Practical Course: Modeling, Simulation, Optimization

Week 5

Daniël Veldman

Chair in Dynamics, Control, and Numerics, Friedrich-Alexander-University Erlangen-Nürnberg

Contents

5.A Solutions Exercise Week 3

Chair
DYNAMICS, CONTROL
AND NUMERICS
FAU

Department of
DATA SCIENCE

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
NATURWISSENSCHAFTLICHE
FAKULTÄT

5.B Existence and uniqueness of minimizers

Existence of the infimum

We consider the minimization of a functional $J : U \to \mathbb{R}$ over a normed space U.
Note: U can be infinite dimensional.

We assume that $J(u) \geq 0$ for all $u \in U$.

We are also given a subset $U_{\mathrm{ad}} \subseteq U$ of admissible values for u.

Existence of the infimum

We consider the minimization of a functional $J : U \to \mathbb{R}$ over a normed space U.
Note: U can be infinite dimensional.

We assume that $J(u) \geq 0$ for all $u \in U$.

We are also given a subset $U_{\text{ad}} \subseteq U$ of admissible values for u.

Then $\{J(u) \mid u \in U_{\text{ad}}\}$ is a subset of $\mathbb{R}$ that is bounded from below (by 0). Therefore,

$$\inf_{u \in U_{\text{ad}}} J(u) = \inf\{J(u) \mid u \in U_{\text{ad}}\},$$

exists.
By definition of the infimum, there thus exists a sequence $u_1, u_2, u_3, \ldots$ in U_{ad} such that

$$J(u_k) \to \inf_{u \in U_{\text{ad}}} J(u).$$

This sequence is called a *minimizing sequence*.

Existence of the minimizer (finite dimensional case)

Question: does

$$\min_{u \in U_{\mathrm{ad}}} J(u)$$

exist? In other words, is there a minimizer $u^* \in U_{\mathrm{ad}}$ such that

$$J(u^*) = \inf_{u \in U_{\mathrm{ad}}} J(u)?$$

Existence of the minimizer (finite dimensional case)

Question: does

$$\min_{u \in U_{\mathrm{ad}}} J(u)$$

exist? In other words, is there a minimizer $u^* \in U_{\mathrm{ad}}$ such that

$$J(u^*) = \inf_{u \in U_{\mathrm{ad}}} J(u)?$$

First consider the case where U is finite dimensional.

Observe, if U_{ad} is closed and the minimizing sequence $u_1, u_2, u_3, \ldots$ is bounded, then it also has a limit in U_{ad}. This limit is a minimizer u^*.

Two important cases:

- U_{ad} is bounded and closed.
  It is immediate that the minimizing sequence is bounded.

Existence of the minimizer (finite dimensional case)

Question: does

$$\min_{u \in U_{\mathrm{ad}}} J(u)$$

exist? In other words, is there a minimizer $u^* \in U_{\mathrm{ad}}$ such that

$$J(u^*) = \inf_{u \in U_{\mathrm{ad}}} J(u)?$$

First consider the case where U is finite dimensional.

Observe, if U_{ad} is closed and the minimizing sequence $u_1, u_2, u_3, \ldots$ is bounded, then it also has a limit in U_{ad}. This limit is a minimizer u^*.

Two important cases:

- U_{ad} is bounded and closed.
  It is immediate that the minimizing sequence is bounded.
- $J(u)$ is coercive, i.e. $J(u_k) \to \infty$ if $|u_k| \to \infty$. Note: it is sufficient that $J(u) \geq |u|^2$.
  Then we can reason as follows.
  Suppose that the minimizing sequence $u_1, u_2, u_3, \ldots$ is unbounded.
  Then there exists a subsequence $u_{k_1}, u_{k_2}, u_{k_3}, \ldots$ such that $|u_{k_j}| \to \infty$.
  But $J(u_{k_j}) > |u_{k_j}|^2$, so also $J(u_{k_j}) \to \infty$.
  But then $J(u_{k_j})$ is not a minimizing sequence. Contradiction.
  Conclusion: the minimizing sequence must be bounded.

Existence of the minimizer (infinite dimensional case)

Question: does

$$\min_{u \in U_{\mathrm{ad}}} J(u)$$

exist? In other words, is there a minimizer $u^* \in U_{\mathrm{ad}}$ such that

$$J(u^*) = \inf_{u \in U_{\mathrm{ad}}} J(u)?$$

The infinite dimensional case is much more subtle.

Problem: We can no longer be sure that a bounded sequence has a (strong) limit. In other words, we do no longer have compactness.

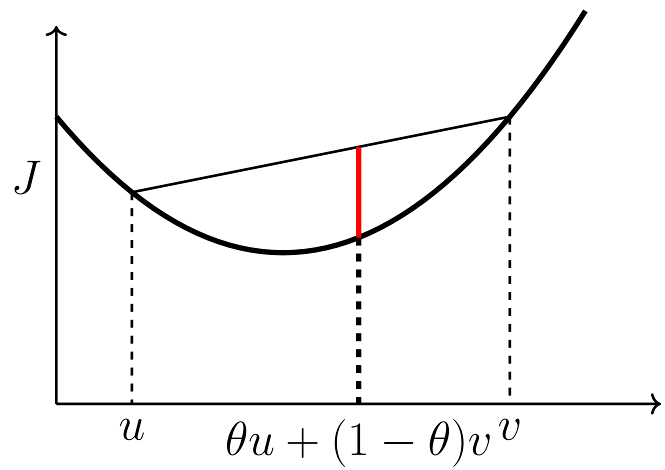
Typical example: consider $U_{\mathrm{ad}} = L^2(0, \pi)$ and consider the sequence $u_k = \sin(kx)$. This sequence converges weakly to zero, but does not have a strong limit.

We will come back to this problem in a few slides.

