Slides have been created by Prof. Michele Focchi

webpage: https://mfocchi.github.io/Teaching/

E4-Numerical Inverse Kinematics

IK : compute q such that $r = f(q)$

$$r \Rightarrow \boxed{IK} \Rightarrow q$$

- at the velocity level is easy:

$$\boxed{\dot{q} = J^{-1}\dot{r}}$$

if J^{-1} exists otherwise we will see some tricks...

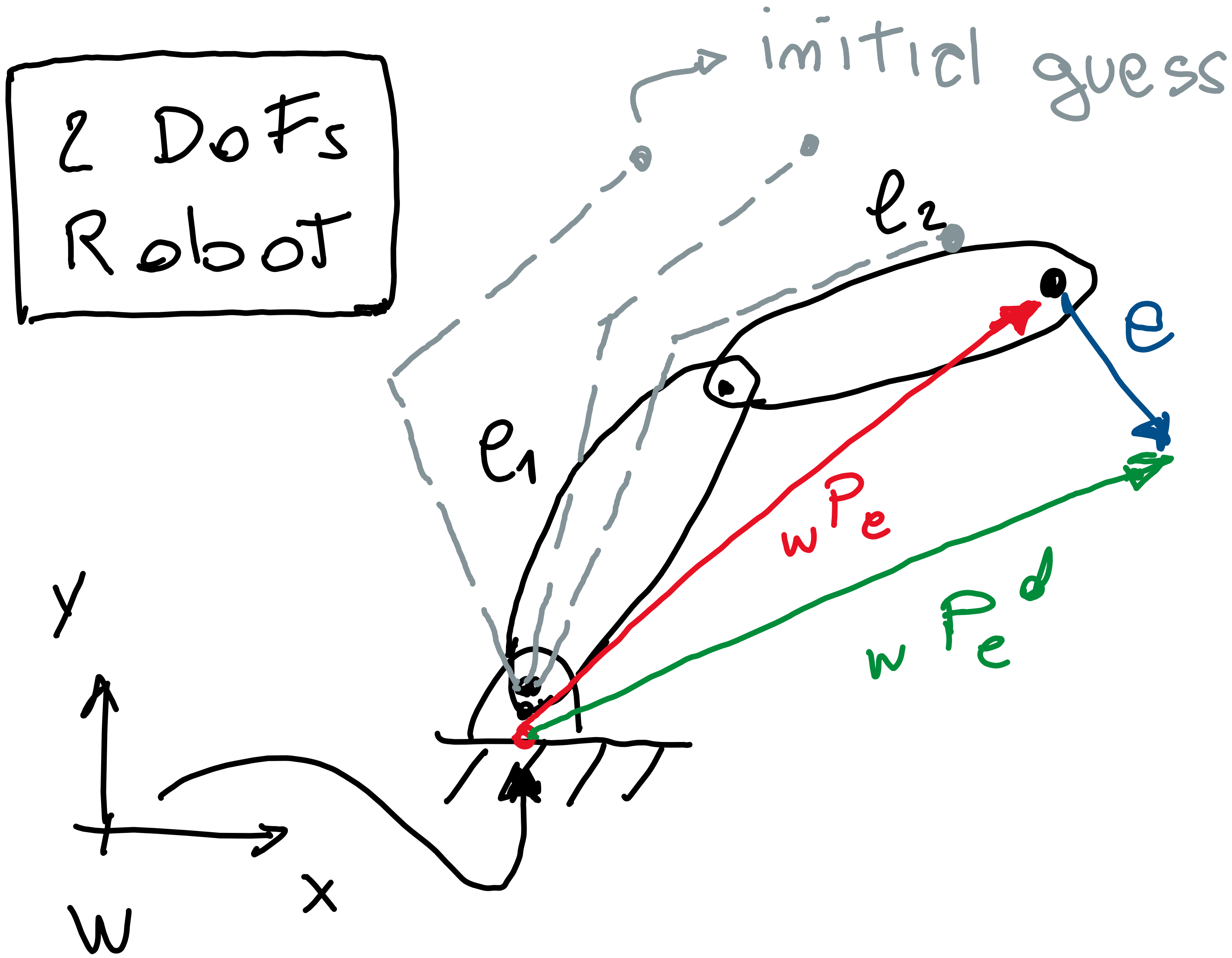
- at the position level ?

numerical approaches are mandatory:

① case of redundant robots

② when a closed form solution does not exists or it is too hard to be found

NUMERICAL INVERSE KINEMATICS



$$ {}_w P_e = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix} $$

NUMERICAL IK STEPS (IMPUT P_e^d, OUTUPUT q)

1. sequence of q^i
2. compute error $e_i = P_e^d - P_{e_i}(q^i)$
3. make $\| e_i \| \to 0$ as $i \to \infty$

NUMERICAL INVERSE KINEMATICS

- require $f(q)$ (DIR. KIN), $J(q)$

- it consist in the solution of an optimization problem

- The variables of the problem are the joint angles we are looking for

- The problem is unconstrained (only cost)

$$q^* = \arg\min_q \frac{1}{2} \| p(q) - p^d \|^2 = \frac{1}{2} \| e(q) \|^2 = c(q)$$

input : p^d = desired end-effector

output: q^* = joint positions

NON CONVEX OPTIMIZATION PROBLEM

$$q^* = \arg\min_q \frac{1}{2}\|p(q) - p^d\|^2 = \frac{1}{2}\|e(q)\|^2 = c(q)$$

- $c(q)$ is non linear and non convex
  - difficult to solve
  - we can find a local optimum (good enough)
  - how to solve it? 1° ORDER NECESSARY CONDITION (FONC)

$$\underbrace{\frac{\partial c}{\partial q}}_{\text{scalar / vector}}(q) = 0 \Rightarrow$$ non linear system of equations

$$c(q) = \frac{1}{2}\|e(q)\|^2 = \frac{1}{2} e(q)^T e(q)$$

$$\frac{\partial c}{\partial q} = \underbrace{e(q)^T}_{\text{row vector}} \underbrace{\frac{\partial e(q)}{\partial q}}_{\text{matrix}} = e(q)^T \Big(\underbrace{\frac{\partial p(q)}{\partial q}}_{J(q)} - \underbrace{\frac{\partial p^d}{\partial q}}_{=0} \Big) =$$

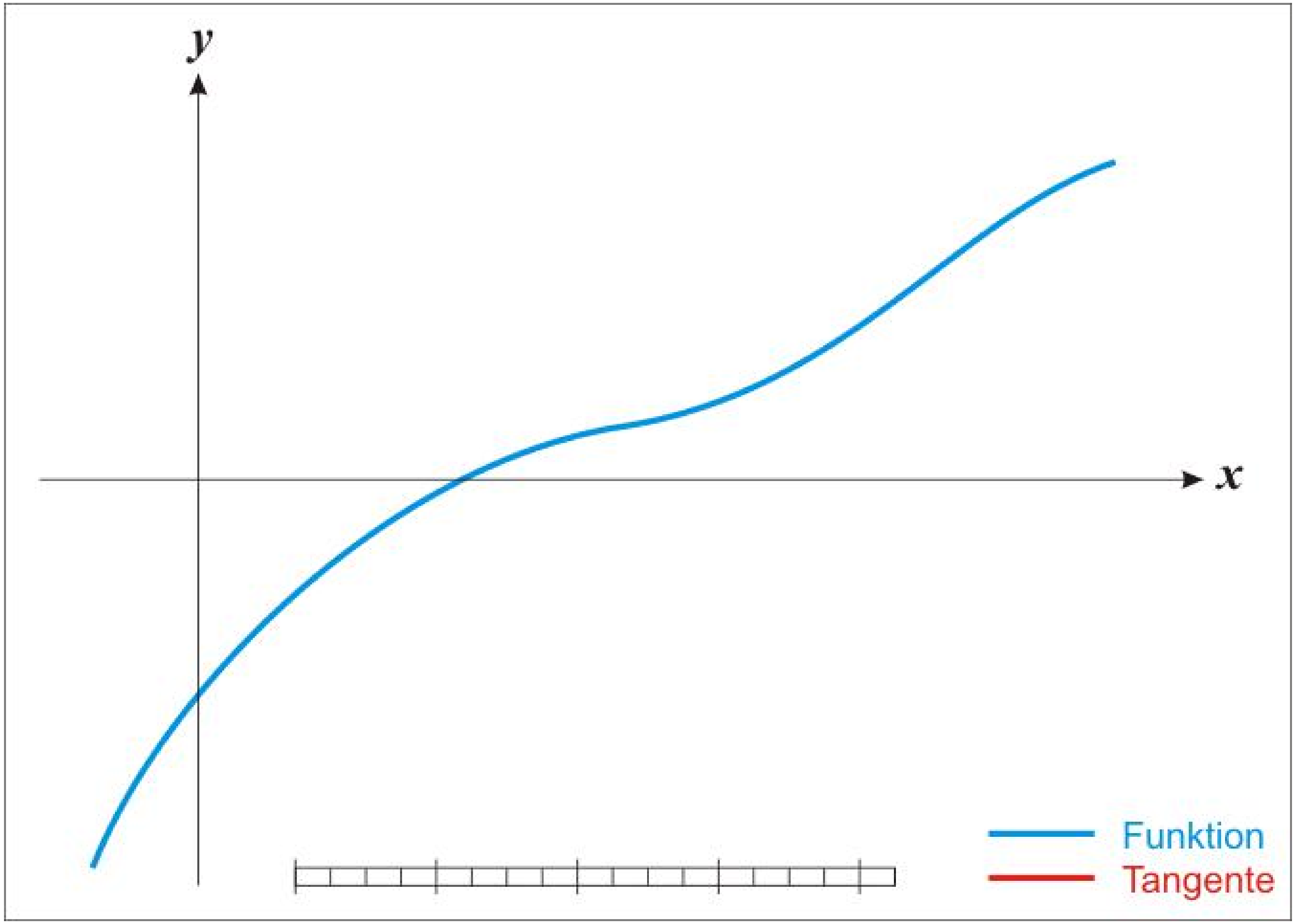
$$= e(q)^T J(q) = \emptyset \iff \boxed{J^T(q)\, e(q) = 0}$$

$r(q)$ set of non linear equations of q

- To find q^* such that $r(q) = 0$ I can use Gauss Newton method
- q^* will be a minimum / maximum for $c(q)$

Method of the tangent

f(x) =0



Gauss-Newton is the generalization to the multidimensional case

GAUSS- NEWTON APPROACH

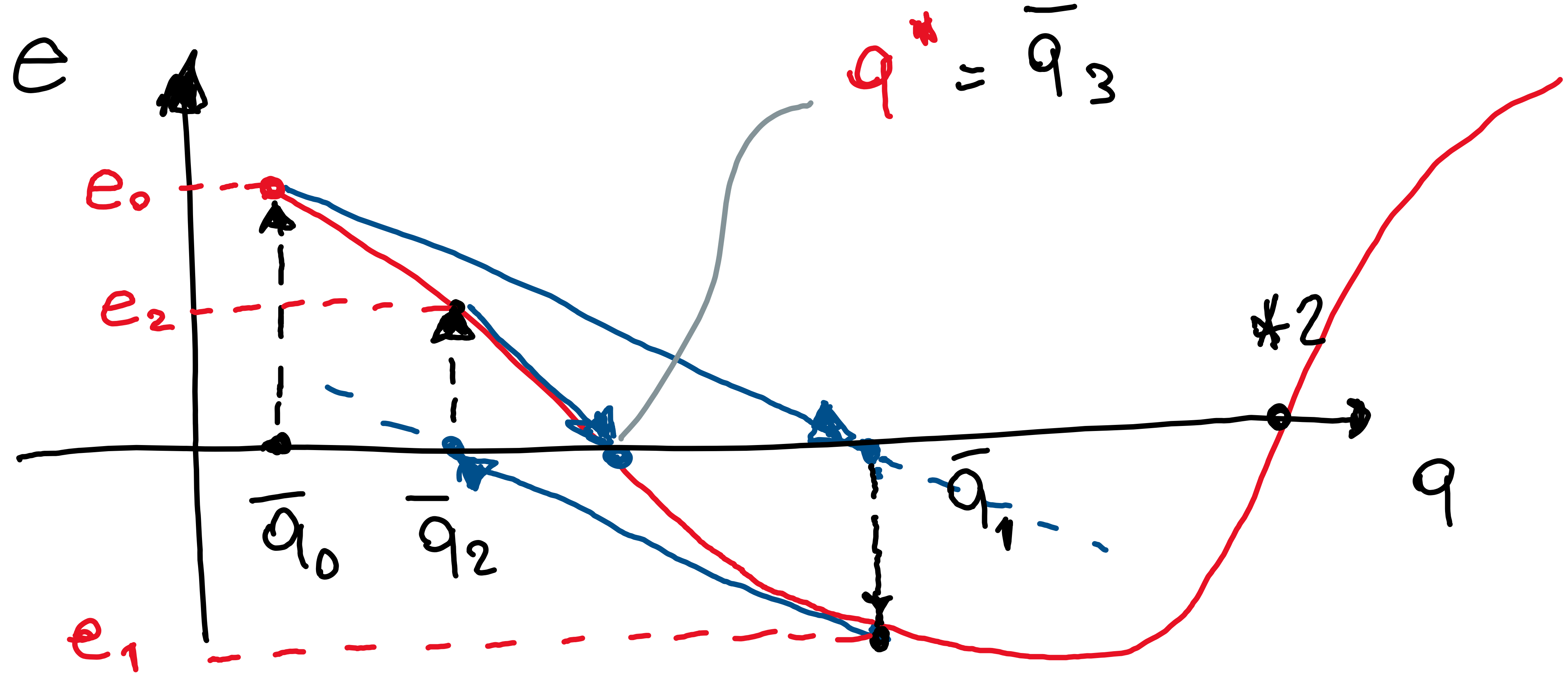
① start with a guess $\bar{q}$

② compute a linear approximation of the problem

③ solve it to find a new guess

③ Iterate till the solution of the linear problem will converge to the solution of the original (non linear) problem

