10.34: Numerical Methods Applied to Chemical Engineering

Lecture 11:
Unconstrained Optimization
Newton-Raphson and trust region methods

Recap

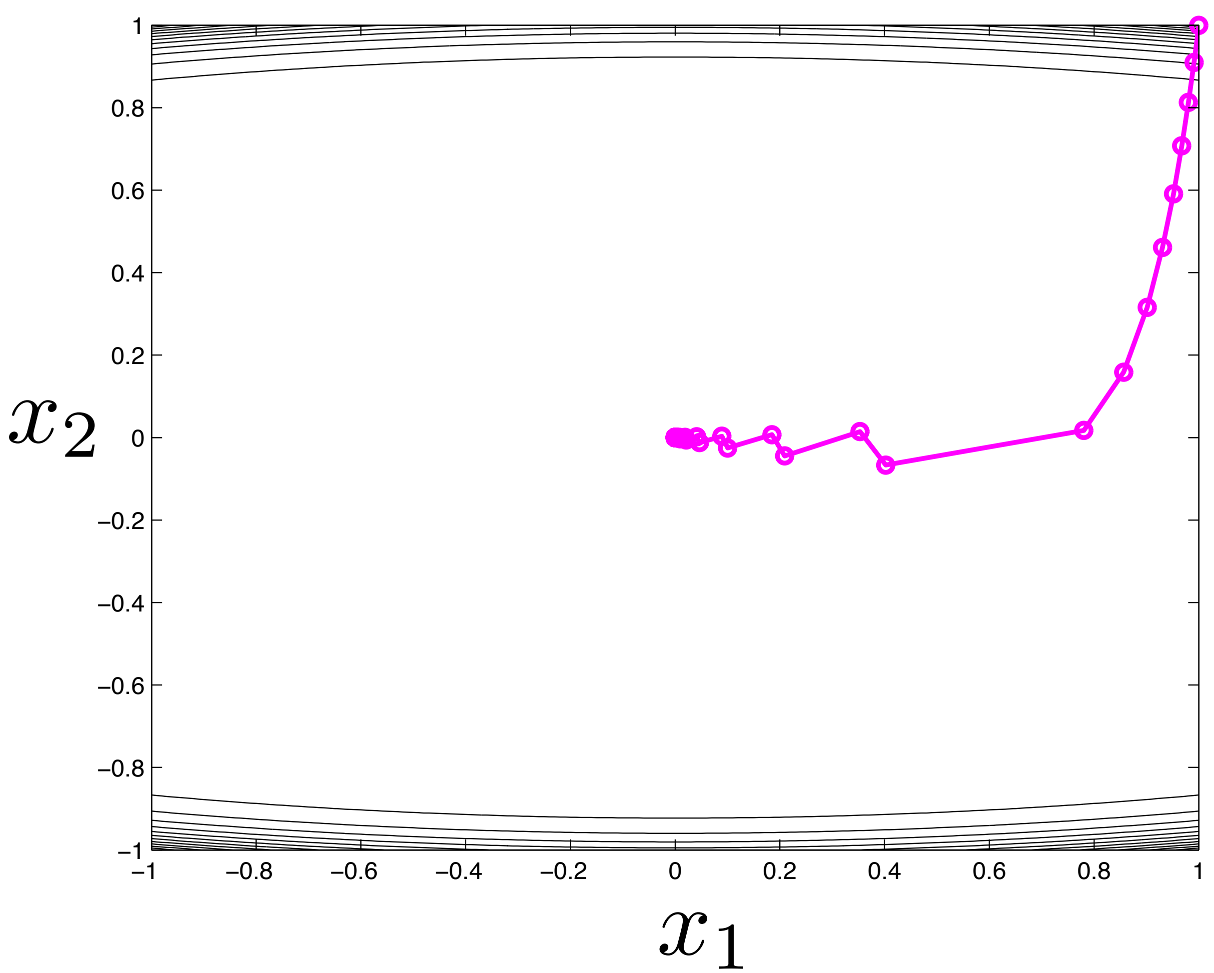
- Optimization
- Steepest descent

Recap

- Method of steepest decent: $\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha_i \mathbf{g}(\mathbf{x}_i)$
  - Estimating an optimal α_i with a Taylor expansion:

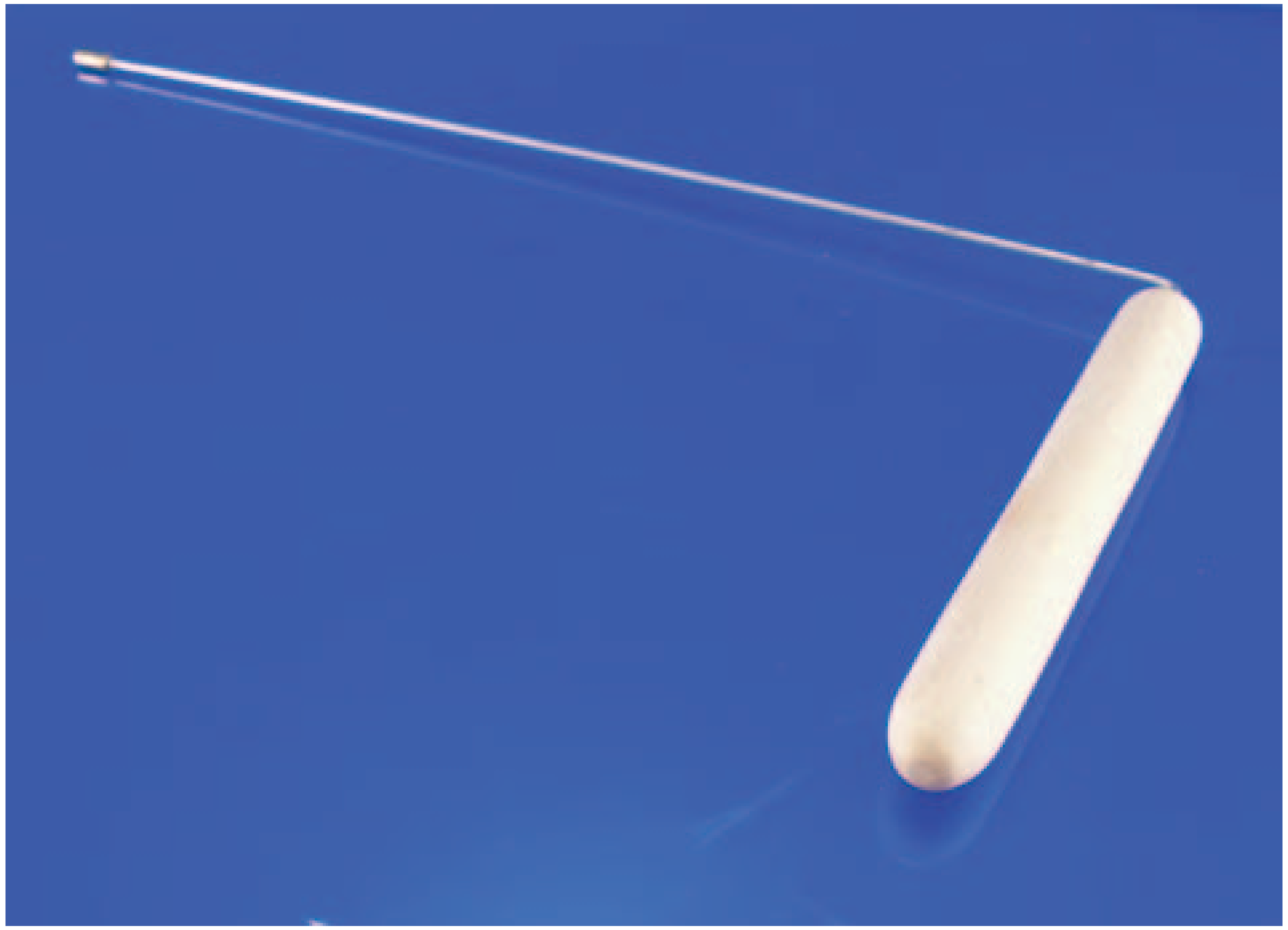
$$f(\mathbf{x}_{i+1}) = f(\mathbf{x}_i) - \alpha_i \mathbf{g}(\mathbf{x}_i)^T \mathbf{g}(\mathbf{x}_i) + \frac{1}{2}\alpha_i^2 \mathbf{g}(\mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i)\mathbf{g}(\mathbf{x}_i) + \ldots$$

