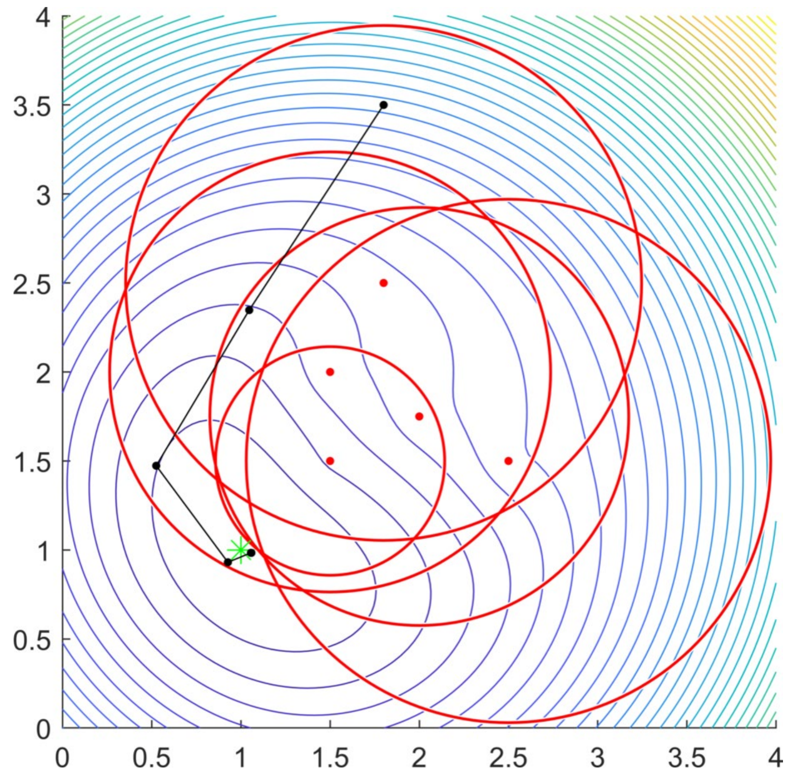
UiO : Department of Technology Systems
University of Oslo

Nonlinear least squares

Trym Vegard Haavardsholm



TEK5030

Problem formulation

Consider a set of *n* possibly nonlinear equations in *m* unknowns $\mathbf{x} = [x_1, \ldots, x_m]^T$ written as

$$e_i(\mathbf{x}) = 0, \quad i = 1, \ldots, n$$

$$e_i : \mathbb{R}^m \rightarrow \mathbb{R}$$

Problem formulation

Consider a set of n possibly nonlinear equations in m unknowns $\mathbf{x} = [x_1, \dots, x_m]^T$ written as

$$\underbrace{e_i(\mathbf{x})}_{i\text{-th equation}} = 0, \quad i = 1, \dots, n$$

$$e_i : \mathbb{R}^m \rightarrow \mathbb{R}$$

Problem formulation

Consider a set of *n* possibly nonlinear equations in *m* unknowns $\mathbf{x} = [x_1, \dots, x_m]^T$ written as

$$\underbrace{e_i(\mathbf{x})}_{i\text{-th } error \text{ or } residual} = 0, \quad i = 1, \dots, n$$

$$e_i : \mathbb{R}^m \to \mathbb{R}$$

Problem formulation

Consider a set of *n* possibly nonlinear equations in *m* unknowns $\mathbf{x} = [x_1, \dots, x_m]^T$ written as

$$e_i(\mathbf{x}) = 0, \quad i = 1, \dots, n \qquad\qquad e_i : \mathbb{R}^m \rightarrow \mathbb{R}$$

We can write these equations on vector form

$$e(\mathbf{x}) = \mathbf{0}, \qquad\qquad e : \mathbb{R}^m \rightarrow \mathbb{R}^n$$

where

$$e(\mathbf{x}) = \begin{bmatrix} e_1(\mathbf{x}) \\ \vdots \\ e_n(\mathbf{x}) \end{bmatrix}$$

Problem formulation

It is often not possible to find an exact solution to this problem.

We can instead seek an approximate solution
that minimizes the sum of squares of the residuals

$$f(\mathbf{x}) = e(\mathbf{x})^T e(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2$$

Problem formulation

It is often not possible to find an exact solution to this problem.

We can instead seek an approximate solution
that minimizes the sum of squares of the residuals

$$\underbrace{f(\mathbf{x})}_{\text{The \textit{objective function}}} = e(\mathbf{x})^T e(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2$$

Problem formulation

It is often not possible to find an exact solution to this problem.

We can instead seek an approximate solution
that minimizes the sum of squares of the residuals

$$f(\mathbf{x}) = e(\mathbf{x})^T e(\mathbf{x}) = \left\| e(\mathbf{x}) \right\|^2$$

This means that we want to find the $\mathbf{x}$ that minimizes the objective function:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\, f(\mathbf{x}) = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| e(\mathbf{x}) \right\|^2$$

Linear least squares

When the equations $e(\mathbf{x})$ are linear,
we can obtain an objective function on the form

$$f(\mathbf{x}) = \|e(\mathbf{x})\|^2 = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

A solution is required to have zero gradient:

$$\nabla f(\mathbf{x}^*) = 2\mathbf{A}^T\left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) = \mathbf{0}$$

This results in the **normal equations**,

$$\mathbf{A}^T\mathbf{A}\mathbf{x}^* = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

which can be solved with for example Cholesky or QR, or SVD.

Linear least squares

When the equations $e(\mathbf{x})$ are linear,
we can obtain an objective function on the form

$$f(\mathbf{x}) = \|e(\mathbf{x})\|^2 = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

A solution is required to have zero gradient:

$$\nabla f(\mathbf{x}^*) = 2\mathbf{A}^T\left(\mathbf{A}\mathbf{x}^* - \mathbf{b}\right) = \mathbf{0}$$

This results in the **normal equations**,

$$\mathbf{A}^T\mathbf{A}\mathbf{x}^* = \mathbf{A}^T\mathbf{b}$$

$$\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

which can be solved with for example Cholesky or QR, or SVD.

Matlab example:

```
x = A\b;
```

Python example:

```
x = numpy.linalg.lstsq(A, b)[0]
```

Eigen example:

```
x = A.colPivHouseholderQr().solve(b);
```

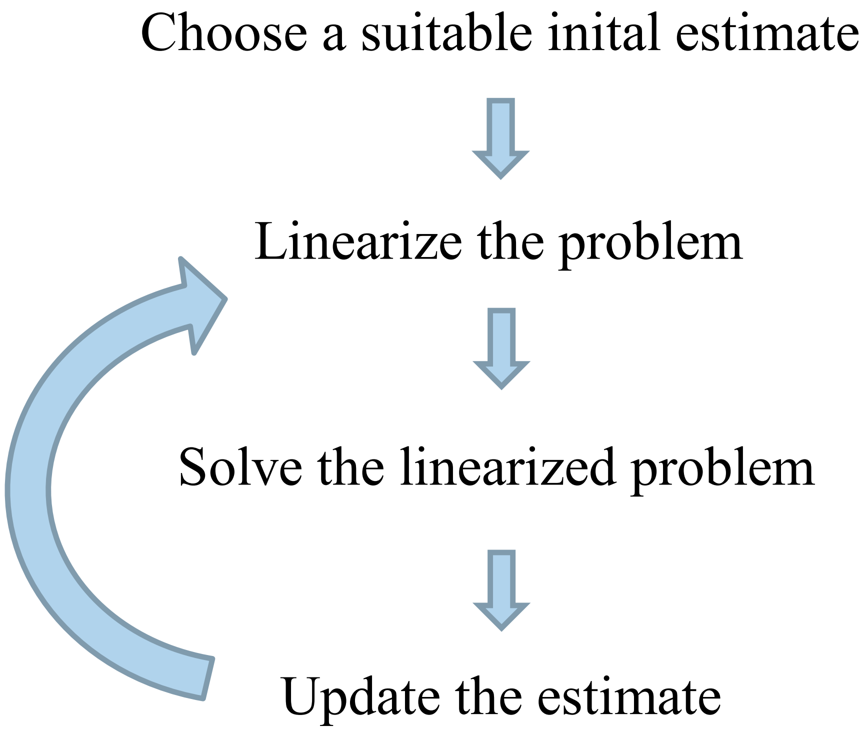
Read more about LLS:

- http://vmls-book.stanford.edu/vmls.pdf

Nonlinear least squares

When the equations $e(\mathbf{x})$ are nonlinear,
we have a **nonlinear least squares** problem.

They cannot be solved directly,
but require an iterative procedure
starting from a suitable initial estimate.

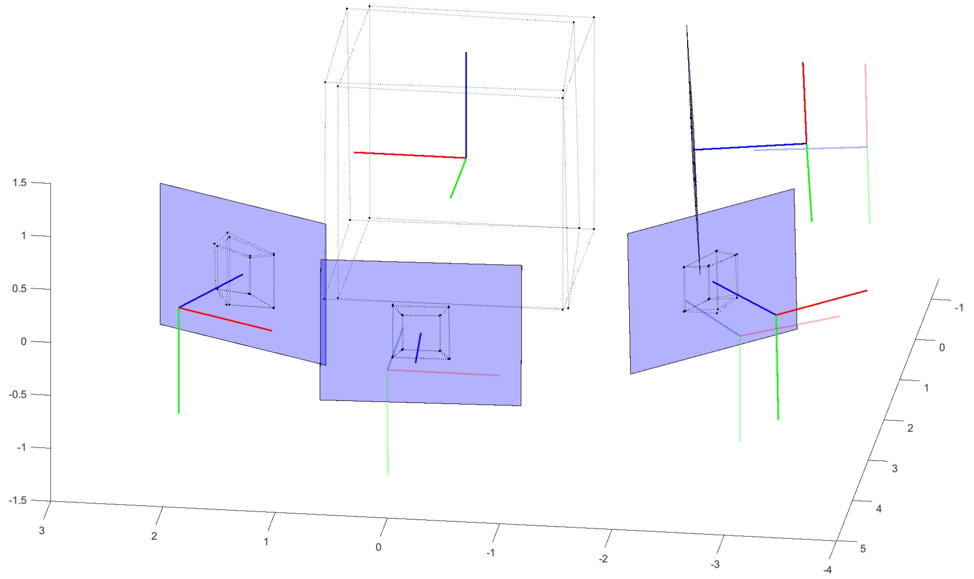


State variables

A **state variable** $\mathbf{x}$ is typically used to describe the physical state of an object.

We can estimate several state variables at once by concatenating all the variables into the vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix}$$



The equations $e_i(\mathbf{x})$ can be defined to operate on one or more of these p state variables.

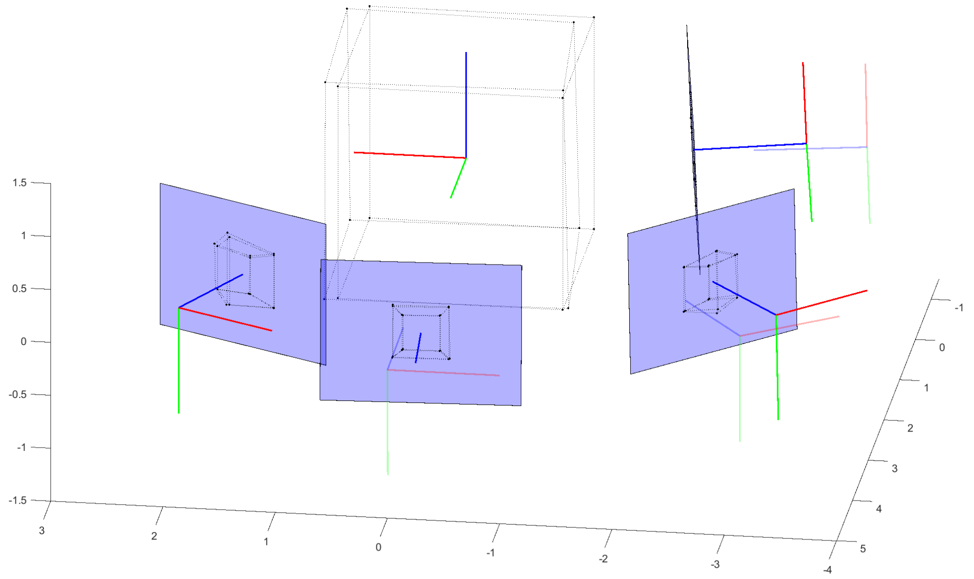
State variables

A **state variable** $\mathbf{x}$ is typically used to describe the physical state of an object.

We can estimate several state variables at once by concatenating all the variables into the vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_p \end{bmatrix}$$

The equations $e_i(\mathbf{x})$ can be defined to operate on one or more of these p state variables.



How can we represent both points and poses as states?

Concatenated set of state variables

Concatenation of state variables over a composite manifold
and the corresponding concatenation of tangent space vectors

$$\underline{\mathcal{X}} \triangleq \begin{Bmatrix} \mathcal{X}_1 \\ \vdots \\ \mathcal{X}_p \end{Bmatrix} \in \mathcal{M} \qquad \underline{\boldsymbol{\tau}} \triangleq \begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_p \end{bmatrix} \in \mathbb{R}^m \qquad \begin{aligned} &\mathcal{X}_i \in \mathcal{M}_i \\ &\mathcal{M} = \left\{\mathcal{M}_1, \dots, \mathcal{M}_p\right\} \\ &\boldsymbol{\tau}_i \in \mathcal{T}\mathcal{M}_i \end{aligned}$$

Plus and minus for the concatenated state variable

$$\underline{\mathcal{X}} \oplus \underline{\boldsymbol{\tau}} \triangleq \begin{Bmatrix} \mathcal{X}_1 \oplus \boldsymbol{\tau}_1 \\ \vdots \\ \mathcal{X}_p \oplus \boldsymbol{\tau}_p \end{Bmatrix} \in \mathcal{M} \qquad \underline{\mathcal{Y}} \ominus \underline{\mathcal{X}} \triangleq \begin{bmatrix} \mathcal{Y}_1 \ominus \mathcal{X}_1 \\ \vdots \\ \mathcal{Y}_p \ominus \mathcal{X}_p \end{bmatrix} \in \mathbb{R}^m$$

Concatenated set of state variables

We define $\underline{\mathcal{X}}_i$ to be the concatenated set of state variables taken as input by the *i*-th equation $e_i(\underline{\mathcal{X}}_i)$.

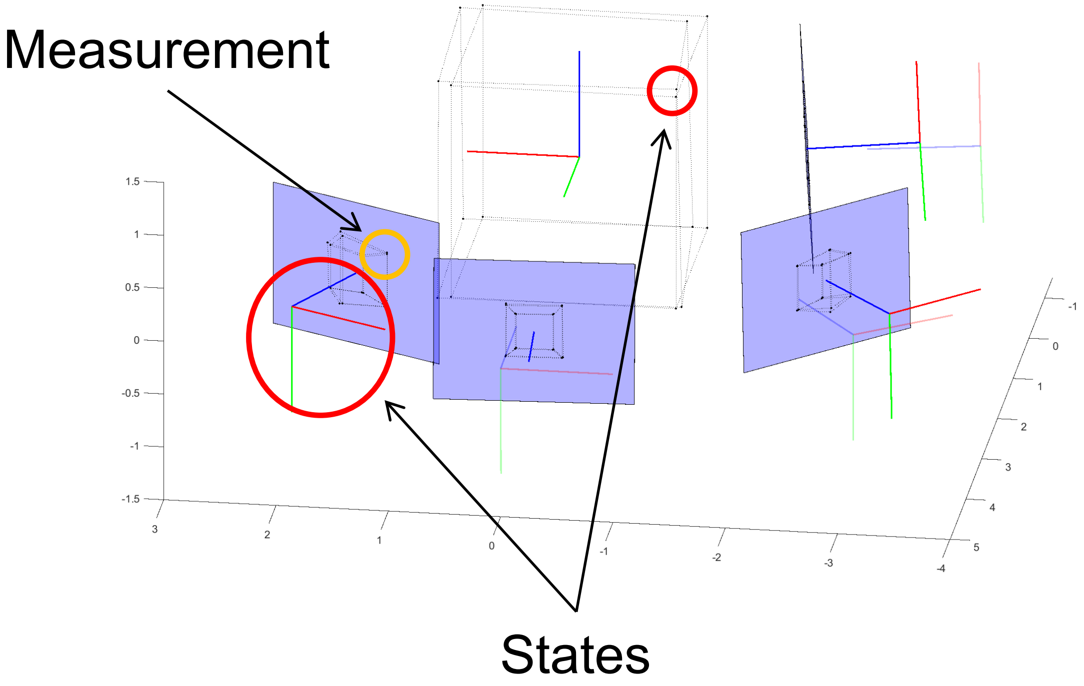
Concatenated set of state variables

We define $\underline{\mathcal{X}}_i$ to be the concatenated set of state variables taken as input by the *i*-th equation $e_i(\underline{\mathcal{X}}_i)$.

Example:

$$e_{ij}(\underline{\mathcal{X}}_{ij}) = e_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w) = \pi(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i$$

$$\underline{\mathcal{X}}_{ij} = \begin{Bmatrix} \mathbf{T}_{wc_i} \\ \mathbf{x}_j^w \end{Bmatrix}$$



Concatenated set of state variables

We define $\underline{\mathcal{X}}_i$ to be the concatenated set of state variables taken as input by the *i*-th equation $e_i(\underline{\mathcal{X}}_i)$.

We can then define the objective function over all state variables

$$f(\underline{\mathcal{X}}) = \left\| e(\underline{\mathcal{X}}) \right\|^2 = \sum_{i=1}^{n} \left\| e_i(\underline{\mathcal{X}}_i) \right\|^2$$

State estimation

We want to solve **state estimation problems**
based on **measurements** and corresponding **measurement models**

Let X be the set of all unknown state variables,
and Z be the set of all measurements.

State estimation

We want to solve **state estimation problems**
based on **measurements** and corresponding **measurement models**

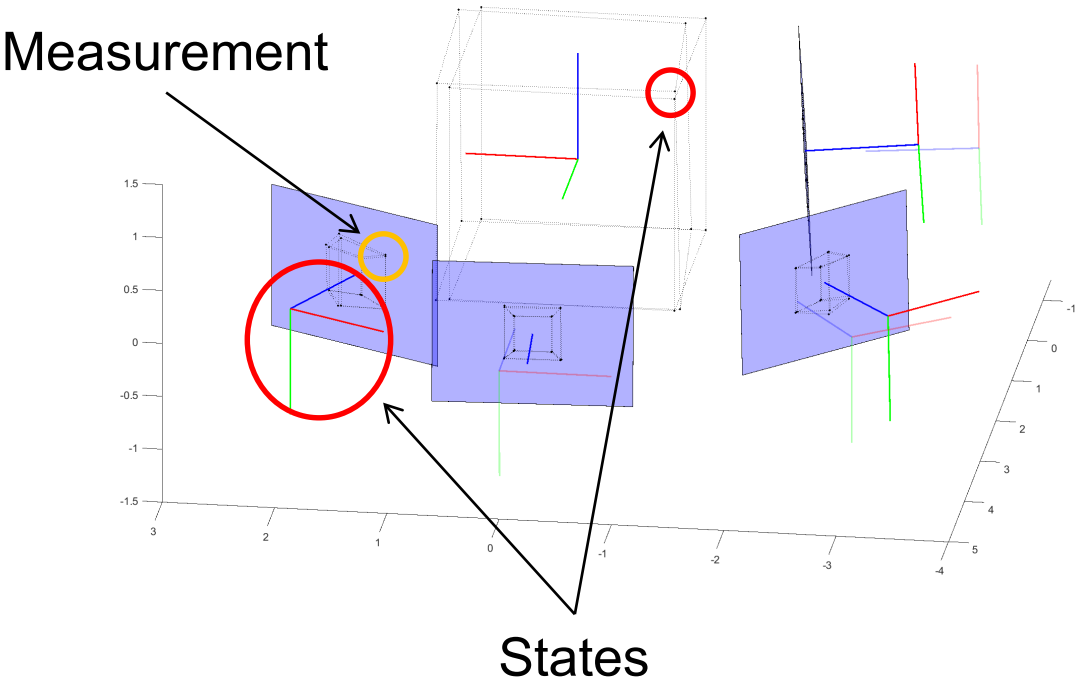
Let X be the set of all unknown state variables,
and Z be the set of all measurements.

Example:

Measurement model

$$e_{ij}(\underline{\mathcal{X}}_{ij}) = e_{ij}(\mathbf{T}_{wc_i}, \mathbf{x}_j^w) = \pi(\mathbf{T}_{wc_i}^{-1} \cdot \mathbf{x}_j^w) - \mathbf{u}_j^i$$

$$\underline{\mathcal{X}}_{ij} = \begin{Bmatrix} \mathbf{T}_{wc_i} \\ \mathbf{x}_j^w \end{Bmatrix} \qquad \mathcal{Z}_{ij} = \{\mathbf{u}_j^i\}$$



Deterministic model for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(\underline{\mathcal{X}}_i) \cancel{+ \eta_i, \qquad \eta_i \sim N(\mathbf{0}, \mathbf{\Sigma}_i)}$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(\underline{\mathcal{X}}_i)$$

Measurement error function:

$$e_i(\underline{\mathcal{X}}_i) = h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i$$

Objective function:

$$f(\underline{\mathcal{X}}) = \sum_{i=1}^{n} \left\| h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i \right\|^2$$

Deterministic model for state estimation

Measurement model:

$$\mathbf{z}_i = h_i(\underline{\mathcal{X}}_i) \cancel{+ \eta_i, \qquad \eta_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i)}$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(\underline{\mathcal{X}}_i)$$

Measurement error function:

$$e_i(\underline{\mathcal{X}}_i) = h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i$$

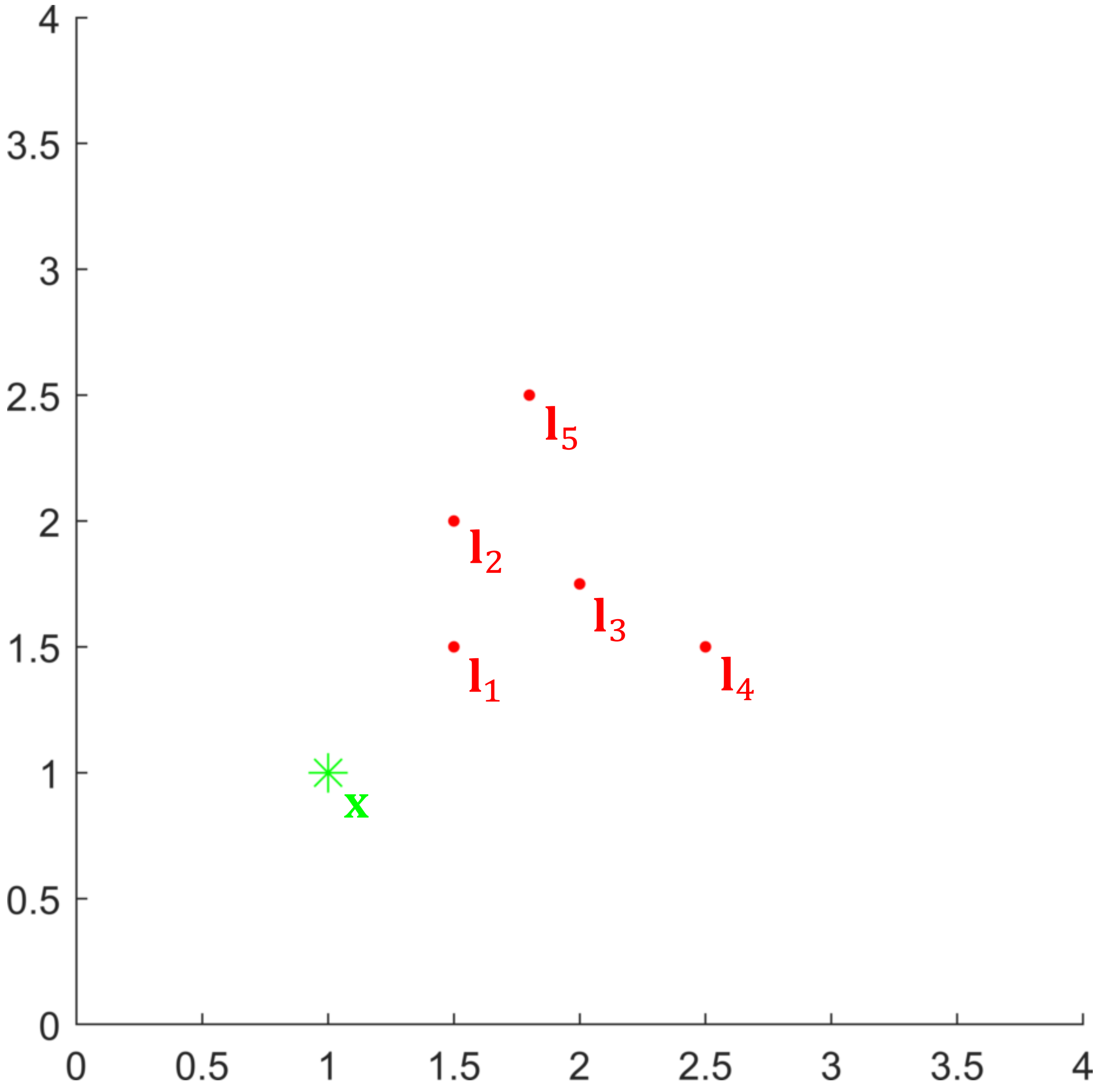
Objective function:

$$f(\underline{\mathcal{X}}) = \sum_{i=1}^{n} \left\| h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i \right\|^2$$