Example with 3 iterations



- Note: if we start from another guess we could converge to another solution (es *2)

② LINEAR APPROXIMATION OF $r(q)$

$$r(q) = r(\bar{q} + \Delta q) = \underbrace{r(\bar{q}) + \nabla_q r(\bar{q})^T \Delta q}_{1^\circ \text{ order Taylor expansion}} = \emptyset$$

$\bar{q}$: current guess

Δq: New variable

$$\nabla_q r = \nabla_q (J^T e) = (\nabla_q J^T) e + J^T \nabla_q e \simeq J^T J$$

$\nabla_q J^T \approx \emptyset$ — Gauss Newton approx

$\nabla_q e \approx J$

Therefore:

$$r(q) = r(\bar{q}) + J^T J \, \Delta q = 0$$

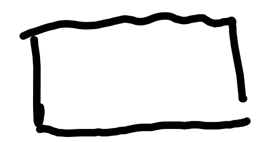
$\quad \hookrightarrow J^T e(\bar{q})$

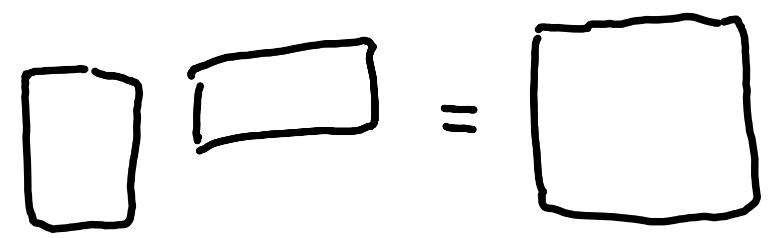
$$J^T J \, \Delta q = - J^T e(\bar{q})$$

$$\boxed{\Delta q = -\overbrace{(J^T J)^{-1} J^T}^{J^\#} \underbrace{e(\bar{q})}_{\bar{e}}}$$

Newton step

INVERTIBILITY OF J^TJ

$J \in \mathbb{R}^{3 \times n}$ 

$J^TJ \in \mathbb{R}^{n \times n}$ 

Typically $n \geq 3$ $\quad$ $\text{rank}(J) \leq 3$

J^TJ is invertible if $\text{rank}(J^TJ) = n$

PROPERTY:

$$\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$$

$$\text{rank}(J^TJ) = \text{rank}(J) \leq 3$$

case 1: $n = 3 \Rightarrow J^TJ$ is invertible

case 2 $n > 3 \Rightarrow J^TJ$ is not invertible

EQUIVALENCE TO A QP

② $$e(q) \cong e(\bar{q}) + \frac{\partial e}{\partial q}\Delta q = e(\bar{q}) + J\Delta q$$ linear approx (Taylor)

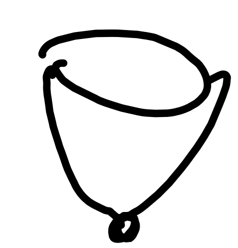
$$\hookrightarrow \frac{\partial p_e}{\partial q} \qquad \cancel{\frac{\partial p_e}{\partial q}}\ 0 = J$$

$$\Delta q^* = \arg\min_{\Delta q} \boxed{\frac{1}{2}\| \overbrace{e(\bar{q})}^{\bar{e}} + \overbrace{J(\bar{q})}^{\bar{J}}\Delta q \|^2} \rightarrow c(q)$$

$$= \arg\min_{\Delta q} \frac{1}{2}(\bar{e} + \bar{J}\Delta q)^T(\bar{e} + \bar{J}\Delta q) =$$

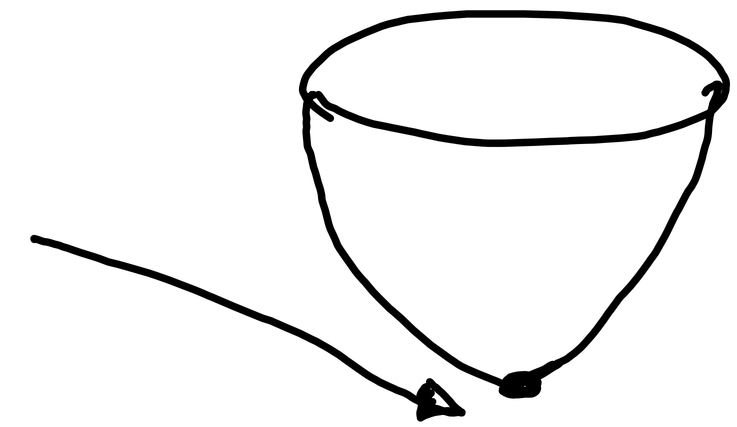
$$\underbrace{\frac{1}{2}\Delta q^T \overbrace{\boxed{\bar{J}^T\bar{J}}}^{\text{Hessian}}\Delta q}_{\text{quadratic Term}} + \underbrace{\overbrace{\boxed{\bar{e}^T\bar{J}}}^{\text{gradient } g^T}\Delta q}_{\text{linear Term}} + \underbrace{\frac{1}{2}\bar{e}^T\bar{e}}_{\text{constant}}$$

The problem is quadratic program, is convex and has only one minimum



The value of Δq That minimizes the linear approx of $\|e(q)\|^2$ is

$$\frac{\partial c(q)}{\partial q} = \bar{J}^T \bar{J} \, \Delta q + \bar{J}^T \bar{e} = 0$$



Note: c is a scalar and The gradient should be a column vector not a row vector so we take The Transpose

$$\boxed{\Delta q = -H^{-1} g = -\left(\bar{J}^T \bar{J}\right)^{-1} \bar{J}^T \bar{e}}$$

Newton step

REGULARIZATION

$$\Delta q^* = -(\bar{J}^T \bar{J} + \lambda I)^{-1} J^T \bar{e} \qquad \lambda > 0 \quad \approx 10^{-8}$$

$$\underbrace{(\bar{J}^T \bar{J} + \lambda I)}_{\text{always invertible and positive definite}}$$

$$= -J_\lambda^\# \bar{e}$$

↳ damped pseudoinverse

DESCENT DIRECTION

Δq is a desent direction for $c(q)$

$\Updownarrow$

$\nabla c^T \Delta q < 0$ → cost decreases if we move along Δq

$$\nabla c = J^T e$$

$$\Delta q = -\underbrace{(J^T J)}_{H}^{-1} J^T e$$

$$\nabla c^T \Delta q = -e^T J H^{-1} J^T e$$ $\rightarrow$ quadratic form

$x^T A x > 0$ if A pos def.

$$-e^T J H^{-1} J^T e < 0$$

if $J H^{-1} J^T$ is positive def. $\Rightarrow$ Δq is a descent direction for $c(q)$ $\forall q$

H^{-1} " "

H " "

SUMMARY REGULARIZATION

- makes $J^T J + \lambda I$ is invertible
- ensures that $J^T J + \lambda I$ remains pos. def.

$\Rightarrow$ Δq is always a descent direction for the cost

③ update guess and evaluate if cost has decreased

$$\bar{q}_{i+1} = \bar{q}_i + \Delta q^*$$

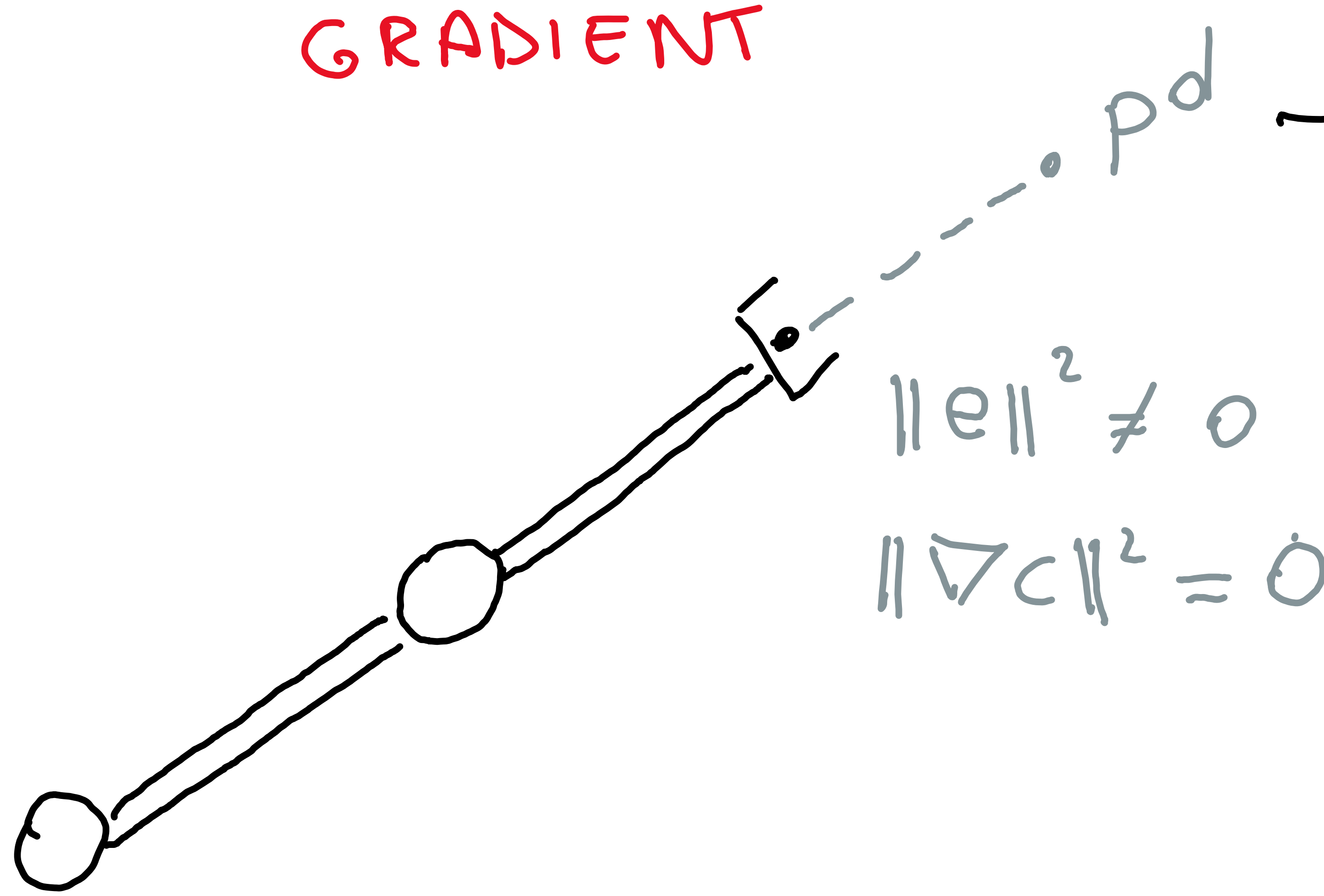
④ if $\underbrace{\|e(q_{i+1})\|^2 \leq \varepsilon}$ stop otherwise go to ②

The algorithm never converges if you ask p^d out of the workspace

$$\boxed{\text{if } \|\nabla c\|^2 = \|J^T e(q_{i+1})\|^2 \leq \varepsilon \quad \text{stop}}$$