  - This is quadratic in α_i, so find the critical point:



$$\alpha_i = \frac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{g}(\mathbf{x}_i)}{\mathbf{g}(\mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i)\mathbf{g}(\mathbf{x}_i)}$$

Recap

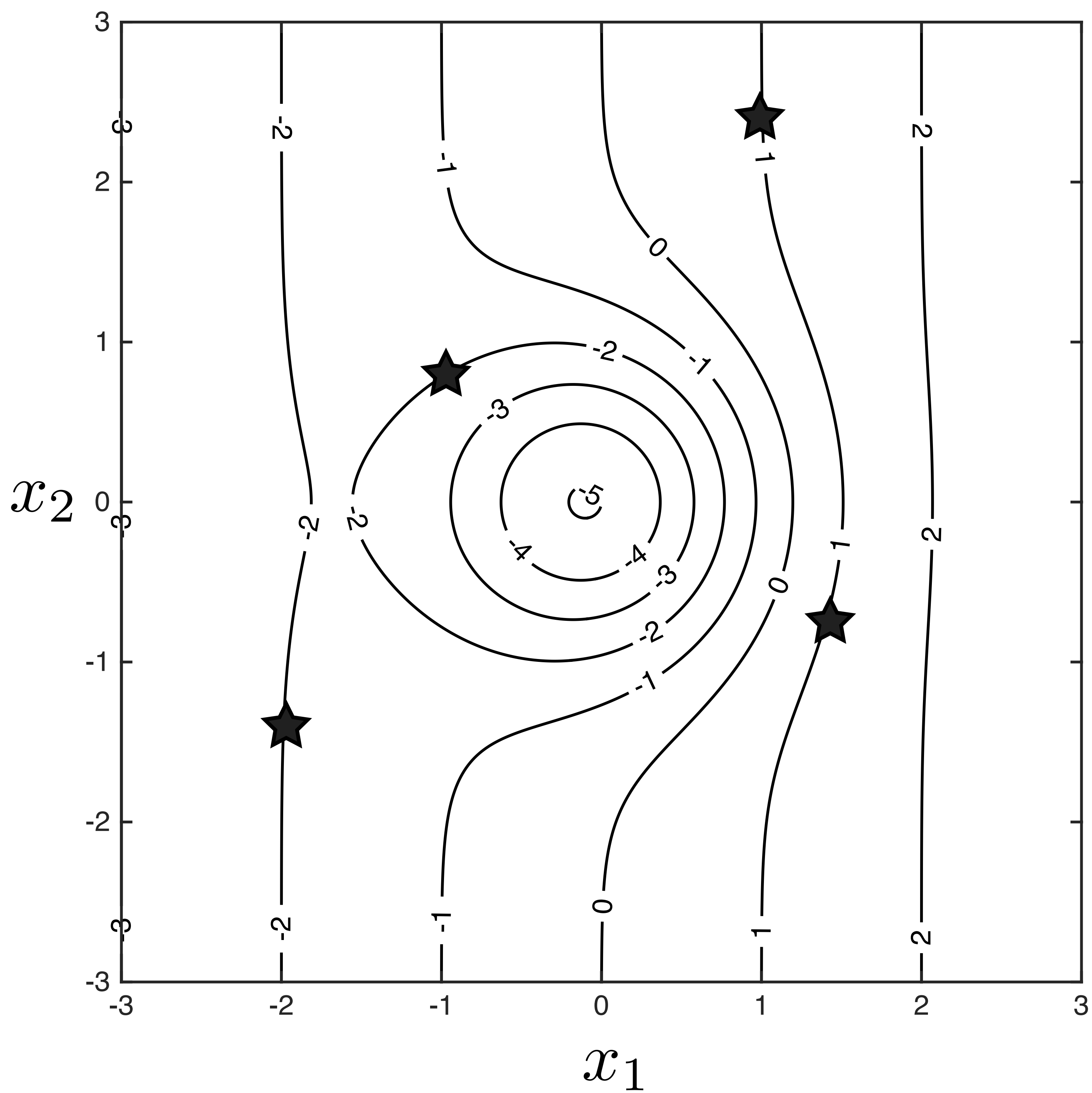


$$m\ddot{\mathbf{x}} = -\gamma\dot{\mathbf{x}} + \mathbf{F}$$

$$m\ddot{\mathbf{x}} = -\gamma\dot{\mathbf{x}} - \nabla U$$

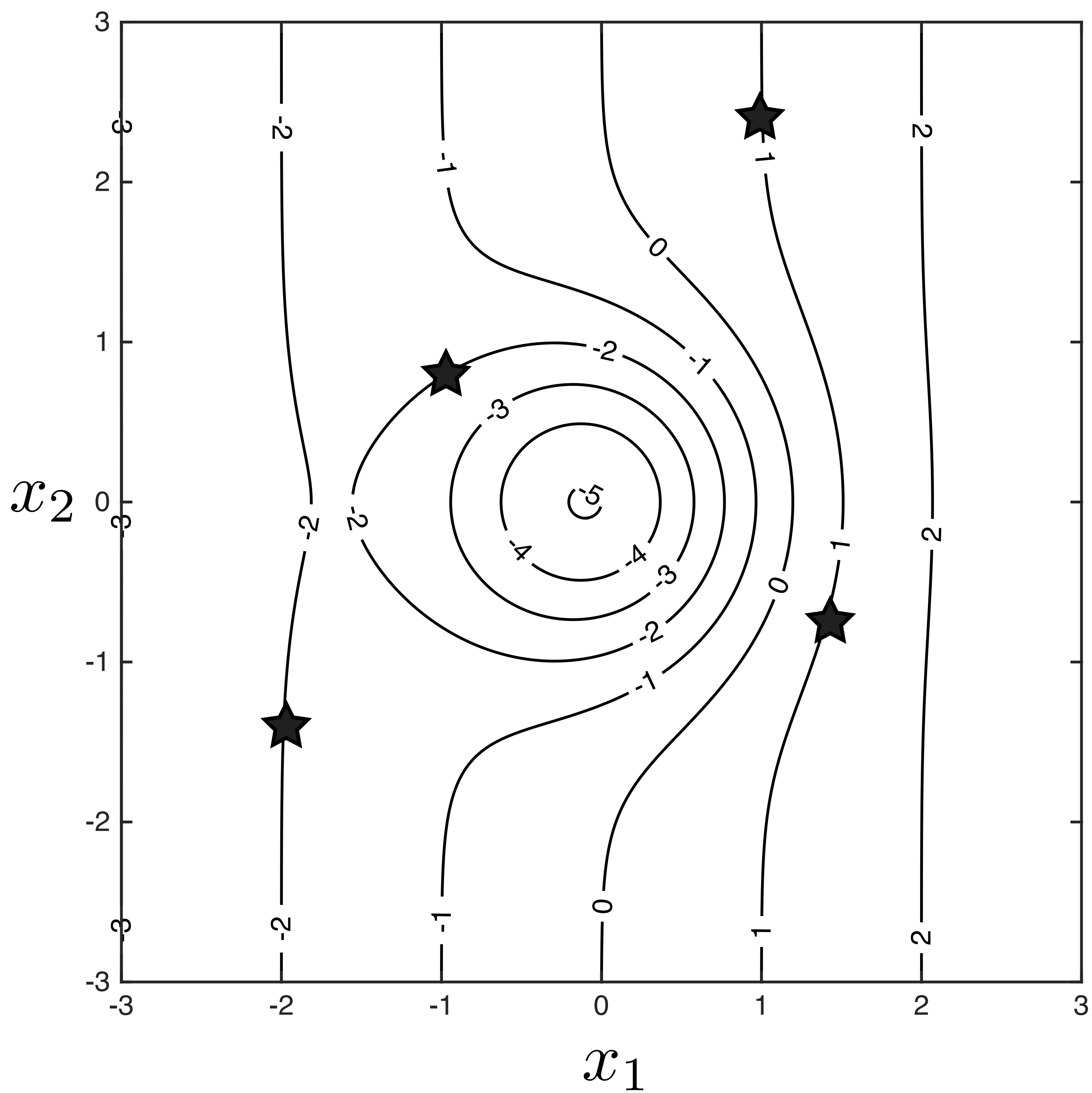
Recap

- Method of steepest decent: $\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha_i \mathbf{g}(\mathbf{x}_i)$



Recap

- Method of steepest decent: $\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha_i \mathbf{g}(\mathbf{x}_i)$



Unconstrained Optimization

- Conjugate gradient method:
  - Consider the minimization of: $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$

    $$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b}$$

    $$\mathbf{H}(\mathbf{x}) = \mathbf{A}$$

    - This has a minimum when?
      - $\mathbf{A}\mathbf{x} = \mathbf{b}$
      - the Hessian, $\mathbf{A}$, is symmetric, positive definite
  - Iterative method: $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$
    - $\mathbf{p}_i$ is a descent dir. but not necessarily the steepest