Uniqueness of the minimizer (convex analysis)

The functional $J(u)$ is called α-convex iff

$$J(\theta u + (1-\theta)v) \leq \theta J(u) + (1-\theta)J(v) - \frac{\alpha\theta(1-\theta)}{2}|u-v|^2, \qquad \theta \in [0,1].$$



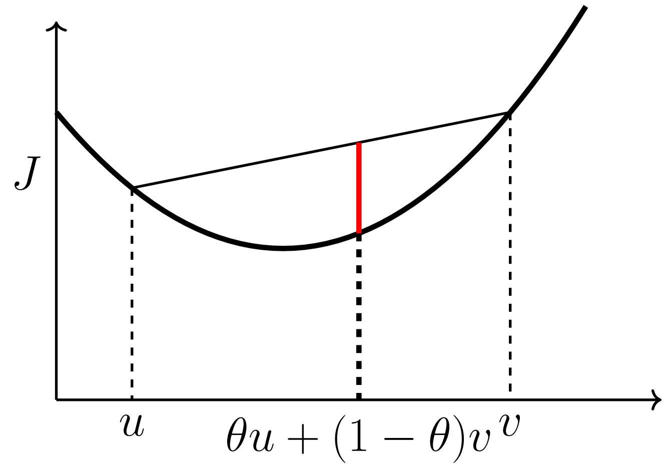
The admissible set U_{ad} is convex when $u, v \in U_{\mathrm{ad}}$

$$\theta u + (1-\theta)v \in U_{\mathrm{ad}}, \qquad \theta \in [0,1].$$

Uniqueness of the minimizer (convex analysis)

The functional $J(u)$ is called α-convex iff

$$J(\theta u + (1-\theta)v) \leq \theta J(u) + (1-\theta)J(v) - \frac{\alpha\theta(1-\theta)}{2}|u-v|^2, \qquad \theta \in [0,1].$$



The admissible set U_{ad} is convex when $u, v \in U_{\text{ad}}$

$$\theta u + (1-\theta)v \in U_{\text{ad}}, \qquad \theta \in [0,1].$$

Uniqueness of the minimizer:
Suppose that there are two points $u, v \in U_{\text{ad}}$ such that $J(u) = J(v) = \min_{u\in U_{\text{ad}}} J(u)$.

$$J(\theta u + (1-\theta)v) \leq \min_{u\in U_{\text{ad}}} J(u) - \frac{\alpha\theta(1-\theta)}{2}|u-v|^2 < \min_{u\in U_{\text{ad}}} J(u),$$

and $\theta u + (1-\theta)v \in U_{\text{ad}}$. Contradiction.

Existence of the minimizer (infinite dimensional case, revisited)

Question: does

$$\min_{u \in U_{\text{ad}}} J(u)$$

exist? In other words, is there a minimizer $u^* \in U_{\text{ad}}$ such that

$$J(u^*) = \inf_{u \in U_{\text{ad}}} J(u)?$$

Consider a minimizing sequence $u_1, u_2, u_3, \dots$.
The minimizing sequence is bounded when U_{ad} is bounded or when J is coercive.
The bounded minimizing sequence $u_1, u_2, u_3, \dots$ has a weak limit v.

Existence of the minimizer (infinite dimensional case, revisited)

Question: does

$$\min_{u \in U_{\text{ad}}} J(u)$$

exist? In other words, is there a minimizer $u^* \in U_{\text{ad}}$ such that

$$J(u^*) = \inf_{u \in U_{\text{ad}}} J(u)?$$

Consider a minimizing sequence $u_1, u_2, u_3, \ldots$.
The minimizing sequence is bounded when U_{ad} is bounded or when J is coercive.
The bounded minimizing sequence $u_1, u_2, u_3, \ldots$ has a weak limit v.

Now three problems remain:

- Is the weak limit $v \in U_{\text{ad}}$?
  If U_{ad} is strongly closed and convex, it is also weakly closed (Hahn-Banach).
- Do we have that $J(v) = \lim_{k \to \infty} J(u_k) = \inf_{u \in U_{\text{ad}}} J(u)$?
  This is achieved by assuming that J is weakly lower semi-continuous (by definition).
- Does the minimizing sequence $u_1, u_2, u_3, \ldots$ also converge strongly to v?
  This follows from the previous point and the strong convexity of J (with $\theta = \frac{1}{2}$):

$$J(v) \leq J(\tfrac{u_k+v}{2}) \leq \frac{J(u_k)+J(v)}{2} - \frac{\alpha}{8}|u_k - v|^2, \quad \Rightarrow \quad \frac{\alpha}{8}|u_k - v|^2 \leq \frac{J(u_k)-J(v)}{2} \to 0.$$

5.C A basic gradient descent algorithm

Gradient descent

Question: How to we compute the minimizer u^* of a (convex) functional $J(u)$.

Basic idea: Start from an initial guess u_0.
Compute iterates by updating u_k in the direction of the steepest descent (i.e. $-\nabla J$),

$$u_{k+1} = u_k - \beta_k \nabla J(u_k), \qquad\qquad \beta_k > 0,$$

where β denotes the step size.

Gradient descent

Question: How to we compute the minimizer u^* of a (convex) functional $J(u)$.

Basic idea: Start from an initial guess u_0.
Compute iterates by updating u_k in the direction of the steepest descent (i.e. $-\nabla J$),

$$u_{k+1} = u_k - \beta_k \nabla J(u_k), \qquad \beta_k > 0,$$

where β denotes the step size.

Three problems:

- How to compute ∇J?
- How to choose the stepsize β_k?
- When do we stop the iterations?

Computation of the gradient/ sensitivity analysis

By definition of the gradient, we have that

$$\langle \nabla J, \tilde{u} \rangle := \lim_{h \to 0} \frac{J(u + h\tilde{u}) - J(u)}{h} = \frac{\partial J}{\partial u}(u)\tilde{u},$$

for all perturbations $\tilde{u}$.

Note:

- $\nabla J(u)$ and $\frac{\partial J}{\partial u}$ are not the same:
  $\nabla J(u)$ is a column vector and $\frac{\partial J}{\partial u}$ is a row vector.
- We can use any innerproduct $\langle \cdot, \cdot \rangle$ at the LHS.
  This will not affect $\frac{\partial J}{\partial u}$ but it will change ∇J!

Computation of the gradient/ sensitivity analysis

By definition of the gradient, we have that

$$\langle \nabla J, \tilde{u} \rangle := \lim_{h \to 0} \frac{J(u + h\tilde{u}) - J(u)}{h} = \frac{\partial J}{\partial u}(u)\tilde{u},$$

for all perturbations $\tilde{u}$.

Note:

- $\nabla J(u)$ and $\frac{\partial J}{\partial u}$ are not the same:
  $\nabla J(u)$ is a column vector and $\frac{\partial J}{\partial u}$ is a row vector.
- We can use any innerproduct $\langle \cdot, \cdot \rangle$ at the LHS.
  This will not affect $\frac{\partial J}{\partial u}$ but it will change ∇J!