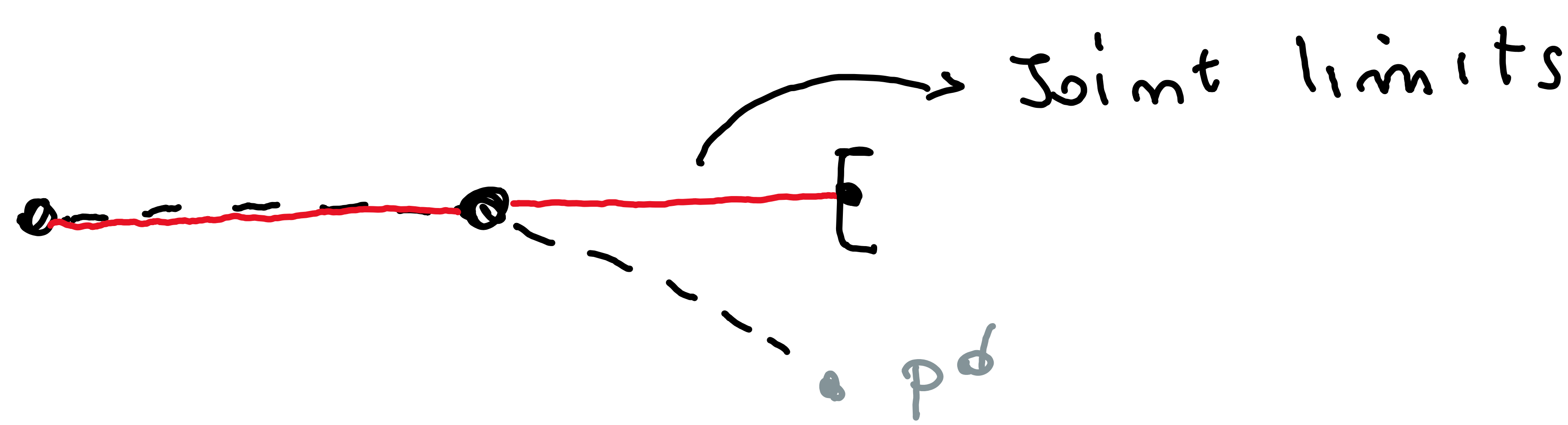
BETTER CONVERGENCE CRITERION

EXAMPLE: CONVERGENCE CRITERIA BASED ON GRADIENT



The algorithm will find the "closest" solution that minimizes the error e

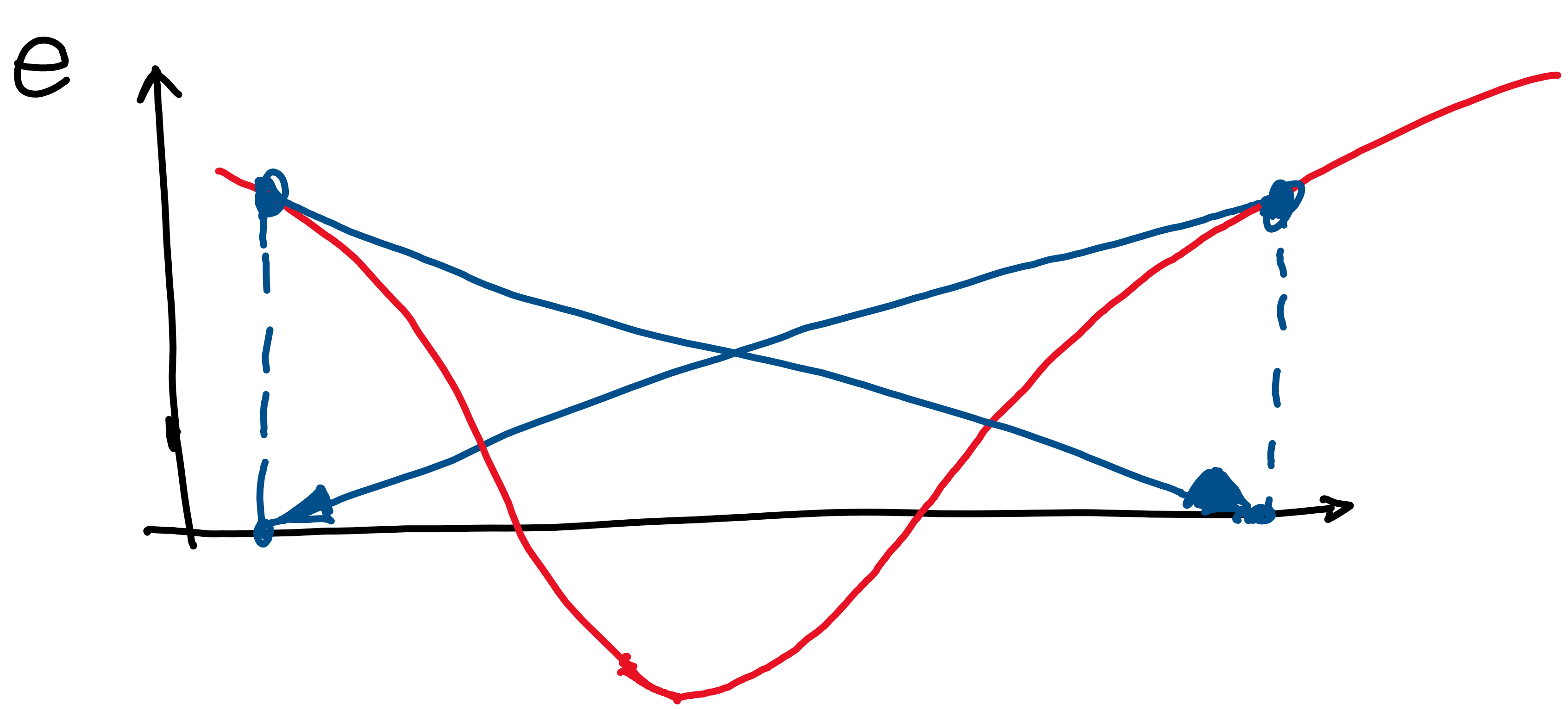
⑤ enforce joint limits

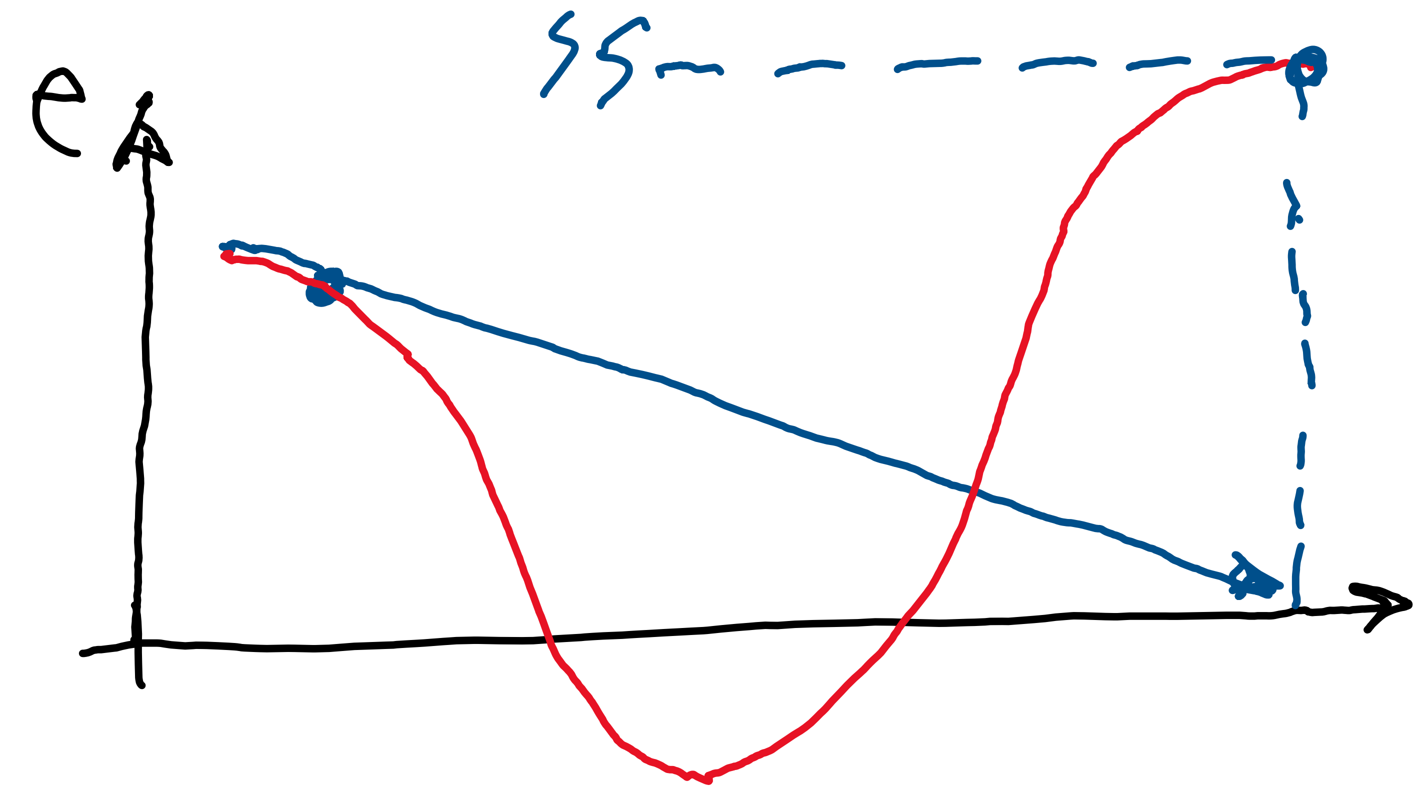


$$\boxed{\begin{array}{l} \text{if } q^* \geq q_{max} \\ \quad q^* = q_{max} \\ \text{if } q^* \leq q_{min} \\ \quad q^* = q_{min} \end{array}}$$

PROBLEM: WHAT IF COST IS NOT DECREASING?

- you can get stuck in cycles or zero gradient





solution. move along the Newton direction Δq but take smaller steps until you see $\|e\|^2$ decreases

step size $\alpha \in [0,1]$

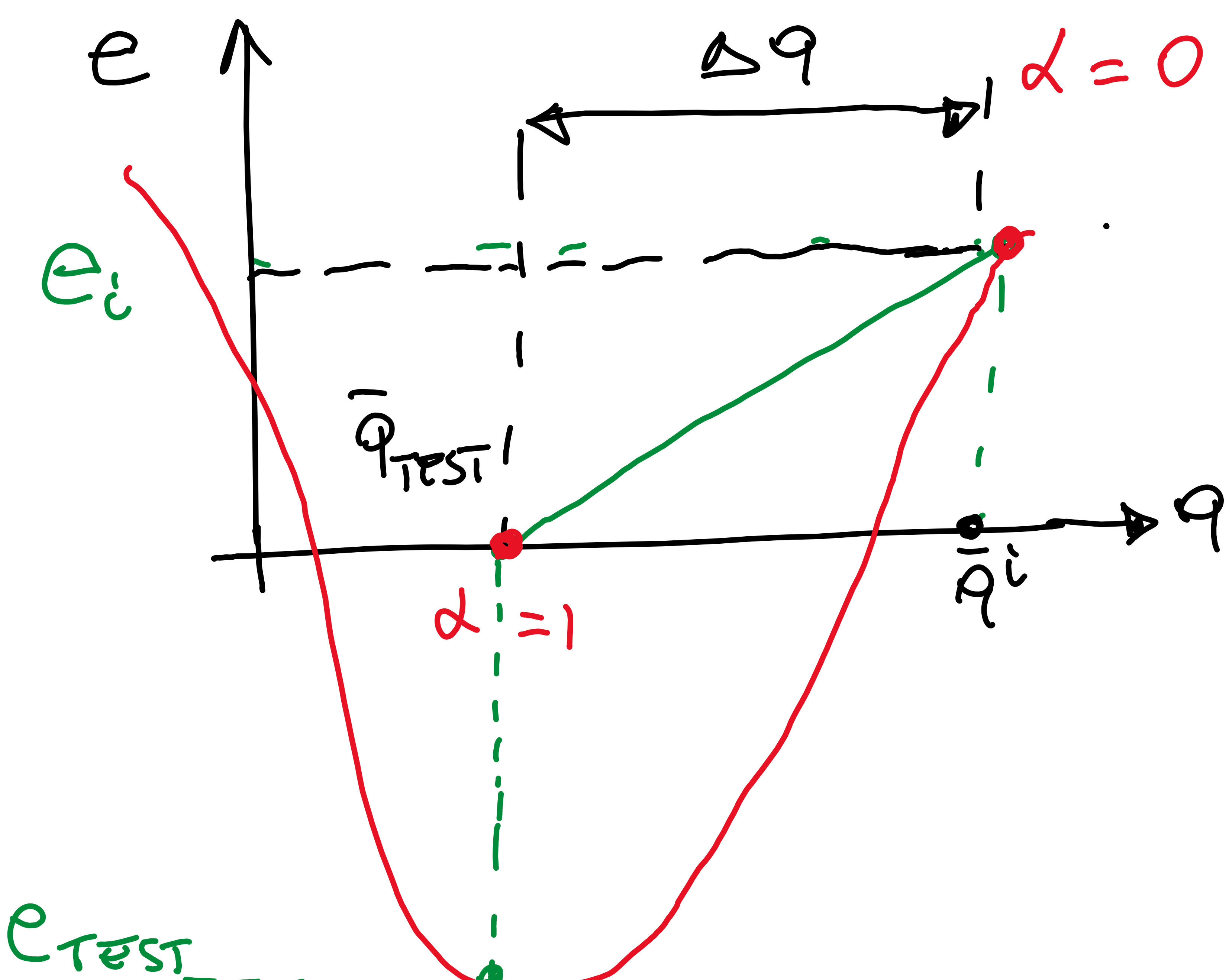
$$q^{i+1} = q^i + \alpha \left[\underbrace{-\left(J^T J + \lambda I\right)^{-1} J^T \left[\overbrace{P(q_i) - P^d}^{e_i} \right]}_{\text{Newton step}} \right]$$

