$$f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta} + \frac{1}{2} \boldsymbol{\Delta}^{\mathsf{T}} \nabla^2 f(\boldsymbol{\theta}) \boldsymbol{\Delta}.$$

For $||\boldsymbol{\Delta}|| < 1$ this model is better, with smaller remainder.

How do we use it to choose step?

We minimise the local quadratic model for $f(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\Delta}$.

Differentiating the local model

$$f(\boldsymbol{\theta} + \boldsymbol{\Delta}) \simeq \tilde{f}(\boldsymbol{\theta} + \boldsymbol{\Delta}) = f(\boldsymbol{\theta}) + \nabla f(\boldsymbol{\theta})^{\mathsf{T}} \boldsymbol{\Delta} + \frac{1}{2} \boldsymbol{\Delta}^{\mathsf{T}} \nabla^2 f(\boldsymbol{\theta}) \boldsymbol{\Delta}.$$

w.r.t. $\boldsymbol{\Delta}$ leads to

$$\nabla_{\boldsymbol{\Delta}} \tilde{f}(\boldsymbol{\theta} + \boldsymbol{\Delta}) = \nabla f(\boldsymbol{\theta}) + \nabla^2 f(\boldsymbol{\theta}) \boldsymbol{\Delta}$$

and setting this to zero leads to

$$\boldsymbol{\Delta} = - \left(\nabla^2 f(\boldsymbol{\theta}) \right)^{-1} \nabla f(\boldsymbol{\theta}).$$

Hence k-th iteration is

$$\boldsymbol{\theta}^{[k+1]} \leftarrow \boldsymbol{\theta}^{[k]} - \left(\nabla^2 f(\boldsymbol{\theta}^{[k]})\right)^{-1} \nabla f(\boldsymbol{\theta}^{[k]}).$$

or

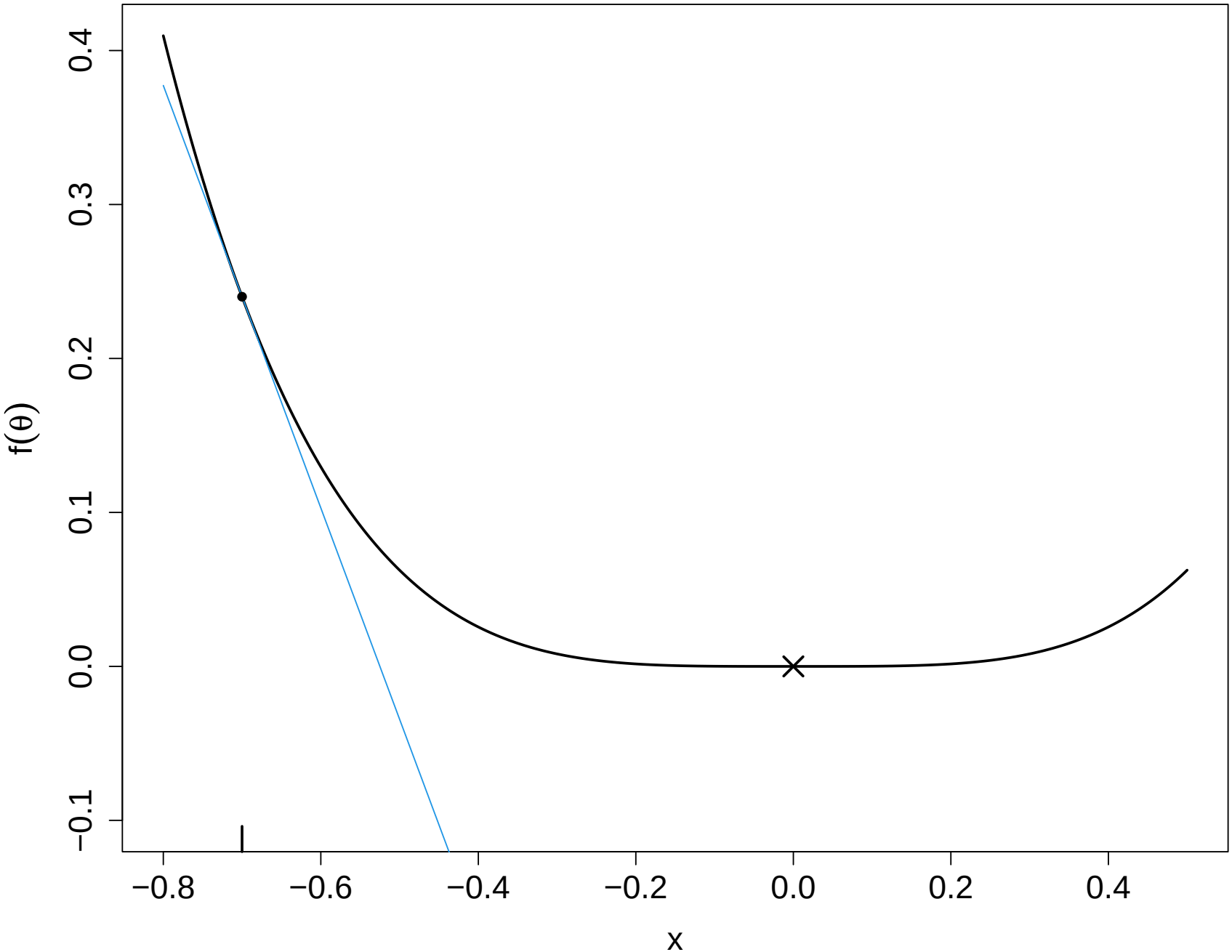
$$\boldsymbol{\theta}^{[k+1]} \leftarrow \boldsymbol{\theta}^{[k]} - \mathbf{A}^{[k]} \nabla f(\boldsymbol{\theta}^{[k]}).$$