Two examples:

- When $\langle x, y \rangle = x^\top y$, i.e. when we use the standard Euclidean inner product

$$\nabla J = \left(\frac{\partial J}{\partial u} \right)^\top.$$

- When we use a weighted inner product $\langle x, y \rangle = x^\top \mathbf{W} y$, for a symmetric and positive definite matrix $\mathbf{W}$, we get that

$$\nabla J = \mathbf{W}^{-1} \left(\frac{\partial J}{\partial u} \right)^\top.$$

Intermezzo: Why the choice of inner product matters/helps

Suppose that $J(u) = \langle u+b, u\rangle = (u+b)^\top \mathbf{W} u$.
(Any quadratic functional with Hessian $\mathbf{W}$ can be written in this form)

$$\begin{aligned}\langle \nabla J, \tilde{u}\rangle &:= \lim_{h\to 0} \frac{J(u+h\tilde{u}) - J(u)}{h} = \lim_{h\to 0} \frac{\langle u+h\tilde{u}+b, u+h\tilde{u}\rangle - \langle u+b, u\rangle}{h}, \\ &= \lim_{h\to 0} \frac{\langle u+b, u\rangle + h\langle u+b, \tilde{u}\rangle + h\langle \tilde{u}, u\rangle + h^2\langle \tilde{u}, \tilde{u}\rangle - \langle u+b, u\rangle}{h} \\ &= \langle 2u+b, \tilde{u}\rangle.\end{aligned}$$

Intermezzo: Why the choice of inner product matters/helps

Suppose that $J(u) = \langle u+b, u\rangle = (u+b)^\top \mathbf{W} u$.
(Any quadratic functional with Hessian $\mathbf{W}$ can be written in this form)

$$\begin{aligned}
\langle \nabla J, \tilde{u}\rangle &:= \lim_{h\to 0} \frac{J(u+h\tilde{u}) - J(u)}{h} = \lim_{h\to 0} \frac{\langle u+h\tilde{u}+b, u+h\tilde{u}\rangle - \langle u+b, u\rangle}{h}, \\
&= \lim_{h\to 0} \frac{\langle u+b, u\rangle + h\langle u+b, \tilde{u}\rangle + h\langle \tilde{u}, u\rangle + h^2\langle \tilde{u}, \tilde{u}\rangle - \langle u+b, u\rangle}{h} \\
&= \langle 2u+b, \tilde{u}\rangle.
\end{aligned}$$

We thus see that

$$\nabla J(u) = 2u + b, \qquad u^* = -\tfrac{1}{2}b.$$

Suppose we have an initial guess u_0 and take the stepsize $\beta_0 = \frac{1}{2}$. Then

$$u_1 = u_0 - \tfrac{1}{2}\nabla J(u_0) = u_0 - \tfrac{1}{2}(2u_0 + b) = -\tfrac{1}{2}b = u^*.$$

Conclusion: when we have a quadratic cost functional with Hessian $\mathbf{W}$ and compute the gradient w.r.t. the inner product $\langle \mathbf{u}, \mathbf{v}\rangle = \mathbf{u}^\top \mathbf{W}\mathbf{v}$, the gradient descent algorithm converges in 1 iteration (with $\beta = \frac{1}{2}$).

However, this idea is not directly applicable: often, the Hessian cannot be computed easily and the considered cost functionals are not quadratic.
Even in these situation, choosing $\mathbf{W}$ well can improve the convergence.

The choice of the step size

We have that

$$\begin{aligned} J(u_{k+1}) = J(u_k - \beta_k \nabla J(u_k)) &= J(u_k) - \beta_k \frac{\partial J}{\partial u_k} \nabla J(u_k) + O(\beta_k^2) \\ &= J(u_k) - \beta_k \langle \nabla J(u_k), \nabla J(u_k) \rangle + O(\beta_k^2). \end{aligned}$$

As long as we are not at a critical point ($\nabla J(u_k) = 0$) $\langle \nabla J(u_k), \nabla J(u_k) \rangle > 0$, so

$$J(u_{k+1}) < J(u_k)$$

for $\beta_k > 0$ small enough.

The choice of the step size

We have that

$$\begin{aligned} J(u_{k+1}) = J(u_k - \beta_k \nabla J(u_k)) &= J(u_k) - \beta_k \frac{\partial J}{\partial u_k} \nabla J(u_k) + O(\beta_k^2) \\ &= J(u_k) - \beta_k \langle \nabla J(u_k), \nabla J(u_k) \rangle + O(\beta_k^2). \end{aligned}$$

As long as we are not at a critical point ($\nabla J(u_k) = 0$) $\langle \nabla J(u_k), \nabla J(u_k) \rangle > 0$, so

$$J(u_{k+1}) < J(u_k)$$

for $\beta_k > 0$ small enough.

We can thus take the following simple but effective approach (used at every iteration).

- Choose a step size $\beta > 0$.
- Compute $J(u_k - \beta \nabla J(u_k))$.
- If $J(u_k - \beta \nabla J(u_k)) < J(u_k)$, we accept this step size.
  If not, we reduce the step size (e.g. by a factor 2) and recompute $J(u_k - \beta \nabla J(u_k))$.

This should always lead to a $\beta_k > 0$ such that $J(u_k - \beta \nabla J(u_k)) < J(u_k)$.
(Provided that $\nabla J(u_k)$ is computed sufficiently accurate)

Improved step size selection

For a convex C^2-functional $J(\mathbf{u})$,
we can estimate the stepsize based on a quadratic approximation:

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k), \qquad \beta_k > 0,$$

$$J(\mathbf{u}_{k+1}) \approx J(\mathbf{u}_k) - \beta_k G + \frac{H}{2}\beta_k^2 + O(\beta_k^3),$$

with

$$G = \langle \nabla J(\mathbf{u}_k), \nabla J(\mathbf{u}_k) \rangle,$$
$$H = \left[\frac{\mathrm{d}^2}{\mathrm{d}\theta^2} J(\mathbf{u}_k + \theta \nabla J(\mathbf{u}_k)) \right]_{\theta=0}.$$

Note: G is positive because we update in a descent direction.
H is positive because J is convex.

