LP, NLP and line search

Onsite workshop · Monday, November 9, 2026

90 min review · 10 min break · 80 min new material

Calculate first; use the demo to test your prediction.

Today’s schedule

0–55 min: LP, duality and shadow prices

55–90 min: NLP, gradients and curvature

100–180 min: line search, demos and practice

Opening questions

Do any resource prices give an upper bound on profit?

Does $\nabla f = 0$ prove that a point is a minimum?

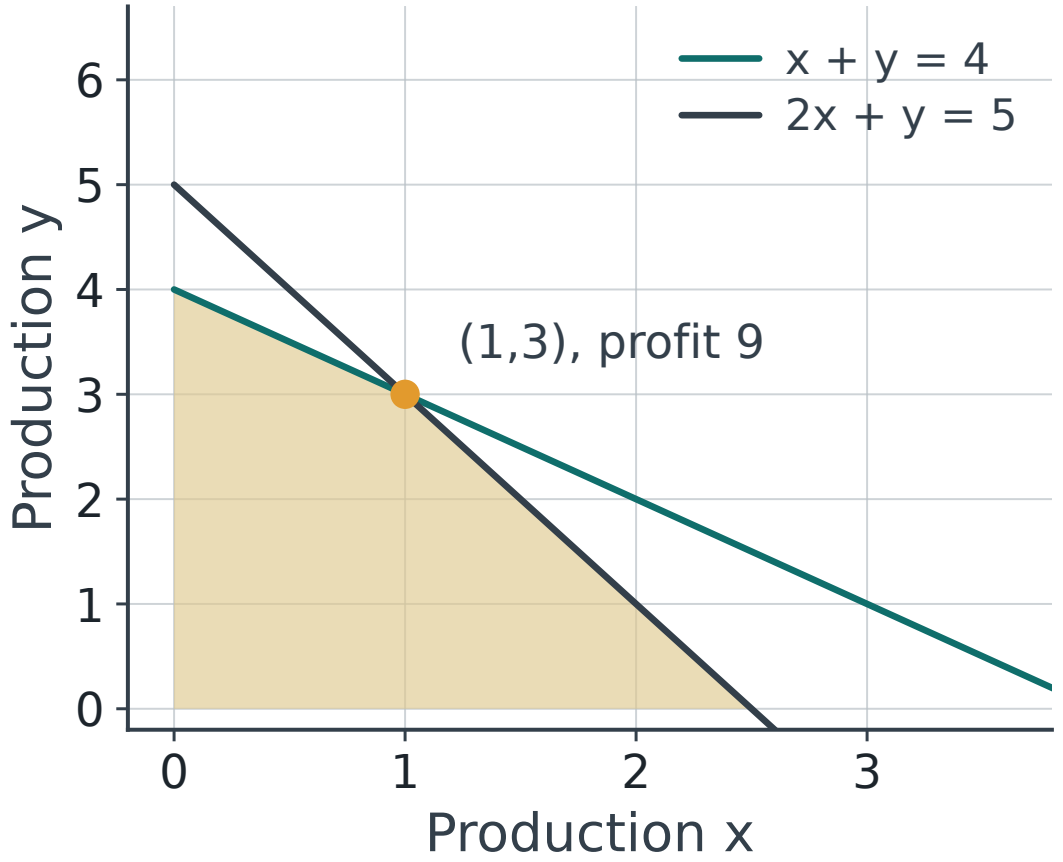
Does a descent direction work with every positive step size?