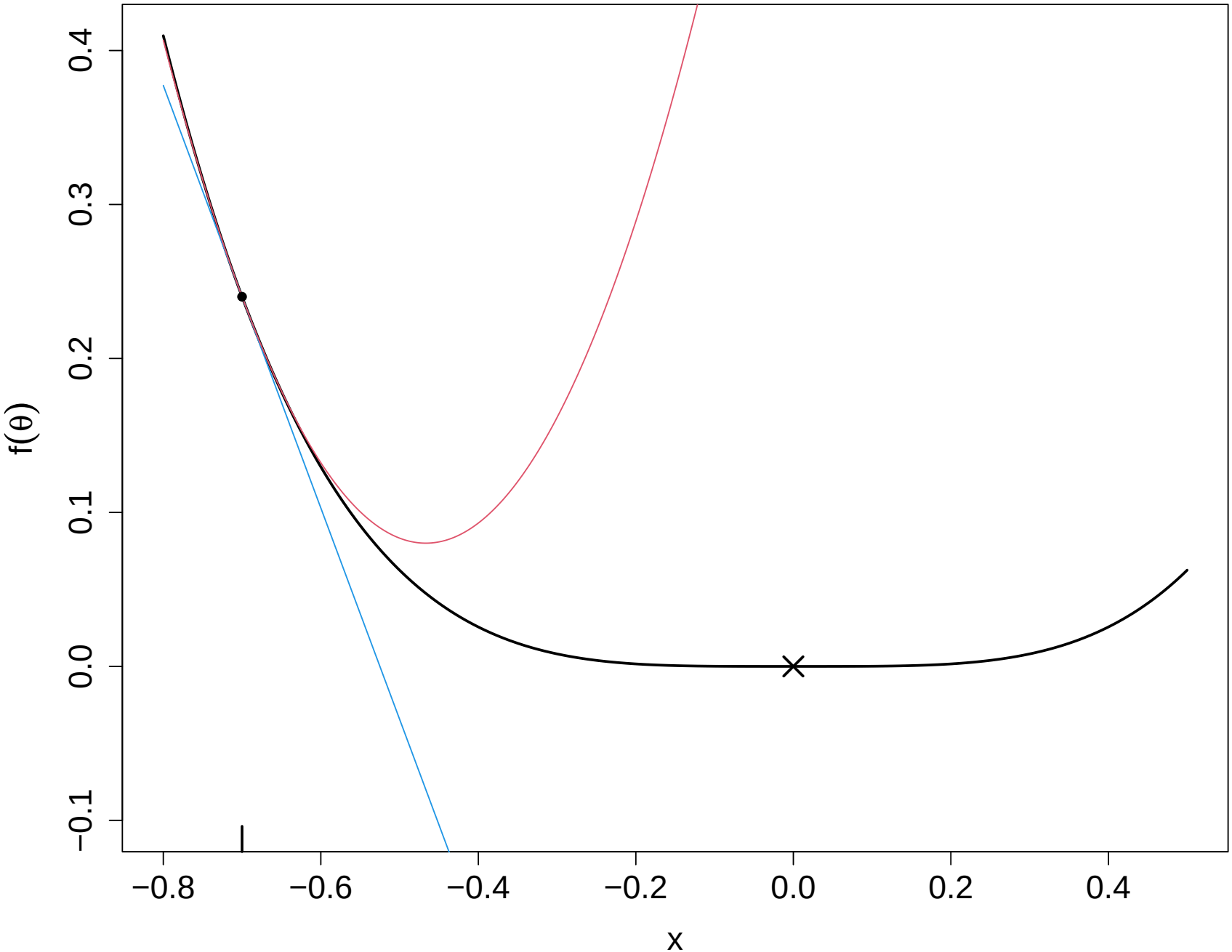
where $\left(\nabla^2 f(\boldsymbol{\theta}^{[k]})\right)^{-1}$ is the preconditioner $\mathbf{A}^{[k]}$.

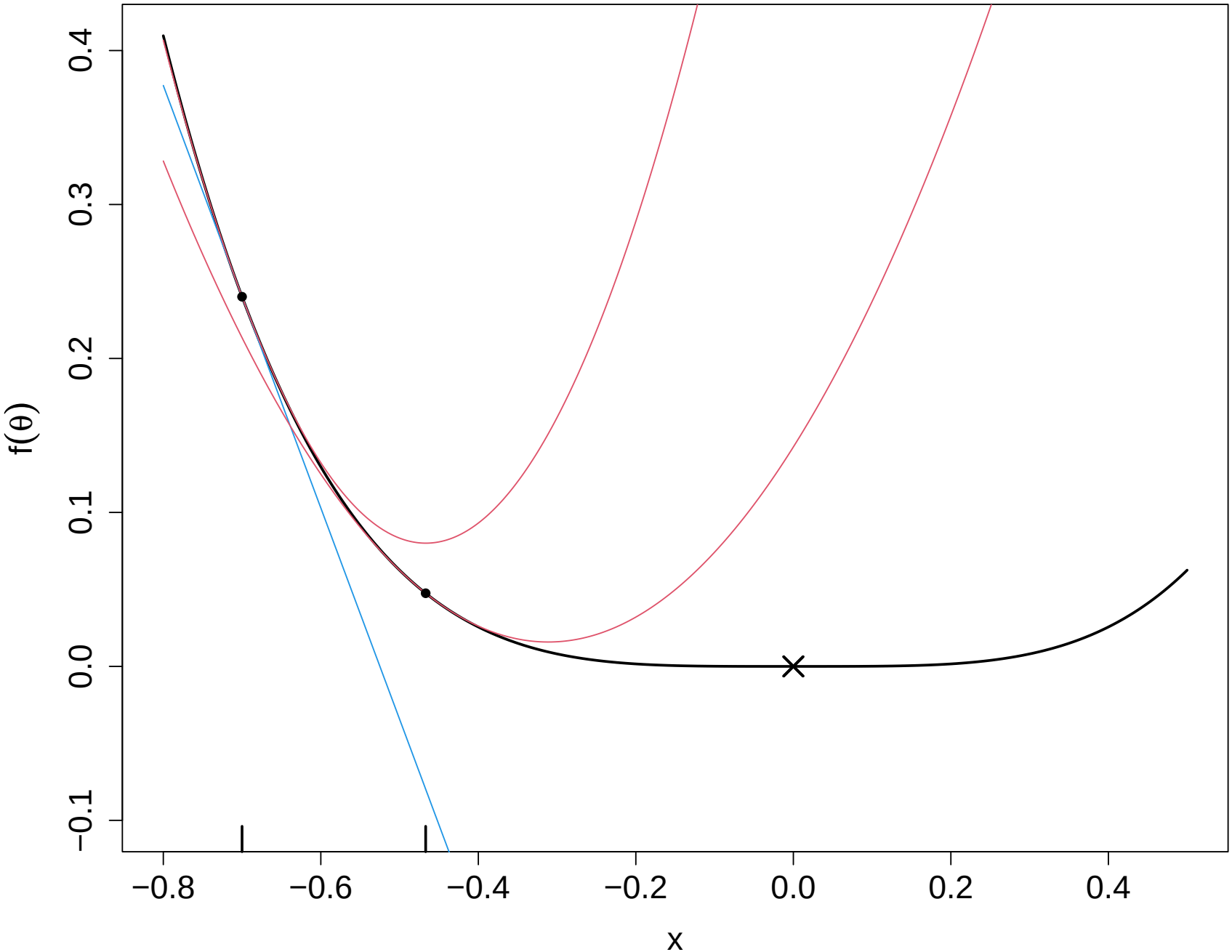
Questions:

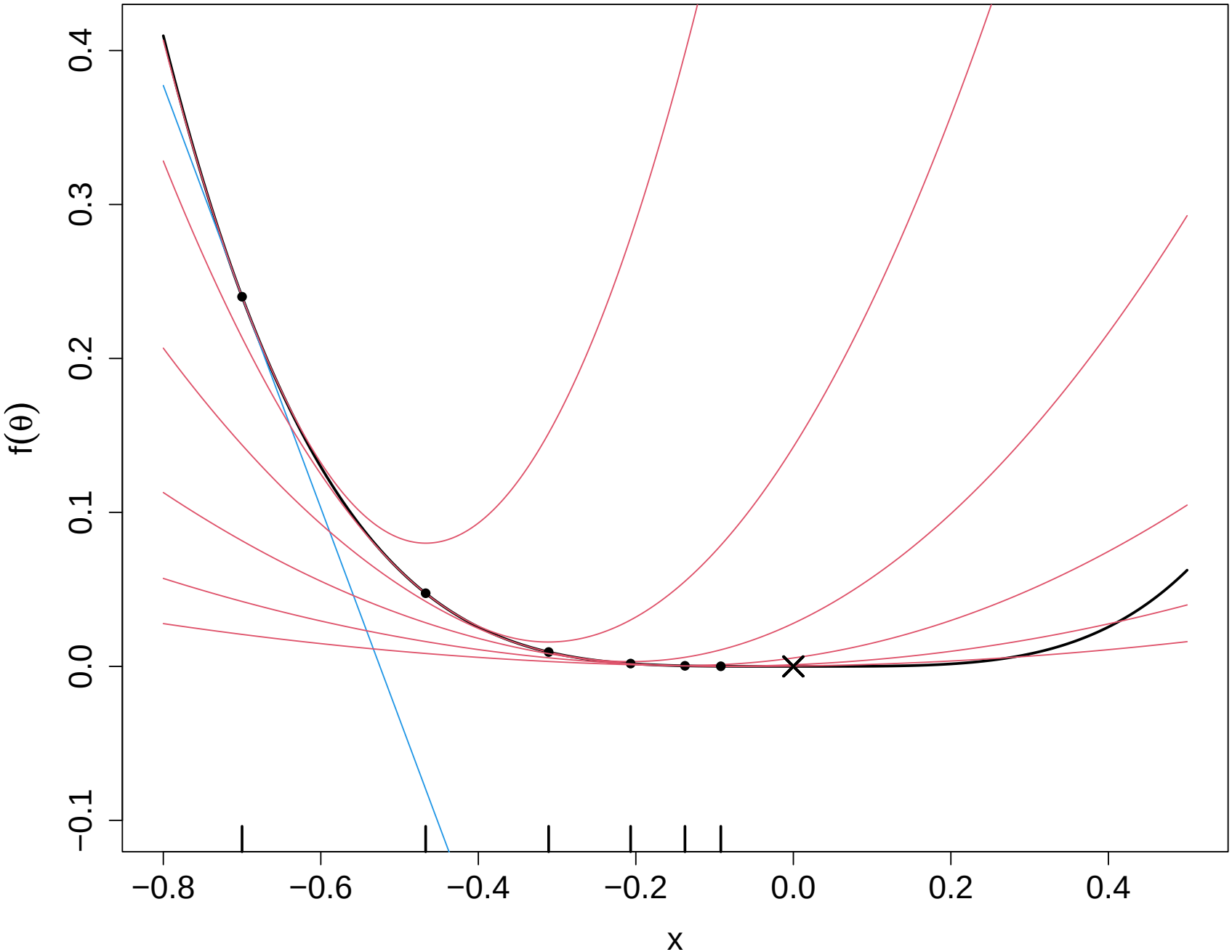
1. Where this the step size α go?
2. What happens if $\mathbf{H} = \nabla^2 f(\boldsymbol{\theta})$ is not positive definite?

Assume 2 is fine, consider 1.


f(θ)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
x


0.4
0.3
0.2
0.1
0.0
−0.1
f(θ)
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
x


f(θ)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
x


f(θ)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
x

So quadratic model suggests iterating using

$$\boldsymbol{\theta}^{[k+1]} \leftarrow \boldsymbol{\theta}^{[k]} - \alpha \boldsymbol{\Delta}^{[k]},$$

with $\alpha = 1$, that is taking full steps.

In practice, Newton should be coupled with a **backtracking** search.

Set $\alpha = \alpha_0 = \mathbf{1}$ (!!) and then **backtrack**:

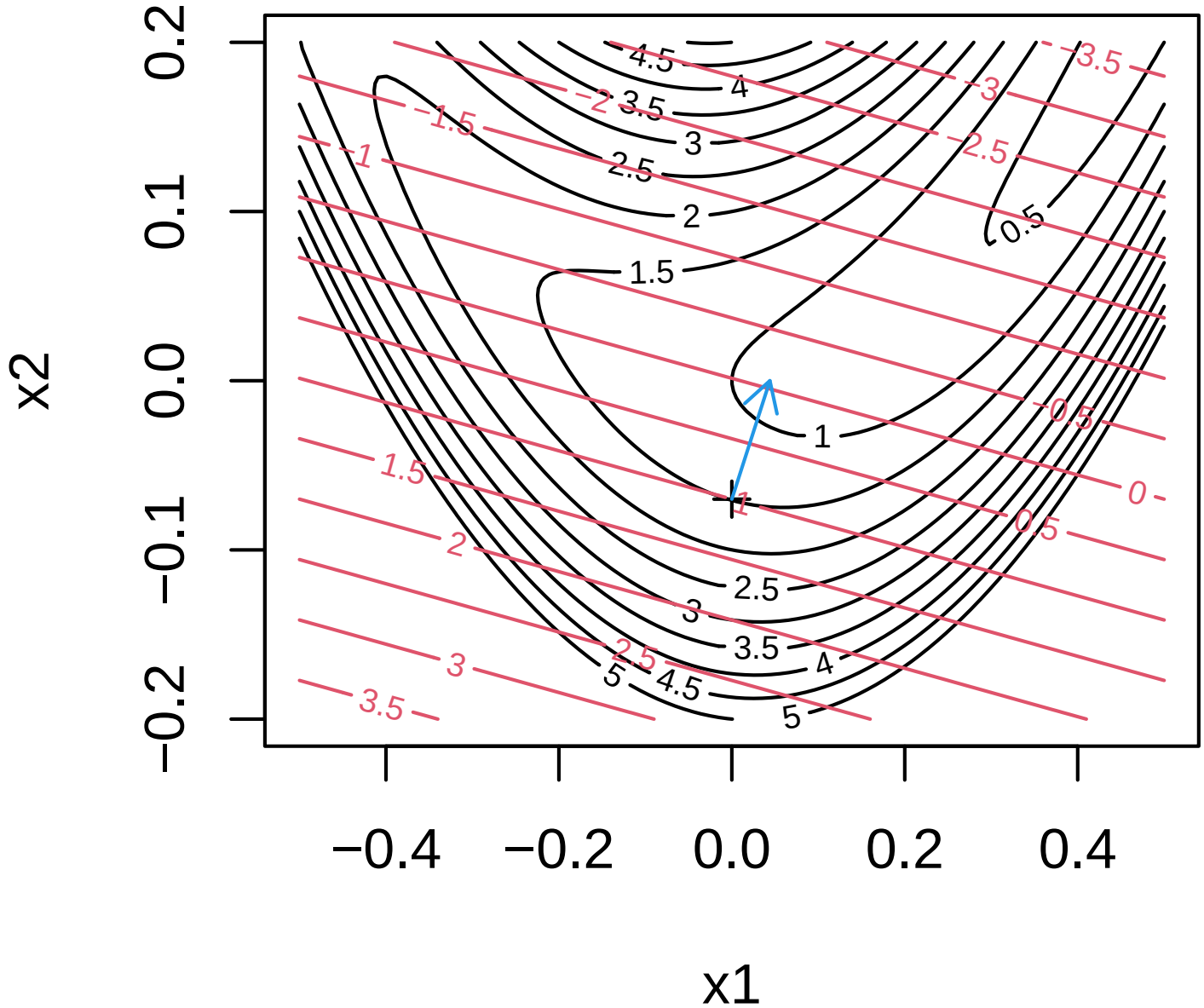
- if $f(\boldsymbol{\theta} + \alpha \boldsymbol{\Delta})$ is not **sufficiently lower** than $f(\boldsymbol{\theta})$ set $\alpha \leftarrow \alpha/2$ and retry.

Sufficiently lower because step might be too long, see Nocedal and Wright 2006, Fig 3.2.