STEP SIZE SELECTION (LINE SEARCH)

how to choose α ? $\quad 0 \leq \alpha \leq 1$

(3.1) $\bar{q}_{TEST} = \bar{q}^i + \alpha \Delta q_i$

check the reduction of e with $\alpha = 1$



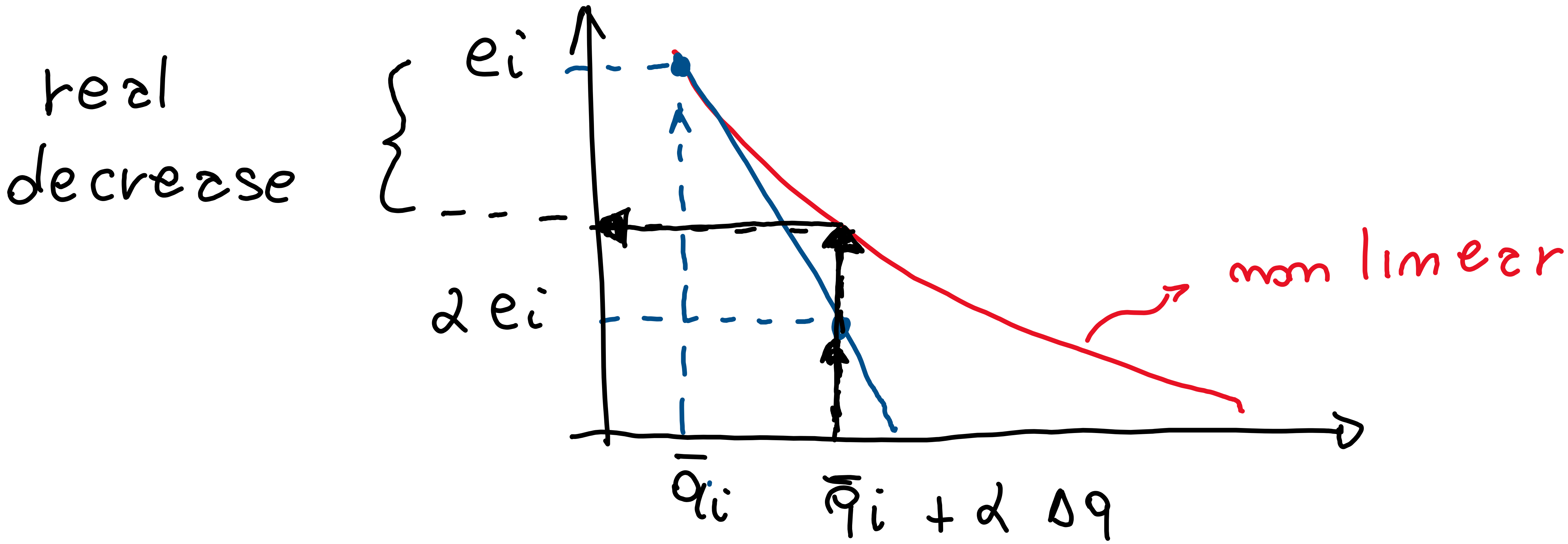
$\bar{q}_{TEST} = \bar{q}^i + 1 \cdot \Delta q_i$

$\| e_i \| < \| e_{TEST} \|$ NOT OK

$\Rightarrow$ Take smaller step

OBSERVATION

- if $e(q)$ was linear $\|e_i\| - \|e_{i+1}\| = \alpha \|e_i\|$



- in practice $e(q)$ is non linear so we are happy with a fraction γ of the error reduction in the linear case

$$\|e_i\| - \|e_{i+1}\| \geq \gamma \, \alpha \|e_i\| \geq 0 \qquad \gamma \in [0,1]$$

e.g. 10^{-2}

- if step size is small convergence can become very <u>slow</u>

STEP SIZE SELECTION (LINE SEARCH PSEUDO CODE)

3.2 $\text{reduction} = \|e(\bar{q}_i)\| - \|e(\bar{q}_{TEST})\|$

3.3 if $![(\text{reduction} > \emptyset)$ or $> \gamma \alpha \|e\|$ (more stringent)

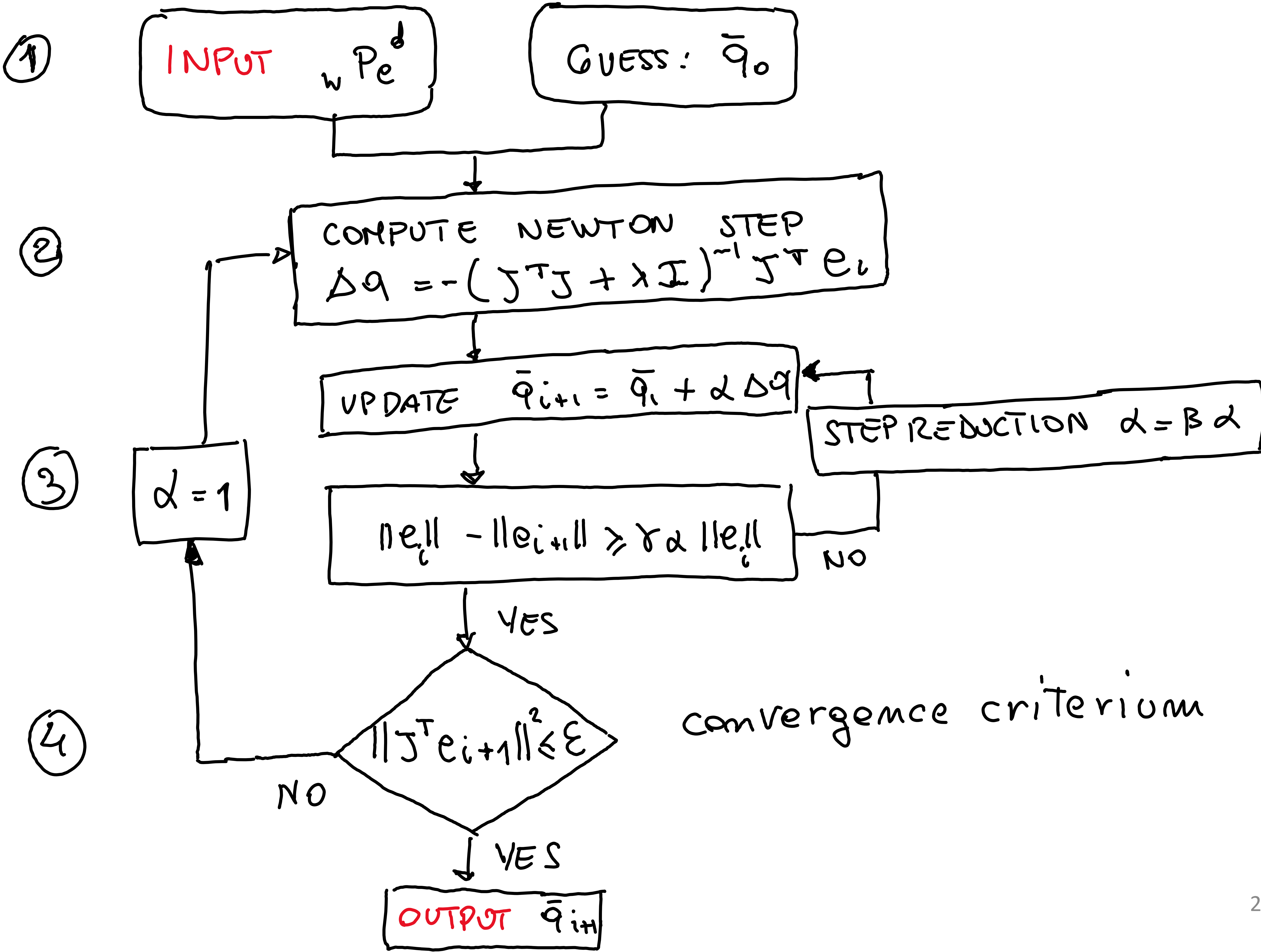
$\alpha = \beta \alpha \qquad \beta \in [0, 1]$ Typically 0,5