Improved step size selection

For a convex C^2-functional $J(\mathbf{u})$,
we can estimate the stepsize based on a quadratic approximation:

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k), \qquad\qquad \beta_k > 0,$$

$$J(\mathbf{u}_{k+1}) \approx J(\mathbf{u}_k) - \beta_k G + \frac{H}{2}\beta_k^2 + O(\beta_k^3),$$

with

$$G = \langle \nabla J(\mathbf{u}_k), \nabla J(\mathbf{u}_k) \rangle,$$
$$H = \left[\frac{\mathrm{d}^2}{\mathrm{d}\theta^2} J(\mathbf{u}_k + \theta \nabla J(\mathbf{u}_k)) \right]_{\theta=0}.$$

Note: G is positive because we update in a descent direction.
H is positive because J is convex.
Set derivative of the quadratic approximation to zero:

$$-G + H\beta_{k,\mathrm{opt}} = 0, \qquad\qquad \beta_{k,\mathrm{opt}} = \frac{G}{H}.$$

Improved step size selection

For a convex C^2-functional $J(\mathbf{u})$,
we can estimate the stepsize based on a quadratic approximation:

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k), \qquad \beta_k > 0,$$

$$J(\mathbf{u}_{k+1}) \approx J(\mathbf{u}_k) - \beta_k G + \frac{H}{2}\beta_k^2 + O(\beta_k^3),$$

with

$$G = \langle \nabla J(\mathbf{u}_k), \nabla J(\mathbf{u}_k) \rangle,$$
$$H = \left[\frac{\mathrm{d}^2}{\mathrm{d}\theta^2} J(\mathbf{u}_k + \theta \nabla J(\mathbf{u}_k)) \right]_{\theta=0}.$$

Note: G is positive because we update in a descent direction.
H is positive because J is convex.
Set derivative of the quadratic approximation to zero:

$$-G + H\beta_{k,\mathrm{opt}} = 0, \qquad \beta_{k,\mathrm{opt}} = \frac{G}{H}.$$

When J is quadratic, $J(\mathbf{u}_k + \beta_{k,\mathrm{opt}} \nabla J(\mathbf{u}_k)) = J(\mathbf{u}_k) - \beta_{k,\mathrm{opt}} G + \frac{H}{2}\beta_{k,\mathrm{opt}}^2 = J(\mathbf{u}_k) - \frac{G^2}{2H}$
When J is not quadratic, there are higher order terms and we cannot guarantee that $J(\mathbf{u}_k + \beta_{k,\mathrm{opt}} \nabla J(\mathbf{u}_k)) \leq J(\mathbf{u}_k)$. We still need to do a line search (starting from $\beta_{k,\mathrm{opt}}$).

Computation of H (example)

Consider the optimization problem

$$\min_{u \in U_{\mathrm{ad}}} \tfrac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \tfrac{1}{2}\mathbf{u}^\top \mathbf{R}\mathbf{u}$$

with $\mathbf{Q} = \mathbf{Q}^\top$, $\mathbf{R} = \mathbf{R}^\top$, $\mathbf{u} \in U_{\mathrm{ad}} \subset \mathbb{R}^M$, and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

As explained before, we can compute the gradient $\nabla J(\mathbf{u}_k)$ at the current iterate $\mathbf{u}_k$. We want to compute

$$H = \left[\frac{\mathrm{d}^2}{\mathrm{d}\theta^2} J(\mathbf{u_k} + \theta \nabla J(\mathbf{u}_k))\right]_{\theta=0}.$$

Observe that

$$\begin{aligned} J(\mathbf{u}_k + \theta\nabla J) &= \tfrac{1}{2}(\mathbf{x}_k + \theta\mathbf{x}_k^\nabla)^\top \mathbf{Q}(\mathbf{x}_k + \theta\mathbf{x}_k^\nabla) + \tfrac{1}{2}(\mathbf{u}_k + \theta\nabla J(\mathbf{u}_k))^\top \mathbf{R}(\mathbf{u}_k + \theta\nabla J(\mathbf{u}_k)) \\ &= \tfrac{1}{2}\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \tfrac{1}{2}\mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k + \theta\left(\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k^\nabla + \mathbf{u}_k^\top \mathbf{R}\nabla J(\mathbf{u}_k)\right) \\ &\qquad \theta^2 \left(\tfrac{1}{2}\left(\mathbf{x}_k^\nabla\right)^\top \mathbf{Q}\mathbf{x}_k^\nabla + \tfrac{1}{2}(\nabla J(\mathbf{u}_k))^\top \mathbf{R}\nabla J(\mathbf{u}_k)\right), \end{aligned}$$

where $\mathbf{x}_k = \mathbf{A}^{-1}\mathbf{B}\mathbf{u}_k$ and $\mathbf{x}_k^\nabla = \mathbf{A}^{-1}\mathbf{B}\nabla J(\mathbf{u}_k)$. Differentiating twice to θ, we obtain

$$H = \left(\mathbf{x}_k^\nabla\right)^\top \mathbf{Q}\mathbf{x}_k^\nabla + (\nabla J(\mathbf{u}_k))^\top \mathbf{R}\nabla J(\mathbf{u}_k).$$

Termination/convergence conditions

Typical convergence conditions:

- Relative decrease in the cost functional is sufficiently small:

$$J(u_k) - J(u_{k+1}) < \texttt{tol} J(u_k).$$

- Relative change in iterates is sufficiently small:

$$|u_{k-1} - u_k| < \texttt{tol}|u_k|.$$

- The gradient is sufficiently small:

$$|\nabla J(u_k)| < \texttt{tol}.$$

In the first two conditions, we typically use $\texttt{tol} \in [10^{-6}, 10^{-3}]$.

Often not all three conditions are checked simultaneously, but only one or two are used.

Note: `tol` in the last condition is an absolute tolerance, while `tol` in the first two conditions is a relative tolerance.
A reasonable magnitude for the absolute tolerance might be difficult to estimate.

Pseudo code of the resulting gradient descent algorithm

- Choose an initial guess u_0
- Choose an initial step size β
- Compute $J_0 = J(u_0)$.
- for i = 1: `max_iters`
-     Compute $g_0 = \nabla J(u_0)$.
-     Set $J_1 = \infty$ and $\beta = 4\beta$.
-     while $J_1 > J_0$
-         Set $\beta = \beta/2$.
-         Set $u_1 = u_0 - \beta g_0$.
-         Compute $J_1 = J(u_1)$.
-     if convergence conditions are satisfied
-         Return u_1, J_1.
-     Set $u_0 = u_1$
-     Set $J_0 = J_1$

Chair DYNAMICS, CONTROL AND NUMERICS FAU

Department of DATA SCIENCE

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

NATURWISSENSCHAFTLICHE FAKULTÄT

5.D Equality constraints

Constrained optimization

Consider the optimization problem

$$\min_{u \in U_{\mathrm{ad}}} J(\mathbf{x}, \mathbf{u})$$

with $\mathbf{u} \in U_{\mathrm{ad}} \subset \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

Assume that $\mathbf{A}$ is invertible such that we can consider $J(\mathbf{x}(\mathbf{u}), \mathbf{u}) =: \tilde{J}(\mathbf{u})$.