Note that for SD **we did have not a clue** about α_0.

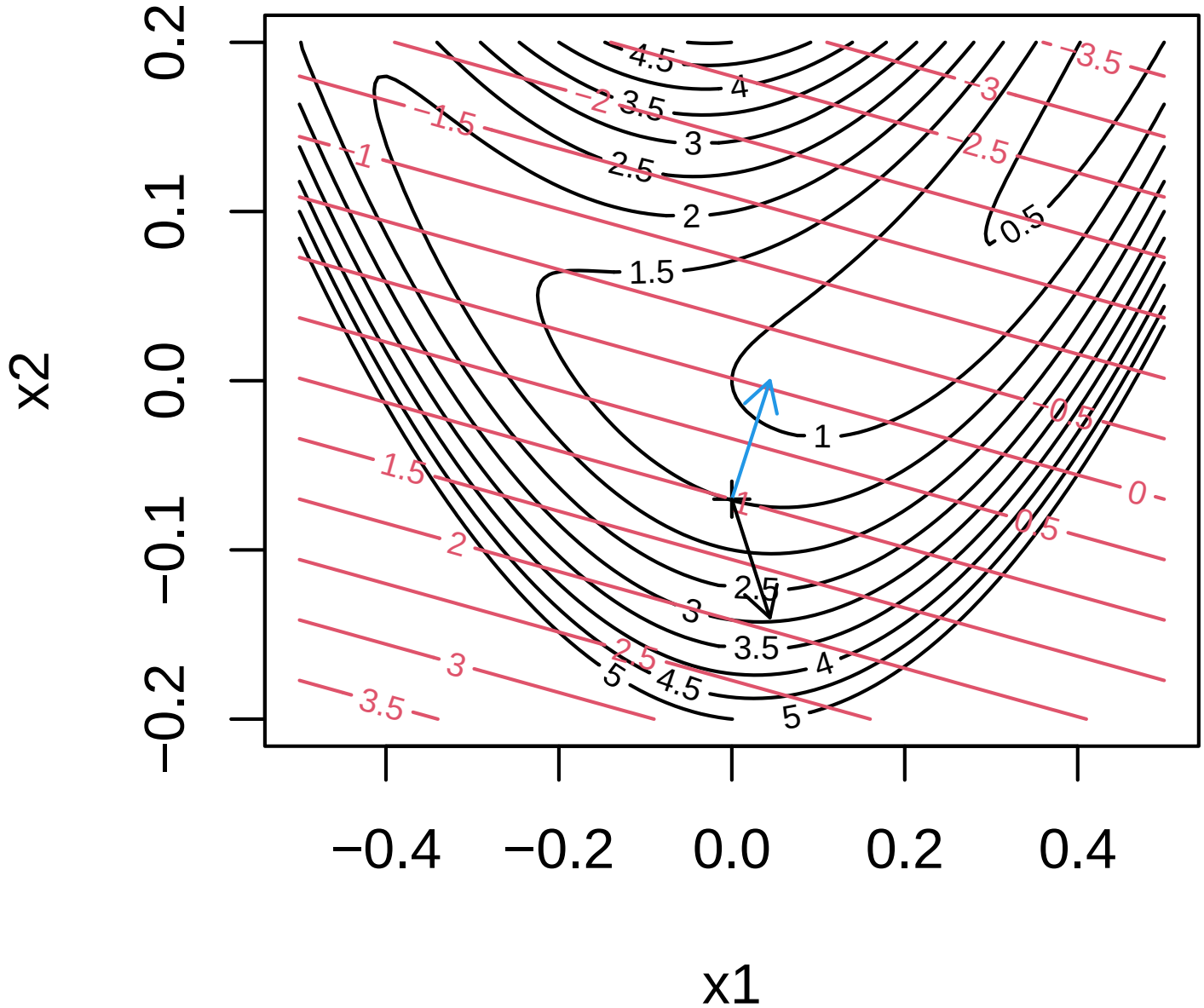
Back to what happens if $\mathbf{H} = \nabla^2 f(\boldsymbol{\theta})$ is not positive definite?

Using **H** directly might sent you uphill!

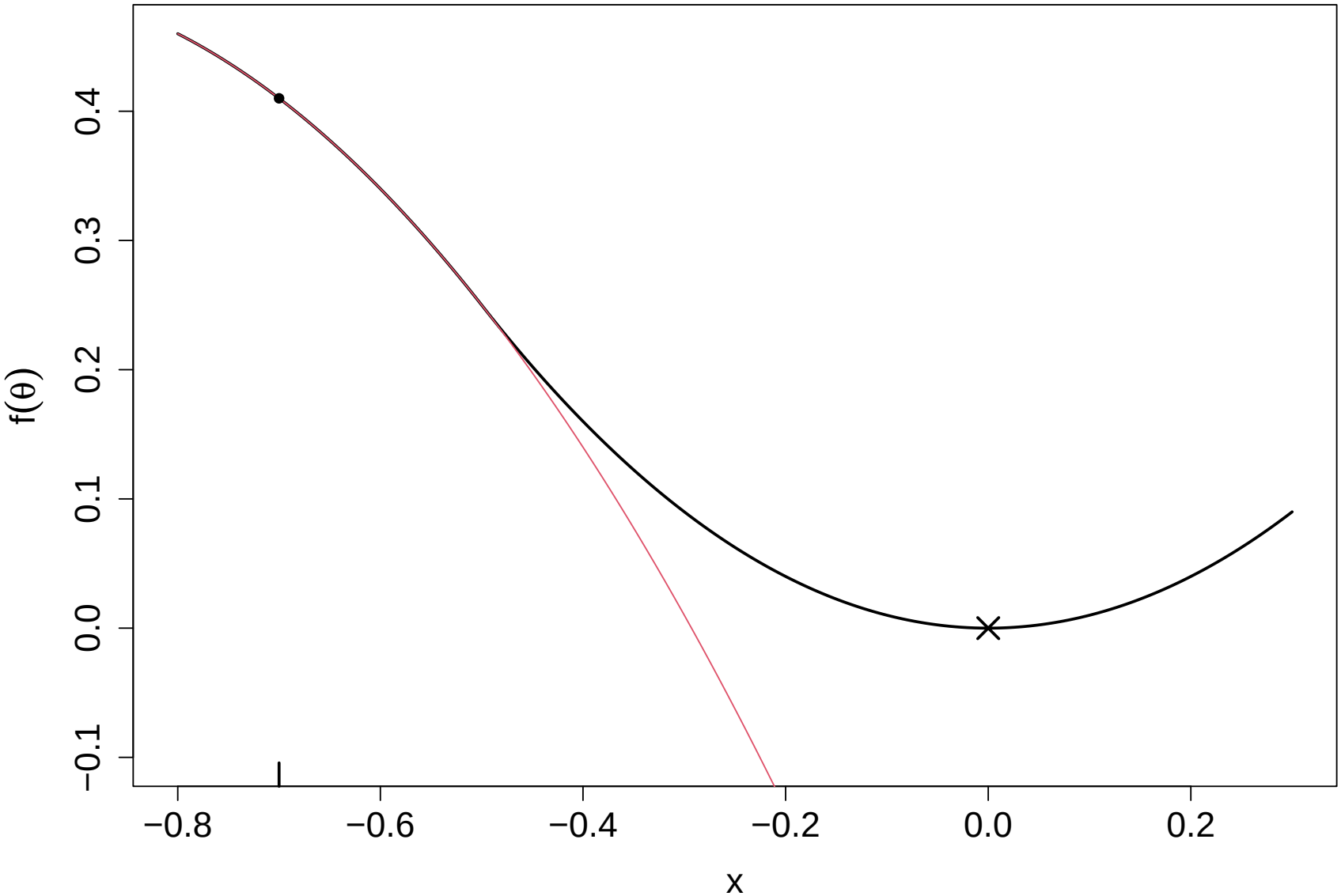


Back to what happens if $\mathbf{H} = \nabla^2 f(\boldsymbol{\theta})$ is not positive definite?