    - Let's determine the optimal α_i for a given $\mathbf{p}_i$

Unconstrained Optimization

- Conjugate gradient method

$$f(\mathbf{x}_{i+1}) = f(\mathbf{x}_i) + \alpha_i \mathbf{g}(\mathbf{x}_i)^T \mathbf{p}_i + \frac{1}{2}\alpha_i^2 \mathbf{p}_i^T \mathbf{A}\mathbf{p}$$

  - $f(\mathbf{x}_{i+1})$ is quadratic in α_i .
  - $f(\mathbf{x}_{i+1})$ is minimized when $\alpha_i = -\dfrac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{p}_i}{\mathbf{p}_i^T \mathbf{A}\mathbf{p}_i}$

  - For a given direction $\mathbf{p}_i$ there is an optimal step size α_i
  - How can we choose the optimal direction?
    - $f(\mathbf{x}_{i+1})$ is already minimized along $\mathbf{p}_i$: $\mathbf{g}(\mathbf{x}_{i+1})^T \mathbf{p}_i = 0$
    - Can this hold for $f(\mathbf{x}_{i+2})$ also?
      - Let $\mathbf{g}(\mathbf{x}_{i+2})^T \mathbf{p}_i = 0$, then:

$$\left[\mathbf{A}\left(\mathbf{x}_{i+1} + \alpha_{i+1}\mathbf{p}_{i+1}\right) - \mathbf{b}\right]^T \mathbf{p}_i = 0 \Rightarrow \mathbf{p}_{i+1}^T \mathbf{A}\mathbf{p}_i = 0$$

Unconstrained Optimization

- Conjugate gradient method:

$$f(\mathbf{x}_{i+1}) = f(\mathbf{x}_i) + \alpha_i \mathbf{g}(\mathbf{x}_i)^T \mathbf{p}_i + \frac{1}{2}\alpha_i^2 \mathbf{p}_i^T \mathbf{A}\mathbf{p}$$