Question: How to compute the Jacobian?

ANSWER 1: By finite differences.
Choose a step size h (typically 10^{-5}) and approximate for every $m \in \{1, 2, \dots, M\}$

$$\left(\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{u}}(\mathbf{u})\right)_m = \frac{\mathrm{d}\tilde{J}}{\mathrm{d}u_m}(\mathbf{u}) \approx \frac{\tilde{J}(\mathbf{u} + h\mathbf{e}_m) - J(\mathbf{u})}{h} = \frac{J(\mathbf{x} + \delta\mathbf{x}_m, \mathbf{u} + h\mathbf{e}_m) - J(\mathbf{x}, \mathbf{u})}{h},$$

where $\delta\mathbf{x}_m$ satisfies

$$\mathbf{A}\delta\mathbf{x}_m + h\mathbf{B}\mathbf{e}_m = \mathbf{0}.$$

Note: we need to solve M linear systems in N unknowns.
This is very time-consuming when M and N are large.

Constrained optimization

Consider the optimization problem

$$\min_{u \in U_{\mathrm{ad}}} J(\mathbf{x}, \mathbf{u})$$

with $\mathbf{u} \in U_{\mathrm{ad}} \subset \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

Assume that $\mathbf{A}$ is invertible such that we can consider $J(-\mathbf{A}^{-1}\mathbf{B}\mathbf{u}, \mathbf{u}) =: \tilde{J}(\mathbf{u})$.

Question: How to compute the Jacobian?

ANSWER 2: Analytically.
Similarly, as in the exercise we can use the chain rule to find

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{u}} = \frac{\partial J}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{u}} + \frac{\partial J}{\partial \mathbf{u}} = -\frac{\partial J}{\partial \mathbf{x}}\mathbf{A}^{-1}\mathbf{B} + \frac{\partial J}{\partial \mathbf{u}}.$$

Constrained optimization

Consider the optimization problem

$$\min_{u \in U_{\mathrm{ad}}} J(\mathbf{x}, \mathbf{u})$$

with $\mathbf{u} \in U_{\mathrm{ad}} \subset \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

Assume that $\mathbf{A}$ is invertible such that we can consider $J(-\mathbf{A}^{-1}\mathbf{B}\mathbf{u}, \mathbf{u}) =: \tilde{J}(\mathbf{u})$.

Question: How to compute the Jacobian?

ANSWER 2: Analytically.
Similarly, as in the exercise we can use the chain rule to find

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{u}} = \frac{\partial J}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{u}} + \frac{\partial J}{\partial \mathbf{u}} = -\frac{\partial J}{\partial \mathbf{x}}\mathbf{A}^{-1}\mathbf{B} + \frac{\partial J}{\partial \mathbf{u}}.$$

The computational cost depends on where you put the brackets:

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{u}} = -\frac{\partial J}{\partial \mathbf{x}}(\mathbf{A}^{-1}\mathbf{B}) + \frac{\partial J}{\partial \mathbf{u}} = -\left(\frac{\partial J}{\partial \mathbf{x}}\mathbf{A}^{-1}\right)\mathbf{B} + \frac{\partial J}{\partial \mathbf{u}}.$$

Note: the first expression requires the solution of M linear system in N unknowns, whereas the second requires requires the solution of 1 linear system in N unknowns.

Constrained optimization

Consider the optimization problem

$$\min_{u \in U_{\mathrm{ad}}} J(\mathbf{x}, \mathbf{u})$$

with $\mathbf{u} \in U_{\mathrm{ad}} \subset \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

Assume that $\mathbf{A}$ is invertible such that we can consider $J(\mathbf{x}(\mathbf{u}), \mathbf{u}) =: \tilde{J}(\mathbf{u})$.

Question: How to compute the Jacobian?

ANSWER 3: Using the Lagrangian.
Introduce the vector of Lagrange multipliers $\boldsymbol{\lambda}$ and form the Lagrangian

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = J(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}).$$

Take the partial derivative w.r.t. $\mathbf{u}$ to find the Jacobian

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{u}} = \frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \frac{\partial J}{\partial \mathbf{u}} + \boldsymbol{\lambda}^\top \mathbf{B}\mathbf{u}.$$

Set the partial derivative w.r.t. $\mathbf{x}$ to zero to determine $\boldsymbol{\lambda}$:

$$\mathbf{0} = \frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial J}{\partial \mathbf{x}} + \boldsymbol{\lambda}^\top \mathbf{A}, \qquad -\boldsymbol{\lambda}^\top = \frac{\partial J}{\partial \mathbf{x}} \mathbf{A}^{-1}, \qquad \boldsymbol{\lambda} = -\left(\mathbf{A}^\top\right)^{-1} \left(\frac{\partial J}{\partial \mathbf{x}}\right)^\top.$$

The result is the same as for answer 2 (with well-placed brackets).

5.E Inequality constraints

Inequality constraints

Consider the optimization problem

$$\min_{u \in U_{\mathrm{ad}}} J(\mathbf{u}) = J(\mathbf{x}(\mathbf{u}), \mathbf{u})$$

with $\mathbf{u} \in U_{\mathrm{ad}} \subset \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

We distinguish between two types of constraints:

- Constraints on $\mathbf{u}$ ('input constraints'), $g(\mathbf{u}) \geq \mathbf{0}$
- Constraints on $\mathbf{x}(\mathbf{u})$ ('state constraints') $h(\mathbf{x}(\mathbf{u})) \geq \mathbf{0}$.

Input constraints can be easily incorporated with the projected gradient method.

Projected gradient method

Suppose we want to solve an optimization problem with the constraints:

$$a \leq u_m \leq b, \qquad m \in \{1, 2, \ldots, M\}.$$

(This thus defines the admissible set U_{ad})

Projected gradient method

Suppose we want to solve an optimization problem with the constraints:

$$a \leq u_m \leq b, \qquad\qquad m \in \{1, 2, \ldots, M\}.$$

(This thus defines the admissible set U_{ad})

Problem: We do not know whether $\mathbf{u}_{k+1} = \mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)$ is in U_{ad}.
(Even when $\mathbf{u}_k \in U_{\mathrm{ad}}$)

Solution: Project $\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)$ onto the U_{ad}, i.e. do the update as

$$\mathbf{u}_{k+1} = \Pi_{U_{\mathrm{ad}}}(\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)) \in U_{\mathrm{ad}}.$$

Projected gradient method

Suppose we want to solve an optimization problem with the constraints:

$$a \le u_m \le b, \qquad m \in \{1, 2, \ldots, M\}.$$

(This thus defines the admissible set U_{ad})

Problem: We do not know whether $\mathbf{u}_{k+1} = \mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)$ is in U_{ad}.
(Even when $\mathbf{u}_k \in U_{\mathrm{ad}}$)

Solution: Project $\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)$ onto the U_{ad}, i.e. do the update as

$$\mathbf{u}_{k+1} = \Pi_{U_{\mathrm{ad}}} (\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)) \in U_{\mathrm{ad}}.$$

In general, the projection onto the admissible set is difficult to compute
(it requires the solution of another optimization problem).