This results in the nonlinear least squares problem:

$$\underline{\mathcal{X}}^* = \underset{\underline{\mathcal{X}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i \right\|^2$$

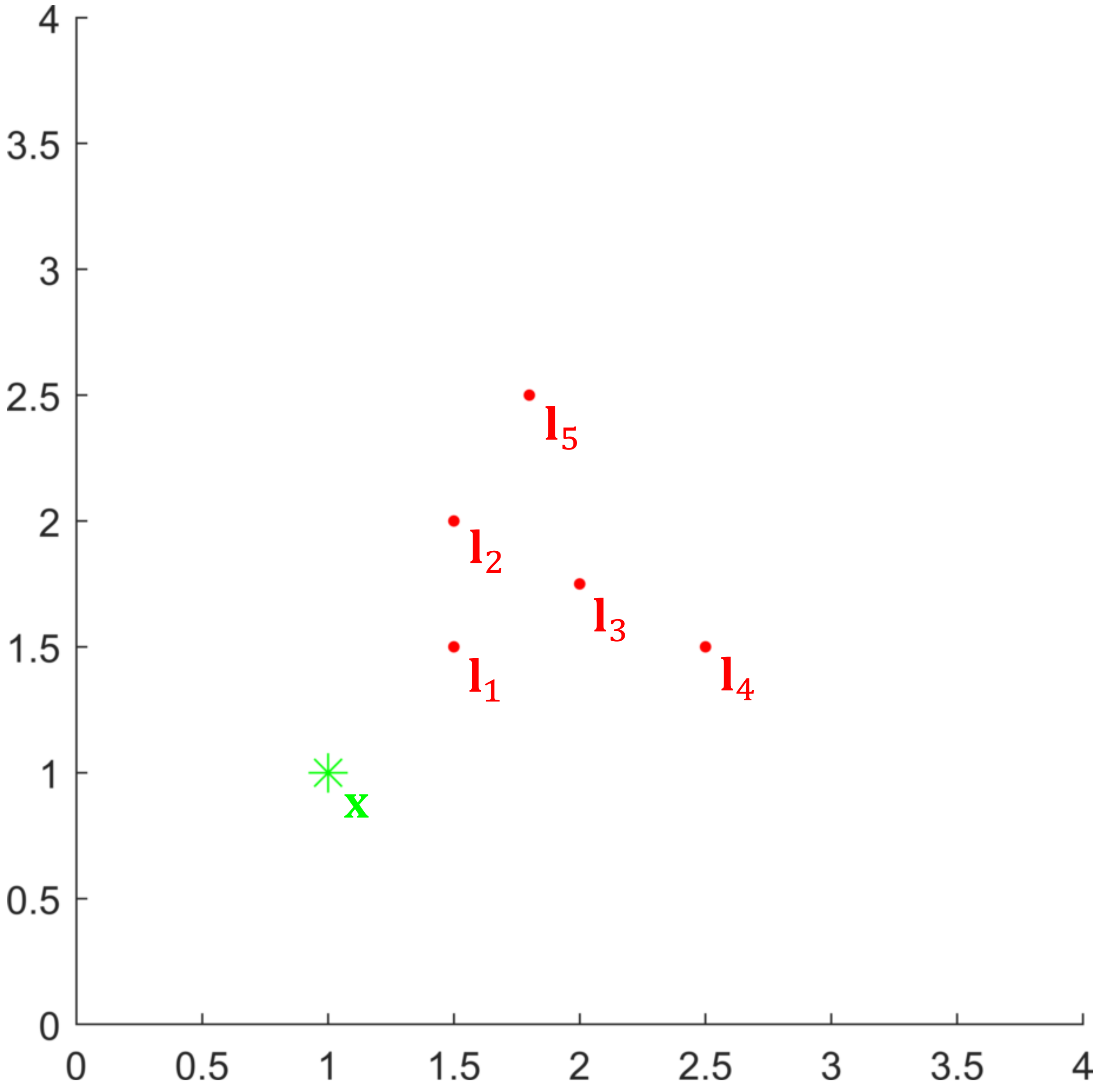
Example:
Range-based localization



Example: Range-based localization

States: Our location

$$\underline{\mathcal{X}} = \mathbf{x}$$



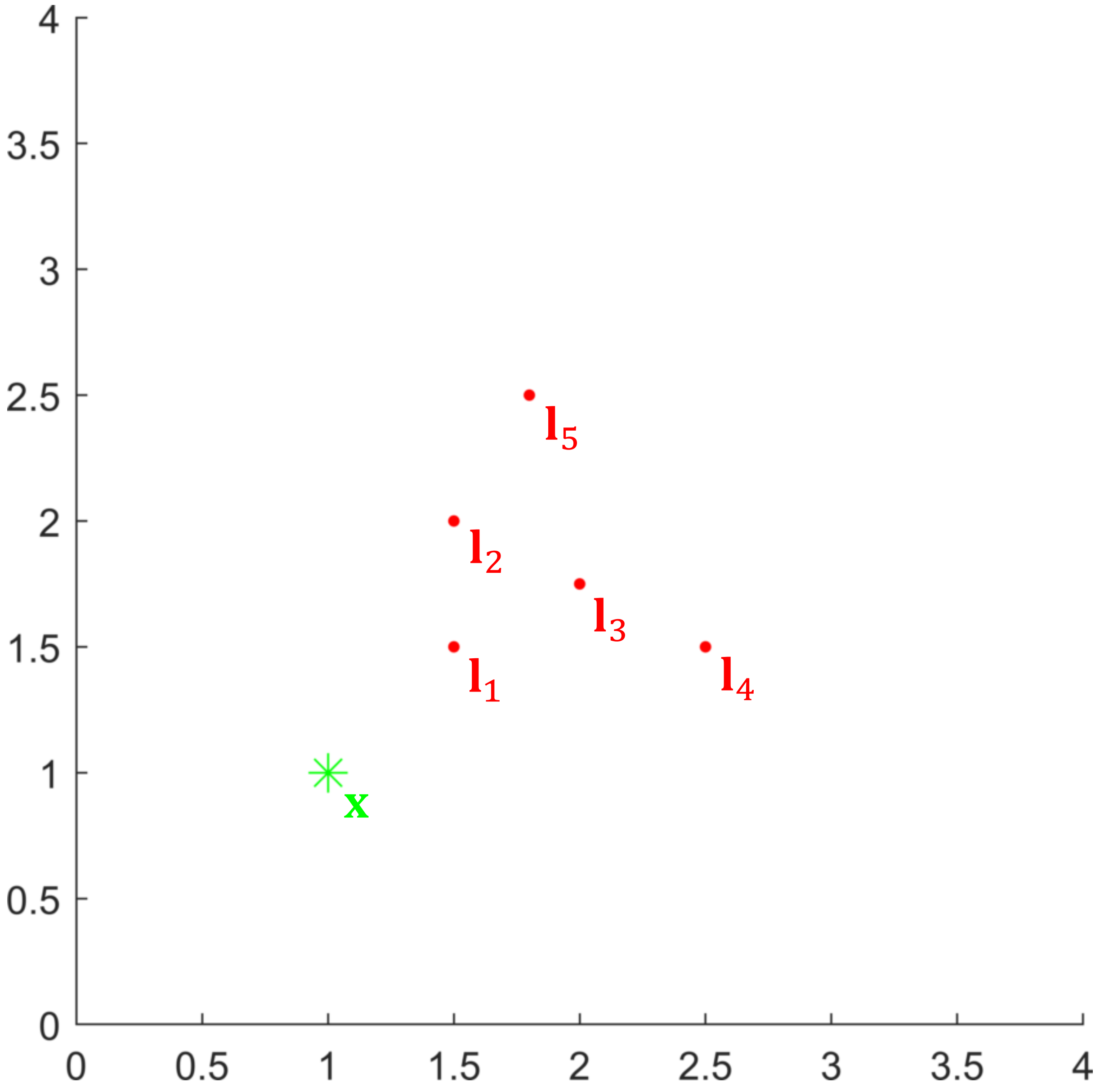
Example:
Range-based localization

States: Our location

$$\underline{\mathcal{X}} = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \dots, \rho_n\}$$



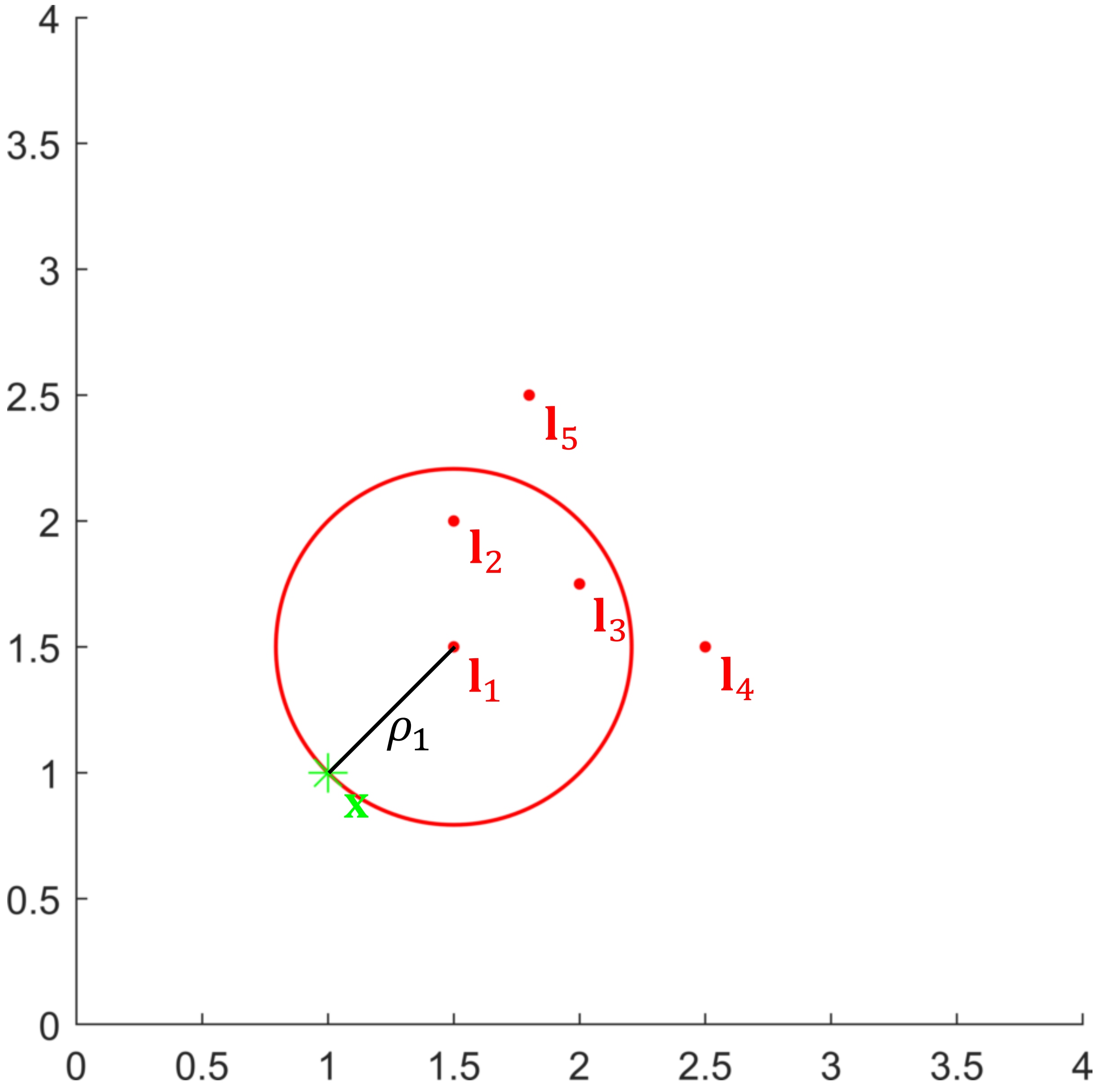
Example:
Range-based localization

States: Our location

$$\underline{\mathcal{X}} = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_n\}$$



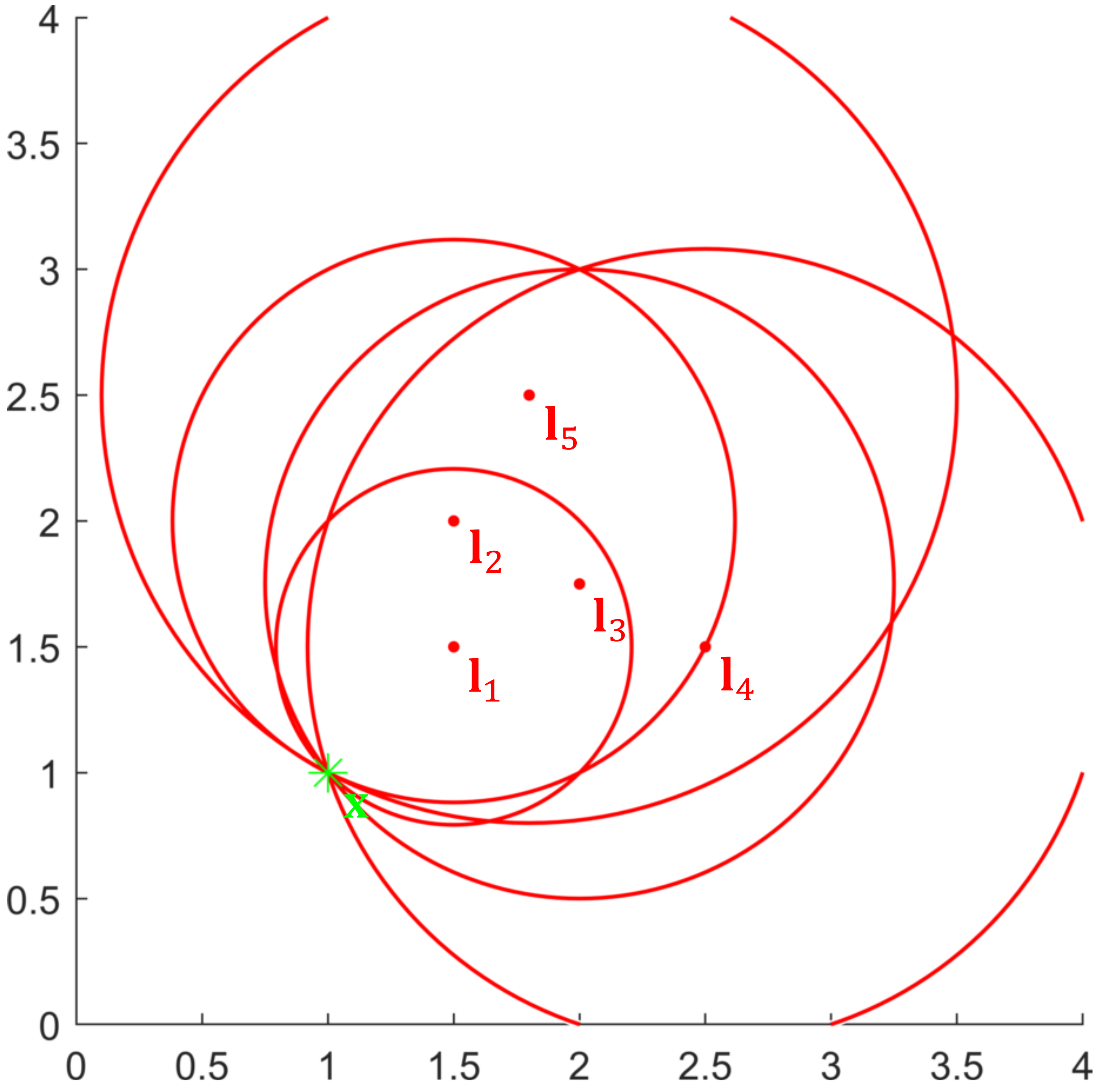
Example:
Range-based localization

States: Our location

$$\underline{\mathcal{X}} = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_n\}$$



Example:
Range-based localization

States: Our location

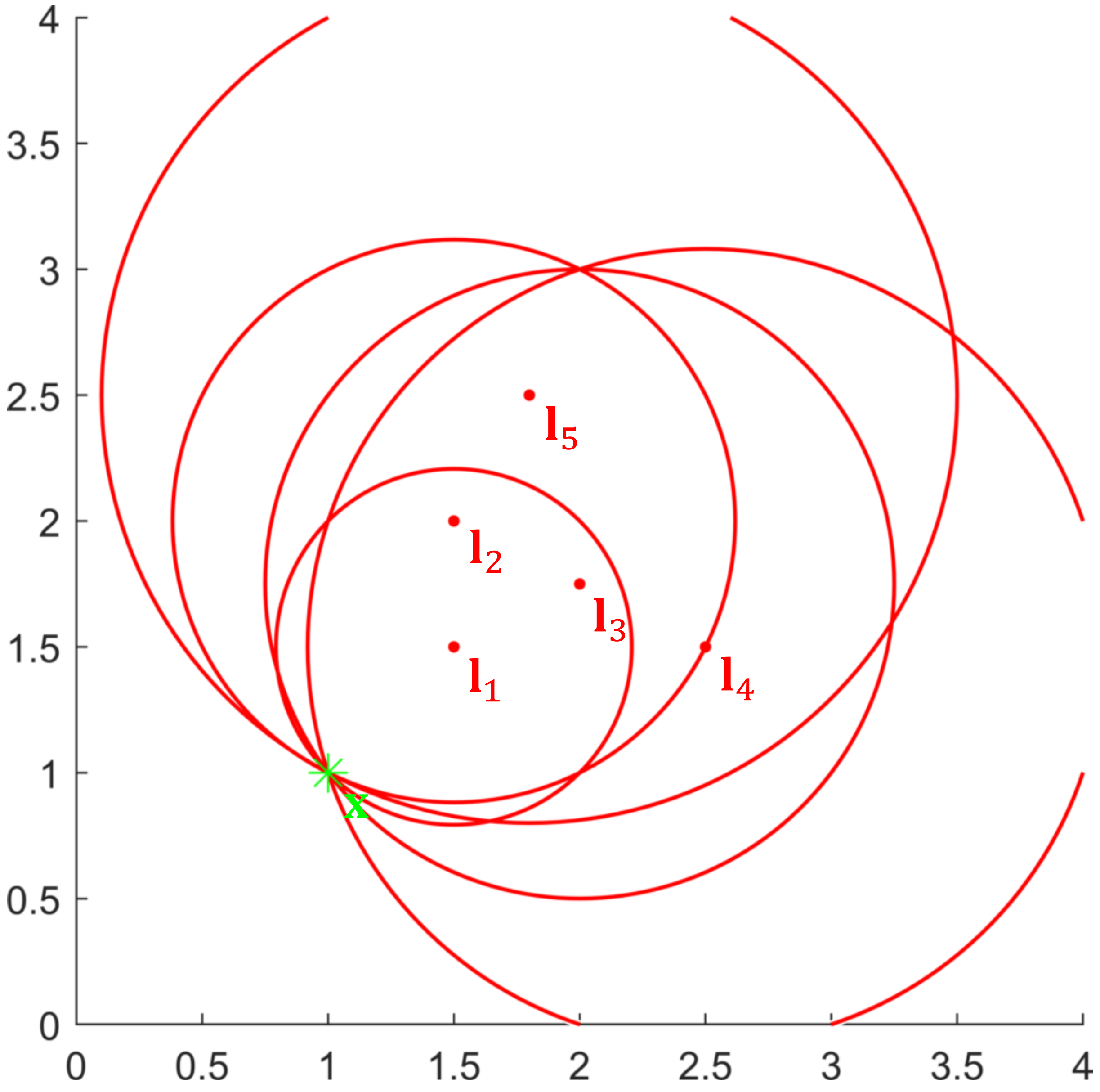
$$\underset{\sim}{\mathcal{X}} = \mathbf{x}$$

Measurements: Range to landmarks

$$Z = \{\rho_1, \ldots, \rho_n\}$$

Measurement model:

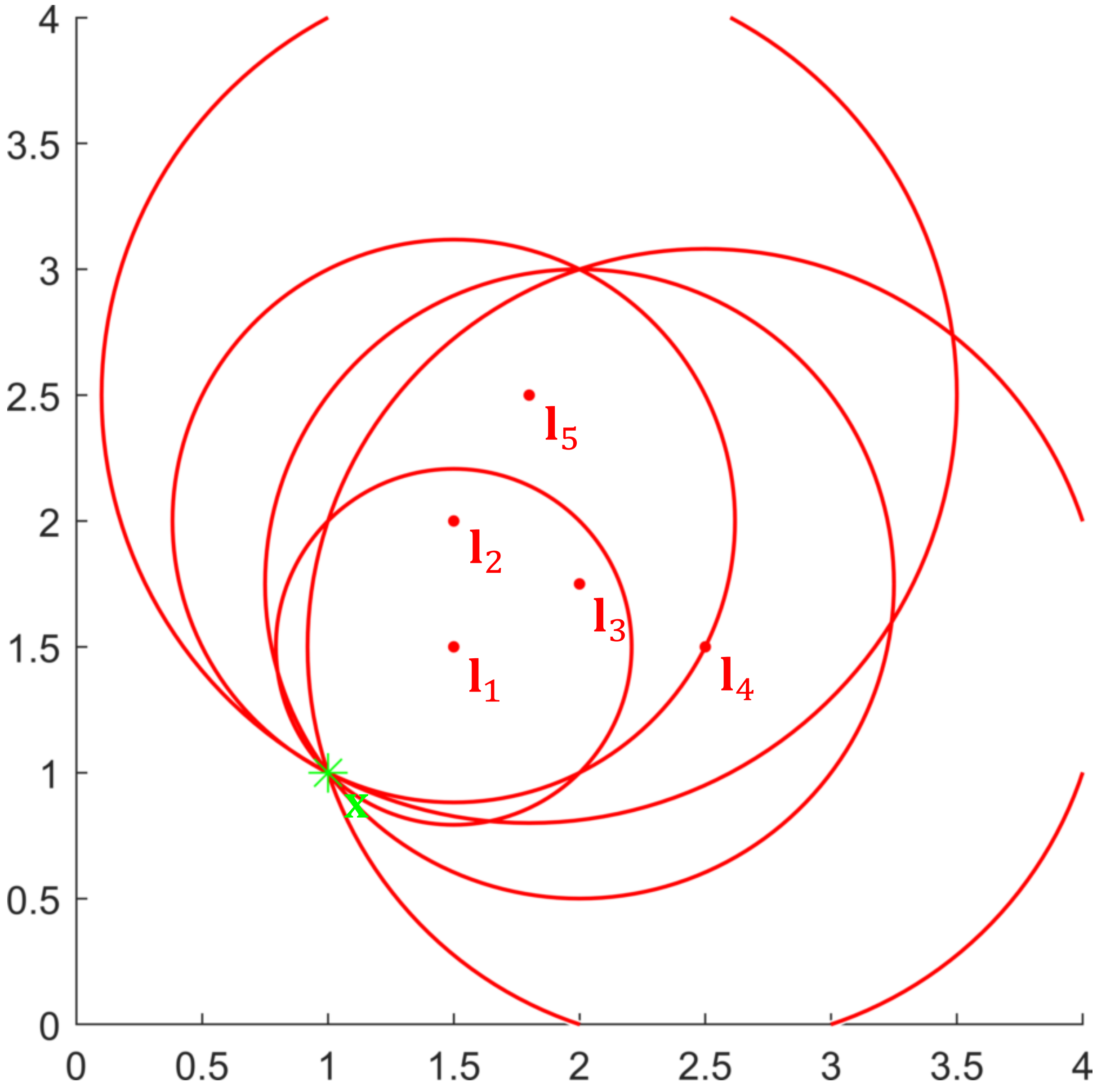
$$\rho_i = \|\mathbf{x} - \mathbf{l}_i\|$$



Example:
Range-based localization

Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x};\mathbf{l}_i) = \left\|\mathbf{x} - \mathbf{l}_i\right\|$$