$\bar{q}_{TEST} = \bar{q}_i + \alpha \Delta q_i$

$\text{reduction} = \|e(\bar{q}_i)\| - \|e(\bar{q}_{TEST})\|$

3.4 $\bar{q}_{i+1} = \bar{q}_{TEST}$

FLOW CHART IK ALGORITHM



EXAMPLE STEP SELECTION WITH $\gamma = 0$ $\beta = 0{,}5$

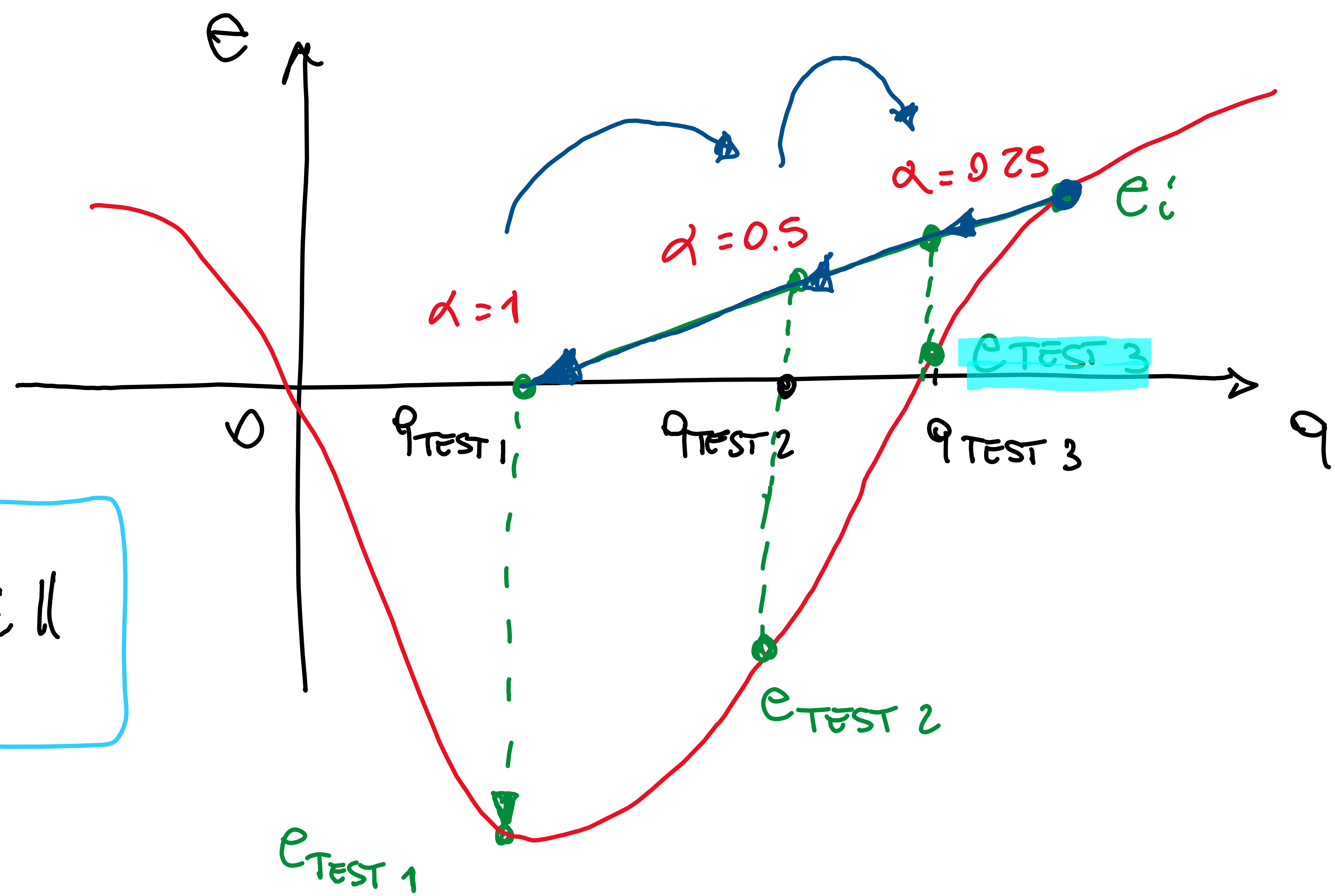
while $\|e_i\| \leq \|e_{TEST}\|$

$\alpha = 0{,}5\,\alpha$

$q_{TEST} = \bar{q}_i + \alpha\, \Delta q_i$

end

$q^{i+1} = q_{TEST}$



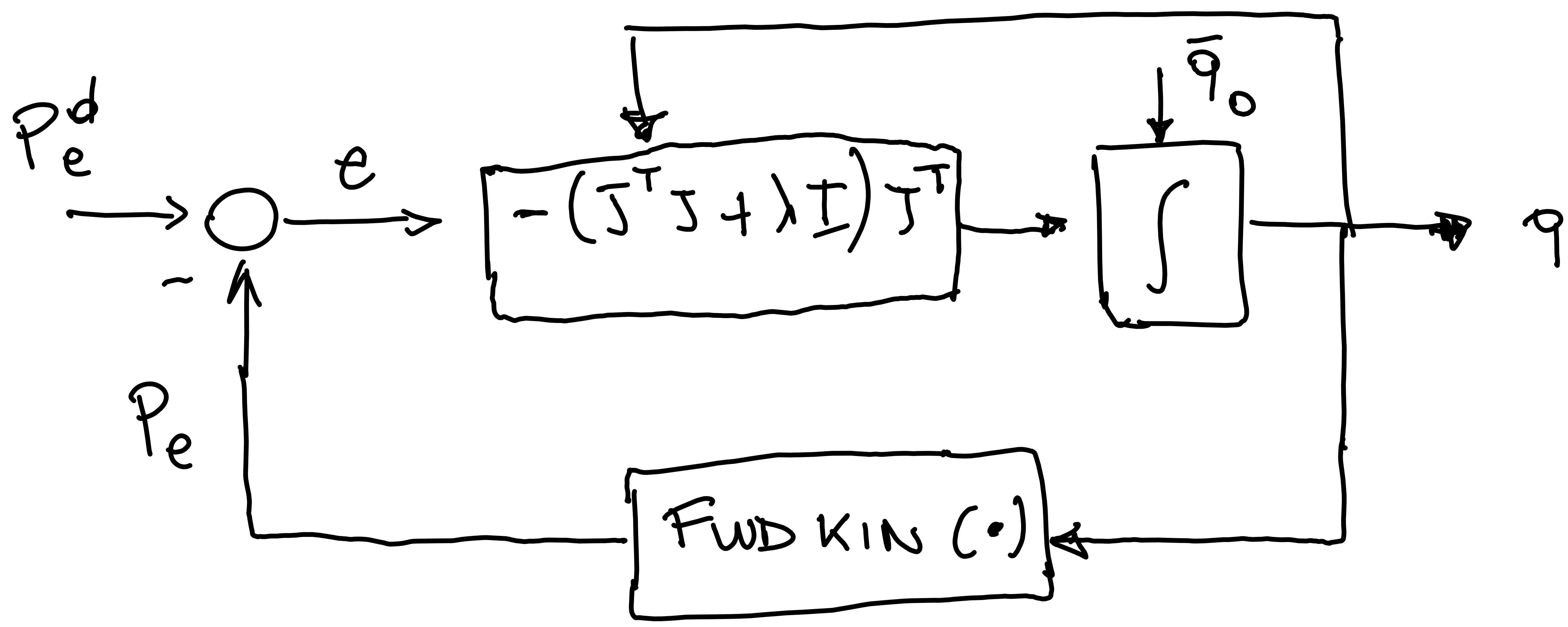
GOAL

$\|e_{TEST}\| \leq \|e_i\|$

$\alpha = 0{,}25$

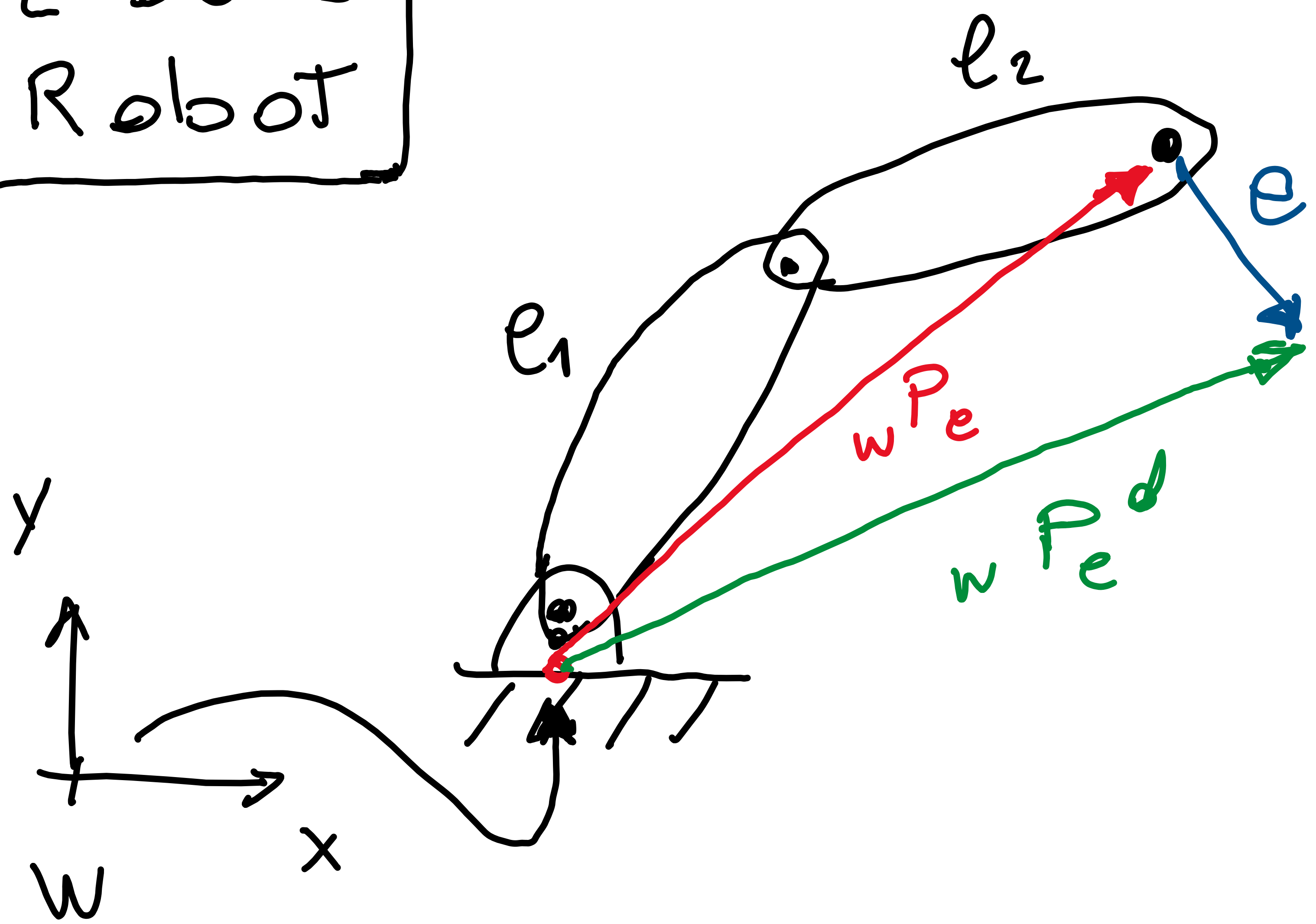
$q^{i+1} = q_{TEST\,3}$

Equivalence to feedback scheme



EXERCISE NUMERICAL IK

2 DoFs Robot



$$
{}_wP_e = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix}
$$

$$
{}_wJ_e = \left[\begin{array}{c|c} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ \hline l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{array}\right]
$$

input: ${}_wP_e = (1, 1)$

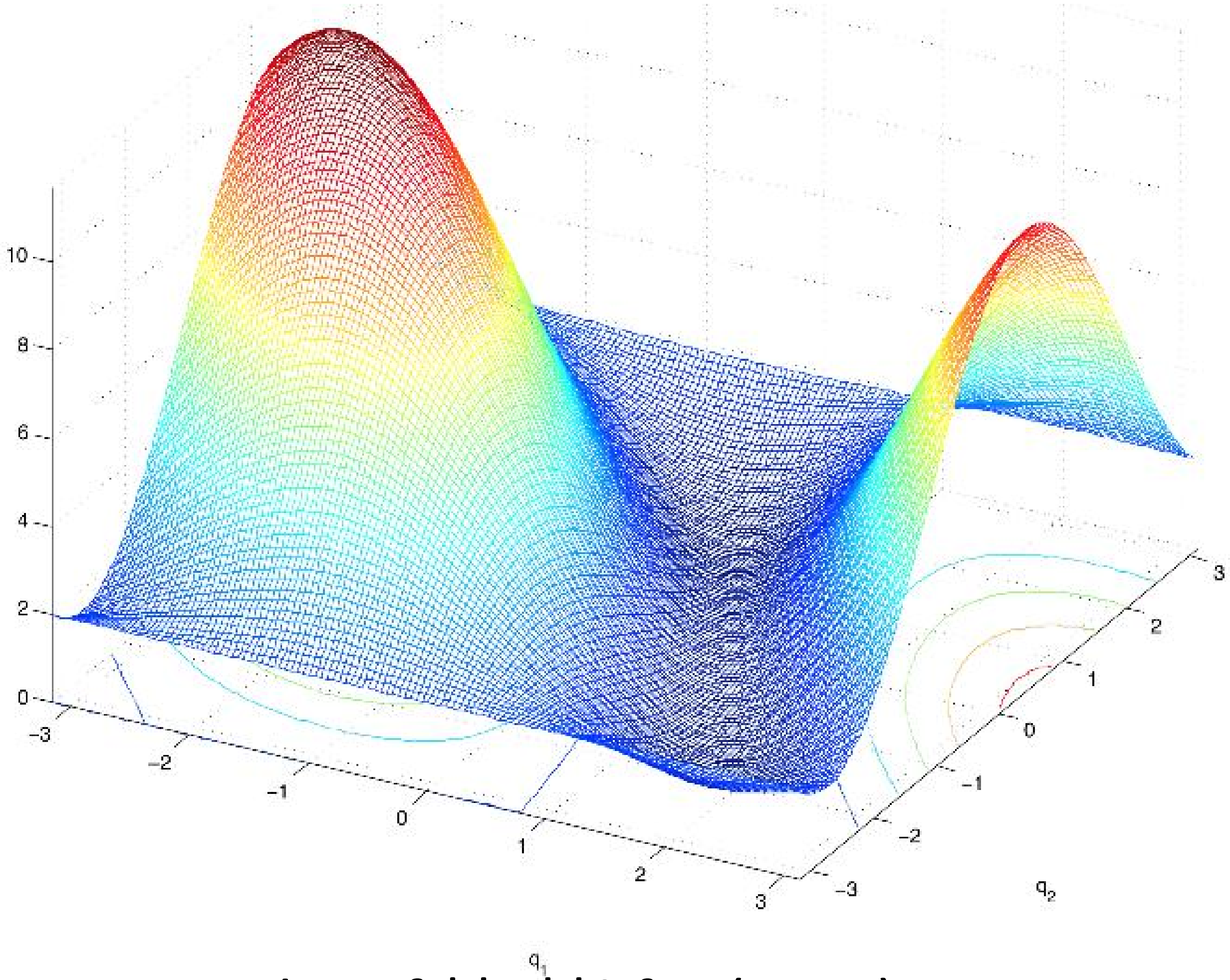
params: $l_1 = 1$ $\quad l_2 = 1$

$\varepsilon = 10^{-5}$

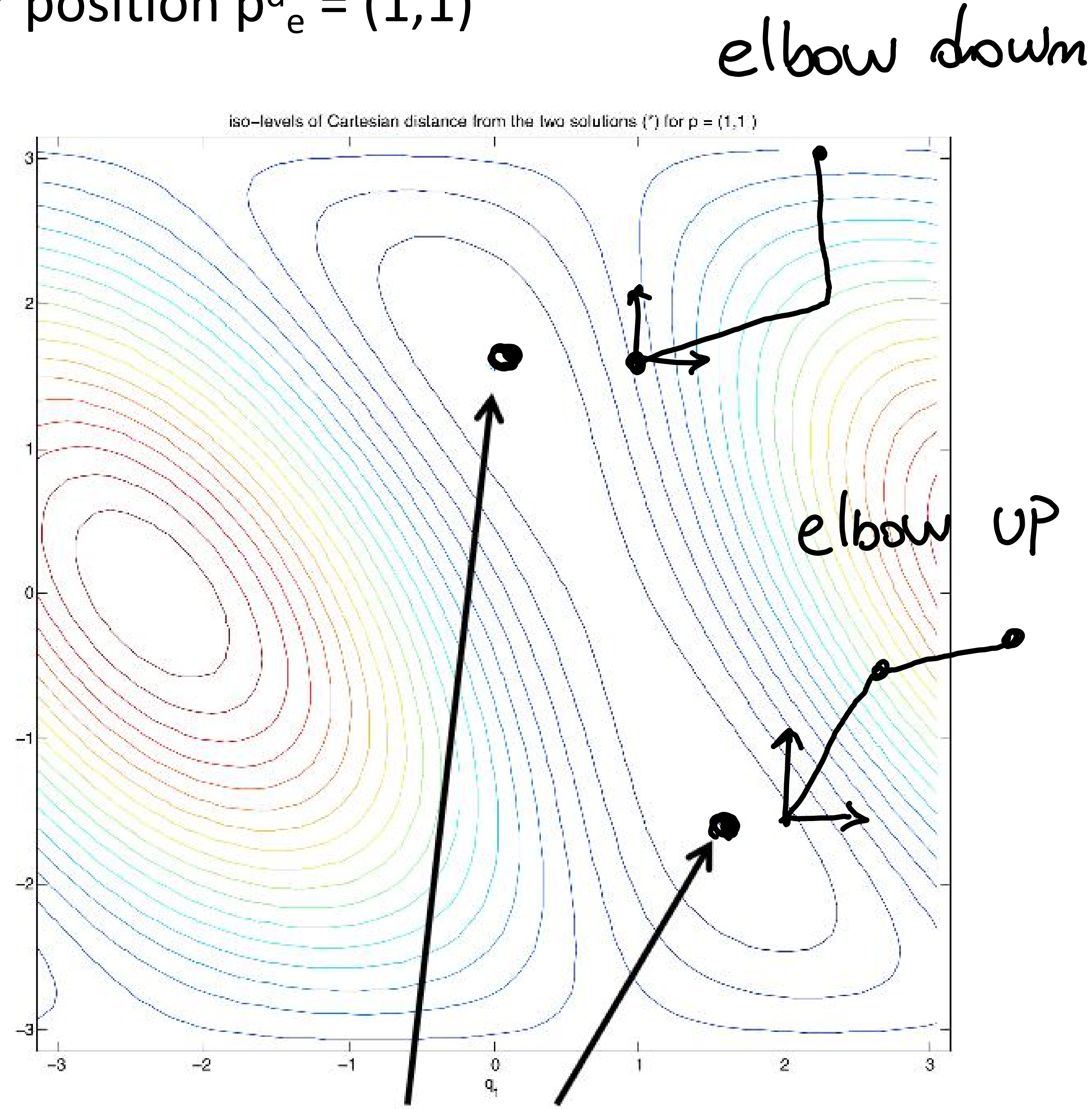
$K_{max} = 10$ max number of iterations

Error function analysis

2R robot with $l_1 = l_2 = 1$ and desired end-effector position $p^d_e = (1,1)$



Plot of $||e||^2$ for (q_1,q_2)



two local minima

IMPLEMENTATION ISSUES AND COMMENTS

1. initial guess $\bar{q}_0$
   - only one solution is generated for each guess
   - multiple initializations $\Rightarrow$ obtain different solutions

2. Joint range limits are considered only at the end

3. if the problem has to be solved online

   $\Rightarrow$ the good choice for the initial guess $\bar{q}_0$ at t_i is the solution of the previous problem at $i-1$

   $\Rightarrow$ small number of iterations expected

② stopping criteria: $\|p^d - f(\bar{q}_i)\| \leq \varepsilon$ or $\|J^T e_i\| \leq \varepsilon$