However, for the considered admissible set, the computation is straightforward:

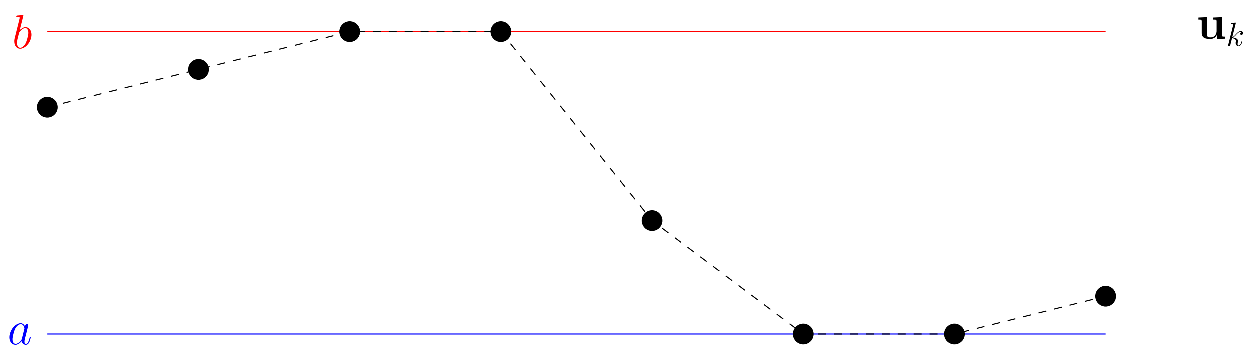
$$\left(\Pi_{U_{\mathrm{ad}}}(\mathbf{u})\right)_m = \begin{cases} a & u_m \le a, \\ u_m & a < u_m < b, \\ b & u_m \ge b. \end{cases}$$

Projected gradient method (graphical illustration)

$$a \leq u_m \leq b, \qquad m \in \{1, 2, \ldots, M\}.$$

$$\mathbf{u}_{k+1} = \Pi_{U_{\mathrm{ad}}} (\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)) \in U_{\mathrm{ad}}$$

$$(\Pi_{U_{\mathrm{ad}}}(\mathbf{u}))_m = \begin{cases} a & u_m \leq a \\ u_m & a < u_m < b \\ b & u_m \geq b \end{cases}$$

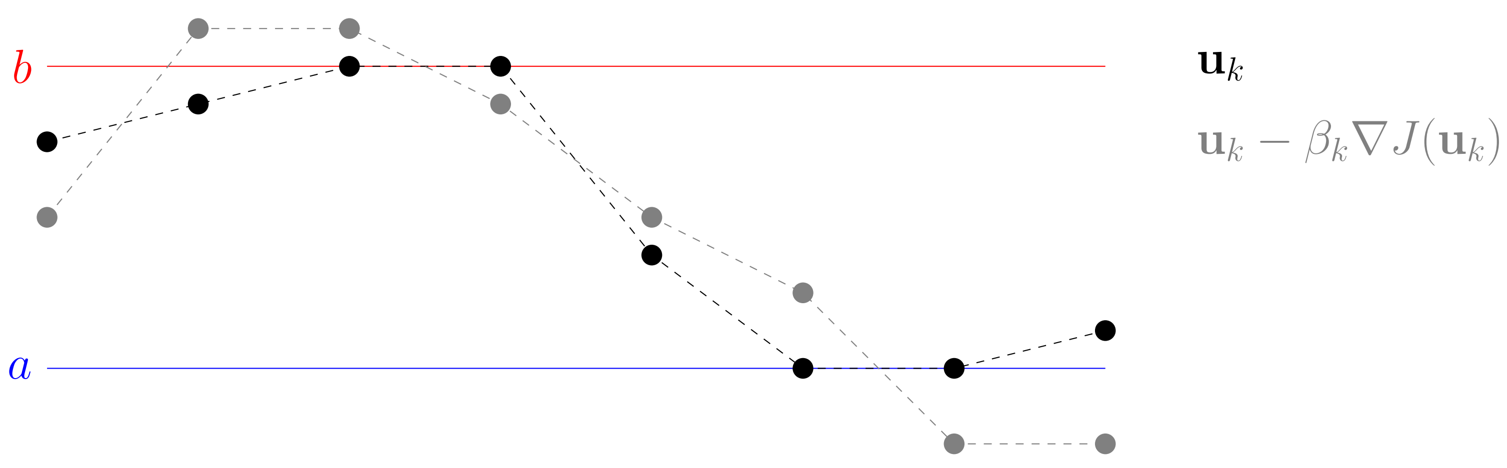


Projected gradient method (graphical illustration)

$$a \leq u_m \leq b, \qquad m \in \{1, 2, \dots, M\}.$$

$$\mathbf{u}_{k+1} = \Pi_{U_{\mathrm{ad}}} (\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k)) \in U_{\mathrm{ad}}$$

$$(\Pi_{U_{\mathrm{ad}}}(\mathbf{u}))_m = \begin{cases} a & u_m \leq a \\ u_m & a < u_m < b \\ b & u_m \geq b \end{cases}$$