Using **H** directly might sent you uphill!



In 1D dimension, consider:



Here setting $\nabla_{\boldsymbol{\Delta}} \tilde{f}(\boldsymbol{\theta} + \boldsymbol{\Delta}) = \mathbf{0}$ sends us far to the left.

If this happens, don't panic.

Hessian might still be useful, but needs some fixing.

$\mathbf{H}$ might be problematic in certain directions, e.g.

$$\mathbf{H} = \begin{bmatrix} H_{11} & 0 & \dots & 0 \\ 0 & H_{22} & & \\ \vdots & & \ddots & \\ 0 & & & H_{pp} \end{bmatrix}.$$

Here Newton is a scaled SD, where $\mathbf{H}^{-1}$ scales each dimension.

But if $H_{jj} < \epsilon$ (where $0 < \epsilon \ll 1$ is a small tolerance) then either

- $|H_{jj}| < \epsilon$ (massive step) $\rightarrow$ set $H_{jj} \leftarrow \epsilon$
- or $H_{jj} < -\epsilon$ (wrong direction) $\rightarrow$ set $H_{jj} \leftarrow |H_{jj}|$

In second case we assume that $|H_{jj}|$ is informative.

"Targeted intervention" idea extends to full $\mathbf{H}$ (correlated $\boldsymbol{\theta}$):

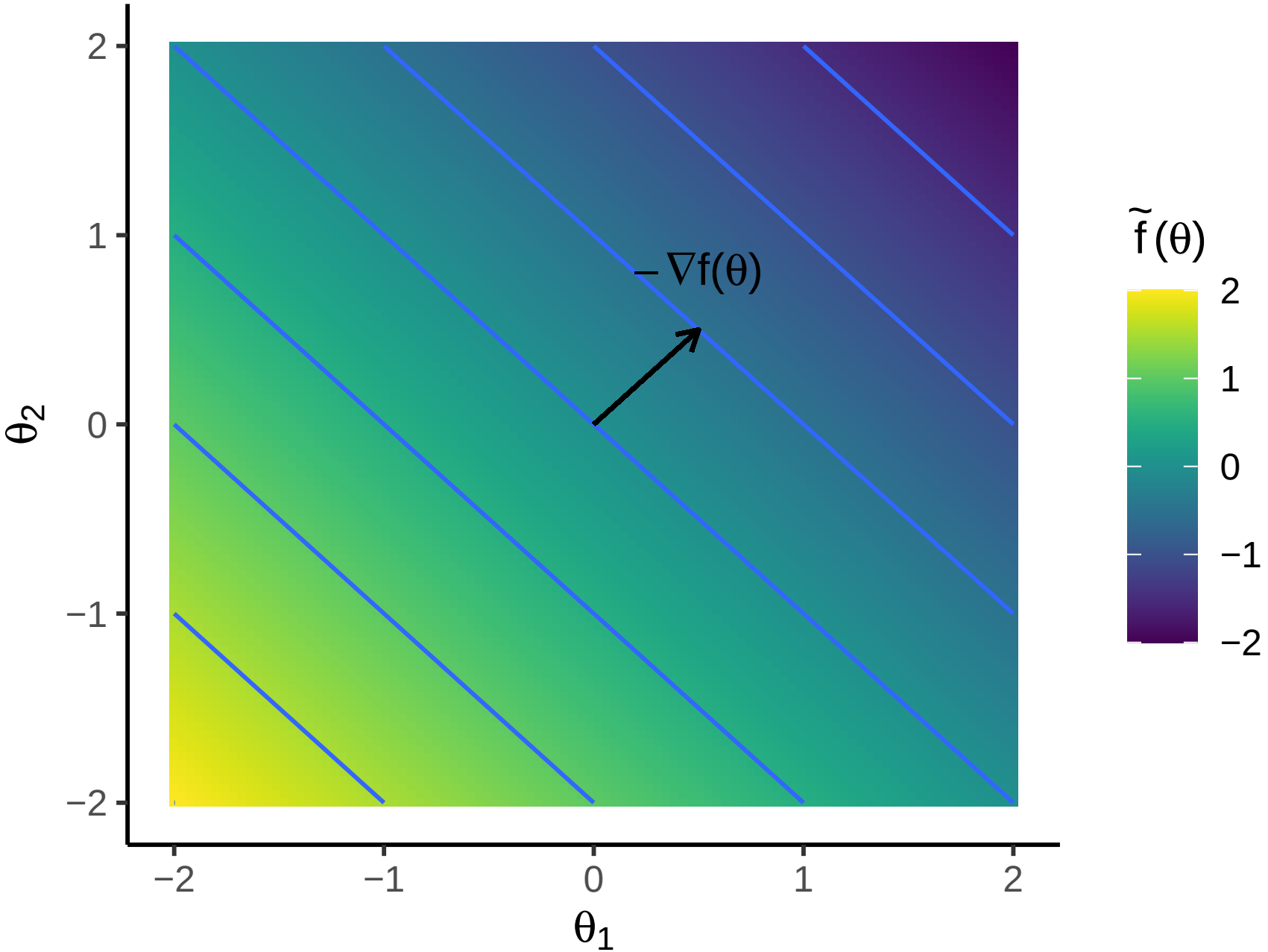
1. Eigen-decompose $\mathbf{H}$.
2. For any eigenvalue $\lambda_j < \epsilon$ then either:
   - $|\lambda_j| < \epsilon$ (indefinite) $\rightarrow$ set $\lambda_j \leftarrow \epsilon$
   - or $\lambda_j < -\epsilon$ (neg. definite) $\rightarrow$ set $\lambda_j \leftarrow |\lambda_j|$

Other solutions:

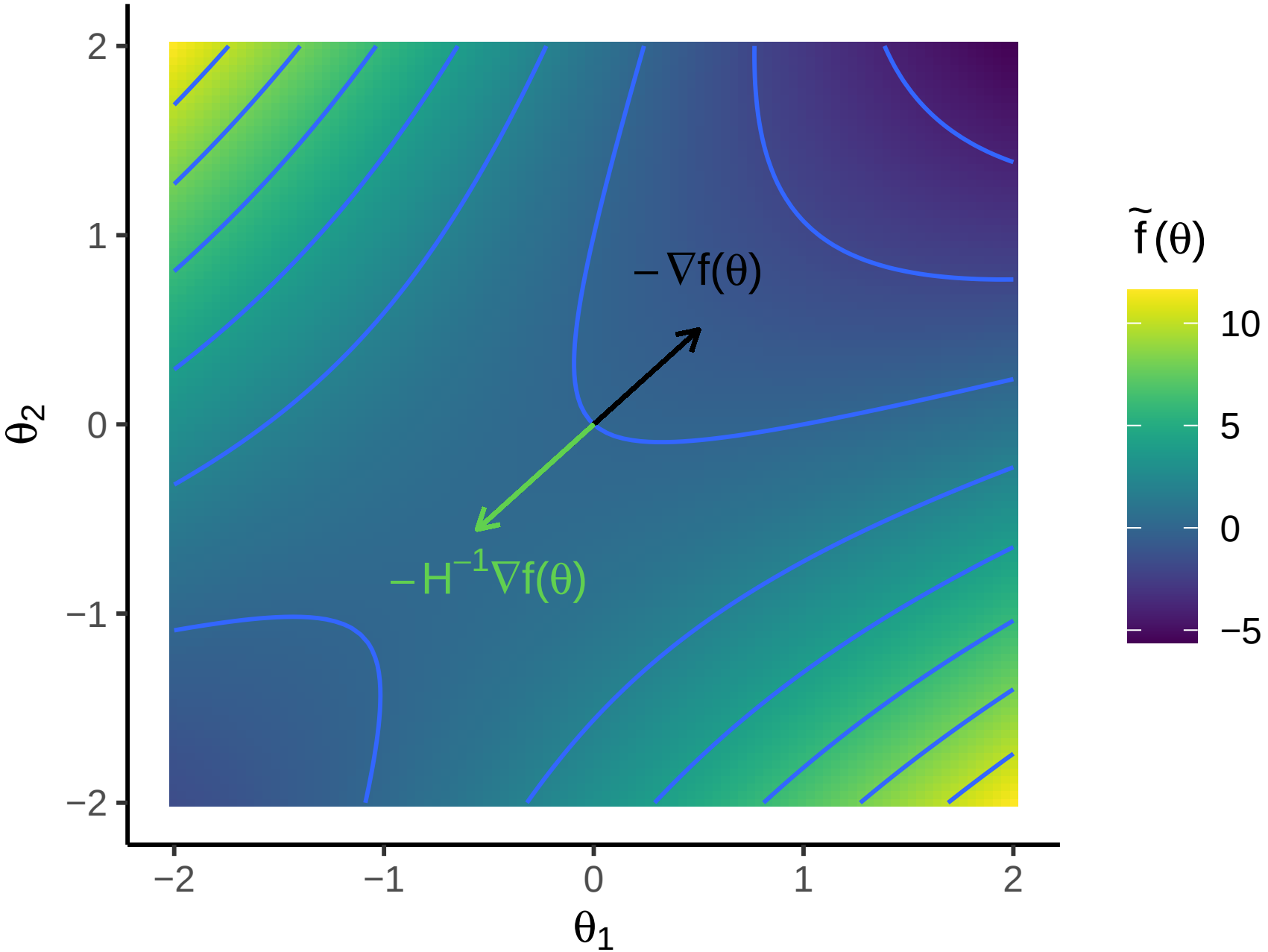
- Set $\mathbf{H} \leftarrow \mathbf{H} + c\,\mathbf{I}$ for large enough $c > 0$
- When $f(\boldsymbol{\theta})$ is neg. log-likelihood use $\mathbb{E}(\mathbf{H})$ (**Fisher scoring**)

How does the "eigen-fix" modify the local model, visually?

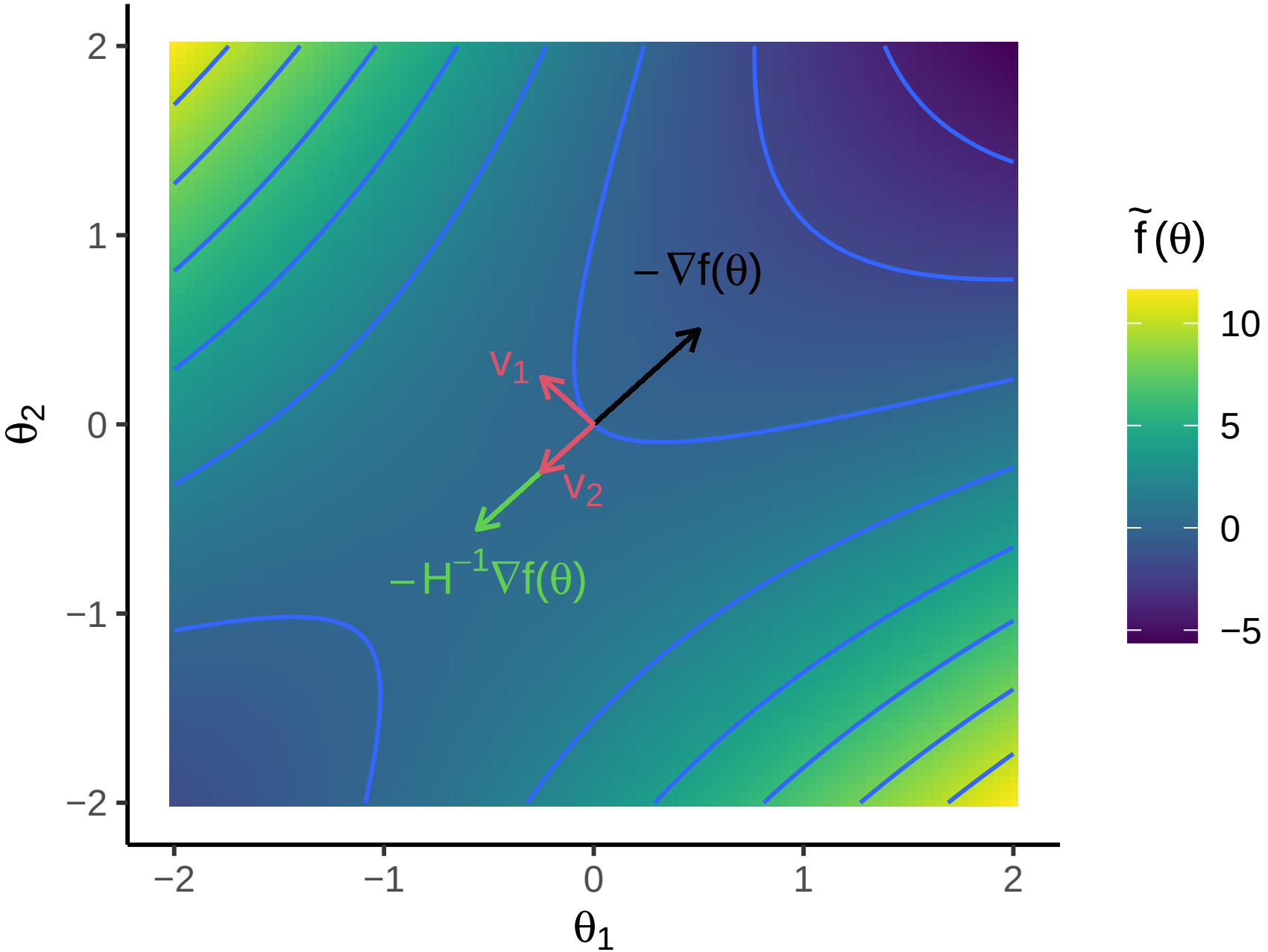
$$\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \; \nabla^2 f(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -1.9 \\ -1.9 & 1 \end{bmatrix}, \; \lambda_1 = 2.9, \; \lambda_2 = -0.9.$$