What the primal variables mean

max z = 3x + 2y; x, y ≥ 0

Resource 1: x + y ≤ 4

Resource 2: 2x + y ≤ 5

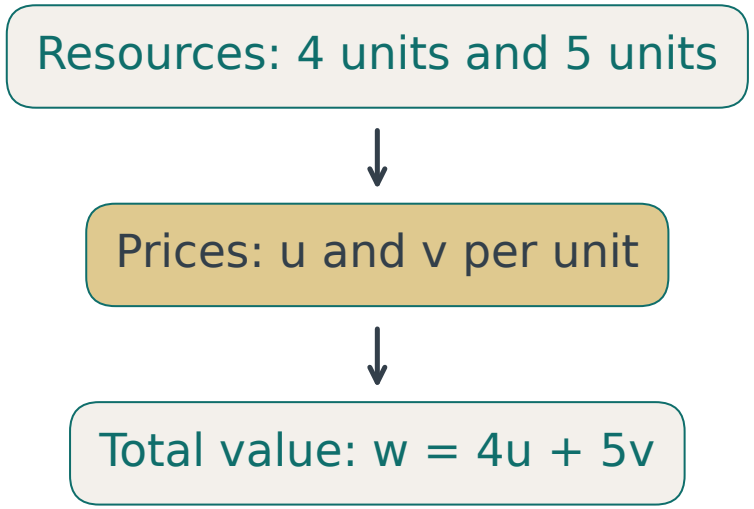


What the dual variables mean

u = price per unit of resource 1

v = price per unit of resource 2

w = 4u + 5v: value of the available resources

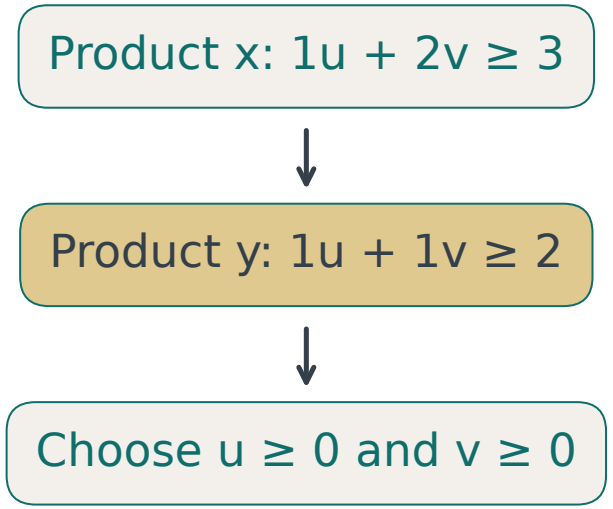


Derive one dual constraint per product

Product x uses (1, 2): cost = u + 2v

Cover its profit: u + 2v ≥ 3

Product y uses (1, 1): u + v ≥ 2



Why the dual gives an upper bound

min w = 4u + 5v; u, v ≥ 0

u + 2v ≥ 3; u + v ≥ 2

For any feasible pair: z ≤ w

$$3x + 2y \le (u + 2v)x + (u + v)y$$

↓

$$= u(x + y) + v(2x + y)$$

↓

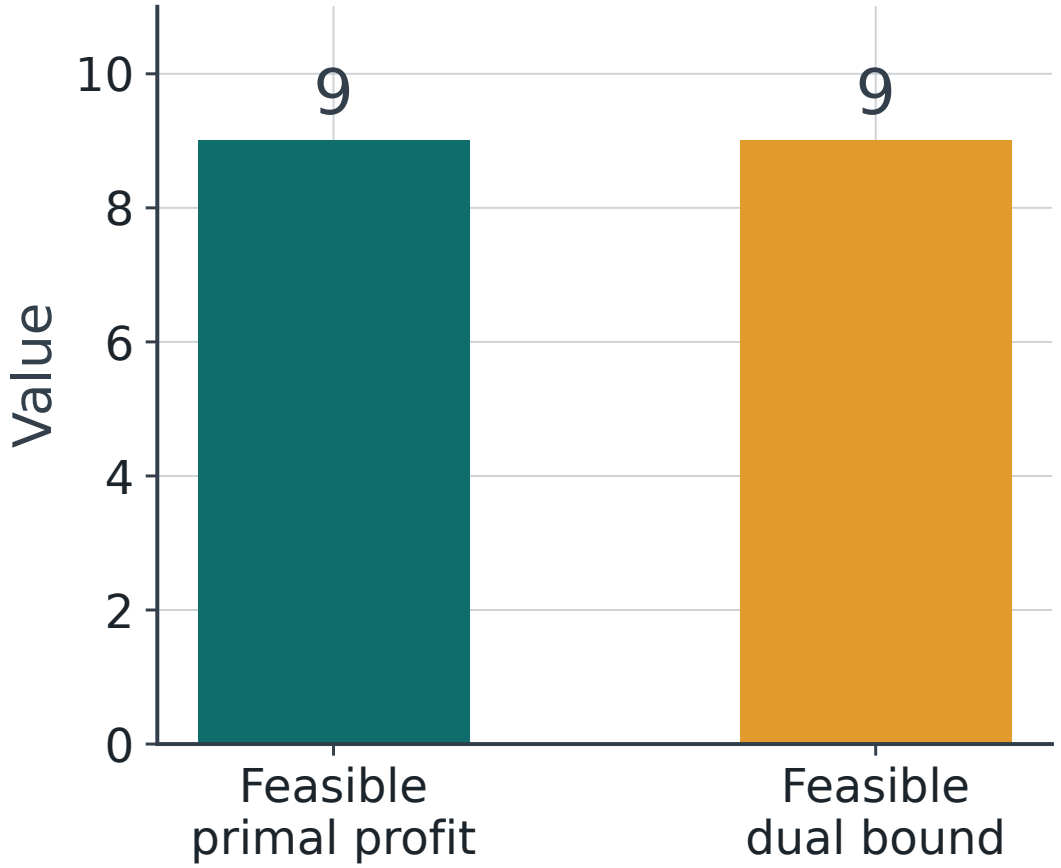
$$\le 4u + 5v$$

An optimality certificate

Primal (x, y) = (1, 3): z = 9

Dual (u, v) = (1, 1): w = 9

Both are feasible; equal values prove optimality.