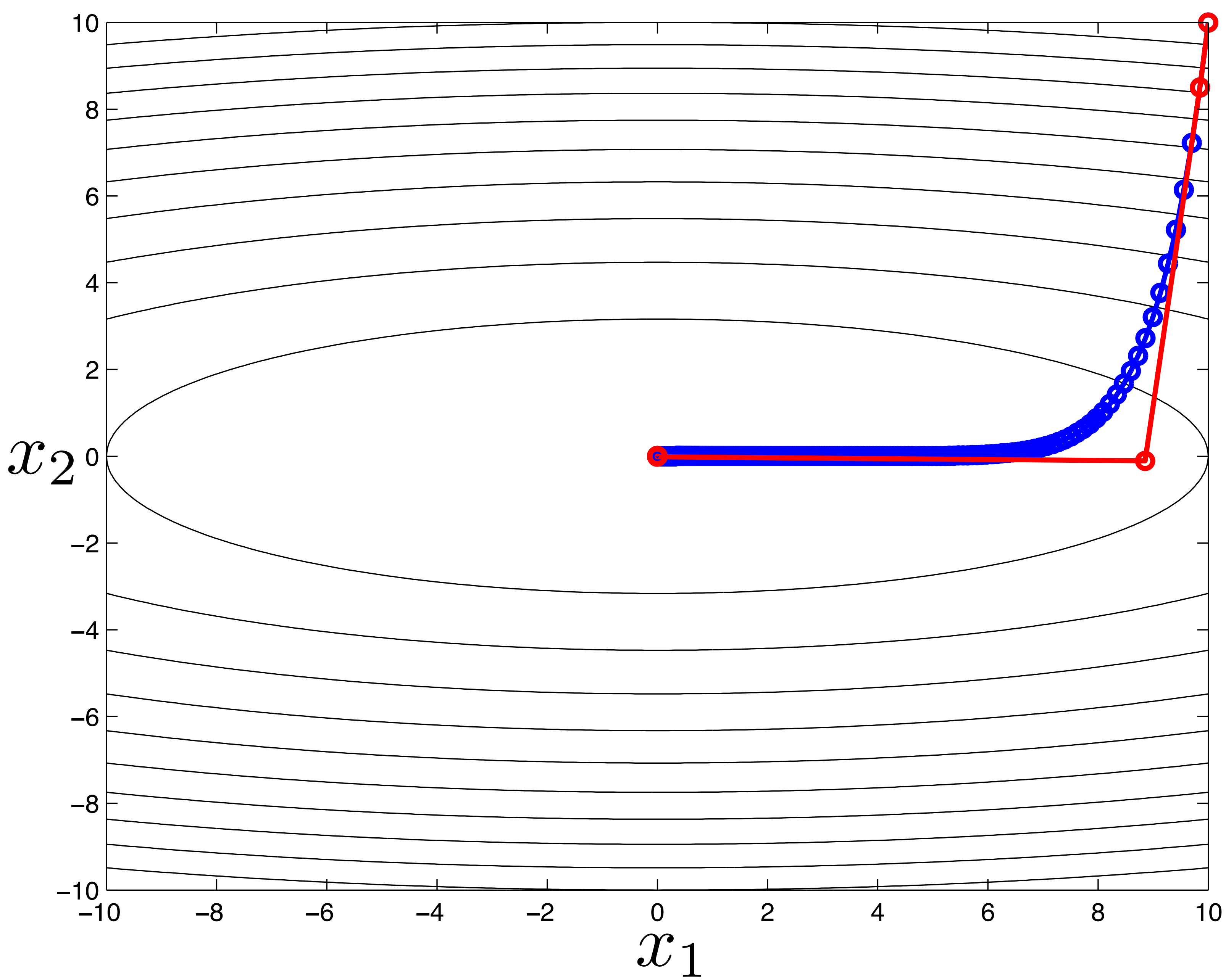
  - $f(\mathbf{x}_{i+1})$ is quadratic in α_i .
  - $f(\mathbf{x}_{i+1})$ is minimized when $\alpha_i = -\dfrac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{p}_i}{\mathbf{p}_i^T \mathbf{A}\mathbf{p}_i}$

- For a given direction $\mathbf{p}_i$ there is an optimal step size α_i
- How can we choose the optimal direction?
  - $f(\mathbf{x}_{i+1})$ is already minimized along $\mathbf{p}_i$: $\mathbf{g}(\mathbf{x}_{i+1})^T \mathbf{p}_i = 0$
  - Can this hold for $f(\mathbf{x}_{i+2})$ also?
    - Let $\mathbf{g}(\mathbf{x}_{i+2})^T \mathbf{p}_i = 0$, then:

$$\mathbf{p}_{i+1} = -\mathbf{g}(\mathbf{x}_{i+1}) + \beta_{i+1}\mathbf{p}_i, \quad \beta_{i+1} = \frac{\mathbf{g}(\mathbf{x}_{i+1})^T \mathbf{A}\mathbf{p}_i}{\mathbf{p}_i^T \mathbf{A}\mathbf{p}_i}$$

Unconstrained Optimization

- Method of steepest decent/conjugate gradient:
  - Example: $f(\mathbf{x}) = x_1^2 + 10x_2^2$ $\quad \mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}, \mathbf{b} = 0$
    - Contours for the function: $\alpha_i = 0.015$ in SD



Unconstrained Optimization

- Conjugate gradient method:
  - Used to solve linear equations with $O(N)$ iterations
  - Requires only the ability to compute the product:
    - The actual matrix is never needed. We only need to compute its action on different vectors, $\mathbf{Ay}$!
  - Only for symmetric, positive definite matrices.
- More sophisticated minimization methods exist for arbitrary matrices.
- Optimization applied to linear equations is the state-of-the-art for solutions of linear equations.

Newton-Raphson

- Finding local minima in unconstrained optimization problems involve solutions of the equation:

$$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = 0$$

  - at minima in $f(\mathbf{x})$

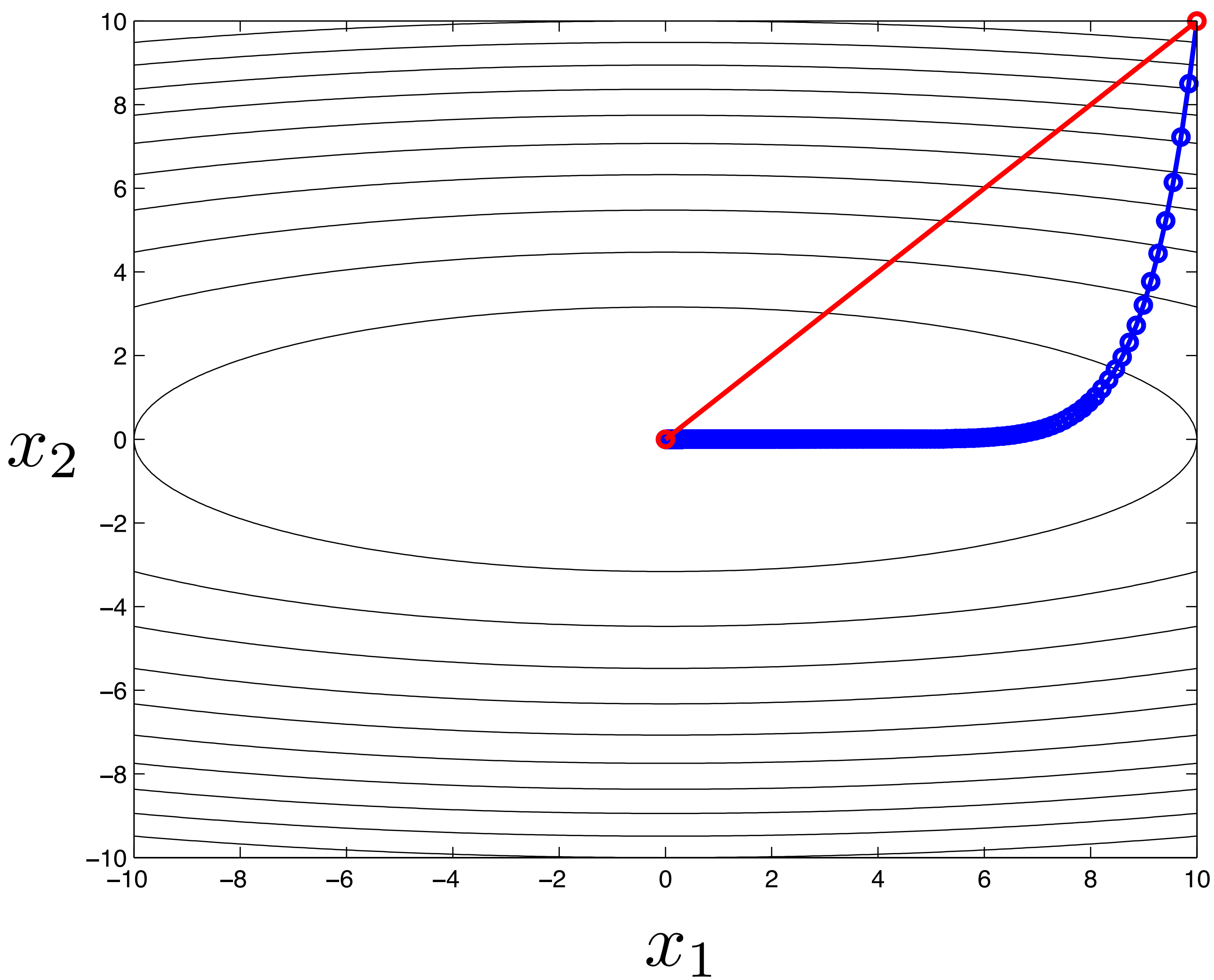
- If we begin close enough to a minimum, can we expect the NR method to converge to that minimum?
  - Yes! NR is locally convergent.
  - Accuracy of the iterates will improve quadratically!

- Newton-Raphson iteration:

- What is the Jacobian of $\mathbf{g}(\mathbf{x})$?

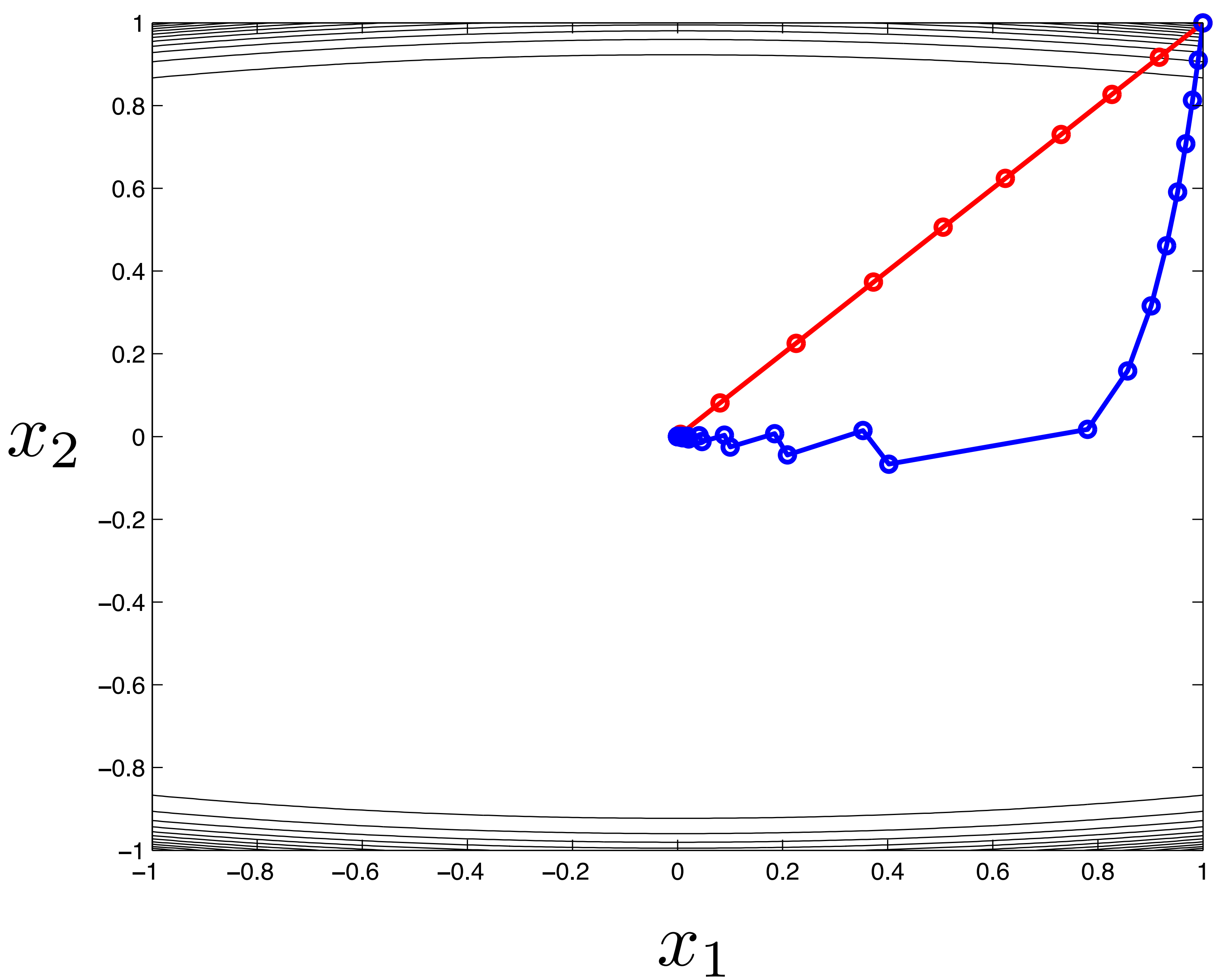
Unconstrained Optimization

- Method of steepest decent/Newton-Raphson:
  - Example: $f(\mathbf{x}) = x_1^2 + 10x_2^2$
    - Contours for the function: $\alpha_i = 0.015$



Unconstrained Optimization

- Method of steepest decent/Newton-Raphson:
  - Example: $\log f(\mathbf{x}) = x_1^2 + 10x_2^2$
    - Contours for the function: $\alpha_i = \dfrac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{g}(\mathbf{x}_i)}{\mathbf{g}(\mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i) \mathbf{g}(\mathbf{x}_i)}$