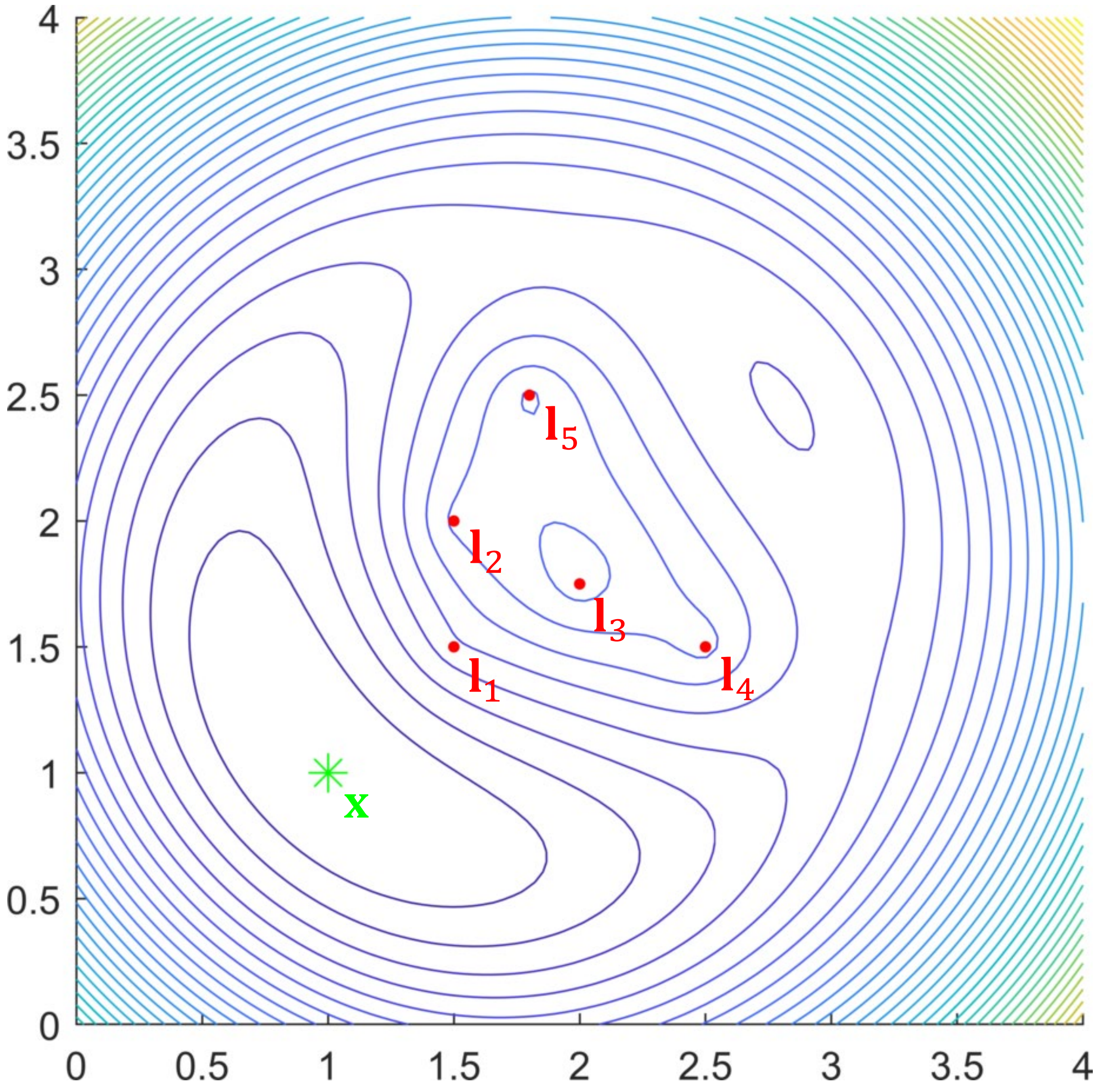
Example:
Range-based localization

Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x};\mathbf{l}_i) = \left\|\mathbf{x} - \mathbf{l}_i\right\|$$

Objective function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left\| h(\mathbf{x};\mathbf{l}_i) - \rho_i \right\|^2$$
$$= \sum_{i=1}^{n} \left(\left\|\mathbf{x} - \mathbf{l}_i\right\| - \rho_i \right)^2$$



Example:
Range-based localization

Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x}; \mathbf{l}_i) = \|\mathbf{x} - \mathbf{l}_i\|$$

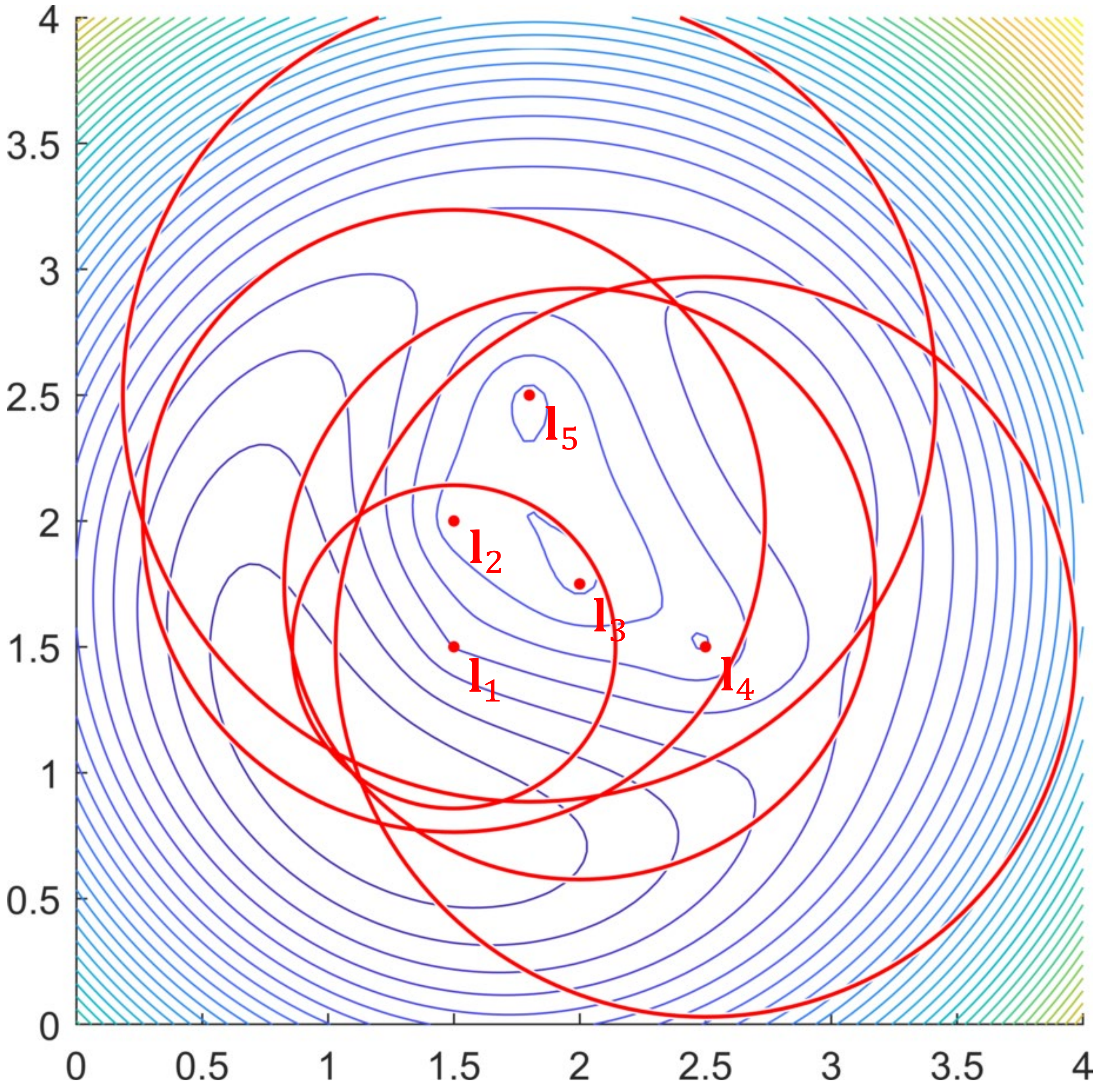
Objective function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \|h(\mathbf{x}; \mathbf{l}_i) - \rho_i\|^2$$

$$= \sum_{i=1}^{n} \left(\|\mathbf{x} - \mathbf{l}_i\| - \rho_i\right)^2$$

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\|\mathbf{x} - \mathbf{l}_i\| - \rho_i\right)^2$$



Nonlinear least squares

When the equations $e_i(\underline{\mathcal{X}}_i) = h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i$ are nonlinear, we have a **nonlinear least squares** problem.

They cannot be solved directly,
but require an iterative procedure
starting from a suitable initial estimate.

Choose a suitable inital estimate

$\Downarrow$

Linearize the problem

$\Downarrow$

Solve the linearized problem

$\Downarrow$

Update the estimate

Linearizing the problem

We can linearize the measurement prediction functions using **first order Taylor expansions** at the current estimates $\underline{\hat{\mathcal{X}}}_i$:

$$h_i(\underline{\mathcal{X}}_i) = h_i(\underline{\hat{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \approx h_i(\underline{\hat{\mathcal{X}}}_i) + \mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i$$

where the **measurement Jacobian** $\mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i}$ is

$$\mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i} \triangleq \left. \frac{\partial h_i(\underline{\mathcal{X}}_i)}{\partial \underline{\mathcal{X}}_i} \right|_{\underline{\hat{\mathcal{X}}}_i}$$

and

$$\underline{\boldsymbol{\tau}}_i \triangleq \underline{\mathcal{X}}_i \ominus \underline{\hat{\mathcal{X}}}_i$$

is the **state update vector**.

Linearizing the problem

This leads to the linearized measurement error function

$$e_i(\underline{\mathcal{X}}_i) = e_i(\hat{\underline{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \approx h_i(\hat{\underline{\mathcal{X}}}_i) + \mathbf{J}^{h_i}_{\hat{\underline{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i$$

Linearizing the problem

The linearized objective function is then given by

$$
\begin{aligned}
f(\underline{\mathcal{X}}) = f(\underline{\hat{\mathcal{X}}} \oplus \underline{\boldsymbol{\tau}}) &= \sum_{i=1}^{n} \left\| e_i(\underline{\hat{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \right\|^2 \\
&\approx \sum_{i=1}^{n} \left\| h_i(\underline{\hat{\mathcal{X}}}_i) + \mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{J}^{h_i}_{\underline{\hat{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i - \left(\mathbf{z}_i - h_i(\underline{\hat{\mathcal{X}}}_i) \right) \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{A}_i \underline{\boldsymbol{\tau}}_i - \mathbf{b}_i \right\|^2 \\
&= \left\| \mathbf{A} \underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2
\end{aligned}
$$

Linearizing the problem

The linearized objective function is then given by

$$
\begin{aligned}
f(\underline{\mathcal{X}}) = f(\hat{\underline{\mathcal{X}}} \oplus \underline{\boldsymbol{\tau}}) &= \sum_{i=1}^{n} \left\| e_i(\hat{\underline{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \right\|^2 \\
&\approx \sum_{i=1}^{n} \left\| h_i(\hat{\underline{\mathcal{X}}}_i) + \mathbf{J}^{h_i}_{\hat{\underline{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{J}^{h_i}_{\hat{\underline{\mathcal{X}}}_i} \underline{\boldsymbol{\tau}}_i - \left(\mathbf{z}_i - h_i(\hat{\underline{\mathcal{X}}}_i) \right) \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{A}_i \underline{\boldsymbol{\tau}}_i - \mathbf{b}_i \right\|^2 \\
&= \left\| \mathbf{A} \underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2
\end{aligned}
$$

$\mathbf{A}_i = \mathbf{J}^{h_i}_{\hat{\underline{\mathcal{X}}}_i}$ are submatrices of the Jacobian $\mathbf{A}$

$\mathbf{b}_i = -e_i(\hat{\underline{\mathcal{X}}}_i)$ are subvectors of the error $\mathbf{b} = -e(\hat{\underline{\mathcal{X}}})$

Solving the linearized problem

The linearized objective function is then given by

$$
\begin{aligned}
f(\underline{\mathcal{X}}) = f(\underline{\hat{\mathcal{X}}} \oplus \underline{\boldsymbol{\tau}}) &= \sum_{i=1}^{n} \left\| e_i(\underline{\hat{\mathcal{X}}}_i \oplus \underline{\boldsymbol{\tau}}_i) \right\|^2 \\
&\approx \sum_{i=1}^{n} \left\| h_i(\underline{\hat{\mathcal{X}}}_i) + \mathbf{J}_{\underline{\hat{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \mathbf{z}_i \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{J}_{\underline{\hat{\mathcal{X}}}_i}^{h_i} \underline{\boldsymbol{\tau}}_i - \left(\mathbf{z}_i - h_i(\underline{\hat{\mathcal{X}}}_i) \right) \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \mathbf{A}_i \underline{\boldsymbol{\tau}}_i - \mathbf{b}_i \right\|^2 \\
&= \left\| \mathbf{A} \underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2
\end{aligned}
$$

We can solve the linearized problem as a linear least squares problem using the normal equations

$$
\mathbf{A}^T \mathbf{A} \underline{\boldsymbol{\tau}}^* = \mathbf{A}^T \mathbf{b}
$$

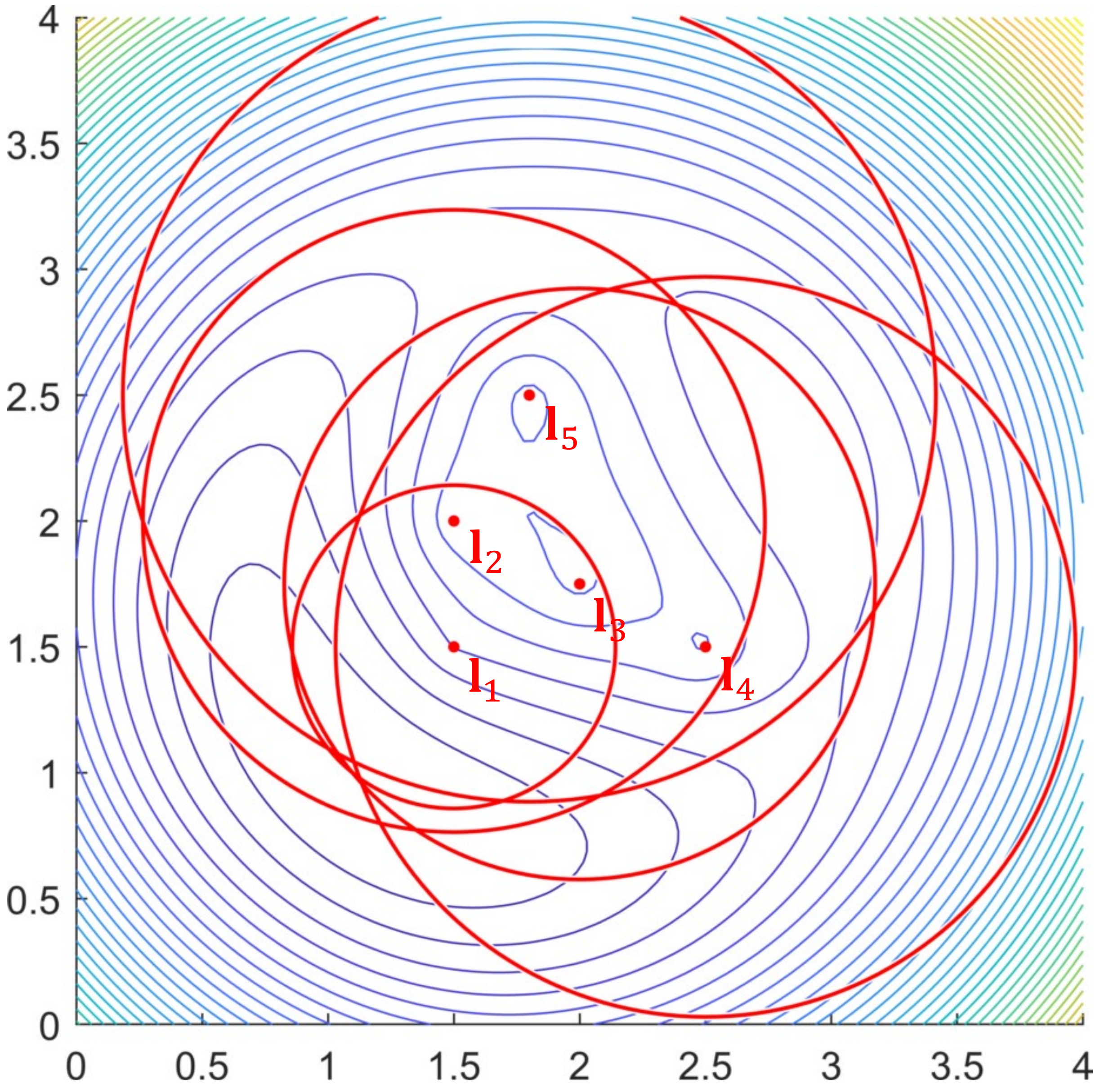
Example:
Range-based localization

Measurement prediction function:

$$\hat{\rho}_i = h(\mathbf{x};\mathbf{l}_i) = \left\|\mathbf{x} - \mathbf{l}_i\right\|$$

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i\right)^2$$



Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

$$h(\mathbf{x};\mathbf{l}_i) = \left\| \mathbf{x} - \mathbf{l}_i \right\|$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}};\mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2$$

$$\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} = \frac{\left(\hat{\mathbf{x}} - \mathbf{l}_i \right)^T}{\left\| \hat{\mathbf{x}} - \mathbf{l}_i \right\|}$$

(See example 5.12 in the compendium)

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}}; \mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}}; \mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}}; \mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}}; \mathbf{l}_i) \right\} \right)^2
\end{aligned}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}};\mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}};\mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2
\end{aligned}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}}; \mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}}; \mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}}; \mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}}; \mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2
\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}};\mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}};\mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2
\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

$$\mathbf{A}_1 = \mathbf{J}_{\mathbf{x}^0}^{h(\mathbf{x}^0;\mathbf{l}_1)} = \frac{\left(\mathbf{x}^0 - \mathbf{l}_1\right)^T}{\left\|\mathbf{x}^0 - \mathbf{l}_1\right\|} = \frac{\left(\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix}\right)^T}{\left\|\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix}\right\|}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x}; \mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}}; \mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}}; \mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2
\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

$$\begin{aligned}
\mathbf{A}_1 = \mathbf{J}_{\mathbf{x}^0}^{h(\mathbf{x}^0;\mathbf{l}_1)} &= \frac{\left(\mathbf{x}^0 - \mathbf{l}_1 \right)^T}{\left\| \mathbf{x}^0 - \mathbf{l}_1 \right\|} = \frac{\left(\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix} \right)^T}{\left\| \begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix} \right\|} \\
&= \frac{\begin{bmatrix} 0.30 & 2.00 \end{bmatrix}}{2.02} = \begin{bmatrix} 0.15 & 0.99 \end{bmatrix}
\end{aligned}$$

Example:
Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^{n} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \operatorname*{argmin}_{\boldsymbol{\delta}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}};\mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \operatorname*{argmin}_{\boldsymbol{\delta}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}};\mathbf{l}_i) \right\} \right)^2 \\
&= \operatorname*{argmin}_{\boldsymbol{\delta}} \sum_{i=1}^{n} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2
\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

$$\begin{aligned}
\mathbf{A}_1 = \mathbf{J}_{\mathbf{x}^0}^{h(\mathbf{x}^0;\mathbf{l}_1)} &= \frac{\left(\mathbf{x}^0 - \mathbf{l}_1\right)^T}{\left\| \mathbf{x}^0 - \mathbf{l}_1 \right\|} = \frac{\left(\begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix} \right)^T}{\left\| \begin{bmatrix} 0.30 \\ 2.00 \end{bmatrix} \right\|} \\
&= \frac{\begin{bmatrix} 0.30 & 2.00 \end{bmatrix}}{2.02} = \begin{bmatrix} 0.15 & 0.99 \end{bmatrix}
\end{aligned}$$

$$\mathbf{b}_1 = \rho_1 - h(\mathbf{x}^0;\mathbf{l}_1) = 0.64 - 2.02 = -1.38$$

Example: Range-based localization

Nonlinear least squares problem:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\mathbf{x};\mathbf{l}_i) - \rho_i \right)^2$$

Linearized problem at $\hat{\mathbf{x}}$:

$$\begin{aligned}
\boldsymbol{\delta}^* &= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(h(\hat{\mathbf{x}};\mathbf{l}_i) + \mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \rho_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{J}_{\hat{\mathbf{x}}}^{h(\hat{\mathbf{x}};\mathbf{l}_i)} \boldsymbol{\delta} - \left\{ \rho_i - h(\hat{\mathbf{x}};\mathbf{l}_i) \right\} \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\mathbf{A}_i \boldsymbol{\delta} - \mathbf{b}_i \right)^2 \\
&= \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2
\end{aligned}$$

$$\mathbf{l}_i = \left\{ \begin{bmatrix} 1.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.50 \\ 2.00 \end{bmatrix}, \begin{bmatrix} 2.00 \\ 1.75 \end{bmatrix}, \begin{bmatrix} 2.50 \\ 1.50 \end{bmatrix}, \begin{bmatrix} 1.80 \\ 2.50 \end{bmatrix} \right\}$$

$$\rho_i = \left\{ 0.64, 1.23, 1.17, 1.47, 1.61 \right\}$$

$$\mathbf{x}^0 = \begin{bmatrix} 1.80 \\ 3.50 \end{bmatrix}$$

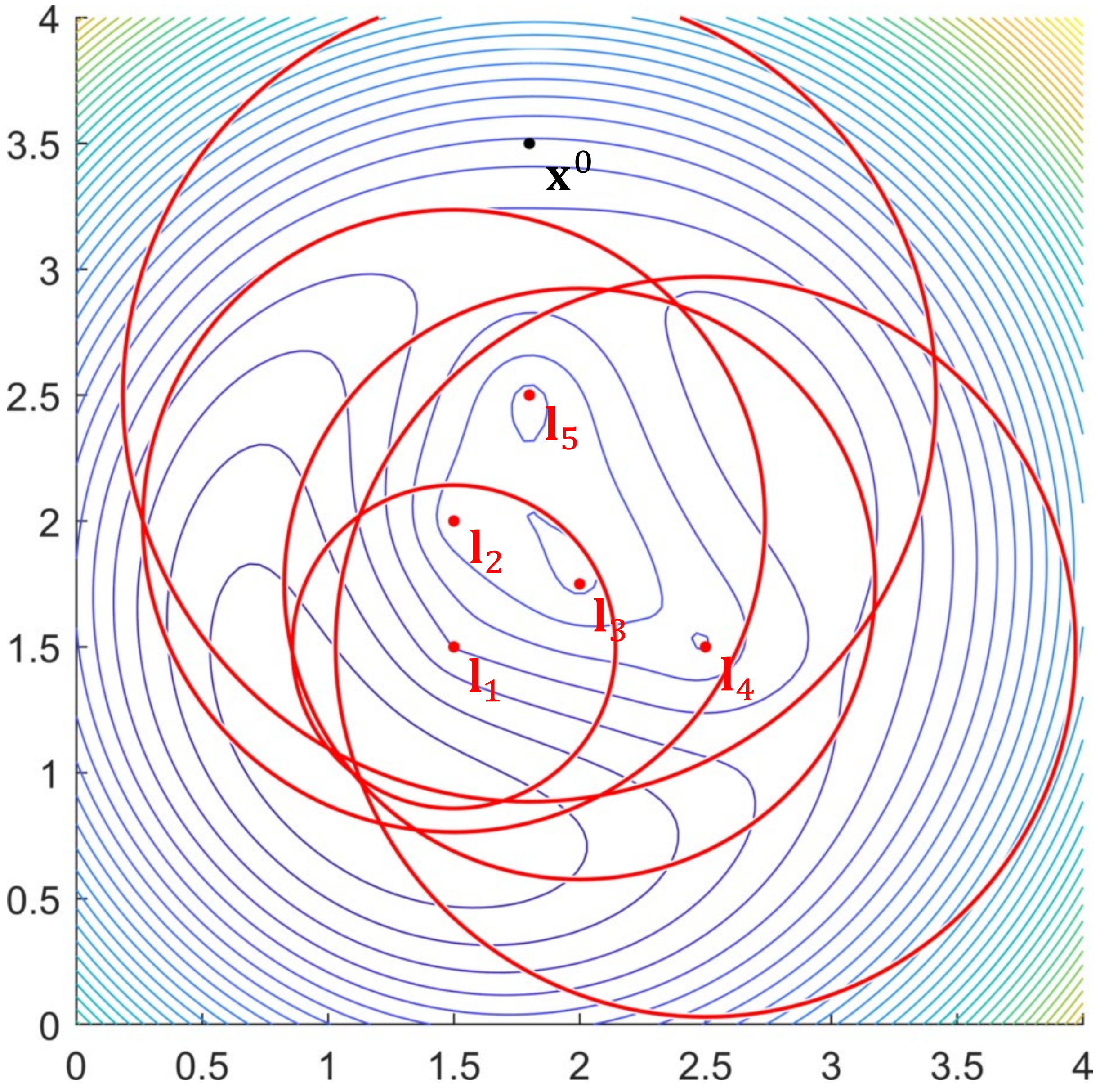
$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$