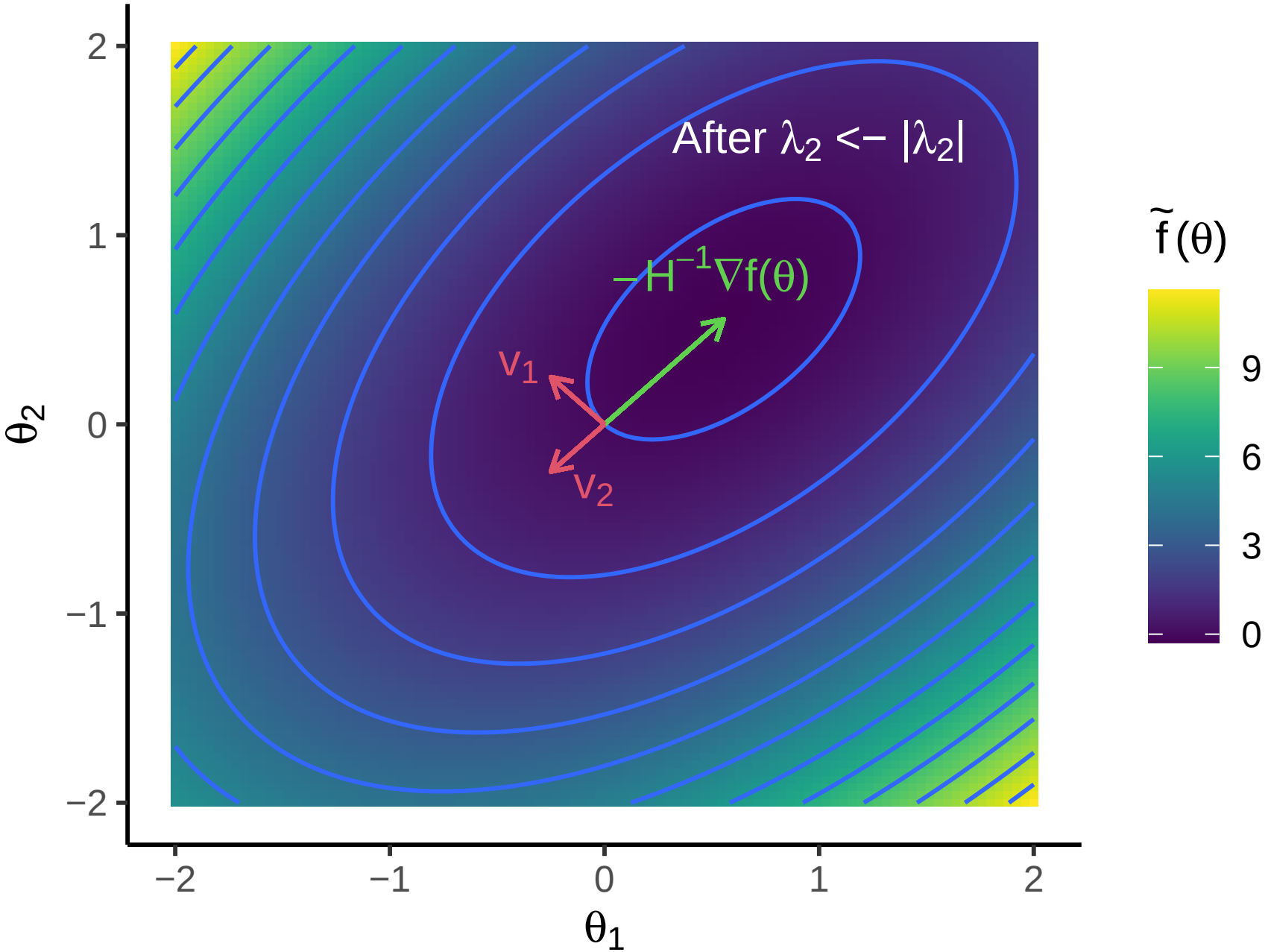
$$\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \; \nabla^2 f(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -1.9 \\ -1.9 & 1 \end{bmatrix}, \; \lambda_1 = 2.9, \; \lambda_2 = -0.9.$$



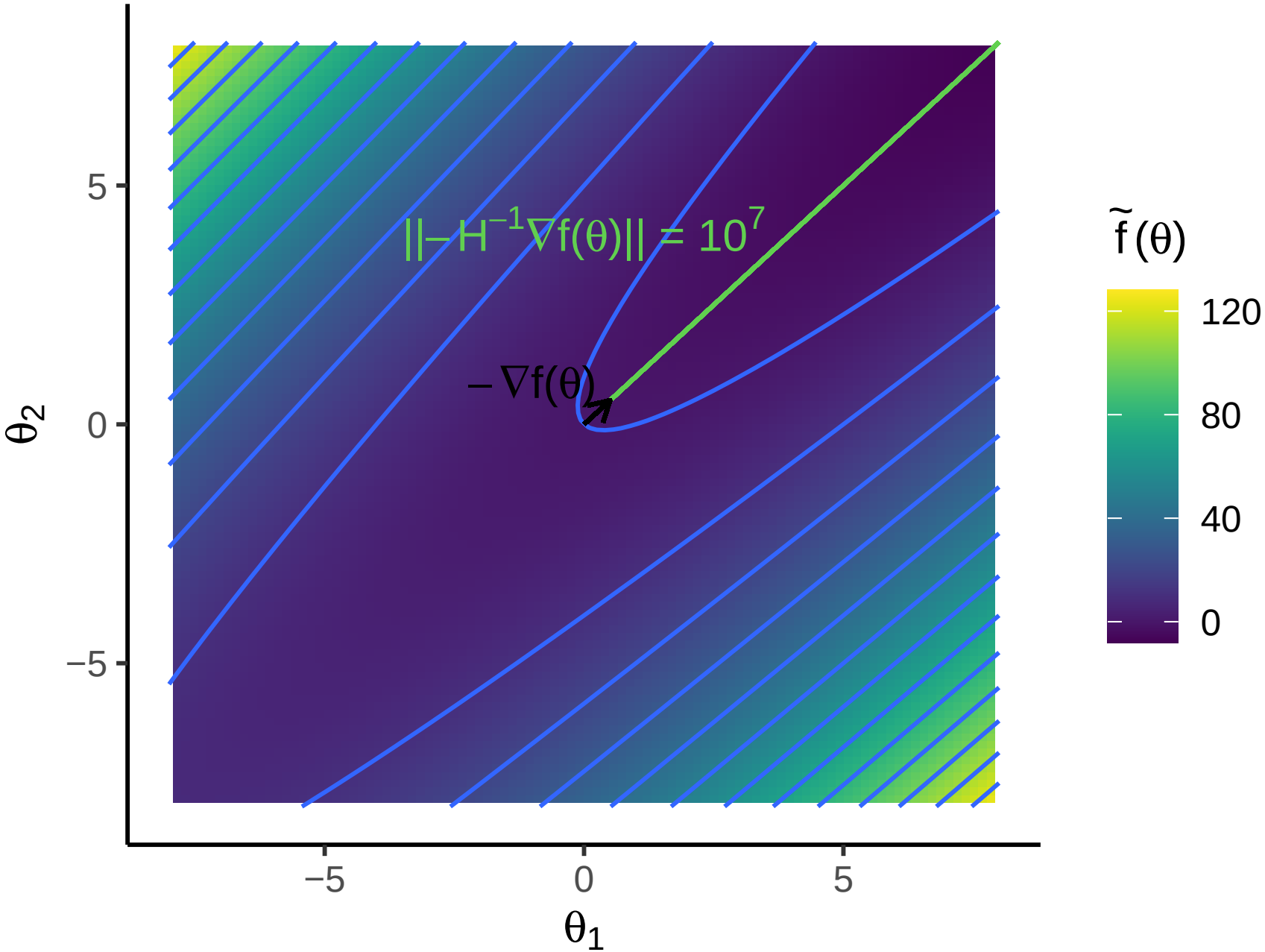
$$\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \; \nabla^2 f(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -1.9 \\ -1.9 & 1 \end{bmatrix}, \; \lambda_1 = 2.9, \; \lambda_2 = -0.9.$$