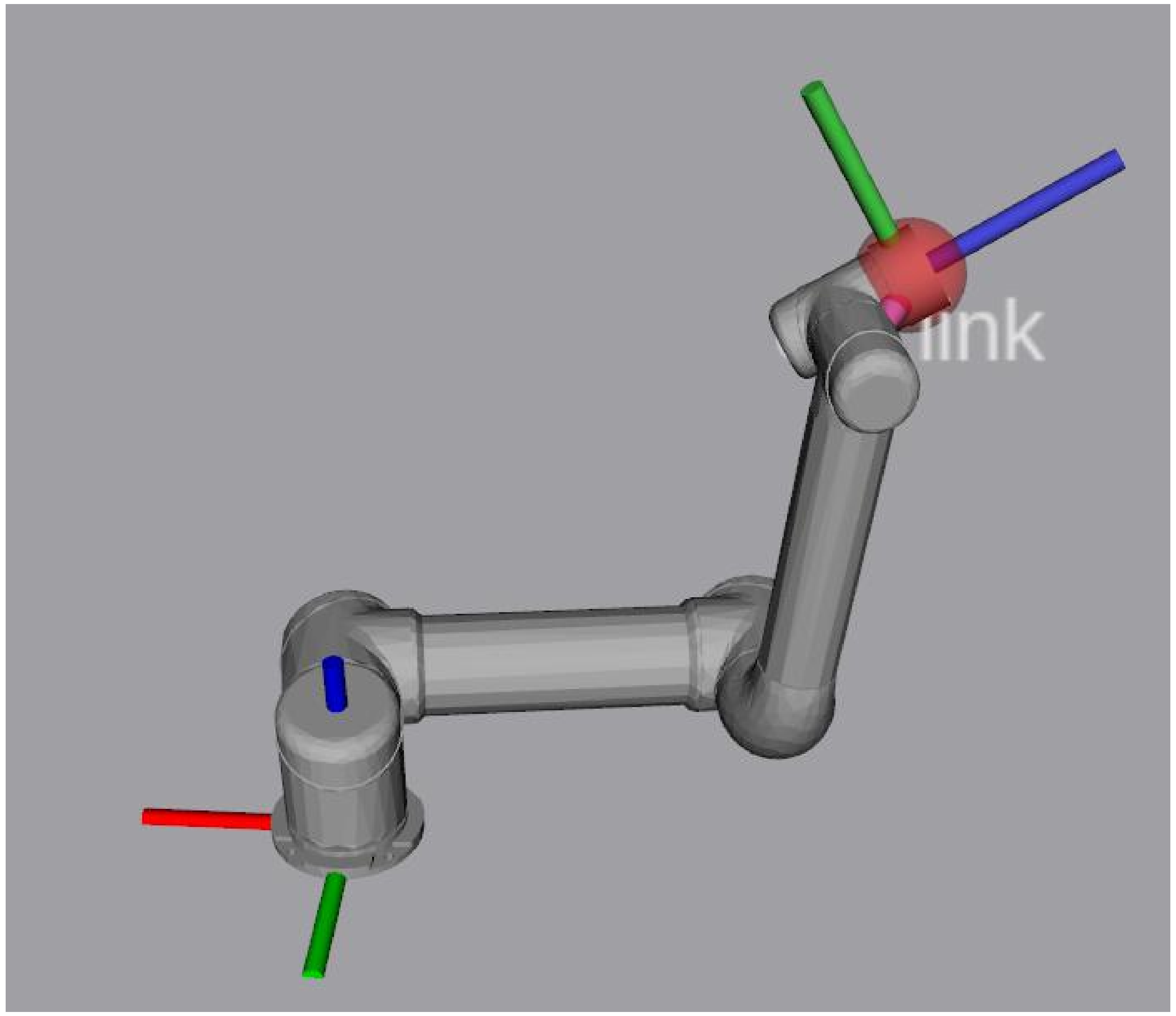
- separate for position and orientation variables

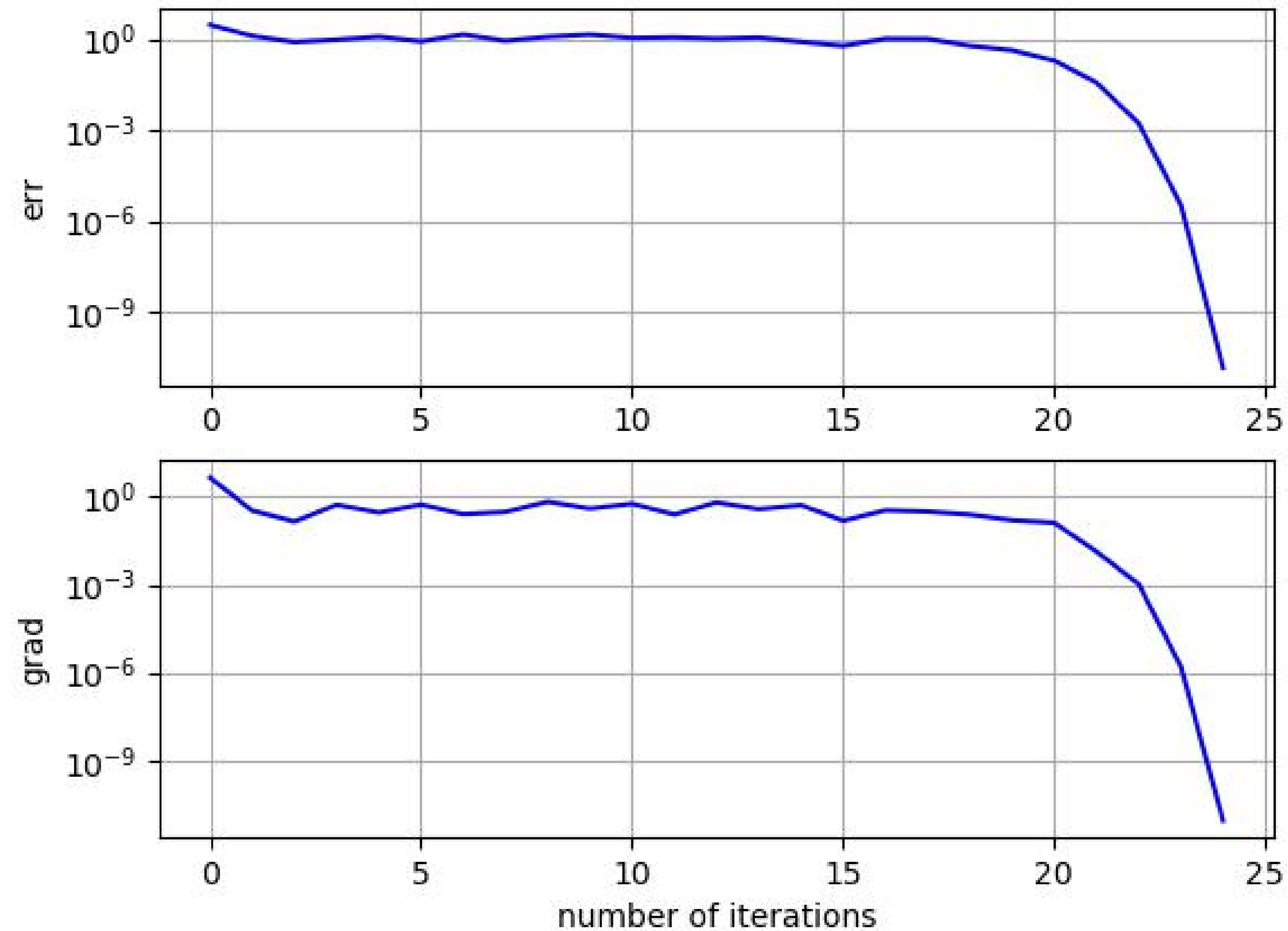
- check closeness to singularity to avoid divergence of NEWTON method