Example:
Range-based localization

Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$



Example:
Range-based localization

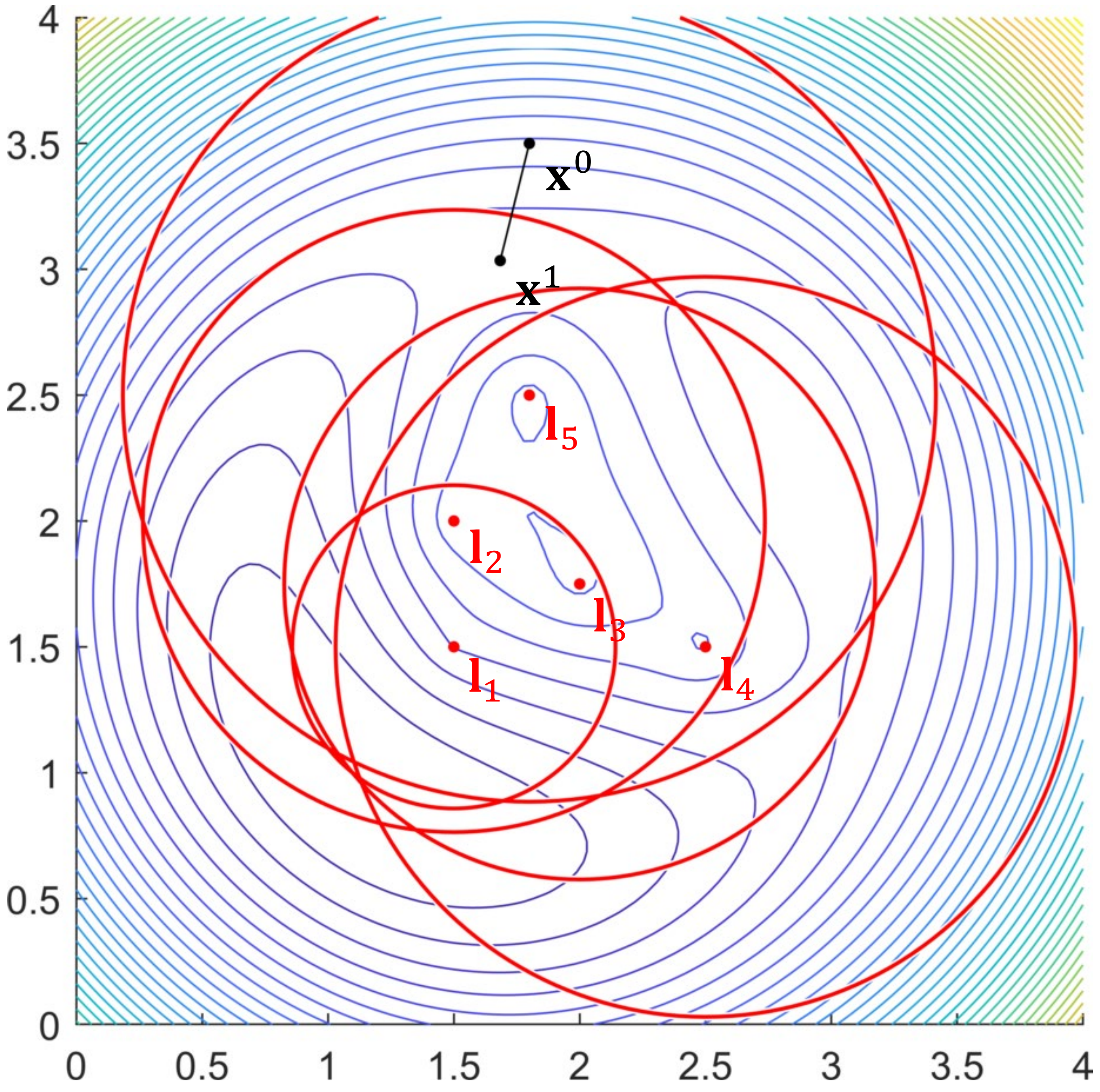
Linearized problem at $\mathbf{x}^0$:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \left\| \mathbf{A}\boldsymbol{\delta} - \mathbf{b} \right\|^2$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.99 \\ 0.20 & 0.98 \\ -0.11 & 0.99 \\ -0.33 & 0.94 \\ 0 & 1.00 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -1.38 \\ -0.29 \\ -0.59 \\ -0.65 \\ 0.62 \end{bmatrix}$$

Solution to the normal equations $\mathbf{A}^T\mathbf{A}\boldsymbol{\delta}^* = \mathbf{A}^T\mathbf{b}$:

$$\boldsymbol{\delta}^* = \begin{bmatrix} -0.12 \\ -0.47 \end{bmatrix} \qquad \mathbf{x}^1 = \mathbf{x}^0 + \boldsymbol{\delta}^* = \begin{bmatrix} 1.68 \\ 3.03 \end{bmatrix}$$



Solving the nonlinear problem

We solve the nonlinear least-squares problem
by iteratively solving the linearized system:

$$\text{Choose a suitable inital estimate } \underline{\hat{\mathcal{X}}}^0$$

$$\Downarrow$$

$$\mathbf{A}, \mathbf{b} \leftarrow \text{Linearize at } \underline{\hat{\mathcal{X}}}^t$$

$$\Downarrow$$

$$\underline{\boldsymbol{\tau}}^* \leftarrow \mathcal{S}\text{olve } \underset{\underline{\boldsymbol{\tau}}}{\operatorname{argmin}} \left\| \mathbf{A}\underline{\boldsymbol{\tau}} - \mathbf{b} \right\|^2$$

$$\Downarrow$$

$$\underline{\hat{\mathcal{X}}}^{t+1} \leftarrow \underline{\hat{\mathcal{X}}}^t \oplus \underline{\boldsymbol{\tau}}^*$$

The Gauss-Newton algorithm

Data: An objective function $f(\underline{\mathcal{X}})$ and a good initial state estimate $\hat{\underline{\mathcal{X}}}^0$
Result: An estimate for the states $\hat{\underline{\mathcal{X}}}$

for $t = 0, 1, \dots, t^{max}$ **do**
  $\mathbf{A}, \mathbf{b} \leftarrow$ Linearise $f(\underline{\mathcal{X}})$ at $\hat{\underline{\mathcal{X}}}^t$
  $\underline{\boldsymbol{\tau}} \leftarrow$ Solve the linearised problem $\mathbf{A}^\top \mathbf{A} \underline{\boldsymbol{\tau}} = \mathbf{A}^\top \mathbf{b}$
  $\hat{\underline{\mathcal{X}}}^{t+1} \leftarrow \hat{\underline{\mathcal{X}}}^t \oplus \underline{\boldsymbol{\tau}}$

  if $f(\hat{\underline{\mathcal{X}}}^{t+1})$ *is very small or* $\hat{\underline{\mathcal{X}}}^{t+1} \approx \hat{\underline{\mathcal{X}}}^t$ **then**
    $\hat{\underline{\mathcal{X}}} \leftarrow \hat{\underline{\mathcal{X}}}^{t+1}$
    return
  end
end

The Gauss-Newton algorithm

Gauss-Newton actually approximates the Hessian of the objective $f(\underline{\mathcal{X}})$ at $\hat{\underline{\mathcal{X}}}$ as

$$\frac{\partial^2 f(\hat{\underline{\mathcal{X}}})}{\partial\hat{\underline{\mathcal{X}}}\partial\hat{\underline{\mathcal{X}}}^T} = \left(\frac{\partial e(\hat{\underline{\mathcal{X}}})}{\partial\hat{\underline{\mathcal{X}}}}\right)^T \left(\frac{\partial e(\hat{\underline{\mathcal{X}}})}{\partial\hat{\underline{\mathcal{X}}}}\right) + \sum_{i=1}^{m} e_i(\hat{\underline{\mathcal{X}}}_i)\left(\frac{\partial^2 e_i(\hat{\underline{\mathcal{X}}}_i)}{\partial\hat{\underline{\mathcal{X}}}_i\partial\hat{\underline{\mathcal{X}}}_i^T}\right) = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{A}$$

This approximation is good if we are near the solution and the objective is nearly quadratic.

The Gauss-Newton algorithm

Gauss-Newton actually approximates the Hessian of the objective $f(\underline{\mathcal{X}})$ at $\hat{\underline{\mathcal{X}}}$ as

$$\frac{\partial^2 f(\hat{\underline{\mathcal{X}}})}{\partial \hat{\underline{\mathcal{X}}} \partial \hat{\underline{\mathcal{X}}}^T} = \left(\frac{\partial e(\hat{\underline{\mathcal{X}}})}{\partial \hat{\underline{\mathcal{X}}}}\right)^T \left(\frac{\partial e(\hat{\underline{\mathcal{X}}})}{\partial \hat{\underline{\mathcal{X}}}}\right) + \sum_{i=1}^{m} e_i(\hat{\underline{\mathcal{X}}}_i) \left(\frac{\partial^2 e_i(\hat{\underline{\mathcal{X}}}_i)}{\partial \hat{\underline{\mathcal{X}}}_i \partial \hat{\underline{\mathcal{X}}}_i^T}\right) = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{A}$$

This approximation is good if we are near the solution and the objective is nearly quadratic.

When the approximation is good:

- The update direction is good
- The update step length is good
- We obtain almost quadratic convergence to a local minimum

The Gauss-Newton algorithm

Gauss-Newton actually approximates the Hessian of the objective $f(\underline{\mathcal{X}})$ at $\hat{\underline{\mathcal{X}}}$ as

$$\frac{\partial^2 f(\hat{\underline{\mathcal{X}}})}{\partial\hat{\underline{\mathcal{X}}}\partial\hat{\underline{\mathcal{X}}}^T} = \left(\frac{\partial e(\hat{\underline{\mathcal{X}}})}{\partial\hat{\underline{\mathcal{X}}}}\right)^T \left(\frac{\partial e(\hat{\underline{\mathcal{X}}})}{\partial\hat{\underline{\mathcal{X}}}}\right) + \sum_{i=1}^{m} e_i(\hat{\underline{\mathcal{X}}}_i)\left(\frac{\partial^2 e_i(\hat{\underline{\mathcal{X}}}_i)}{\partial\hat{\underline{\mathcal{X}}}_i\partial\hat{\underline{\mathcal{X}}}_i^T}\right) = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{A}$$

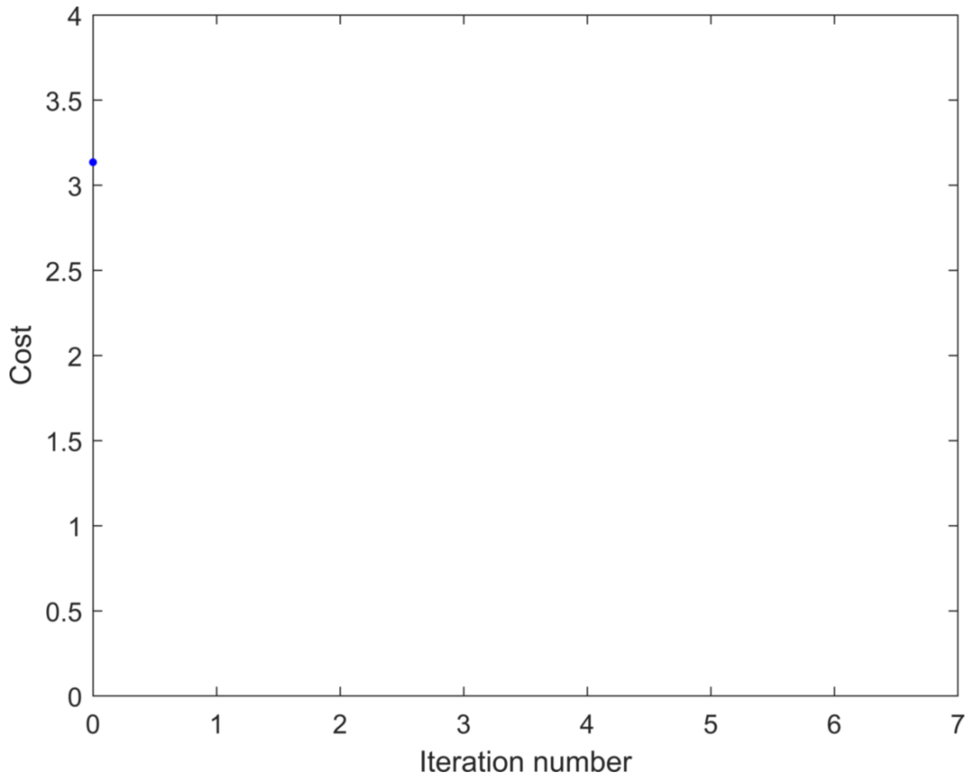
This approximation is good if we are near the solution and the objective is nearly quadratic.

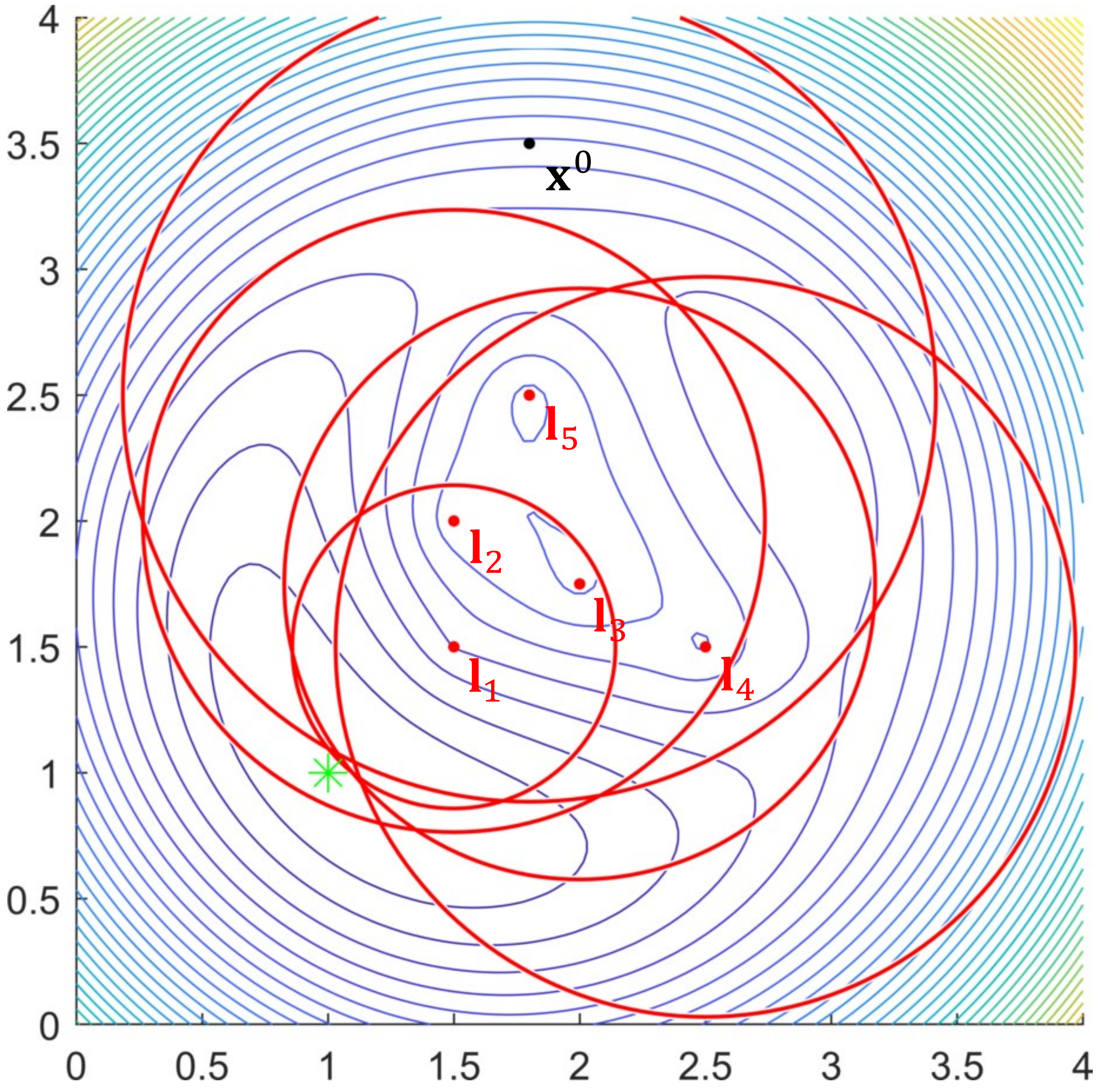
When the approximation is poor:

- The update direction is typically still decent
- The update step length may be bad
- The convergence is slower, and we may even diverge