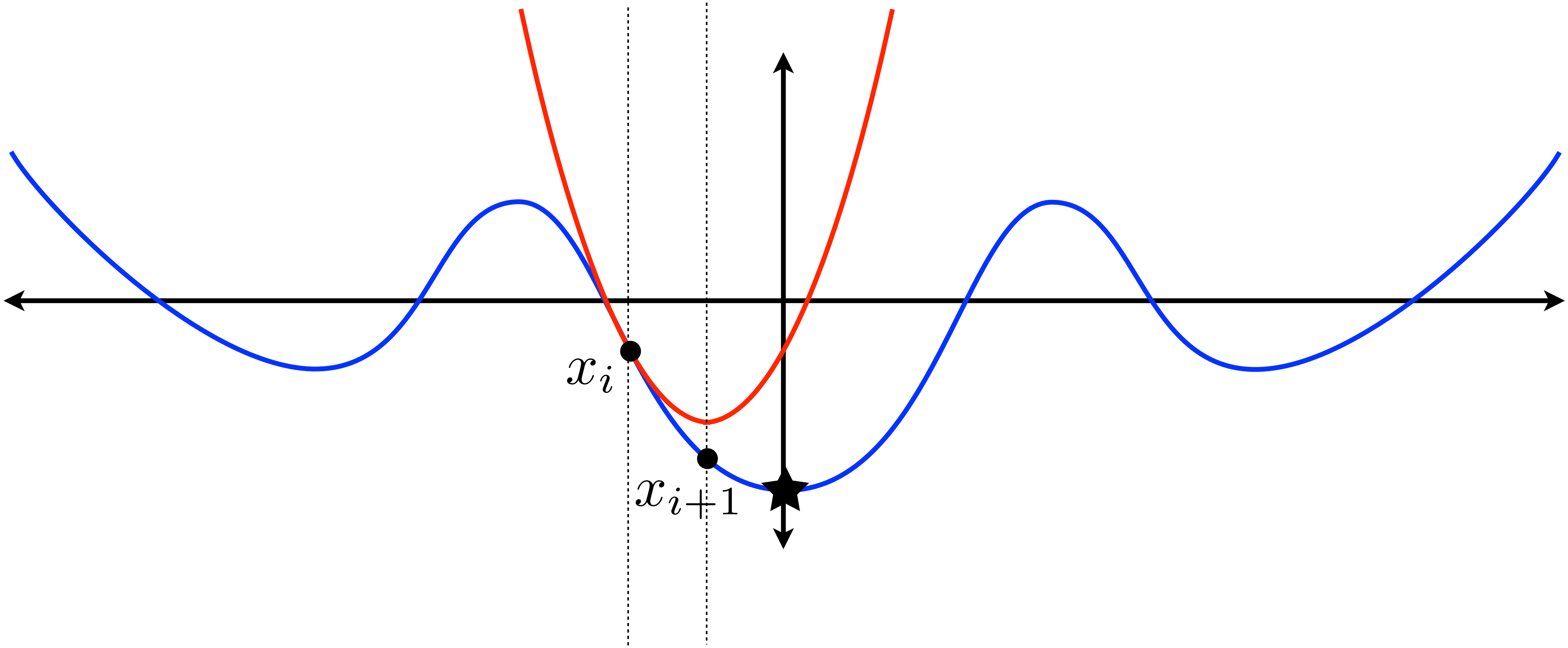


Newton-Raphson

- Compare:
  - Optimized steepest decent:
    - $\mathbf{x}_{i+1} = \mathbf{x}_i - \alpha_i \mathbf{g}(\mathbf{x}_i)$ with $\alpha_i = \dfrac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{g}(\mathbf{x}_i)}{\mathbf{g}(\mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i) \mathbf{g}(\mathbf{x}_i)}$
  - Newton-Raphson:
    - $\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{H}(\mathbf{x}_i)^{-1} \mathbf{g}(\mathbf{x}_i)$
  - What is the difference?
- What are the strengths of Newton-Raphson?
- What are the weaknesses of Newton-Raphson?
- What are the strengths of steepest descent?
- What are the weaknesses of steepest decent?

Trust-Region Methods

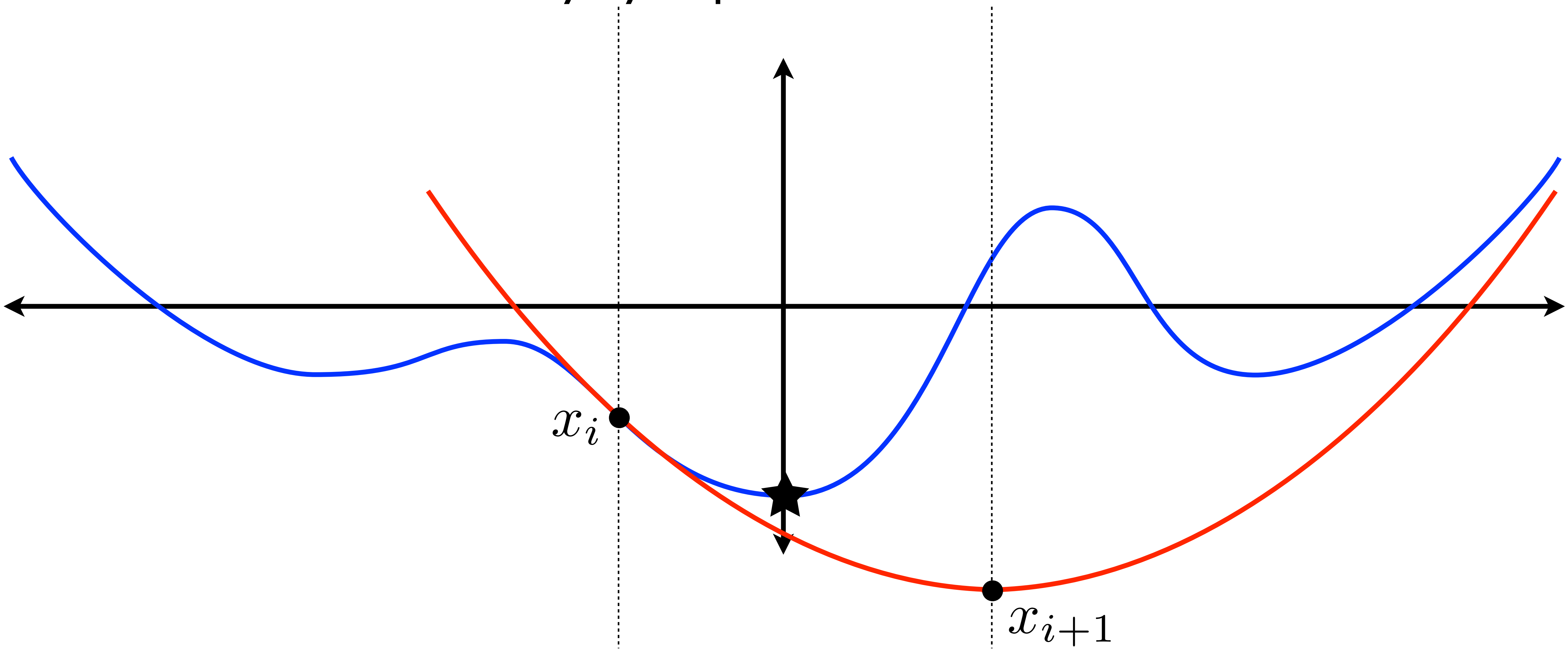
- Both Newton-Raphson and the optimized steepest descent methods assume the objective function can be described locally by a quadratic function.