  $\sigma_{min}(J(\bar{q}_i)) \geq \sigma_0$ good numerical conditioning of J

  or for $m = n$ $\det(J) \geq \sigma_0$

Vanilla newton method: reachable position





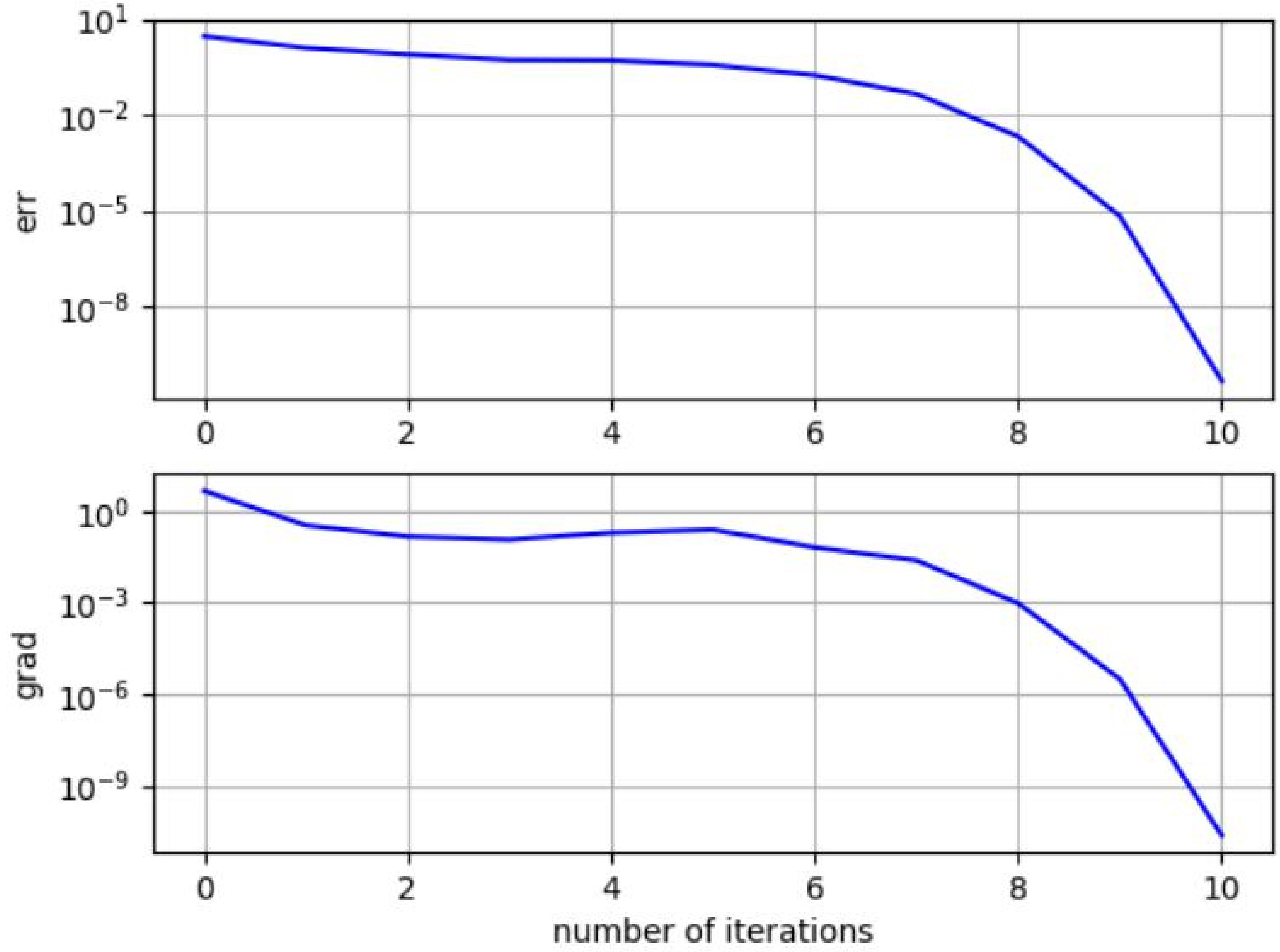
No line search

reachable position

Tolerance: $\varepsilon=10^{-6}$

Hessian regularization: $\lambda=10^{-8}$

With line search



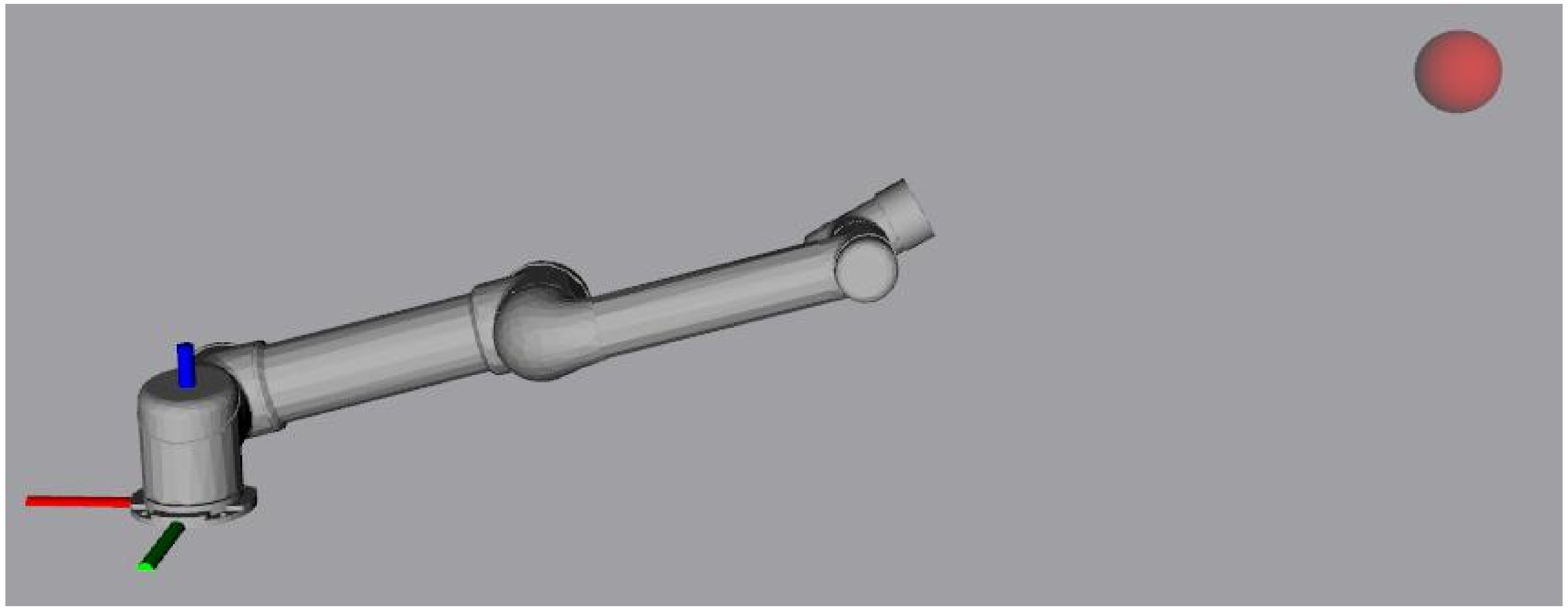
with line search

reachable position

Tolerance: $\varepsilon=10^{-6}$

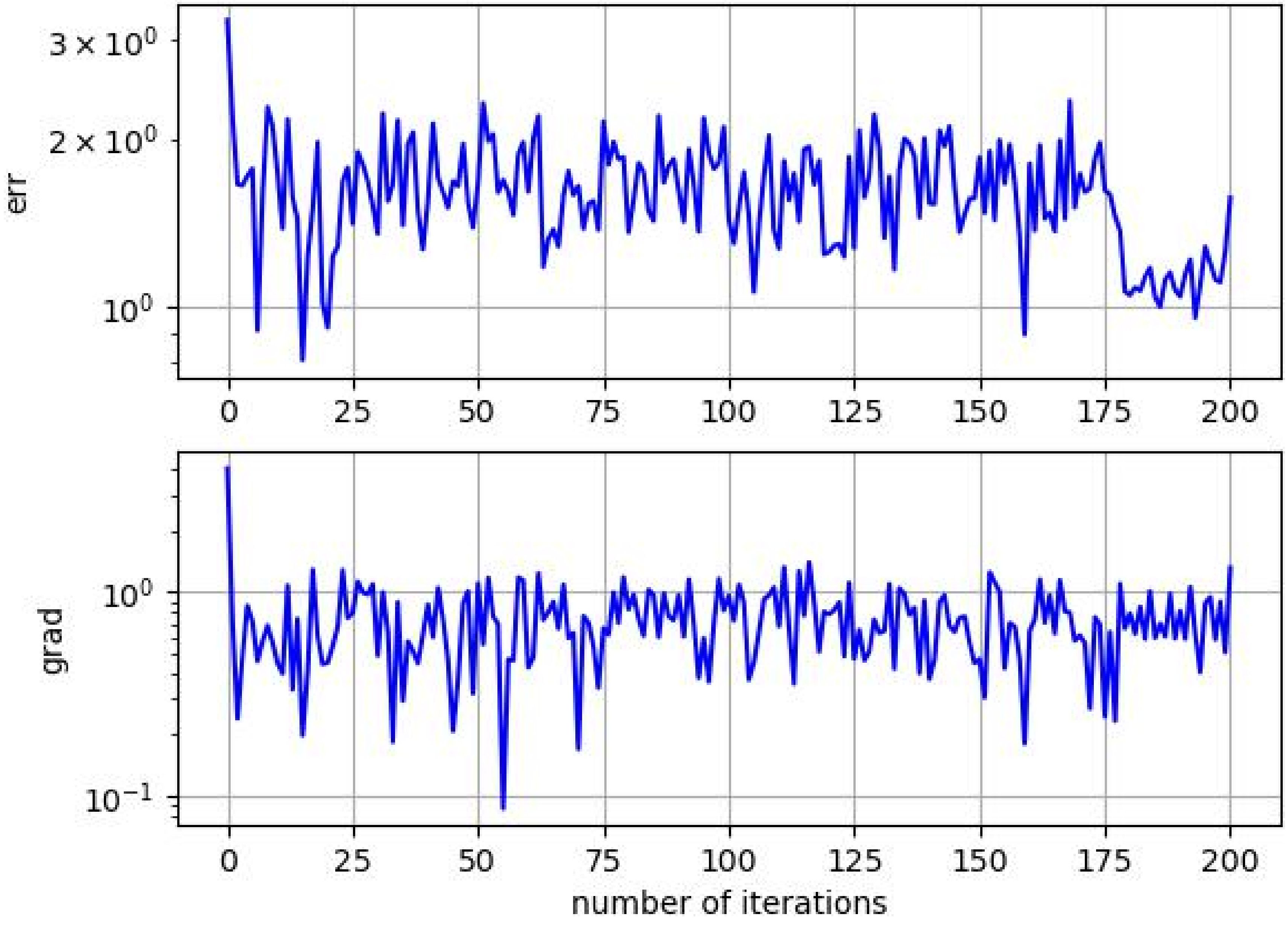
Hessian regularization: $\lambda=10^{-8}$

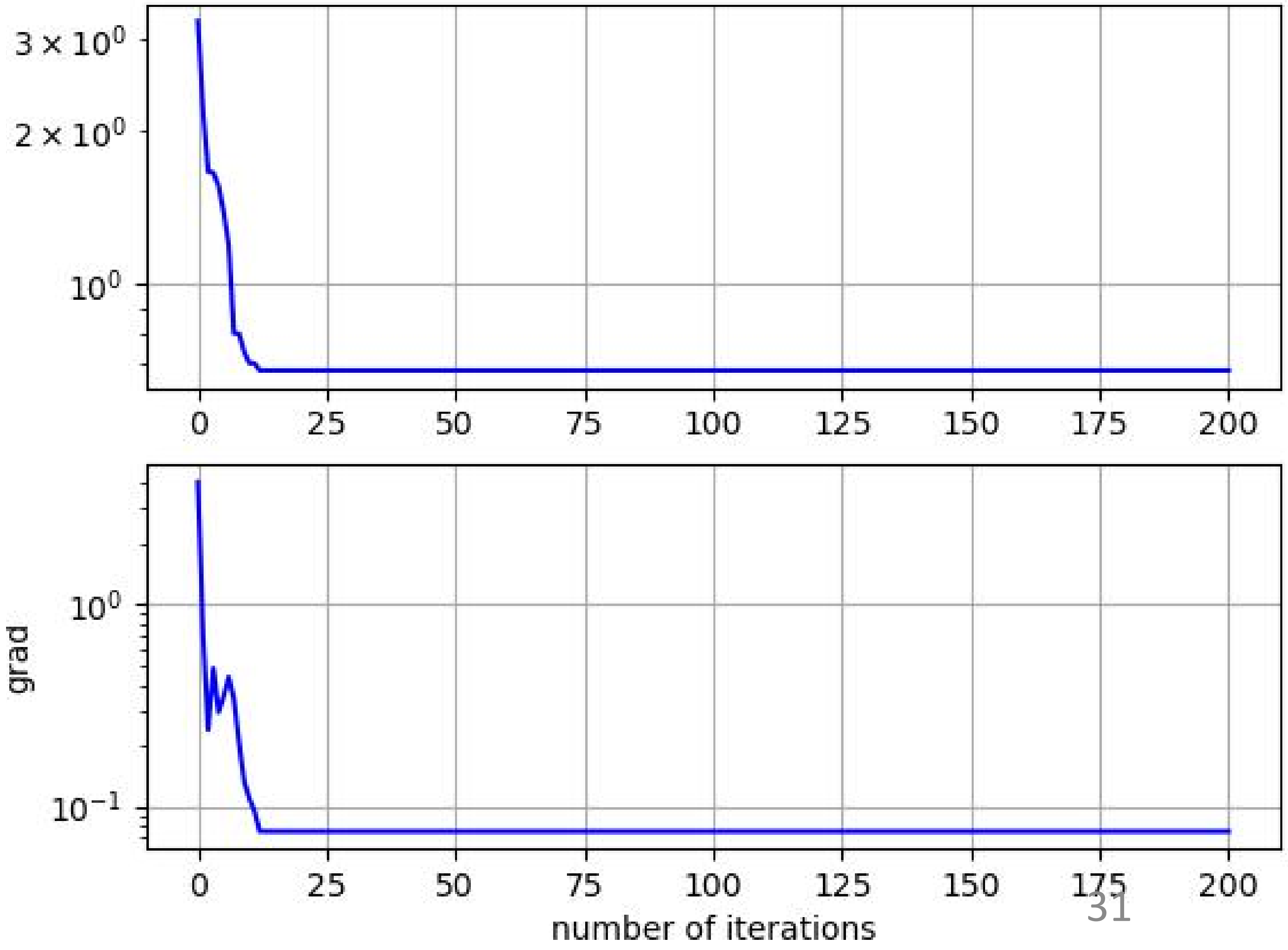
Vanilla newton method: not reachable position