Example: Range-based localization

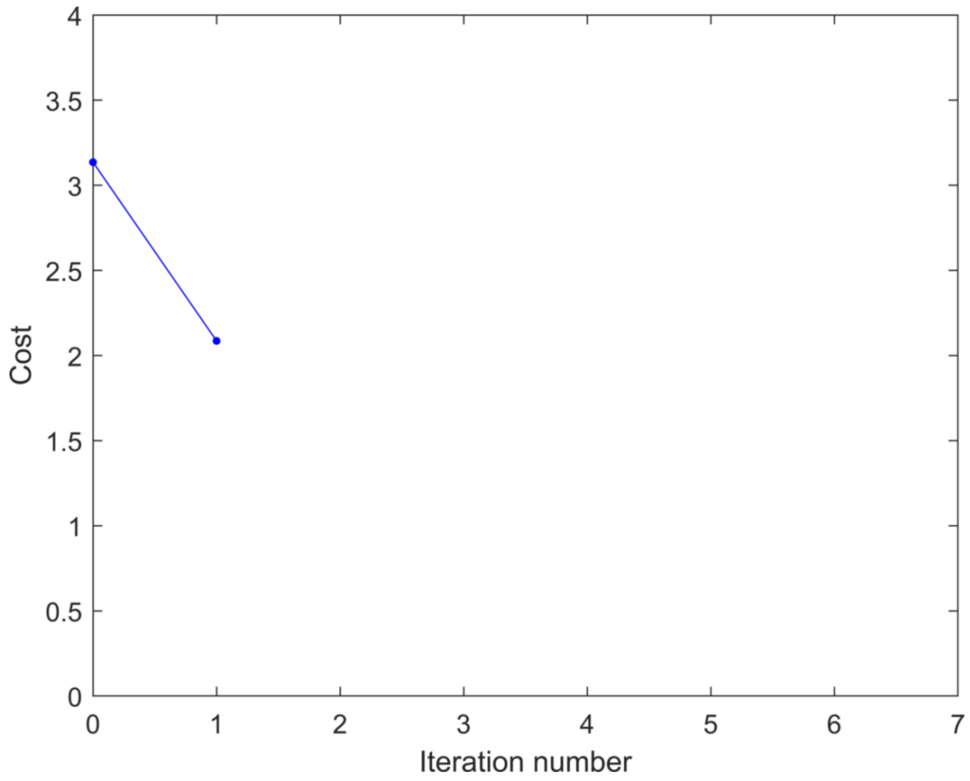
Gauss-Newton optimization

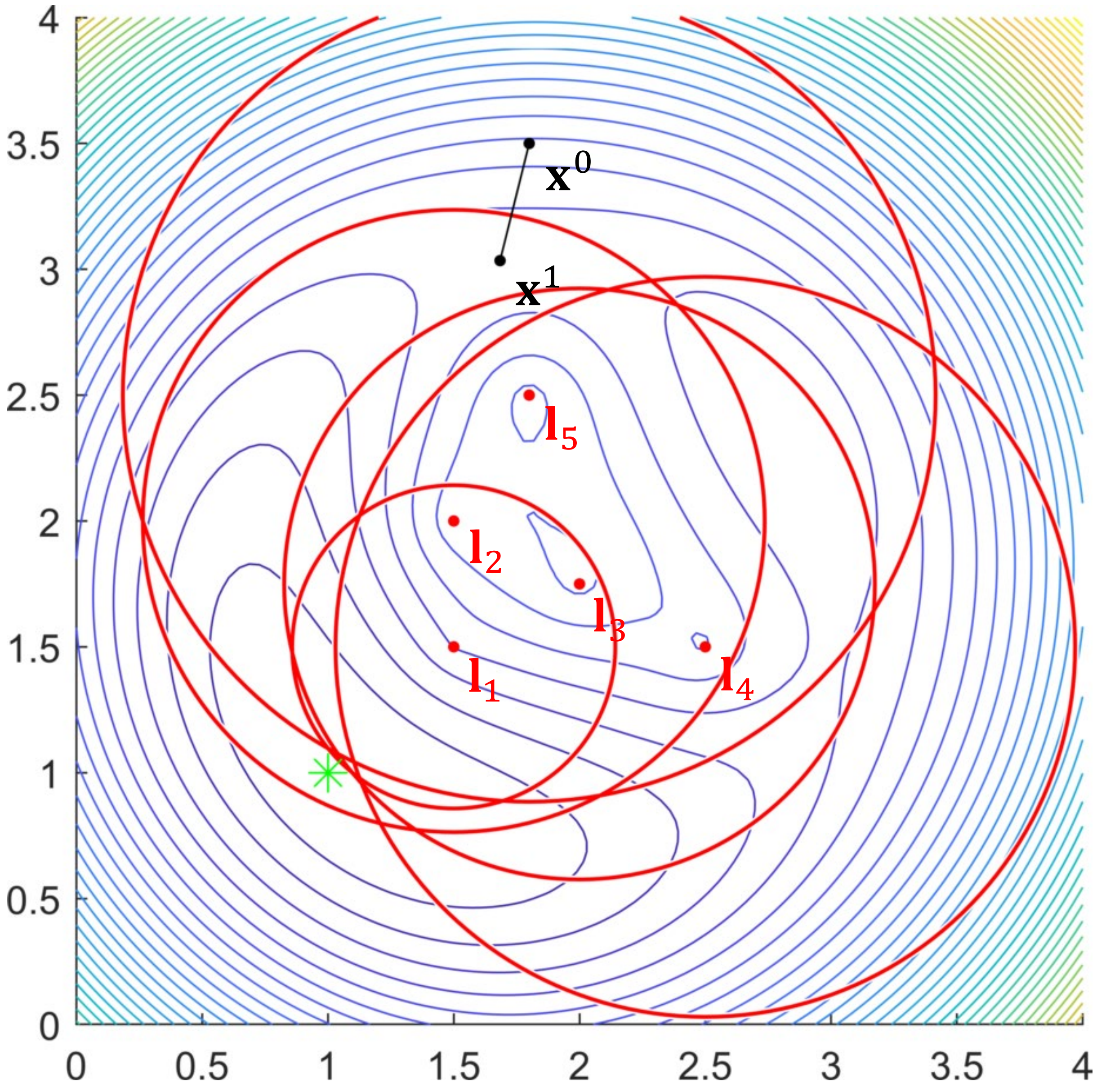




Example:
Range-based localization

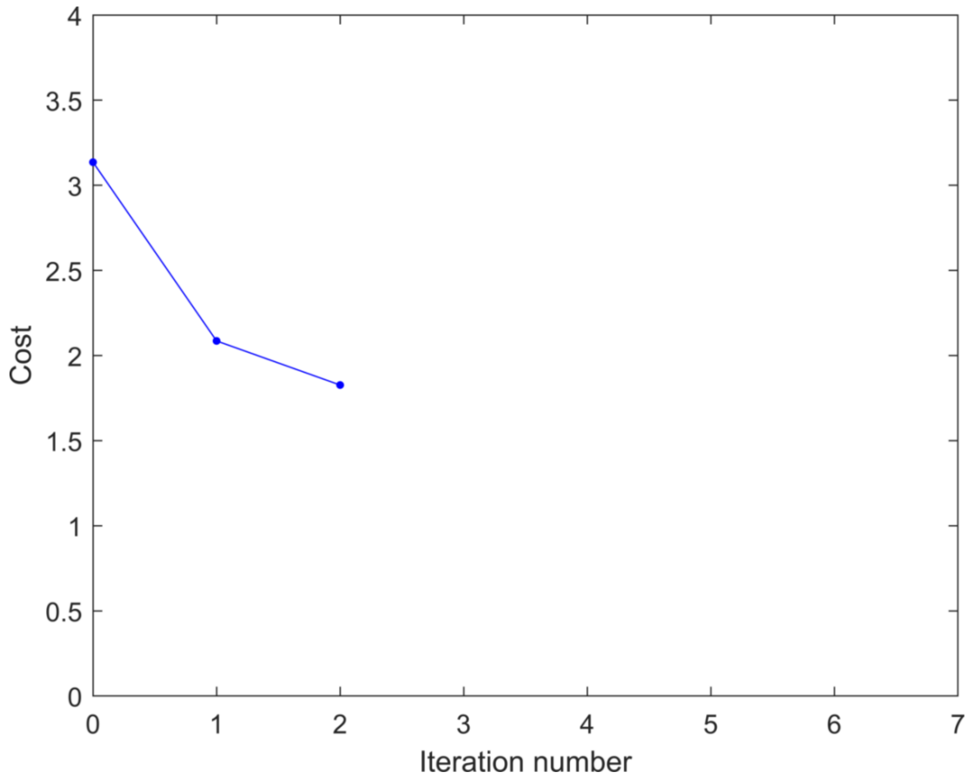
Gauss-Newton optimization

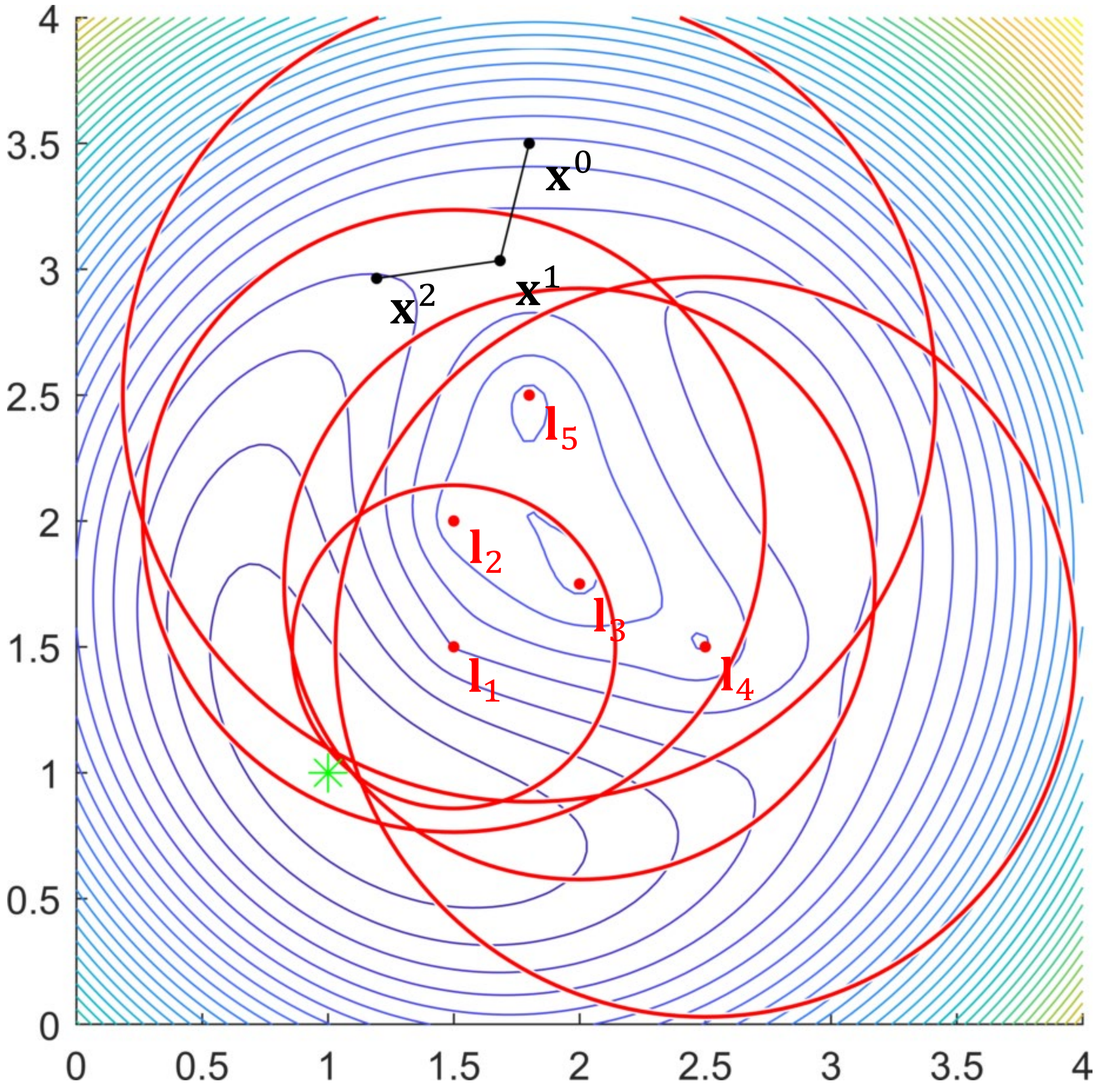




Example:
Range-based localization

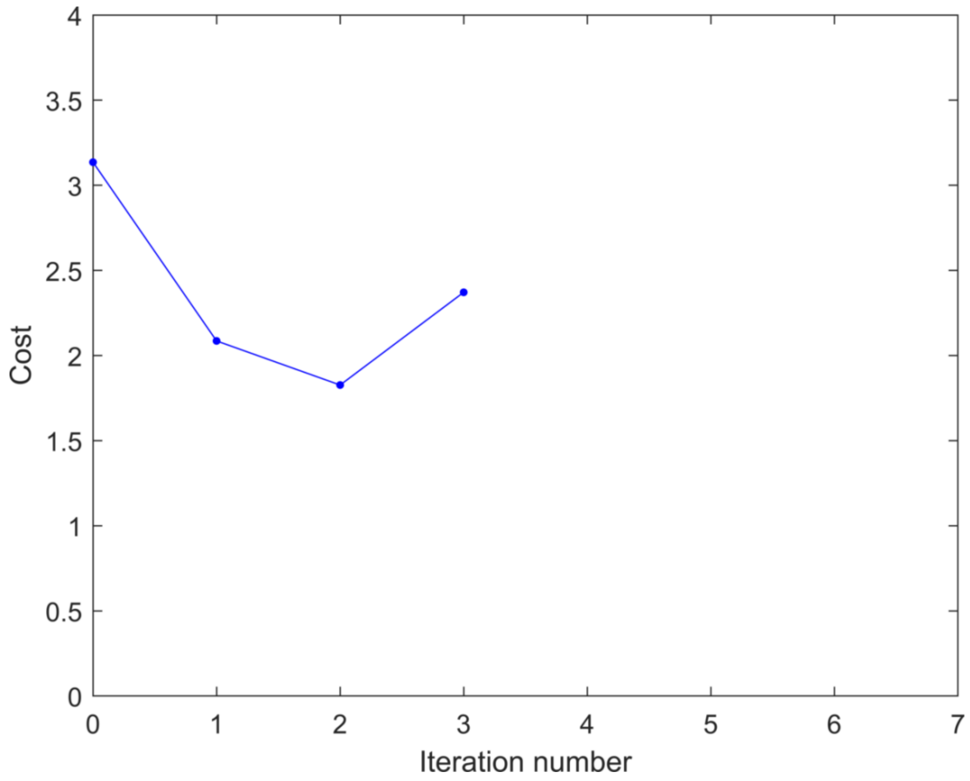
Gauss-Newton optimization

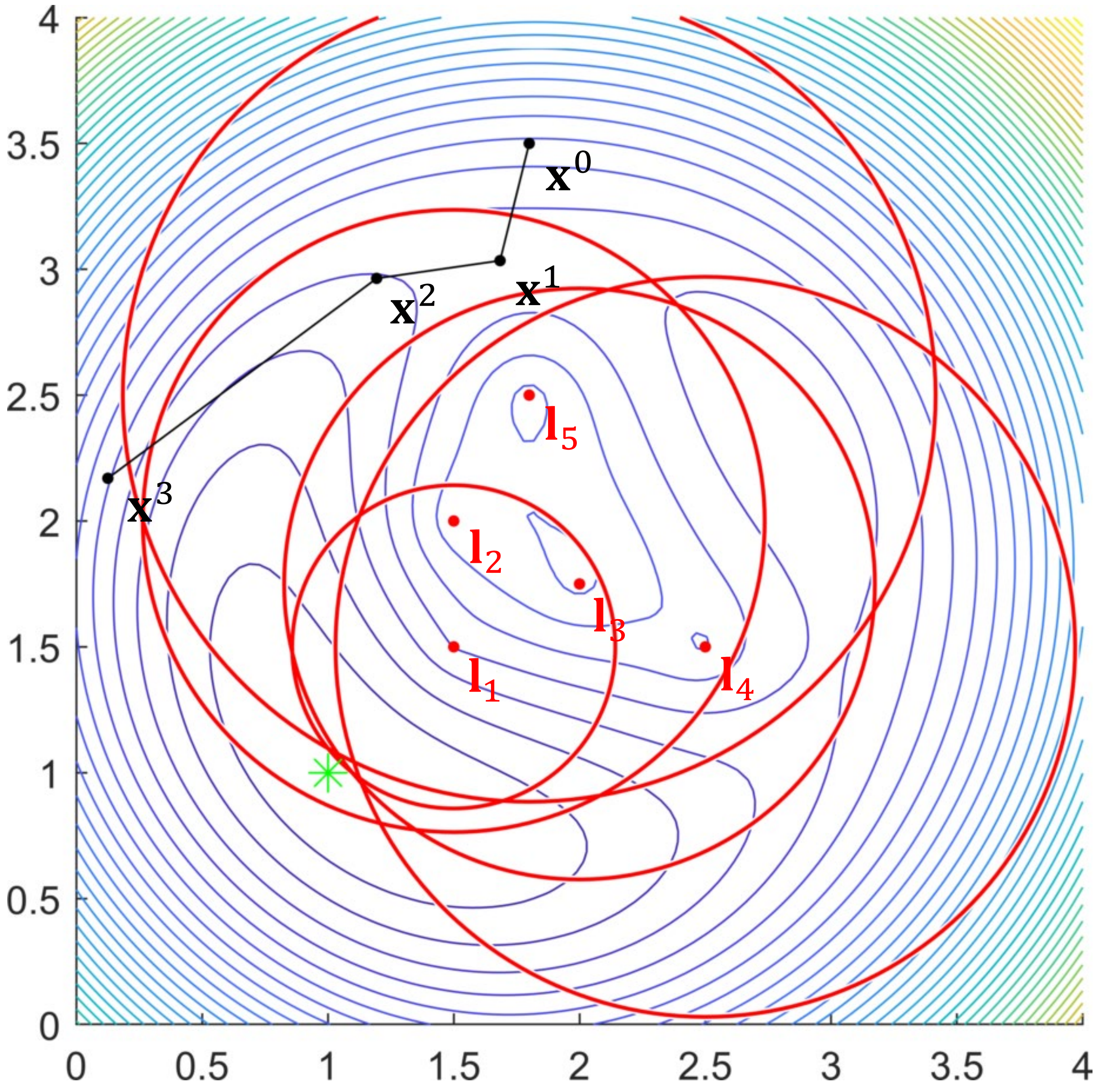




Example:
Range-based localization

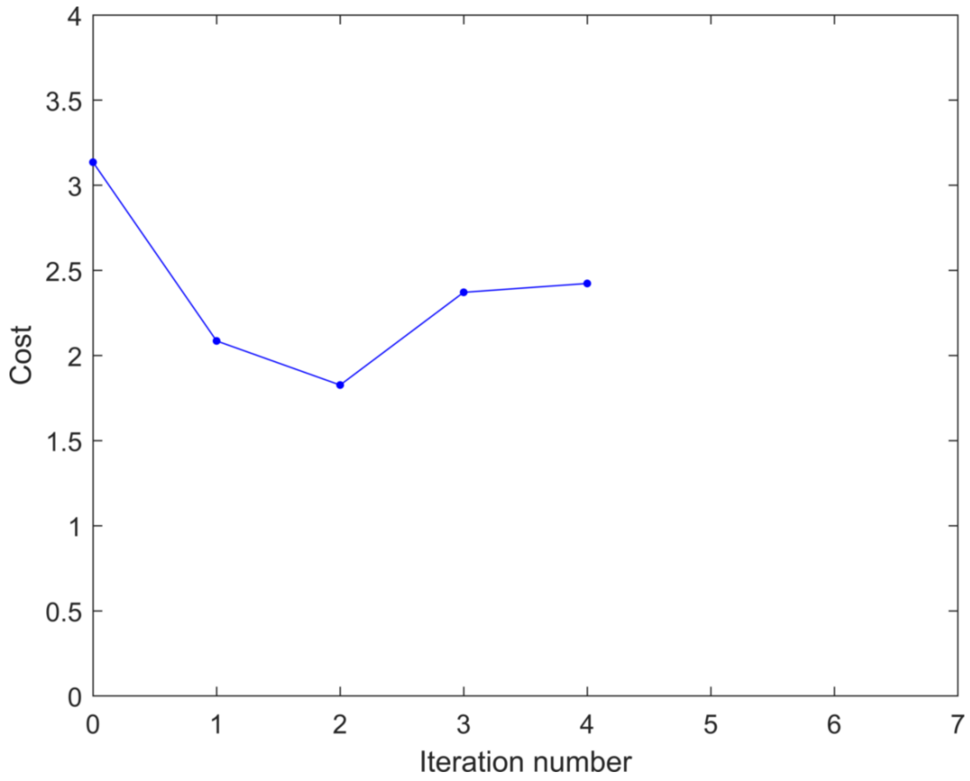
Gauss-Newton optimization

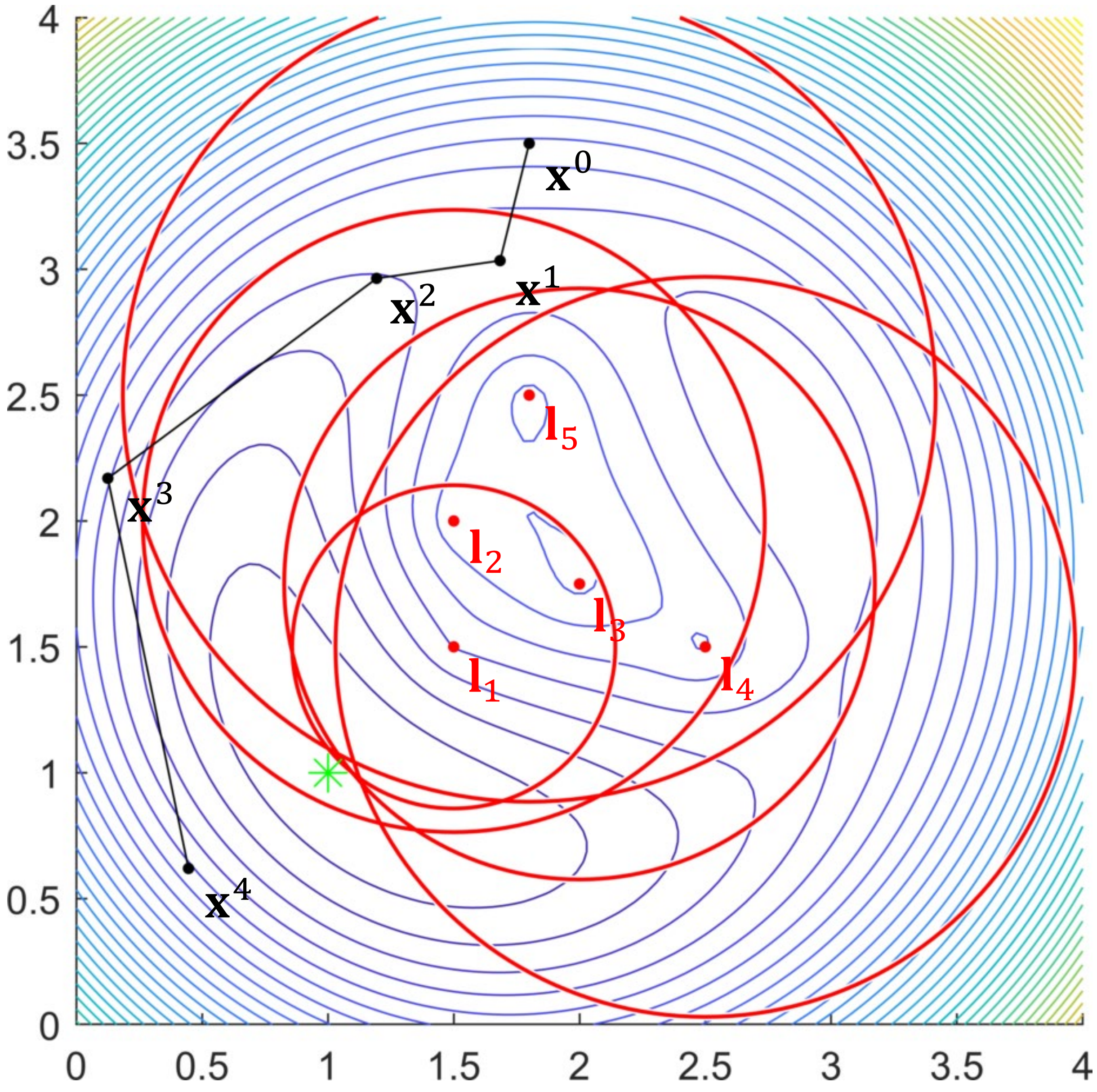




Example:
Range-based localization

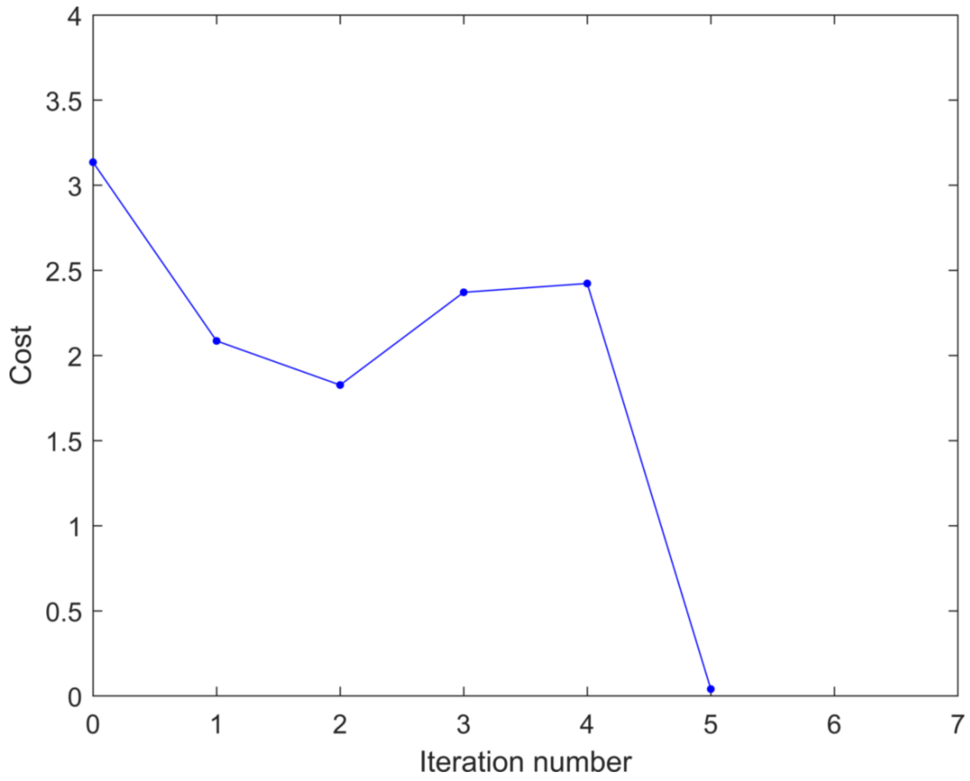
Gauss-Newton optimization

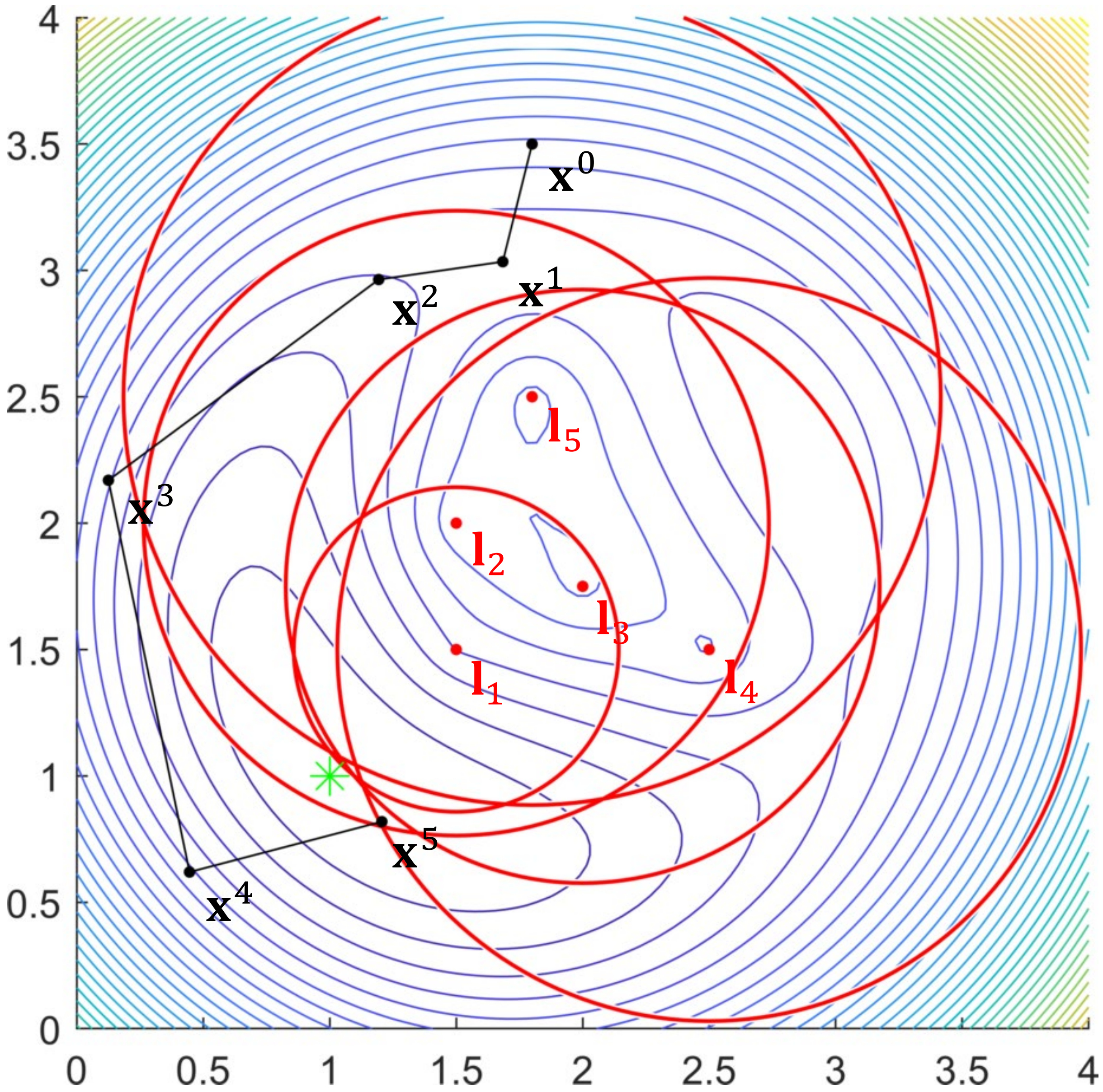




Example:
Range-based localization

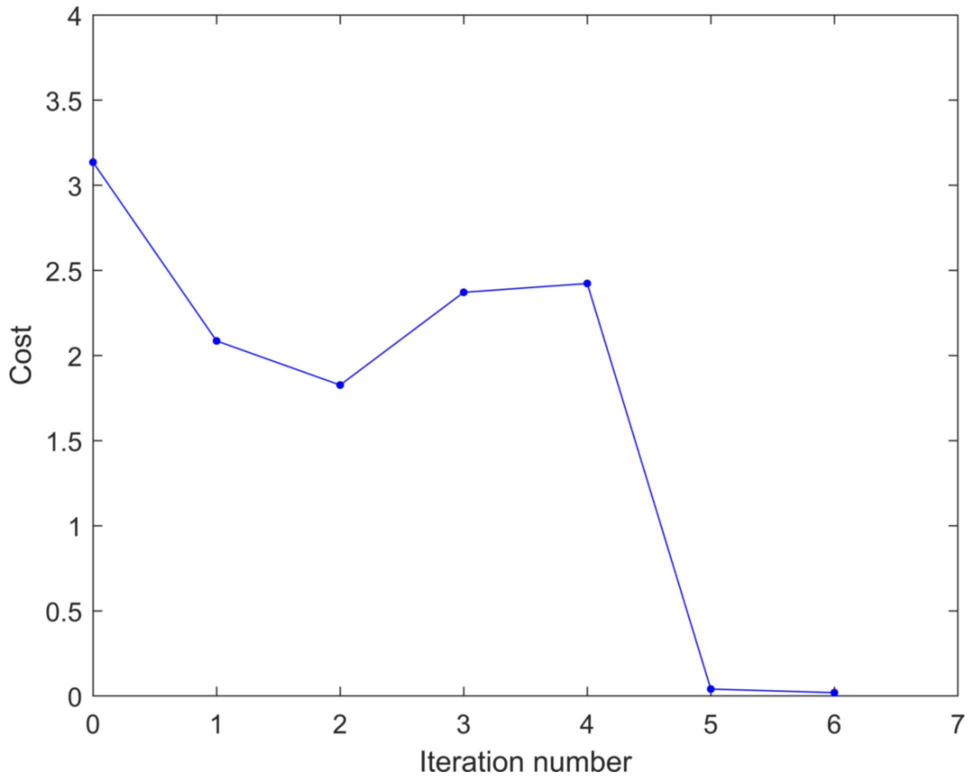
Gauss-Newton optimization

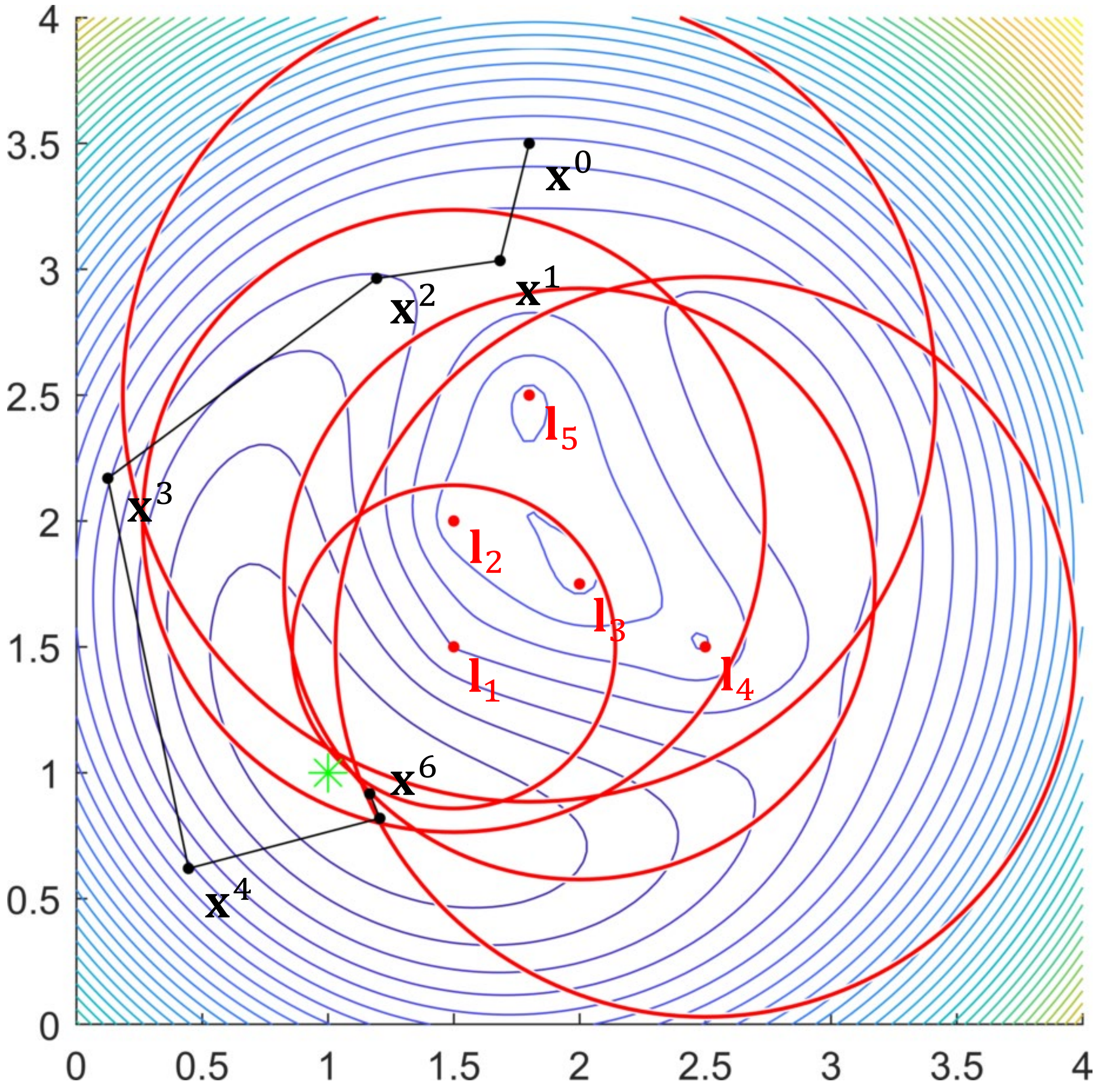




Example:
Range-based localization

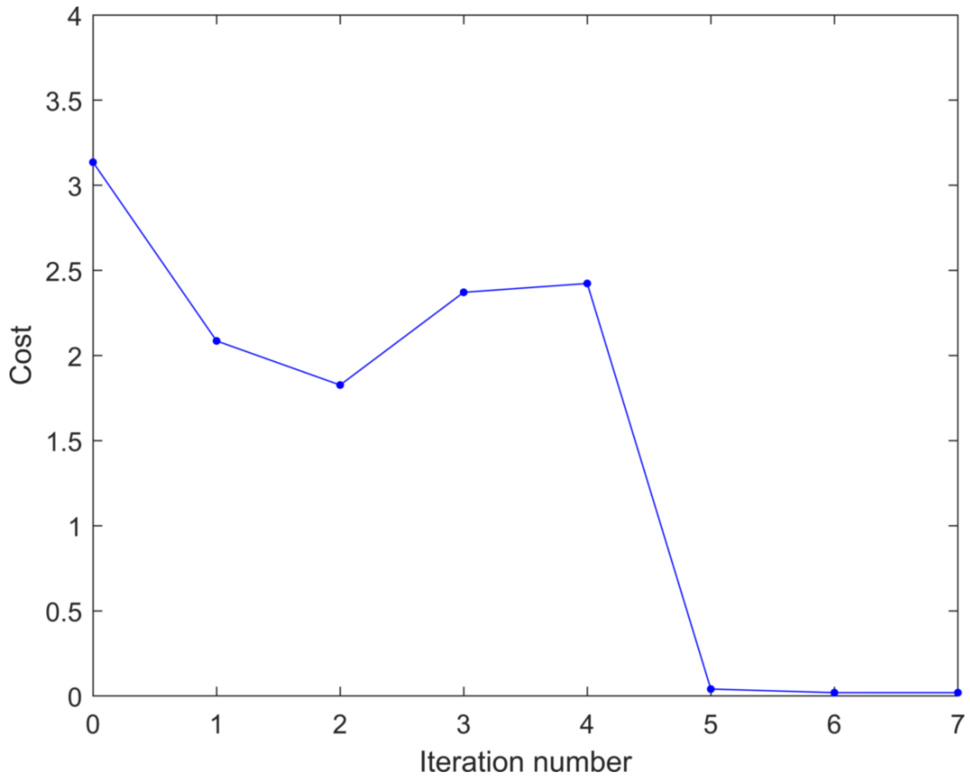
Gauss-Newton optimization

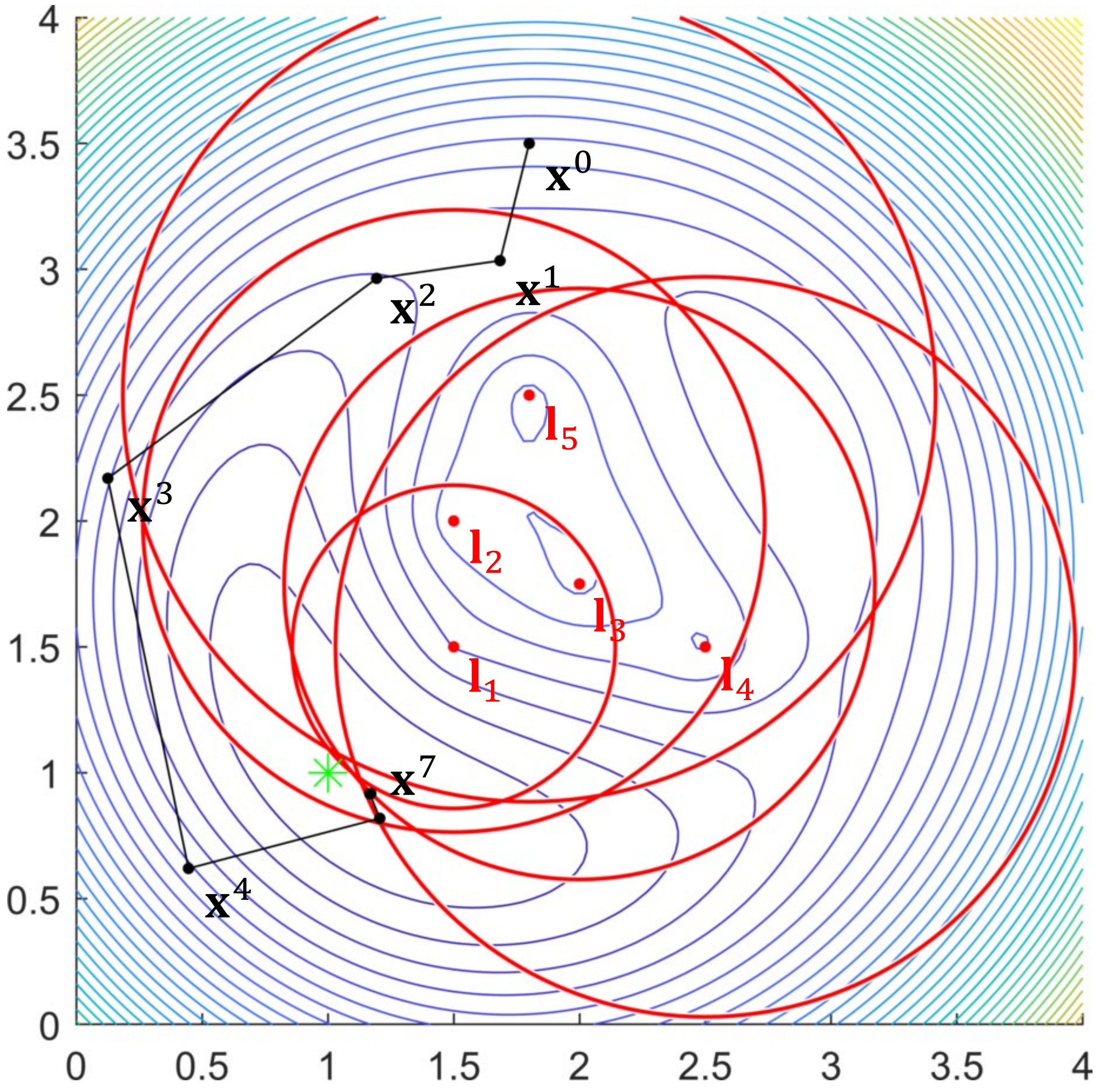




Example: Range-based localization

Gauss-Newton optimization





Trust region

- The Gauss-Newton method is not guaranteed to converge because of the approximate Hessian matrix
- Since the update directions typically are decent, we can help with convergence by limiting the step sizes
  - More conservative towards robustness, rather than speed