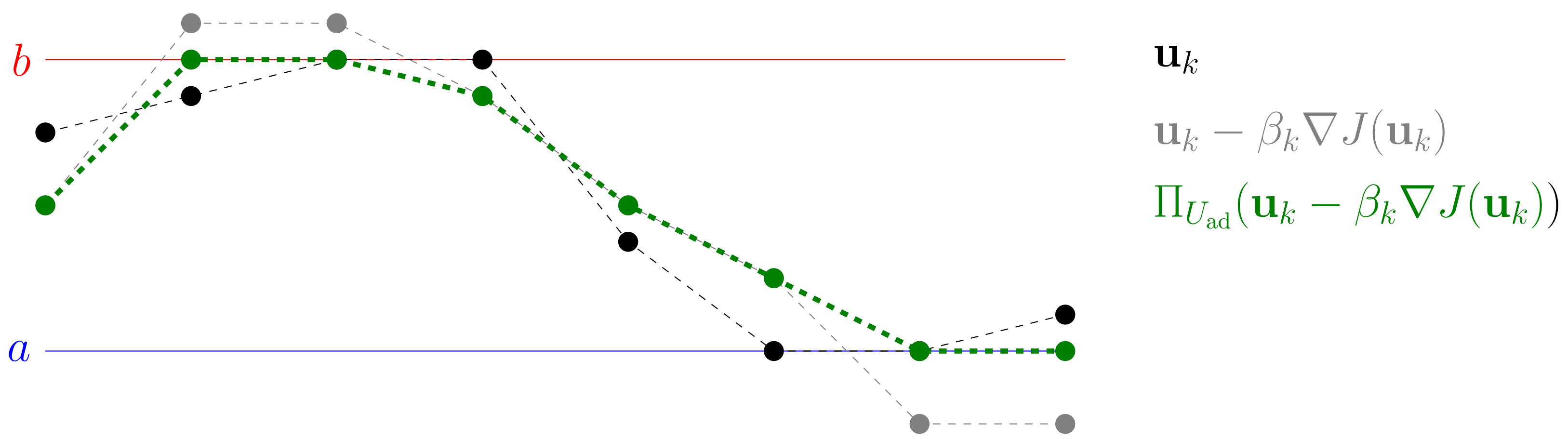


Projected gradient method (graphical illustration)

$$a \leq u_m \leq b, \qquad m \in \{1, 2, \ldots, M\}.$$

$$\mathbf{u}_{k+1} = \Pi_{U_{\mathrm{ad}}} \left(\mathbf{u}_k - \beta_k \nabla J(\mathbf{u}_k) \right) \in U_{\mathrm{ad}}$$

$$\left(\Pi_{U_{\mathrm{ad}}} (\mathbf{u}) \right)_m = \begin{cases} a & u_m \leq a \\ u_m & a < u_m < b \\ b & u_m \geq b \end{cases}$$



Quadratic approximation for the projected gradient

We replace $\nabla J(\mathbf{u}_k)$ by

$$\nabla\Pi J(\mathbf{u}_k) = -\lim_{h\downarrow 0} \frac{\Pi(\mathbf{u}_k - h\nabla J(\mathbf{u}_k)) - \mathbf{u}_k}{h}$$

$\nabla\Pi J(\mathbf{u}_k)$ is equal to $\nabla J(\mathbf{u}_k)$ except for entries where the $-\nabla J(\mathbf{u}_k)$ is pointing out of the admissible set.

Explicitly,

$$(\nabla\Pi J(\mathbf{u}_k))_m = \begin{cases} 0 & (\mathbf{u}_k)_m = a \text{ and } (\nabla J(\mathbf{u}_k))_m \geq 0 \\ & \quad\text{or } (\mathbf{u}_k)_m = b \text{ and } (\nabla J(\mathbf{u}_k))_m \leq 0 \\ (\nabla J(\mathbf{u}_k))_m & \text{otherwise.} \end{cases}$$

Quadratic approximation for the projected gradient

We replace $\nabla J(\mathbf{u}_k)$ by

$$\nabla \Pi J(\mathbf{u}_k) = -\lim_{h \downarrow 0} \frac{\Pi(\mathbf{u}_k - h\nabla J(\mathbf{u}_k)) - \mathbf{u}_k}{h}$$

$\nabla \Pi J(\mathbf{u}_k)$ is equal to $\nabla J(\mathbf{u}_k)$ except for entries where the $-\nabla J(\mathbf{u}_k)$ is pointing out of the admissible set.

Explicitly,

$$(\nabla \Pi J(\mathbf{u}_k))_m = \begin{cases} 0 & (\mathbf{u}_k)_m = a \text{ and } (\nabla J(\mathbf{u}_k))_m \geq 0 \\ & \quad \text{or } (\mathbf{u}_k)_m = b \text{ and } (\nabla J(\mathbf{u}_k))_m \leq 0 \\ (\nabla J(\mathbf{u}_k))_m & \text{otherwise.} \end{cases}$$

We then can use the quadratic approximation:

$$J(\mathbf{u}_{k+1}) \approx J(\mathbf{u}_k) - \beta_k G + \frac{H}{2}\beta_k^2 + O(\beta_k^3)$$

with

$$G = \langle \nabla J(\mathbf{u}_k), \nabla \Pi J(\mathbf{u}_k) \rangle$$
$$H = \left[\frac{\mathrm{d}^2}{\mathrm{d}\theta^2} J(\mathbf{u_k} + \theta \nabla \Pi J(\mathbf{u}_k)) \right]_{\theta=0} .$$

Computation of H with projected gradient (example)

Consider the optimization problem

$$\min_{u \in U_{\text{ad}}} \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} + \frac{1}{2}\mathbf{u}^\top \mathbf{R}\mathbf{u}$$

with $\mathbf{Q} = \mathbf{Q}^\top$, $\mathbf{R} = \mathbf{R}^\top$, $\mathbf{u} \in U_{\text{ad}} \subset \mathbb{R}^M$, and $\mathbf{x} \in \mathbb{R}^N$ subject to

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{0}.$$

We have the 'projected gradient' (which is a bad name) $\nabla \Pi J(\mathbf{u}_k)$.

Compute the state resulting from the projected gradient

$$\mathbf{x}_k^{\nabla \Pi} = -\mathbf{A}^{-1}\left(\mathbf{B}\nabla \Pi J(\mathbf{u}_k)\right).$$

We can then compute

$$H = \left(\mathbf{x}_k^{\nabla \Pi}\right)^\top \mathbf{Q}\mathbf{x}_k^{\nabla \Pi} + \left(\nabla \Pi J(\mathbf{u}_k)\right)^\top \mathbf{R}\nabla \Pi J(\mathbf{u}_k).$$

State constraints

For state constraints (i.e. constraints on $\mathbf{x}(\mathbf{u})$), it is not so straightforward to determine the projection on the admissible set.

State constraints can for example be included using a penalty function method, but we will not discuss this further in this course.

5.F Convergence analysis for gradient descent

Main result

We return to the more abstract optimization problem:

$$\min_{u \in \mathbb{R}^M} J(u).$$

Denote the minimizer by u^*.