- That quadratic approximation may be good or bad

Trust-Region Methods

- Both Newton-Raphson and the optimized steepest descent methods assume the objective function can be described locally by a quadratic function.

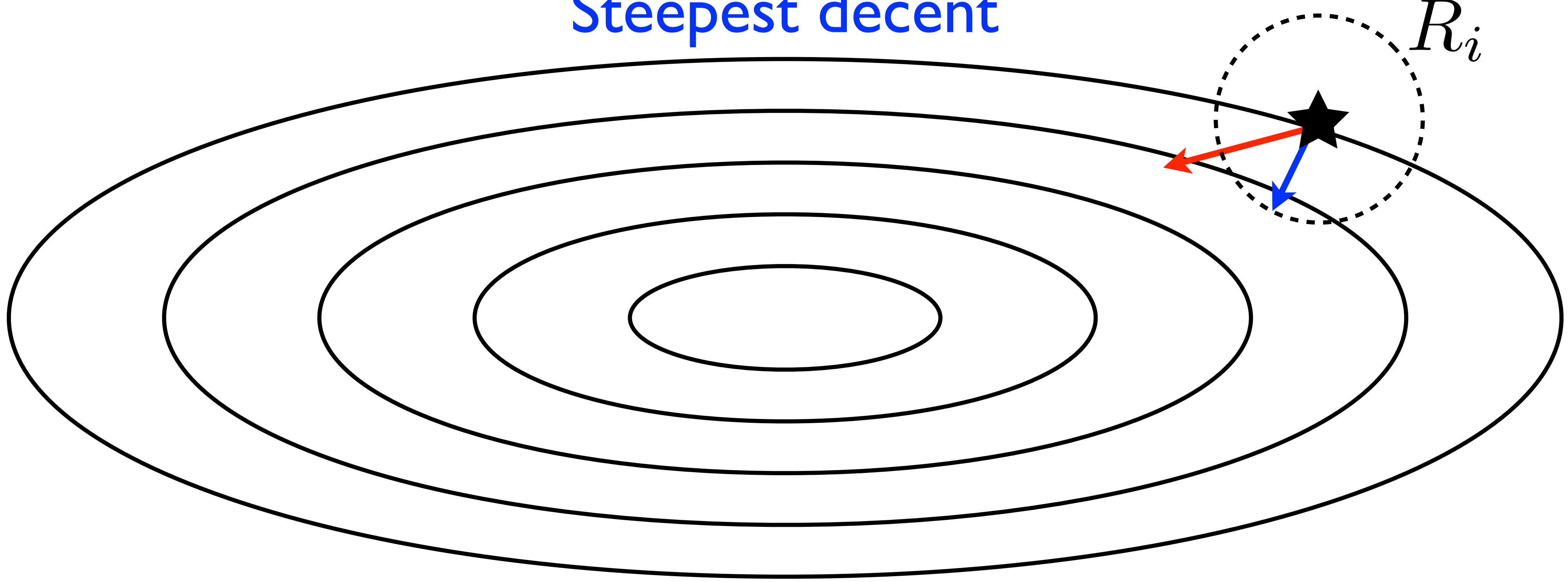


- That quadratic approximation may be good or bad

Trust-Region Methods

- Trust region methods choose between the Newton-Raphson direction when the quadratic approximation is good and the steepest decent direction when it is not.

Newton-Raphson
Steepest decent

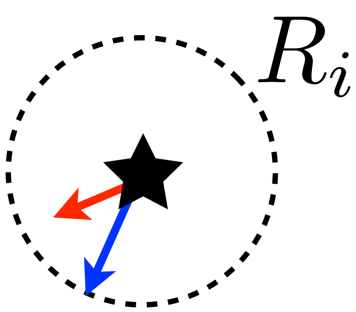


- This choice is based on whether the Newton-Raphson step is too large.

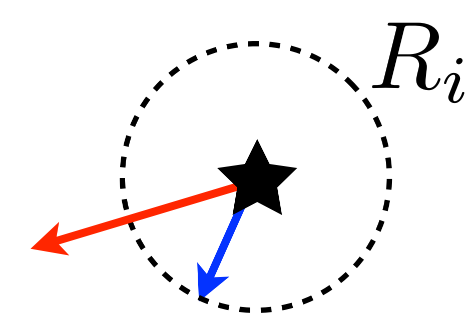
Trust-Region Methods

Newton-Raphson
Steepest decent

- Newton step: $\mathbf{d}_i^{NR} = -\mathbf{H}(\mathbf{x}_i)^{-1}\mathbf{g}(\mathbf{x}_i)$
- Steepest decent: $\mathbf{d}_i^{SD} = -\alpha_i \mathbf{g}(\mathbf{x}_i)$



- If $\|\mathbf{d}_i^{NR}\|_2 < R_i$ and $f(\mathbf{x}_i + \mathbf{d}_i^{NR}) < f(\mathbf{x}_i)$
  - Take the Newton-Raphson step
- Else
  - Take a step in the steepest descent direction
    - If $\|\mathbf{d}_i^{SD}\|_2 < R_i$ and $f(\mathbf{x}_i + \mathbf{d}_i^{SD}) < f(\mathbf{x}_i)$ with optimal step size
      - Take the optimal steepest descent
    - Else step to the trust boundary using:
      $\alpha_i = R_i / \|\mathbf{g}(\mathbf{x}_i)\|_2$



Trust-Region Methods

- The size of the trust region can be set arbitrarily initially.
- The trust region grows or shrinks depending on which of the two steps we choose.
- If the Newton-Raphson step was chosen:
  - The quadratic approximation has minimum value:

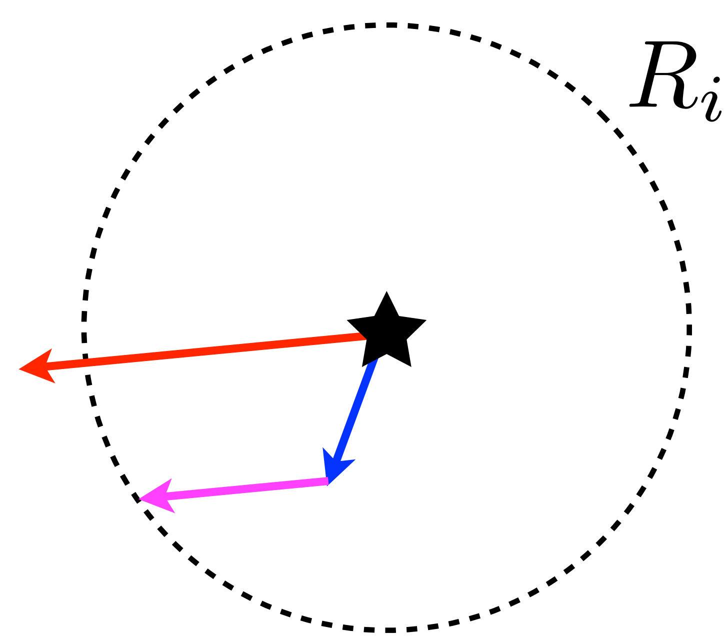
$$\phi = f(\mathbf{x}_i) + \mathbf{g}(\mathbf{x}_i)^T \mathbf{d}_i + \frac{1}{2}\mathbf{d}_i^T \mathbf{H}(\mathbf{x}_i)\mathbf{d}_i$$

  - GROW the trust-radius when $\phi > f(\mathbf{x}_i + \mathbf{d}_i)$, because the function was smaller than predicted
  - otherwise, SHRINK the trust-radius.
- If the steepest descent step was chosen, keep the trust radius the same.

Trust-Region Methods

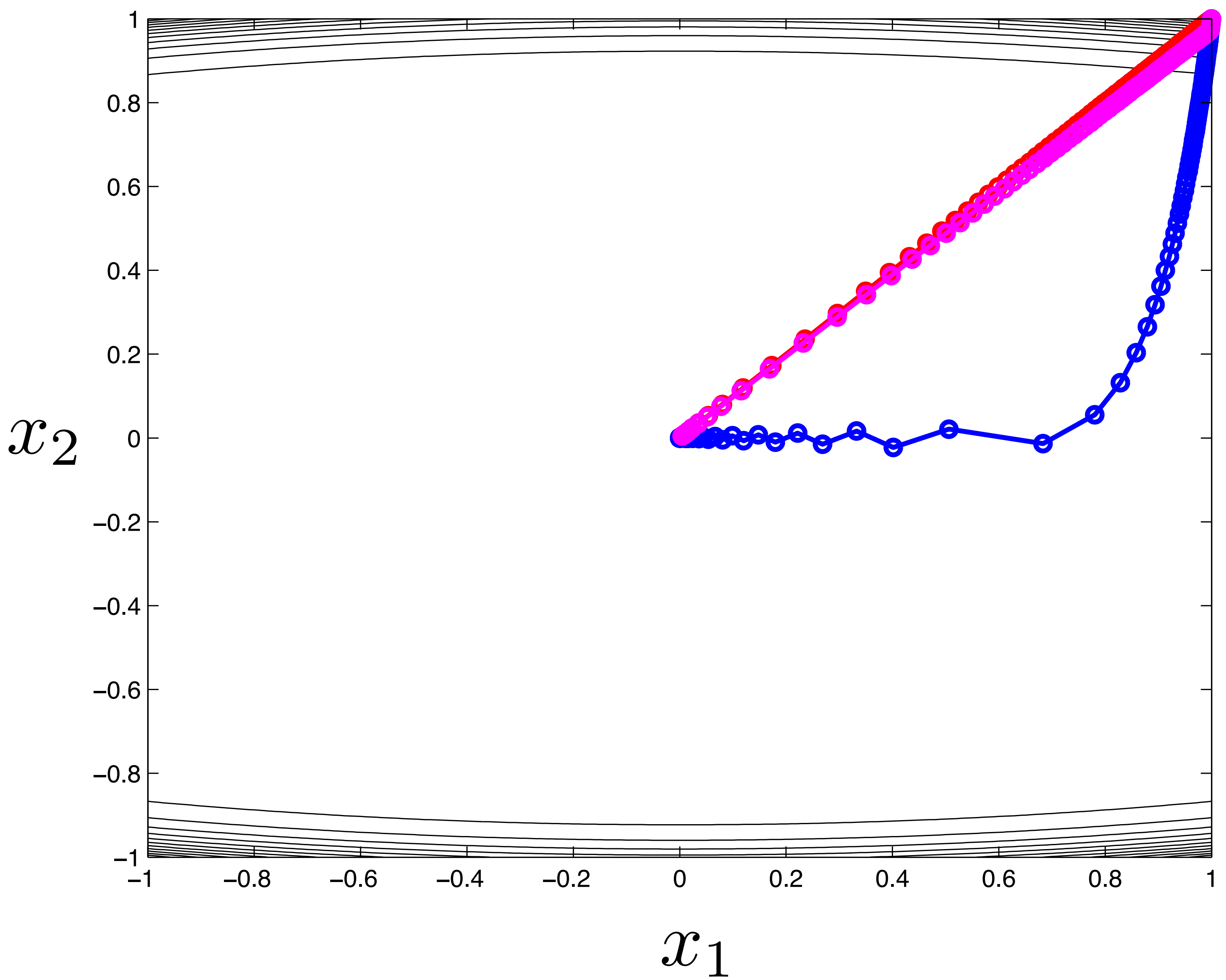
- What is a good value of the trust-region radius?
  - MATLAB uses one initially!
- Variations on the trust-region method exist as well.
  - MATLAB uses the dog-leg step instead of the optimal steepest descent step:

Newton-Raphson
Optimal steepest decent
Dog-leg step



Unconstrained Optimization

- Method of steepest decent/Newton-Raphson/Trust-Region:
  - Example: $\log f(x) = (x_1^2 + 10x_2^2)^2$
    - Contours for the function: $\alpha_i = \dfrac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{g}(\mathbf{x}_i)}{\mathbf{g}(\mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i) \mathbf{g}(\mathbf{x}_i)}$



Unconstrained Optimization

- Method of steepest decent/Newton-Raphson/Trust-Region:
  - Example: $\log f(x) = (x_1^2 + 10x_2^2)^2$
    - Contours for the function: $\alpha_i = \dfrac{\mathbf{g}(\mathbf{x}_i)^T \mathbf{g}(\mathbf{x}_i)}{\mathbf{g}(\mathbf{x}_i)^T \mathbf{H}(\mathbf{x}_i) \mathbf{g}(\mathbf{x}_i)}$