- Such methods are often called **trust region methods**, and one example is **Levenberg-Marquardt**

The Levenberg–Marquardt algorithm

Data: An objective function $f(\mathcal{X})$ and a good initial state estimate $\mathcal{X}^0$
Result: An estimate for the states $\hat{\mathcal{X}}$

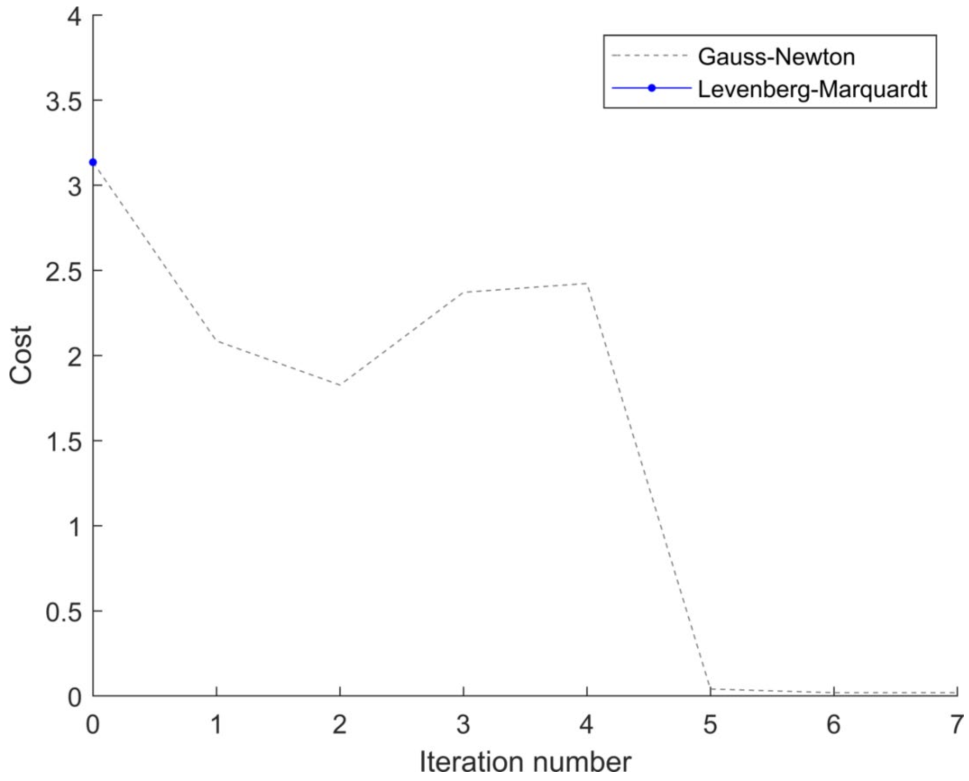
$\lambda \leftarrow 10^{-4}$

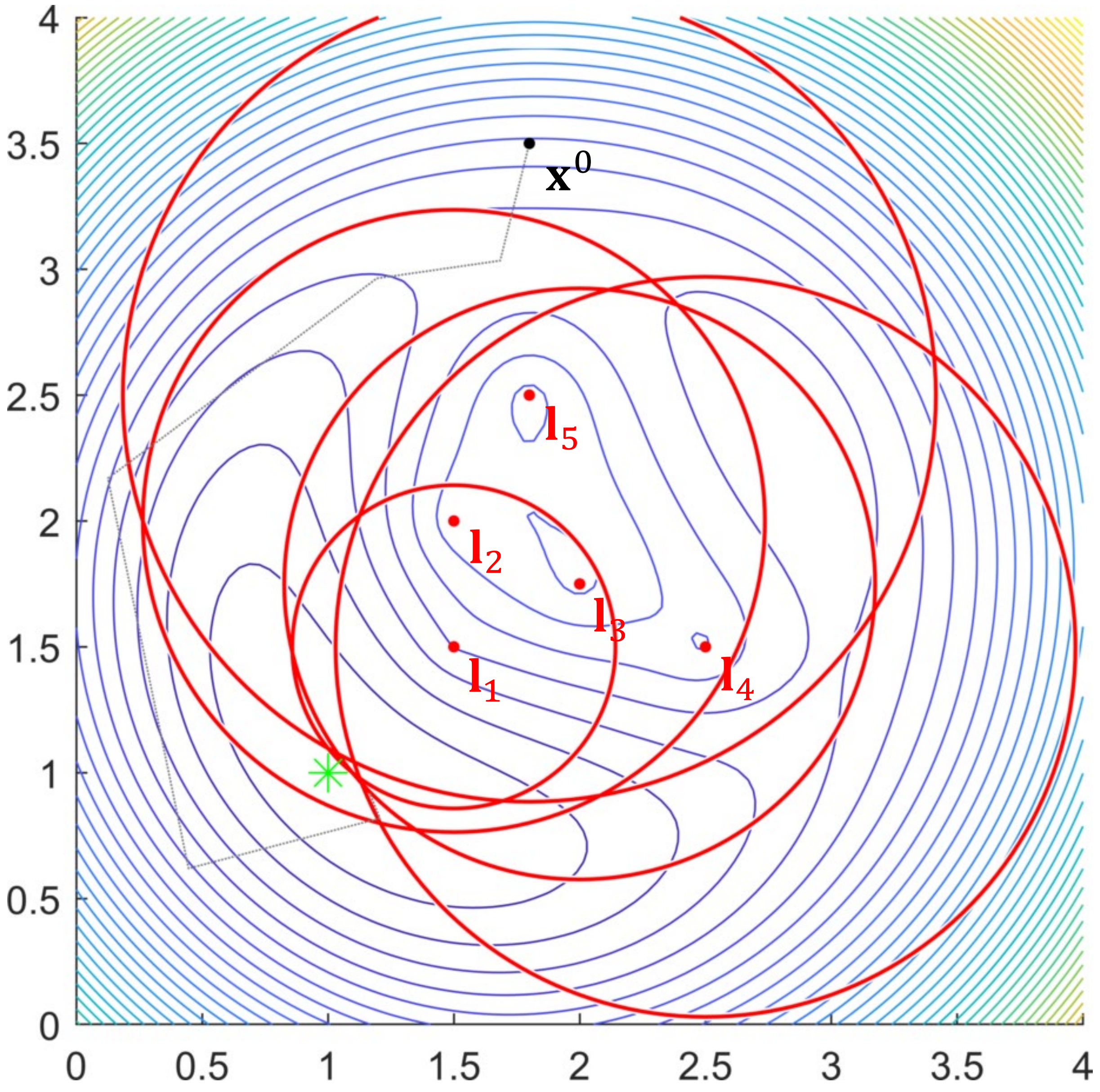
for $t = 0, 1, \ldots, t^{max}$ **do**
- $\mathbf{A}, \mathbf{b} \leftarrow$ Linearise $f(\mathcal{X})$ at $\hat{\mathcal{X}}^t$
- $\boldsymbol{\tau} \leftarrow$ Solve the linearised problem $(\mathbf{A}^\top\mathbf{A} + \lambda\,\mathrm{diag}(\mathbf{A}^\top\mathbf{A}))\boldsymbol{\tau} = \mathbf{A}^\top\mathbf{b}$
- **if** $f(\hat{\mathcal{X}}^t \oplus \boldsymbol{\tau}) < f(\hat{\mathcal{X}}^t)$ **then**
  - Accept update, increase trust region
  - $\hat{\mathcal{X}}^{t+1} \leftarrow \hat{\mathcal{X}}^t \oplus \boldsymbol{\tau}$
  - $\lambda \leftarrow \lambda/10$
- **else**
  - Reject update, reduce trust region
  - $\hat{\mathcal{X}}^{t+1} \leftarrow \hat{\mathcal{X}}^t$
  - $\lambda \leftarrow \lambda * 10$
- **end**
- **if** $f(\hat{\mathcal{X}}^{t+1})$ *is very small or* $\hat{\mathcal{X}}^{t+1} \approx \hat{\mathcal{X}}^t$ **then**
  - $\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}}^{t+1}$
  - **return**
- **end**