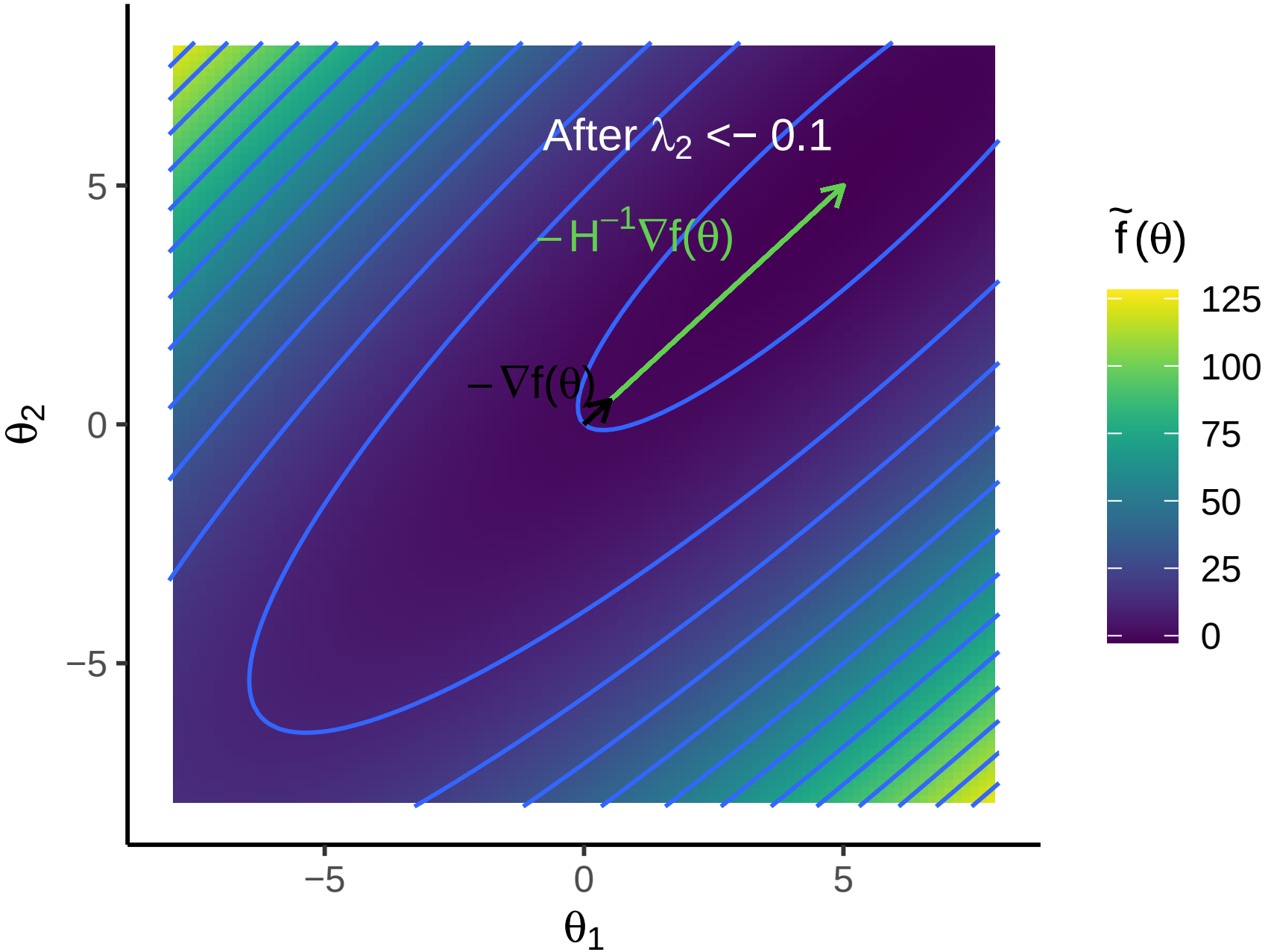
$$\nabla f(\boldsymbol{\theta}) = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \nabla^2 f(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -1.9 \\ -1.9 & 1 \end{bmatrix}, \lambda_1 = 2.9, \lambda_2 = -0.9.$$



If $\nabla^2 f(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -(1-10^{-7}) \\ -(1-10^{-7}) & 1 \end{bmatrix}$ then $\lambda_2 = 10^{-7}$.



If $\nabla^2 f(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -(1-10^{-7}) \\ -(1-10^{-7}) & 1 \end{bmatrix}$ then $\lambda_2 = 10^{-7}$.



We have a "bullet proof" implementation of Newton's method.

At $k = 0$ we start at $\boldsymbol{\theta}^{[0]}$ and iteratively:

1. evaluate $f(\boldsymbol{\theta}^{[k]})$, $\nabla f(\boldsymbol{\theta}^{[k]})$ and $\nabla^2 f(\boldsymbol{\theta}^{[k]})$.
2. Test whether $\boldsymbol{\theta}^{[k]}$ is a minimum and, if it is, stop here.
3. If $\nabla^2 f(\boldsymbol{\theta}^{[k]})$ is not p.d., perturb it to make it p.d..
4. Solve
$$\nabla^2 f(\boldsymbol{\theta}^{[k]})\boldsymbol{\Delta} = -\nabla f(\boldsymbol{\theta}^{[k]}).$$
to find the search direction $\boldsymbol{\Delta}$.
5. If $f(\boldsymbol{\theta}^{[k]} + \boldsymbol{\Delta})$ is not sufficiently lower than $f(\boldsymbol{\theta}^{[k]})$, repeatedly halve $\boldsymbol{\Delta}$ until it is.
6. Set $\boldsymbol{\theta}^{[k+1]} \leftarrow \boldsymbol{\theta}^{[k]} + \boldsymbol{\Delta}$ and back to step 1.

Let's looks at Newton's performance using
`FLtools::optimisation()`

Conclusion on SD and Newton's method

Weaknesses of steepest descent:

- Provides direction but no information on step-length;
- Tends to zig-zag.

Improved by Newton's method via a better local model, but:

1. Needs to compute the Hessian matrix $\mathbf{H}$;
2. Need to ensure its positive definiteness.

Both need line search to guarantee convergence.

Stochastic SD is widely used, stochastic Newton less so.

References

On optimisation methods:

- Nocedal and Wright (2006) Numerical Optimization, 2nd ed.

On problems when fitting chaotic models:

- Fasiolo, M., Pya, N. and Wood, S.N., 2016. A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology. Statistical Science, pp.96-118.