For simplicity, we consider a gradient descent algorithm with a fixed step size β

$$u_{k+1} = u_k - \beta \nabla J(u_k).$$

Main result

We return to the more abstract optimization problem:

$$\min_{u \in \mathbb{R}^M} J(u).$$

Denote the minimizer by u^*.

For simplicity, we consider a gradient descent algorithm with a fixed step size β

$$u_{k+1} = u_k - \beta \nabla J(u_k).$$

Two assumptions:

- The functional J is α-convex, i.e.

$$J(\theta u + (1-\theta)v) \leq \theta J(u) + (1-\theta)J(v) - \frac{\alpha\theta(1-\theta)}{2}|u-v|^2, \qquad \theta \in [0,1].$$

- The gradient $\nabla J(u)$ is Lipschitz, i.e. there is an $L > 0$ such that for all u and v

$$|\nabla J(u) - \nabla J(v)| \leq L|u-v|.$$

Theorem

$$|u_k - u^*|^2 \leq \left(1 - 2\alpha\beta + \beta^2 L^2\right)^k |u_0 - u^*|^2$$

Observation 1

The functional J is α-convex:

$$J(\theta u + (1-\theta)v) \leq \theta J(u) + (1-\theta)J(v) - \frac{\alpha\theta(1-\theta)}{2}|u-v|^2.$$

Subtract expand the brackets on the LHS and subtract $J(v)$ on both sides:

$$J(v + \theta(u-v)) - J(v) \leq \theta J(u) - \theta J(v) - \frac{\alpha\theta(1-\theta)}{2}|u-v|^2.$$

Divide by θ and take the limit $\theta \to 0$:

$$\langle \nabla J(v), u-v \rangle = \lim_{\theta\to 0} \frac{J(v+\theta(u-v)) - J(v)}{\theta} \leq J(u) - J(v) - \frac{\alpha}{2}|u-v|^2.$$

We conclude

$$\langle \nabla J(v), u-v \rangle \leq J(u) - J(v) - \frac{\alpha}{2}|u-v|^2.$$

Observation 2

From the previous slide:

$$\langle \nabla J(v), u - v \rangle \leq J(u) - J(v) - \frac{\alpha}{2}|u - v|^2.$$

Because this holds for all u and v, we may interchange u and v to obtain:

$$\langle \nabla J(u), v - u \rangle \leq J(v) - J(u) - \frac{\alpha}{2}|v - u|^2.$$

Adding these two equations, we find

$$\langle \nabla J(v) - \nabla J(u), u - v \rangle \leq -\alpha|u - v|^2.$$

Proof

Theorem

$$|u_k - u^*|^2 \leq \left(1 - 2\alpha\beta + \beta^2 L^2\right)^k |u_0 - u^*|^2$$

$$\begin{aligned}
|u_{k+1} - u^*|^2 &= \langle u_{k+1} - u^*, u_{k+1} - u^* \rangle \\
&= \langle u_k - \beta \nabla J(u_k) - u^*, u_k - \beta \nabla J(u_k) - u^* \rangle \\
&= \langle u_k - u^*, u_k - u^* \rangle - 2\beta \langle \nabla J(u_k), u_k - u^* \rangle + \beta^2 \langle \nabla J(u_k), \nabla J(u_k) \rangle
\end{aligned}$$

Proof

Theorem

$$|u_k - u^*|^2 \leq \left(1 - 2\alpha\beta + \beta^2 L^2\right)^k |u_0 - u^*|^2$$

$$\begin{aligned}
|u_{k+1} - u^*|^2 &= \langle u_{k+1} - u^*, u_{k+1} - u^* \rangle \\
&= \langle u_k - \beta \nabla J(u_k) - u^*, u_k - \beta \nabla J(u_k) - u^* \rangle \\
&= \langle u_k - u^*, u_k - u^* \rangle - 2\beta \langle \nabla J(u_k), u_k - u^* \rangle + \beta^2 \langle \nabla J(u_k), \nabla J(u_k) \rangle
\end{aligned}$$

Using that $\nabla J(u^*) = 0$ and Observation 2, we find

$$-\langle \nabla J(u_k), u_k - u^* \rangle = -\langle \nabla J(u_k) - \nabla J(u^*), u_k - u^* \rangle \leq -\alpha |u_k - u^*|^2.$$

Again using that $\nabla J(u^*) = 0$ and the Lipschitz continuity of $\nabla J(u)$, we also have that

$$\langle \nabla J(u_k), \nabla J(u_k) \rangle = |\nabla J(u_k) - \nabla J(u^*)|^2 \leq L^2 |u_k - u^*|^2.$$

Proof

Theorem

$$|u_k - u^*|^2 \leq \left(1 - 2\alpha\beta + \beta^2 L^2\right)^k |u_0 - u^*|^2$$

$$\begin{aligned}
|u_{k+1} - u^*|^2 &= \langle u_{k+1} - u^*, u_{k+1} - u^* \rangle \\
&= \langle u_k - \beta \nabla J(u_k) - u^*, u_k - \beta \nabla J(u_k) - u^* \rangle \\
&= \langle u_k - u^*, u_k - u^* \rangle - 2\beta \langle \nabla J(u_k), u_k - u^* \rangle + \beta^2 \langle \nabla J(u_k), \nabla J(u_k) \rangle
\end{aligned}$$

Using that $\nabla J(u^*) = 0$ and Observation 2, we find

$$-\langle \nabla J(u_k), u_k - u^* \rangle = -\langle \nabla J(u_k) - \nabla J(u^*), u_k - u^* \rangle \leq -\alpha |u_k - u^*|^2.$$

Again using that $\nabla J(u^*) = 0$ and the Lipschitz continuity of $\nabla J(u)$, we also have that

$$\langle \nabla J(u_k), \nabla J(u_k) \rangle = |\nabla J(u_k) - \nabla J(u^*)|^2 \leq L^2 |u_k - u^*|^2.$$

Inserting these two results back into the original expression, we conclude

$$|u_{k+1} - u^*|^2 \leq \left(1 - 2\alpha\beta + \beta^2 L^2\right) |u_k - u^*|^2$$

The result now follows by induction over k.

Other algorithms

There are many more gradient-based algorithms.
Gradient-descent/steepest descent is the simplest one.
For quadratic problems, the Conjugate Gradient (CG) method is the best method.
When optimizing $u \in \mathbb{R}^M$, it converges in at most M iterations to the minimizer.
For nonquadratic problems, other algorithms can be more effective.

see e.g. Ascher, The chaotic nature of faster gradient descent methods