end

Example:
Range-based localization

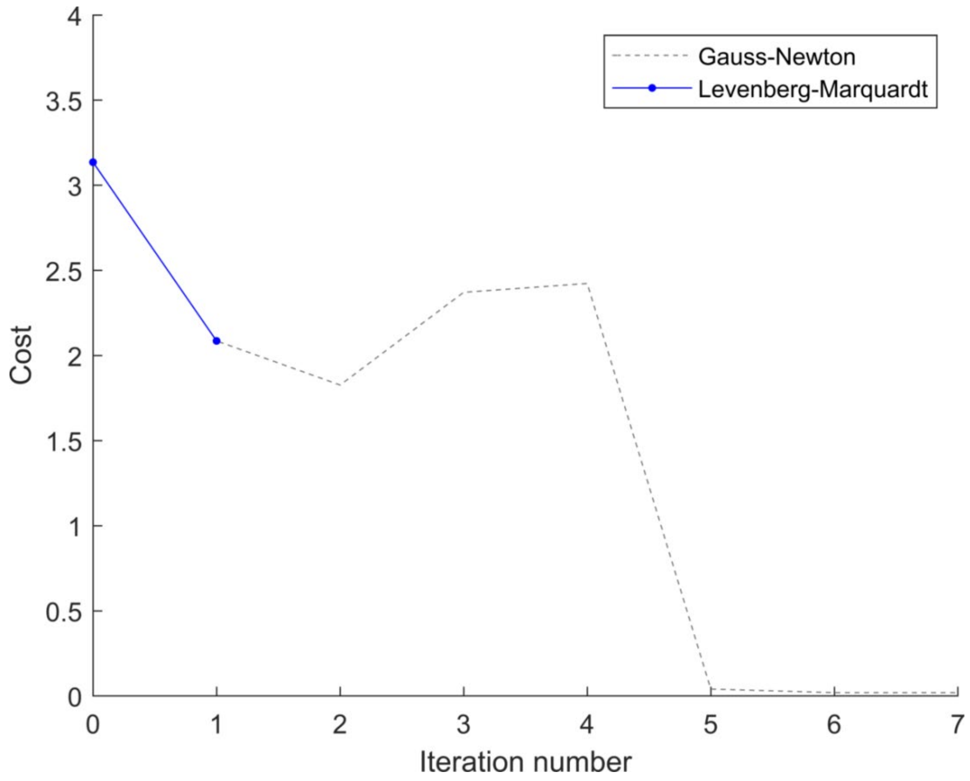
Levenberg–Marquardt optimization

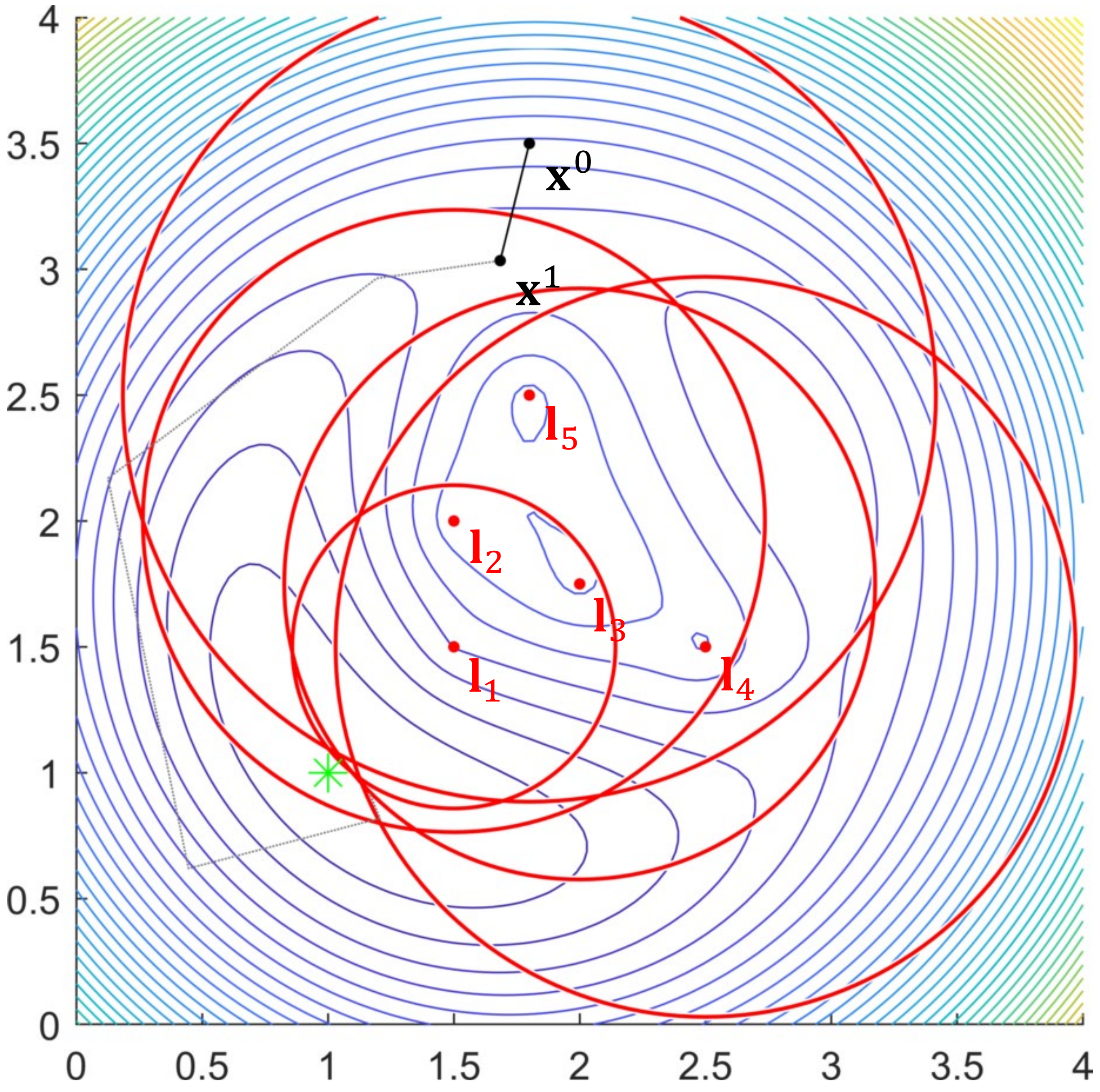




Example:
Range-based localization

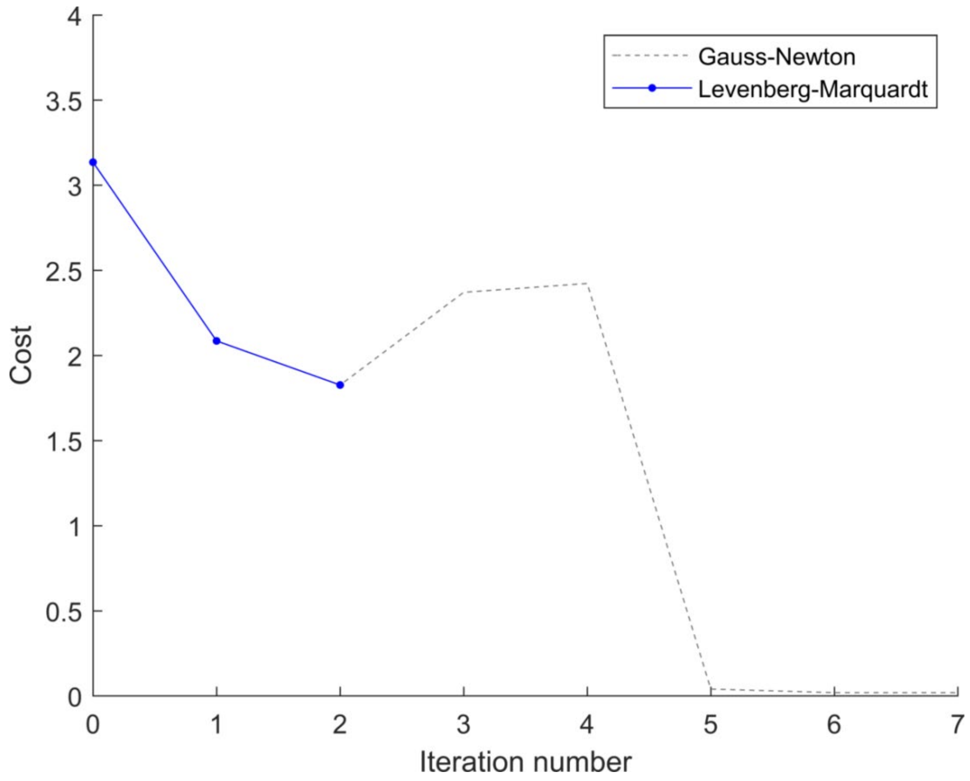
Levenberg–Marquardt optimization

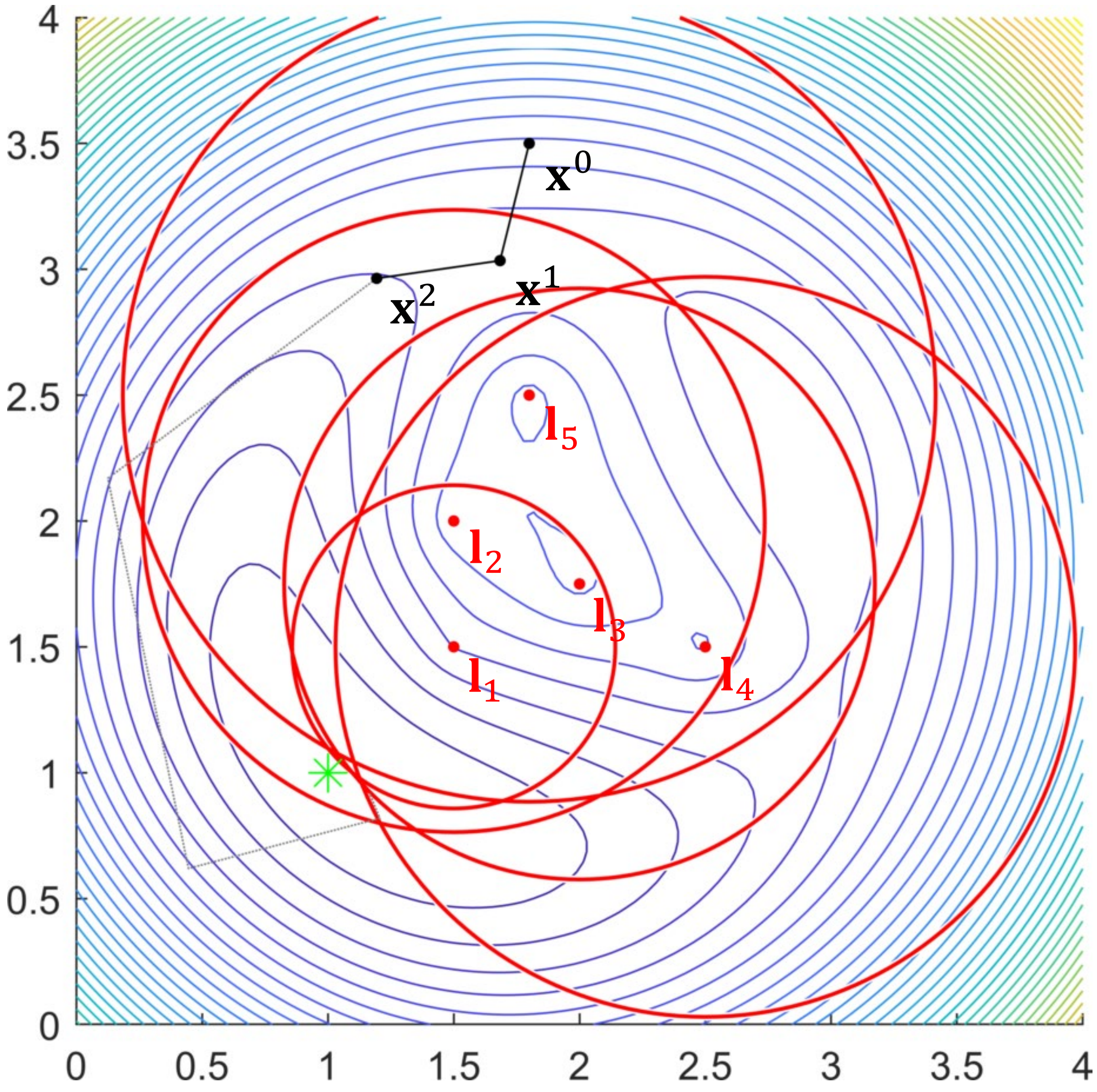




Example:
Range-based localization

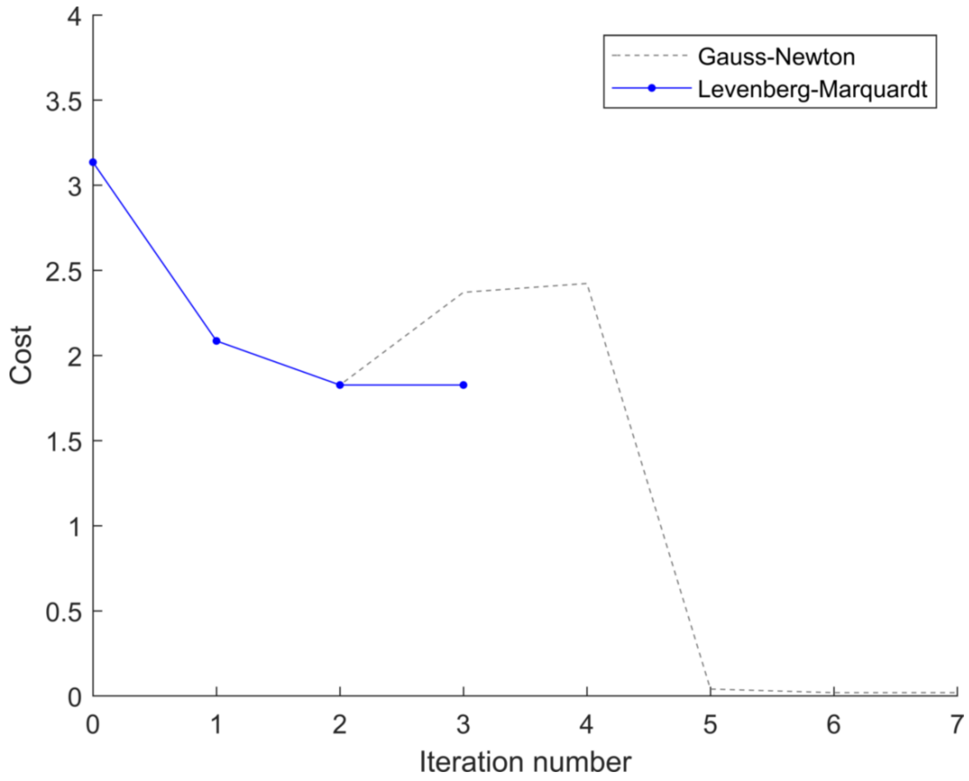
Levenberg–Marquardt optimization

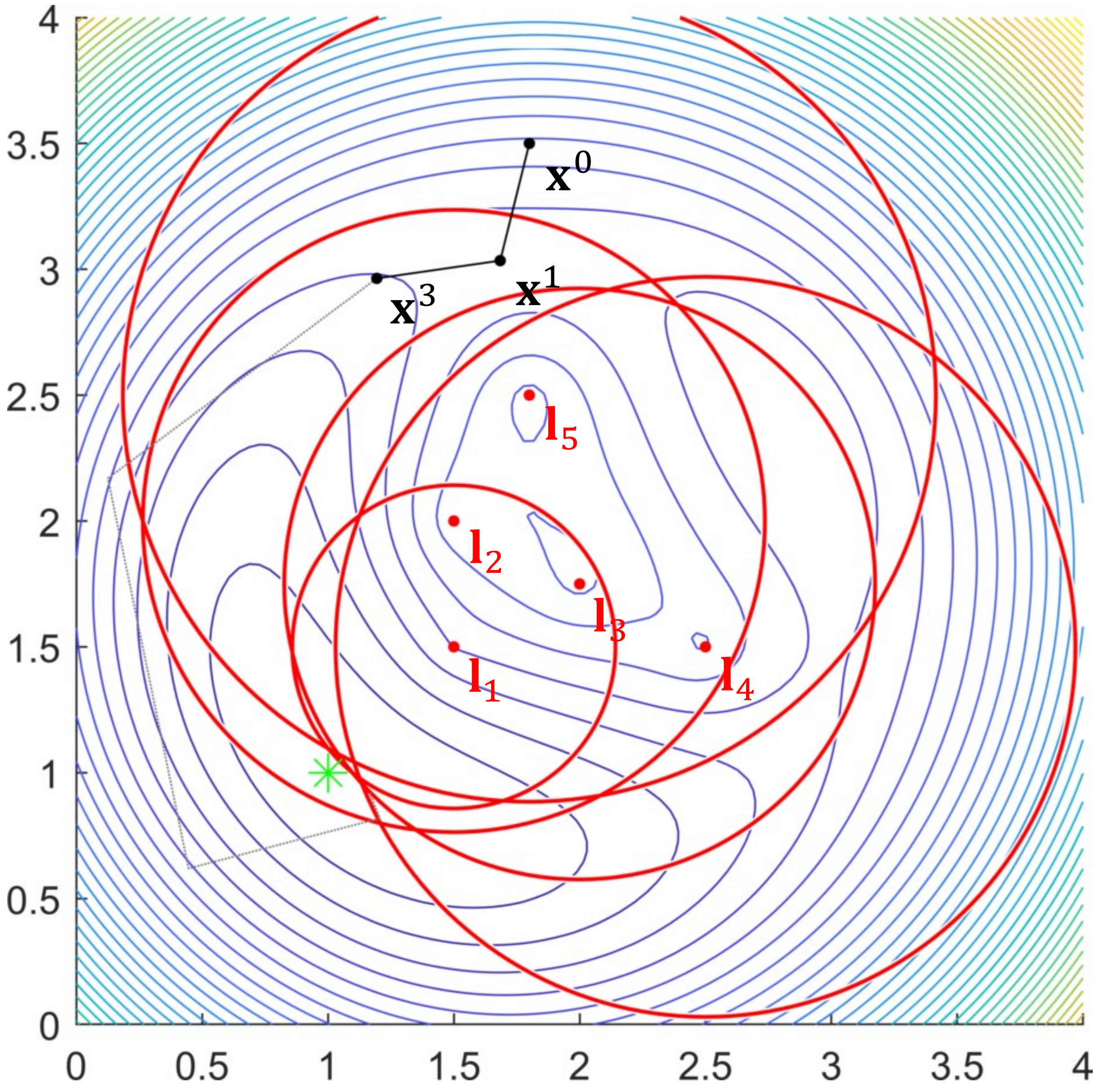




Example:
Range-based localization

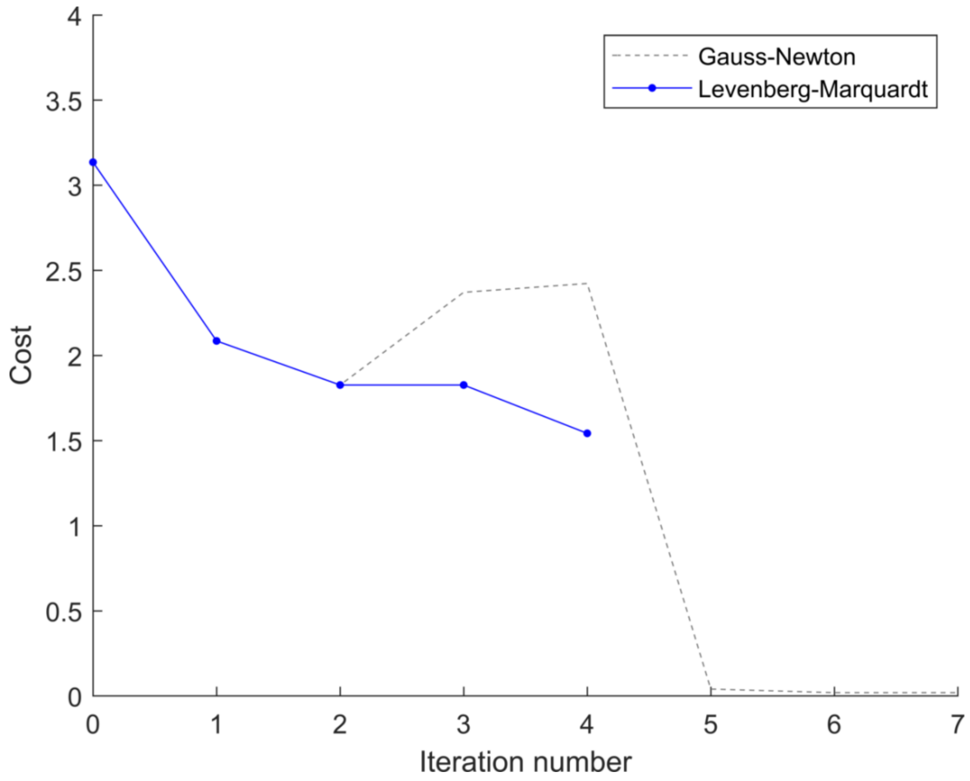
Levenberg–Marquardt optimization

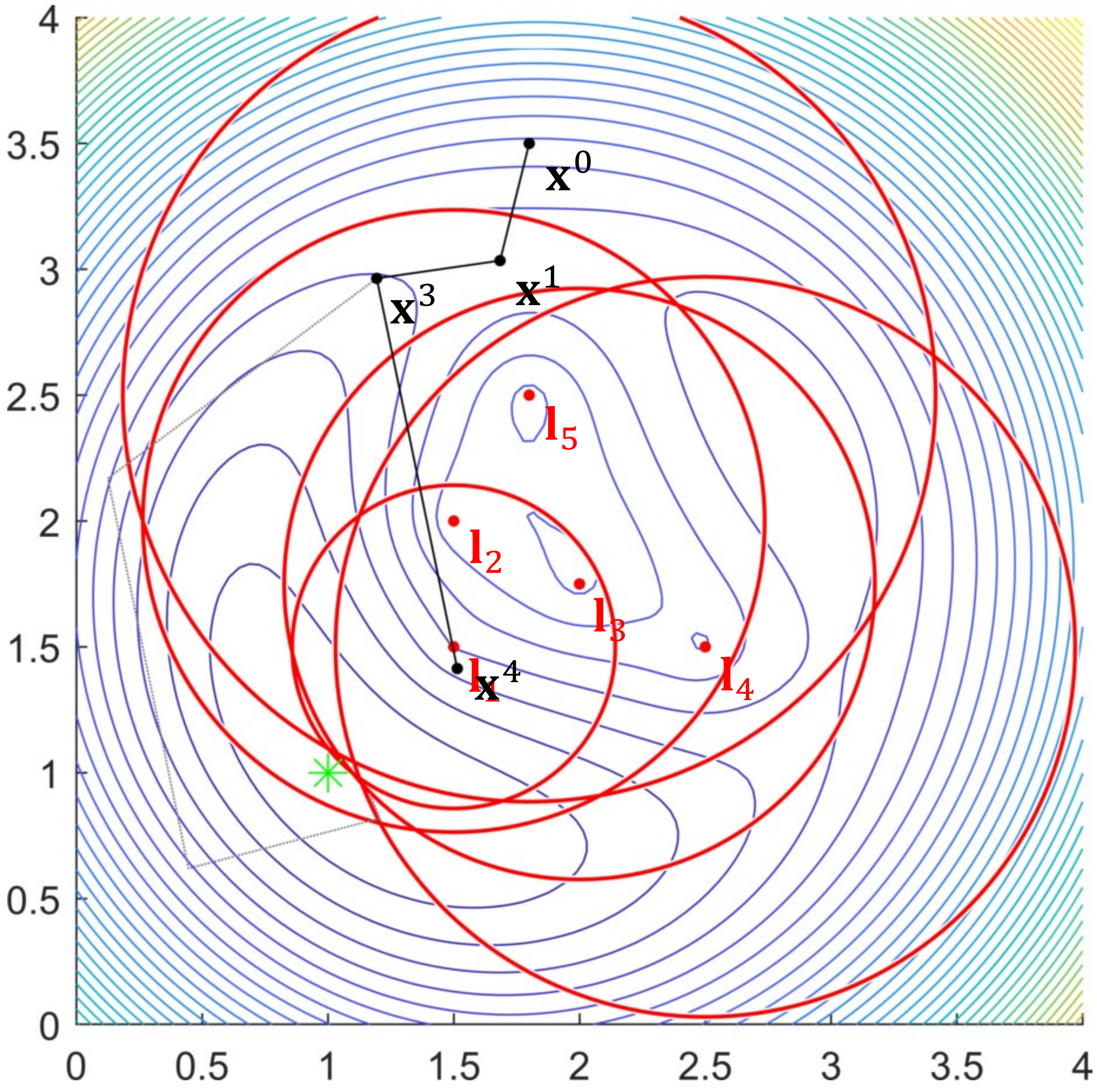




Example:
Range-based localization

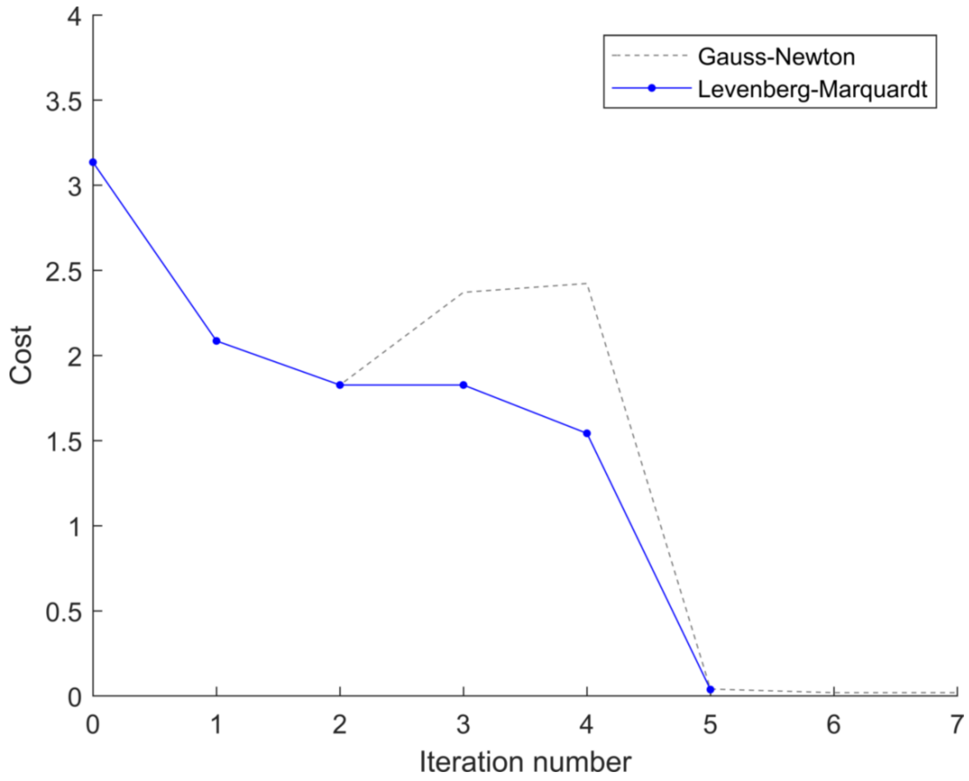
Levenberg–Marquardt optimization

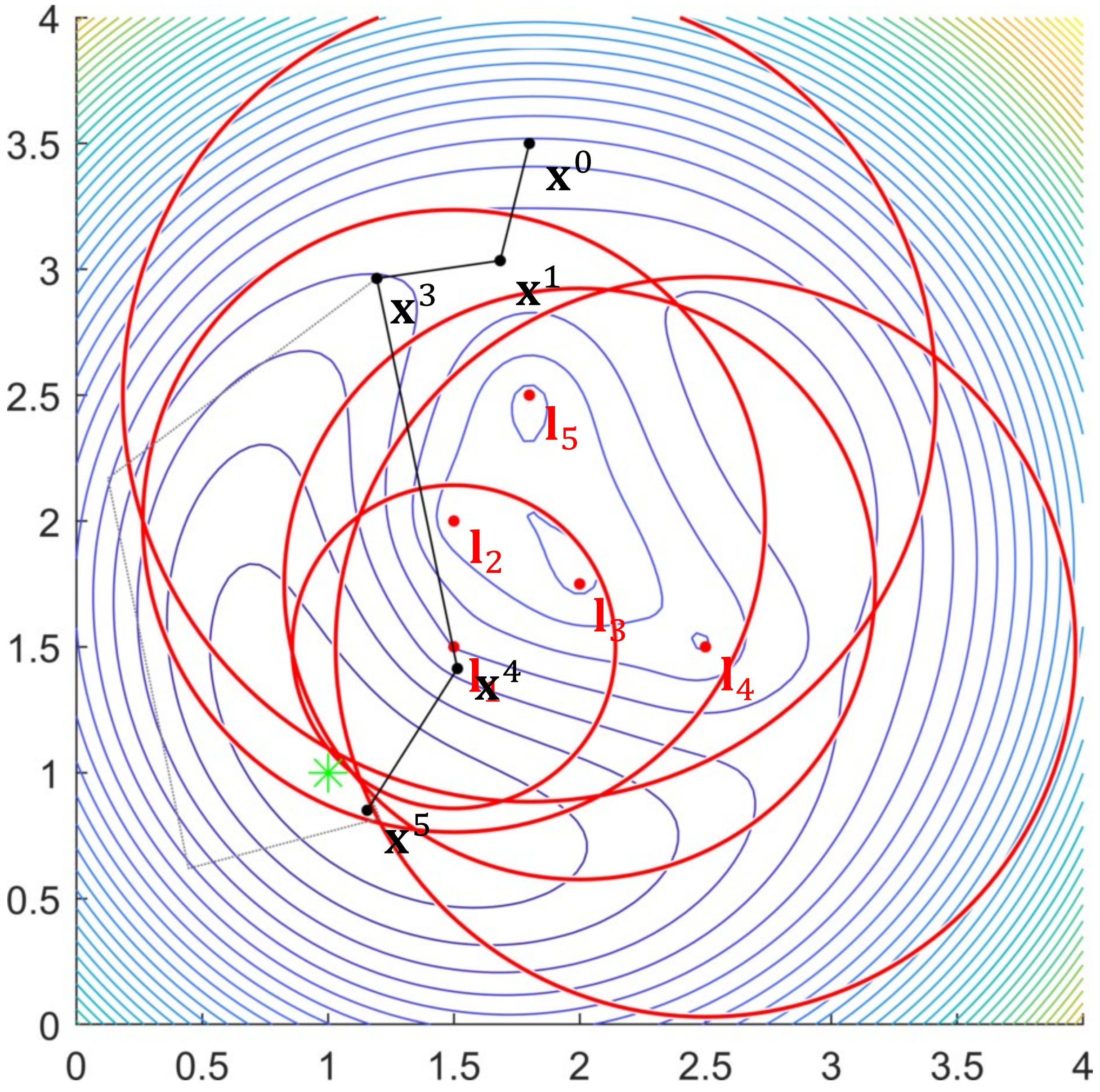




Example:
Range-based localization

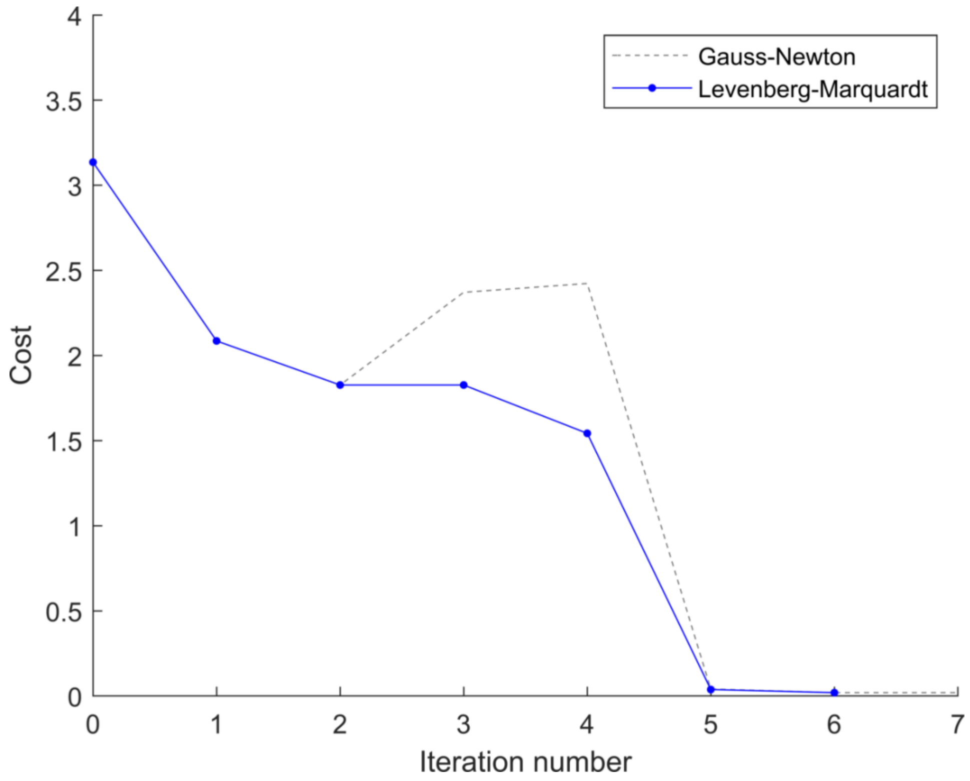
Levenberg–Marquardt optimization

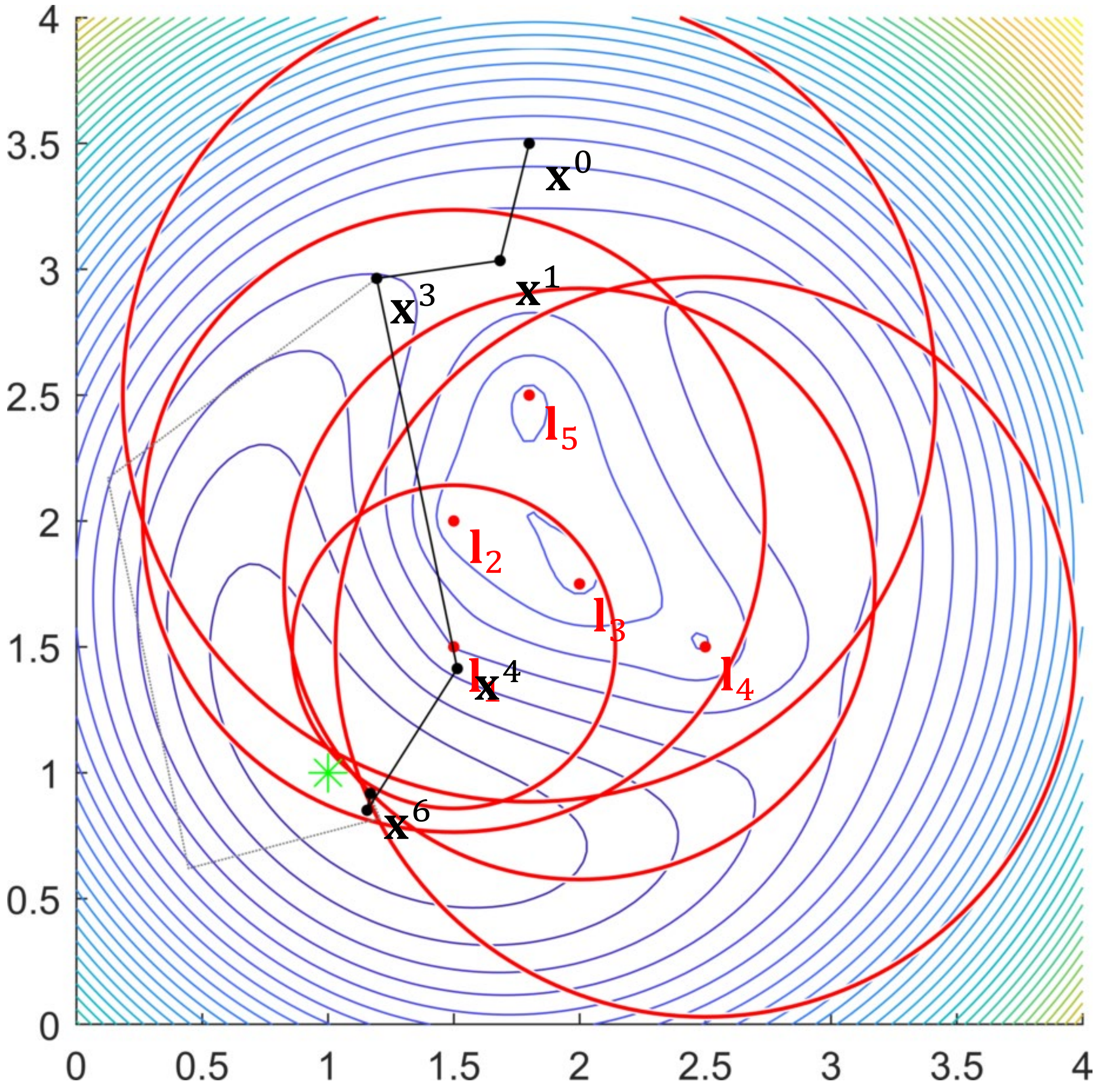




Example:
Range-based localization

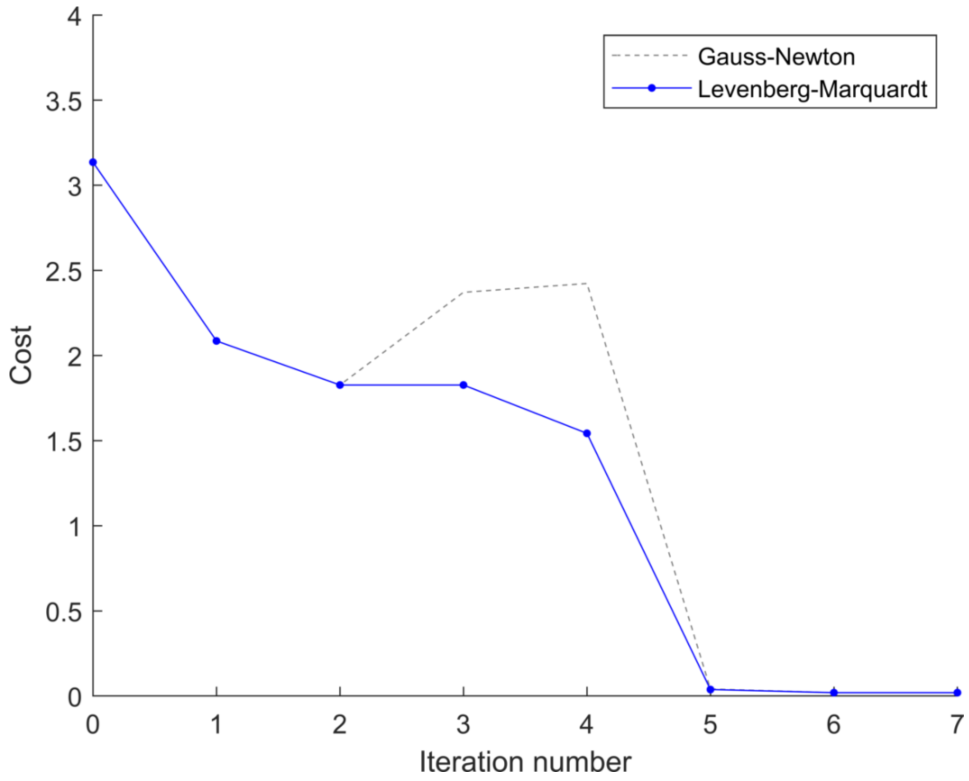
Levenberg–Marquardt optimization

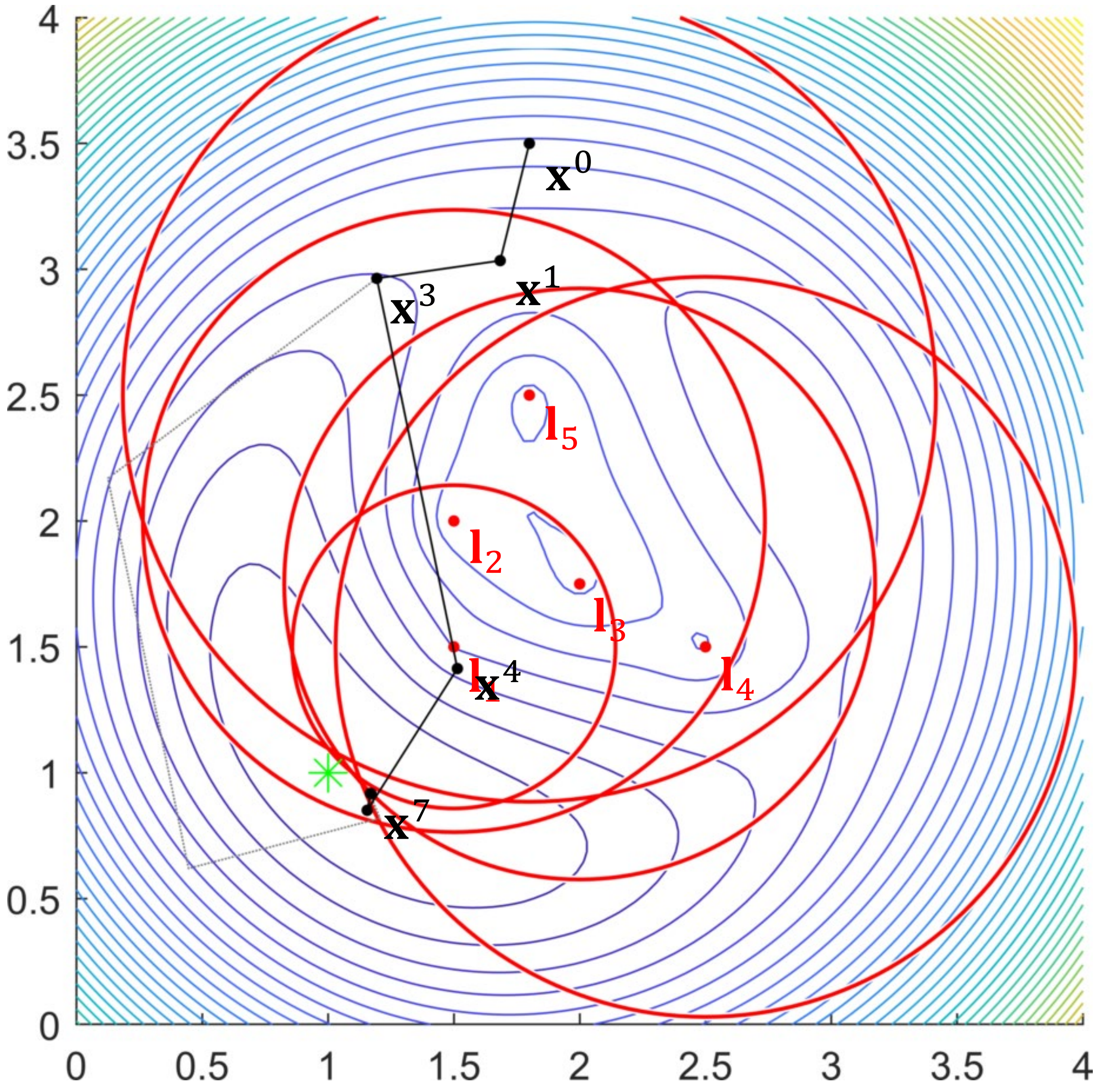




Example: Range-based localization

Levenberg–Marquardt optimization

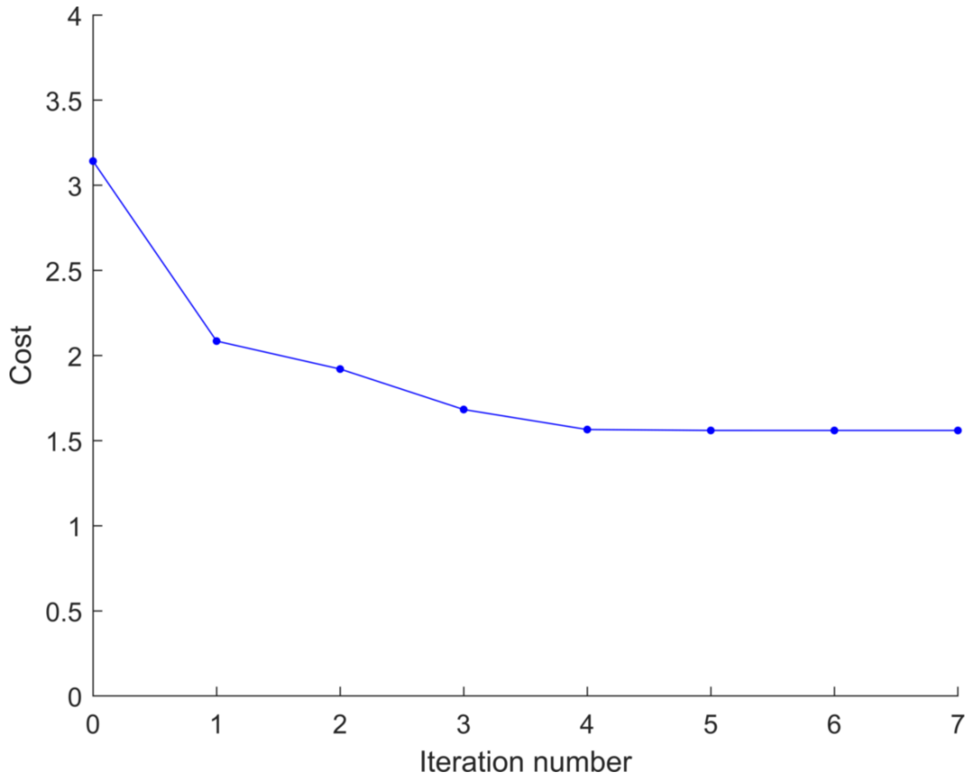