No line search

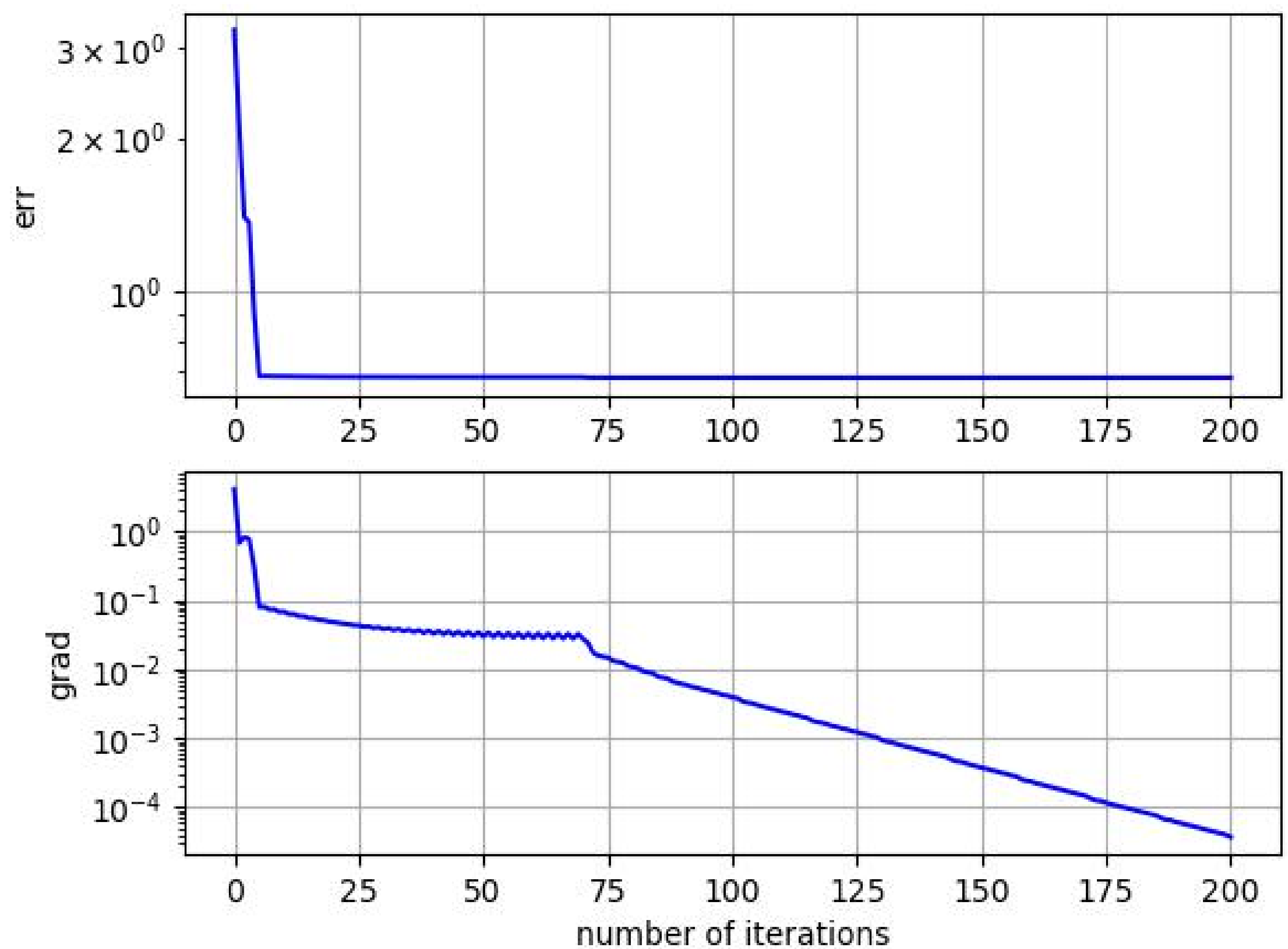
With line search



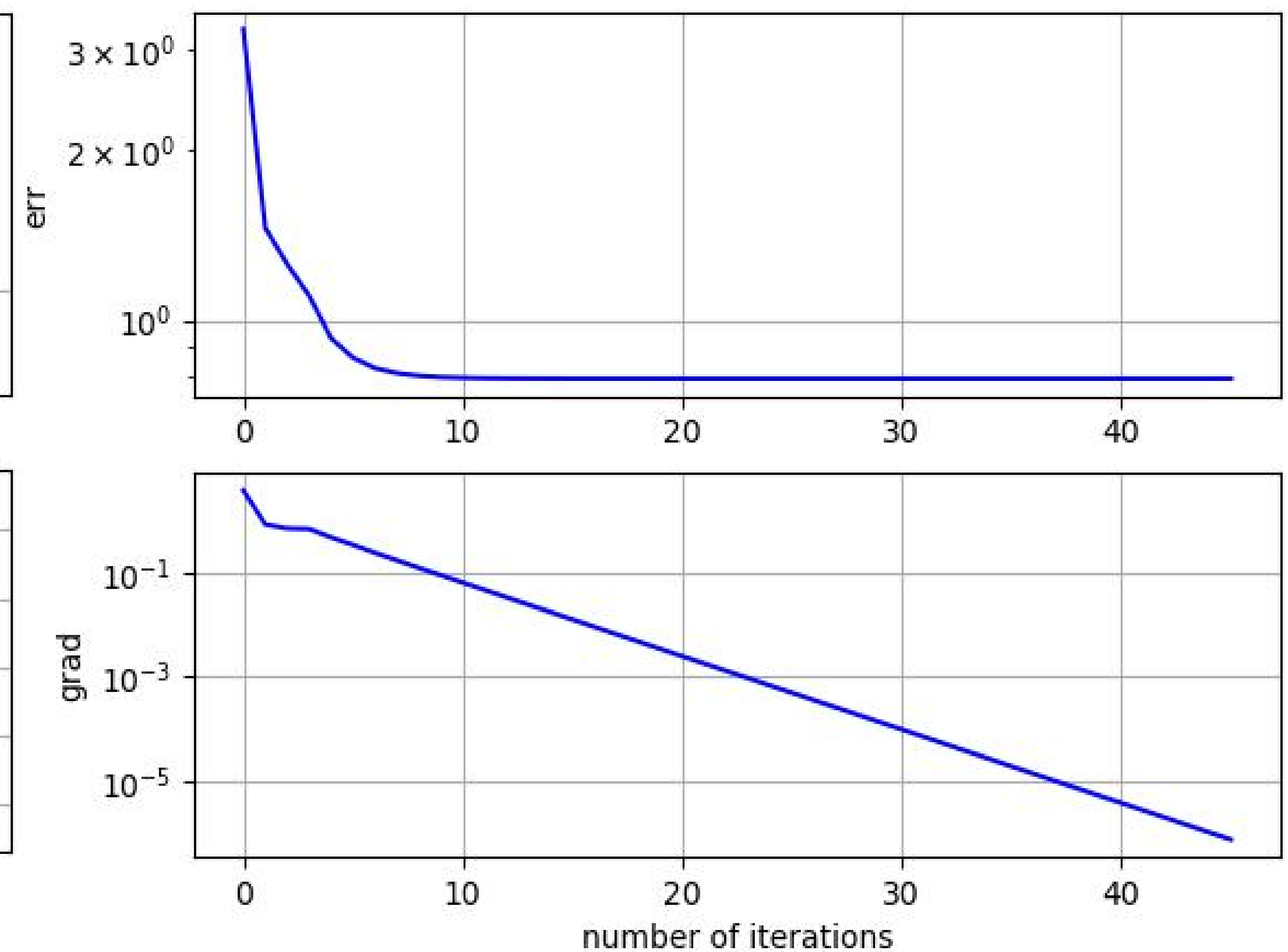


Playing with regularization

Hessian regularization: $\lambda=10^{-3}$

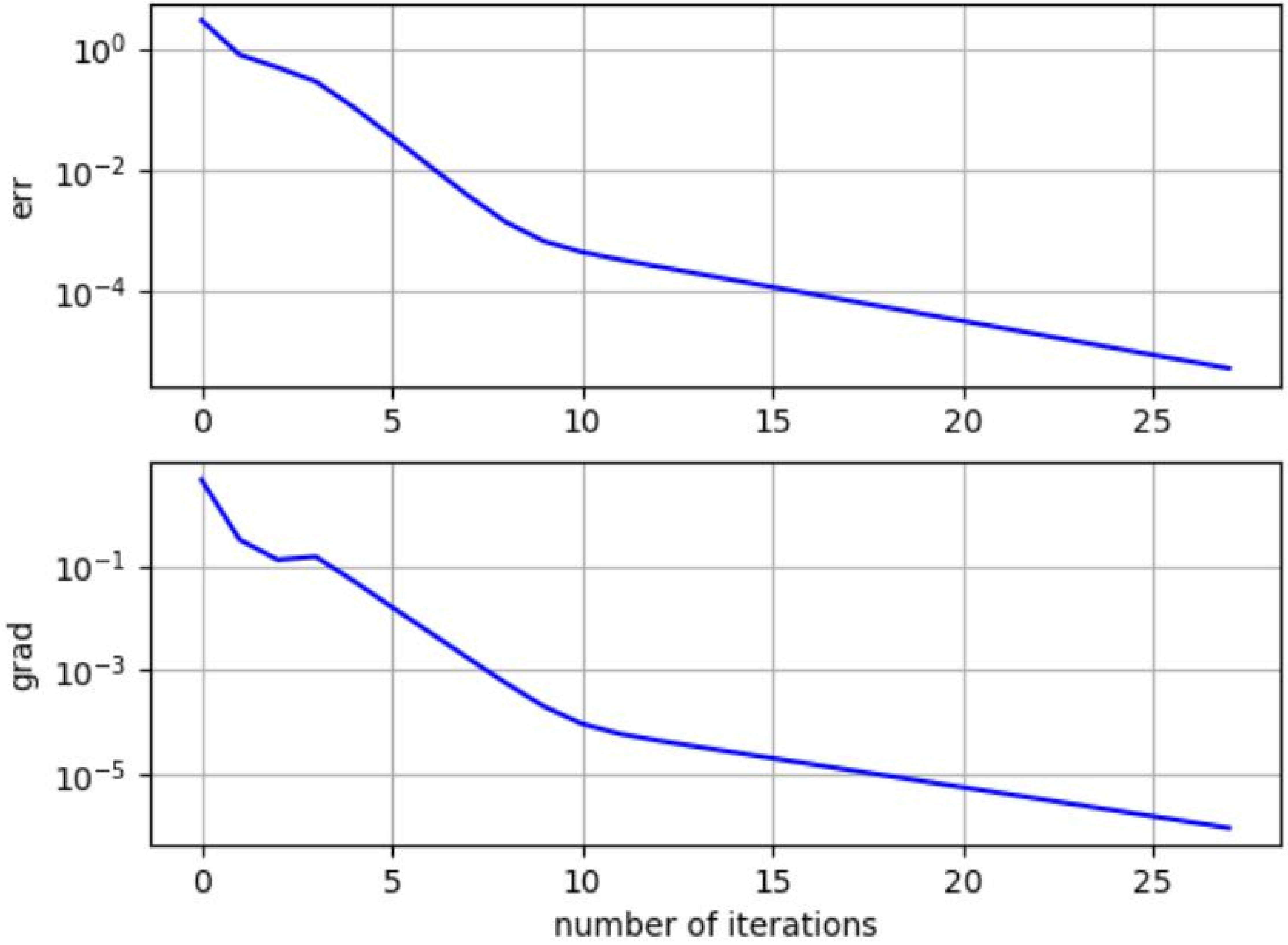


Hessian regularization: $\lambda=10^{-1}$



High regularization with the reachable case

Hessian regularization: $\lambda=10^{-3}$



Slows down convergence: 29 iterations instead of 10

Take home messages

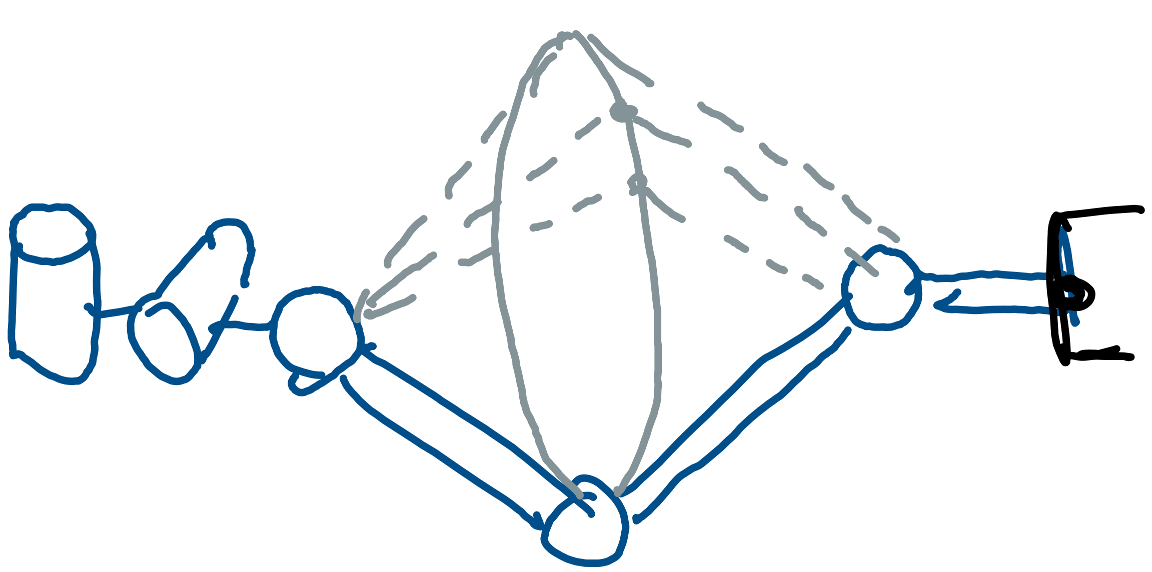
- Line search is needed to ensure convergence (even though sometimes it converges even without)
- Regularization helps speeding up convergence (e.g. in the case of unreachable position)
- Too much regularization can slow down convergence
- Optimization algorithm usually have automatic methods to adjust regulations

Postural task for redundant robots

$m = 3$, $n > 3$ $\Rightarrow$ we have ∞ solutions to the IK that depend on $\bar{q}_0$

goal: have solution independent of initial guess $\bar{q}_0$ (solve issue 1)

idea: reformulate the cost to have the solution closer to a default configuration q^P (postural)



$$\tilde{q}^* = \arg\min_q \frac{1}{2} \Bigg\| \underbrace{\begin{bmatrix} p(q) \\ wq \end{bmatrix} - \begin{bmatrix} p^d \\ wq^d \end{bmatrix}}_{\tilde{e}(q) \in \mathbb{R}^{3+m}} \Bigg\|^2 = C(q)$$

$$C(q) = \frac{1}{2} \| \tilde{e}(q) \|^2 = \frac{1}{2} \tilde{e}(q)^T \tilde{e}(q)$$

the gradient of cost is:

$$\frac{\partial C}{\partial q} = \tilde{e}(q)^T \frac{\partial \tilde{e}}{\partial q} = \tilde{e}(q)^T \begin{bmatrix} \frac{\partial p}{\partial q} - \cancel{\frac{\partial p^d}{\partial q}} \\ w\frac{\partial q}{\partial q} - w\cancel{\frac{\partial q^d}{\partial q}} \end{bmatrix} = \tilde{e}(q)^T \underbrace{\begin{bmatrix} J \\ wI \end{bmatrix}}_{\tilde{J}}$$

$$= \begin{bmatrix} J^T & wI \end{bmatrix} \begin{bmatrix} p - p^d \\ w(q - q^d) \end{bmatrix} = \tilde{J}^T \tilde{e} = r(q)$$

residual we want to nullify

similarly to what we did before we can compute the Newton step considering a linear approximation of $r(q)$

$$r(q) \approx r(\bar{q}) + \nabla_q r(\bar{q})\,\Delta q = 0$$

- $$\nabla_q r(\bar{q}) = \underbrace{(\nabla_q \tilde{J}^T)}_{\approx 0}\tilde{e} + \tilde{J}^T \nabla_q e \approx \tilde{J}^T \tilde{J}$$

$$r(q) \approx \tilde{J}^T \tilde{e}(\bar{q}) + \tilde{J}^T \tilde{J}\,\Delta q = 0$$

Δq such that $r(q) = 0$ is:

$$\Delta q^* = -(\tilde{J}^T \tilde{J})^{-1} \tilde{J}^T \tilde{e}(\bar{q})$$

$$= -\left(\begin{bmatrix} J^T & wI \end{bmatrix} \begin{bmatrix} J \\ wI \end{bmatrix}\right)^{-1} (J^T e_x + w^2 e_q)$$

$$\Delta q^{*} = -\underbrace{(J^T J + w^2 I)^{-1}}_{\text{always invertible and positive definite}} (J^T e_x + w^2 e_q)$$

The postural task:

1. ensures the final solution is independent of initial guess
2. acts as a regularization
3. should consider $\tilde{e} \in \mathbb{R}^{3+n}$ instead of $e \in \mathbb{R}^3$ in line search
4. should consider $\tilde{J}^T \tilde{e}$ in convergence criteria

References:

- D. E. Whitney, Resolved *motion rate* control of manipulators and human, 1969.
- O. Kathib, A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation, 1987.
- P. Wensig, AME 50551 – Introduction To Robotics (L36): http://sites.nd.edu/pwensing/ame-50551-introduction-to-robotics/