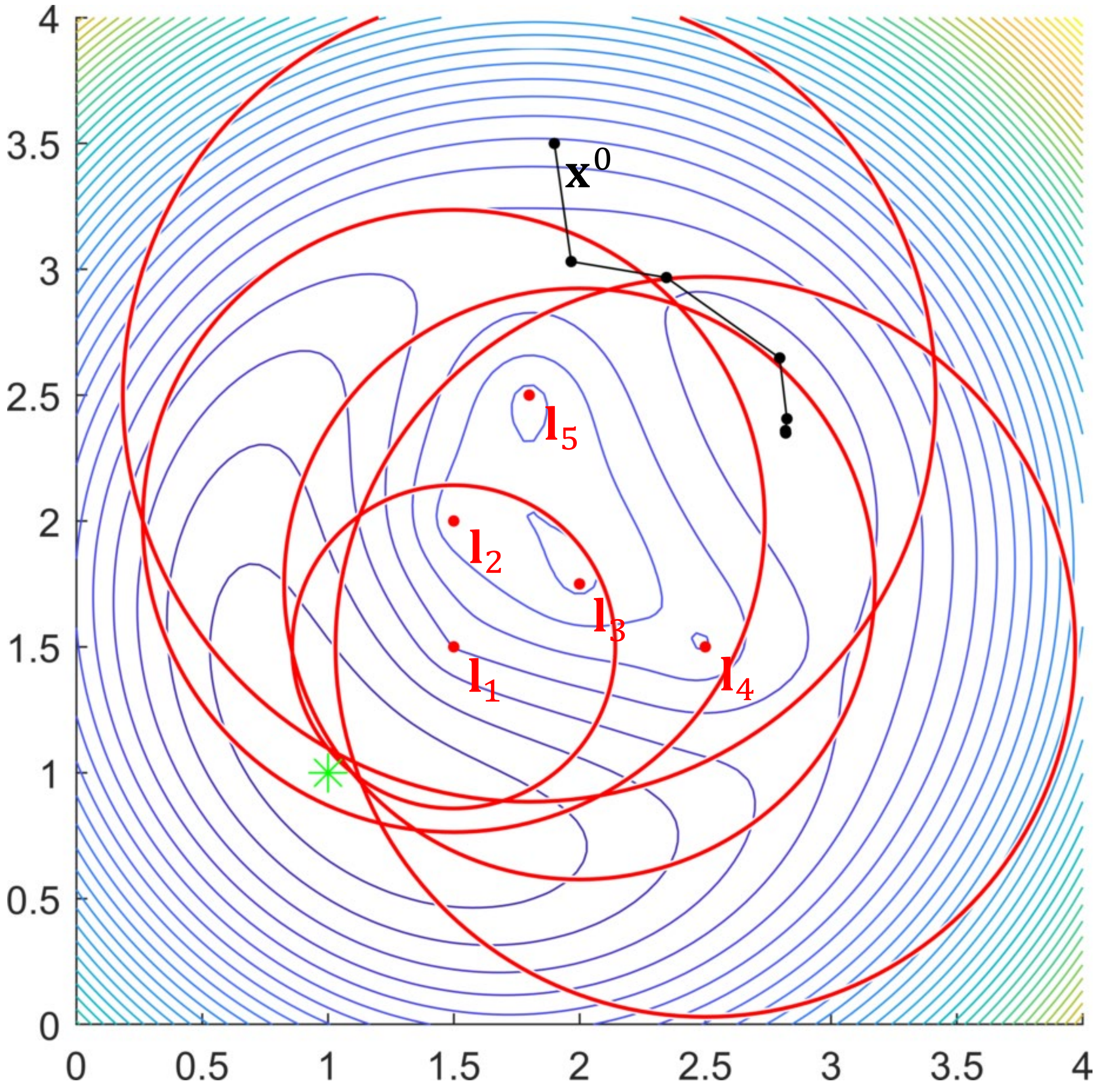


Example:
Range-based localization

Levenberg–Marquardt optimization

- Slightly different initial estimate

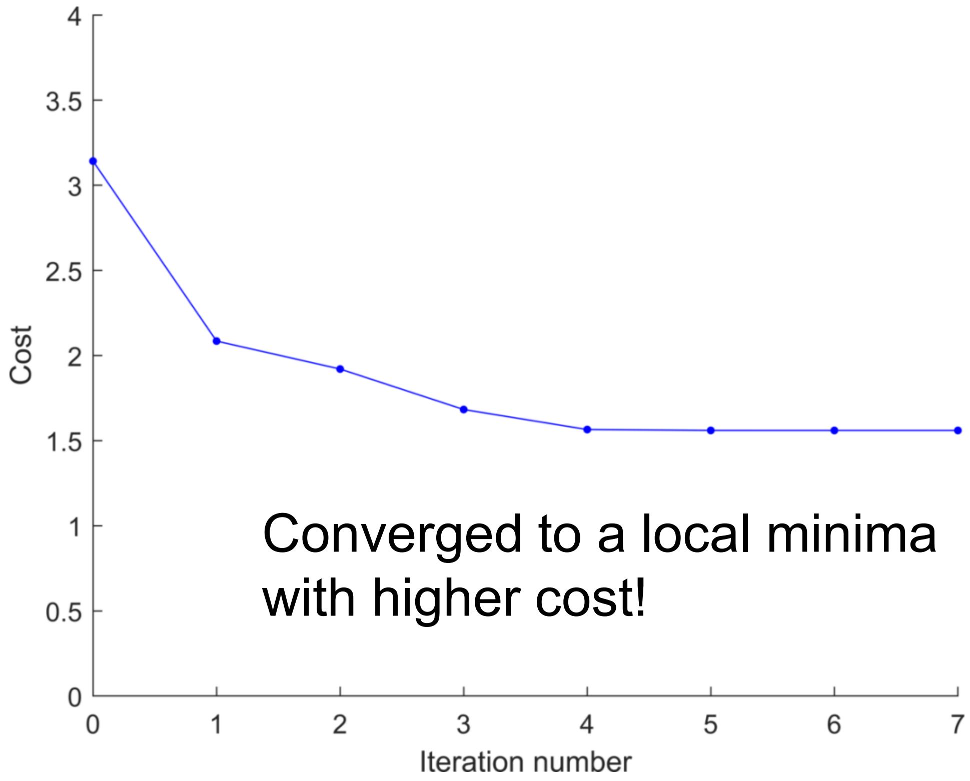




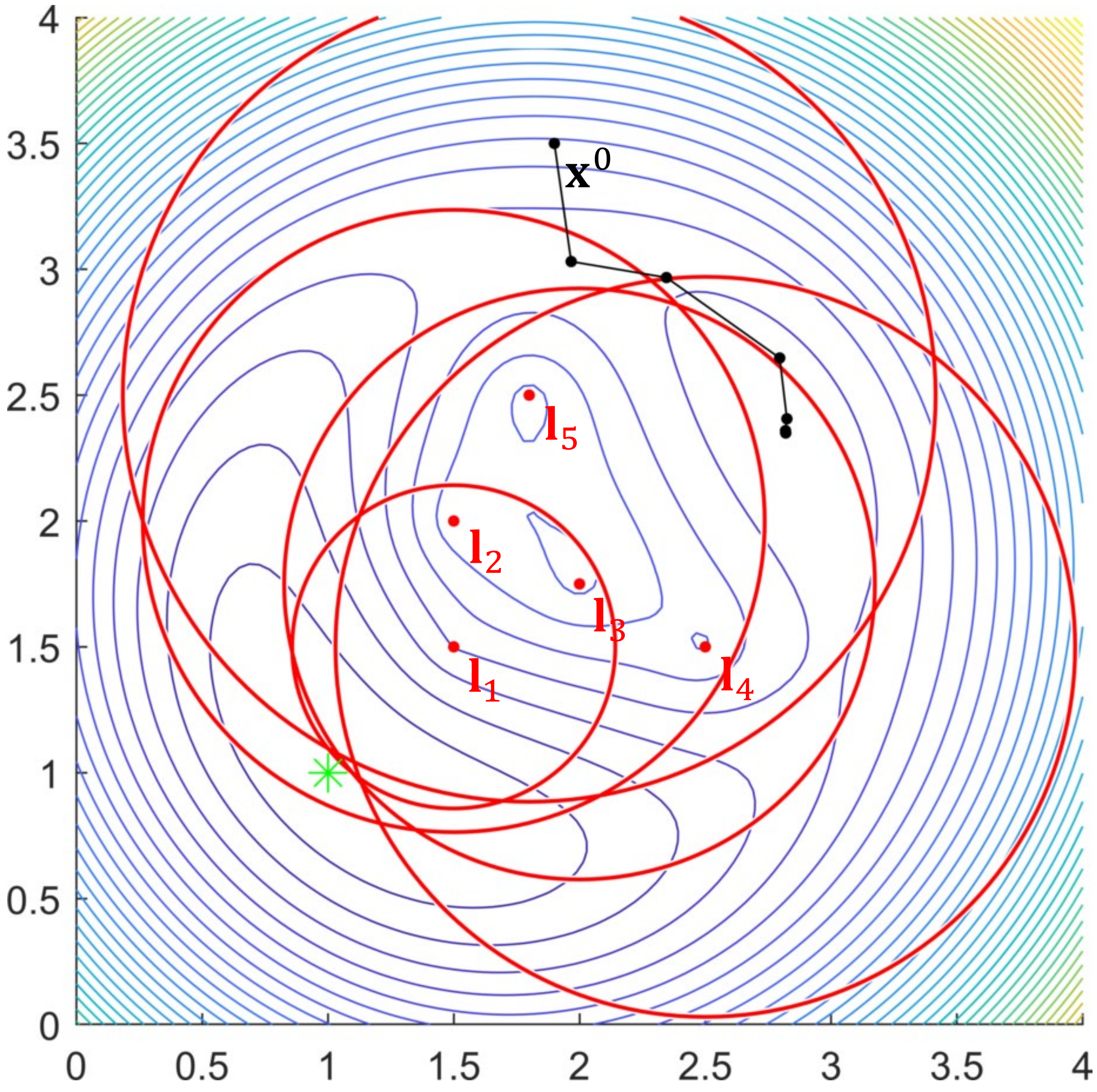
Example:
Range-based localization

Levenberg–Marquardt optimization

- Slightly different initial estimate



Converged to a local minima with higher cost!



Next: Take measurement noise into account!

Measurement model:

$$\mathbf{z}_i = h_i(\underline{\mathcal{X}}_i) \cancel{+ \eta_i, \qquad \eta_i \sim N(\mathbf{0}, \mathbf{\Sigma}_i)}$$

Measurement prediction function:

$$\hat{\mathbf{z}}_i = h_i(\underline{\mathcal{X}}_i)$$

Measurement error function:

$$e_i(\underline{\mathcal{X}}_i) = h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i$$

Objective function:

$$f(\underline{\mathcal{X}}) = \sum_{i=1}^{n} \left\| h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i \right\|^2$$

This results in the nonlinear least squares problem:

$$\underline{\mathcal{X}}^* = \underset{\underline{\mathcal{X}}}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| h_i(\underline{\mathcal{X}}_i) - \mathbf{z}_i \right\|^2$$