Complementary slackness

Resource slack: $s = b - Ax \geq 0$

Product surplus: $t = A^{\mathsf{T}}\lambda - c \geq 0$

At optimal feasible pairs: $\lambda_i s_i = 0$ and $x_j t_j = 0$.

Pair	Zero-product rule
Resource i	price × unused amount = 0
Product j	quantity × surplus = 0

Check both zero products

At (x,y)=(1,3), resource slacks are (0,0).

At (u,v)=(1,1), dual surpluses are (0,0).

All four products vanish; z = w = 9.

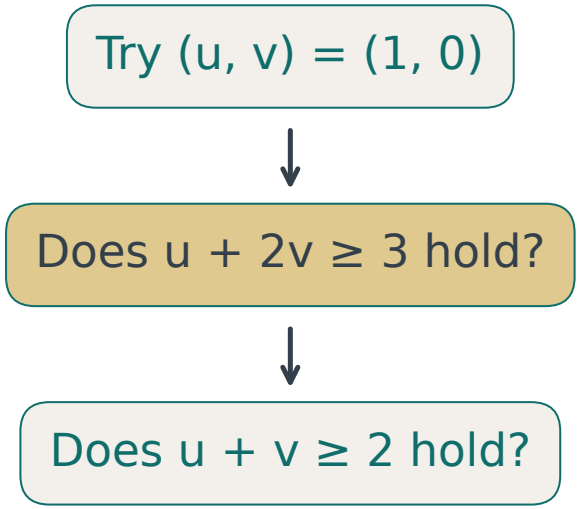
Resource	Slack	Price
1	0	1
2	0	1

Practice: test a proposed price

Proposed prices: (u, v) = (1, 0)

Check both dual constraints and compute w.

Can w bound every feasible primal profit?

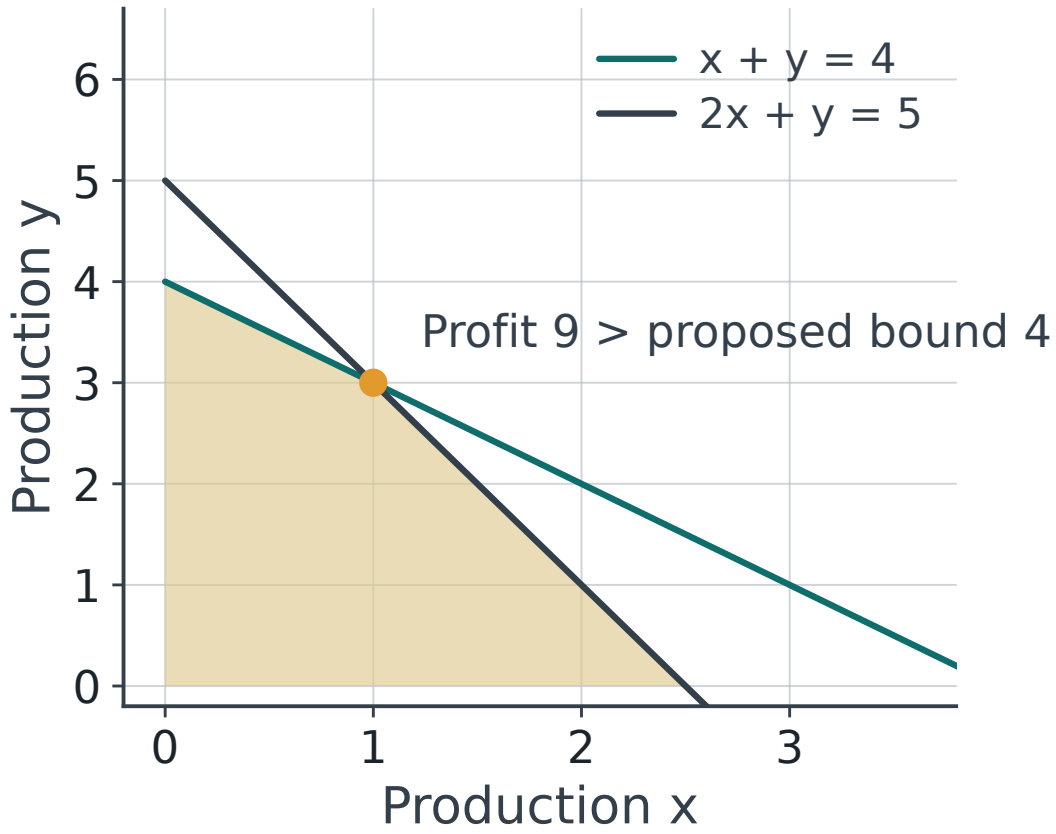


Solution: feasibility comes first

$1 + 2(0) = 1 < 3$; $1 + 0 = 1 < 2$

$w = 4$, but feasible $(1, 3)$ earns $z = 9$.

The proposed prices are not dual feasible.

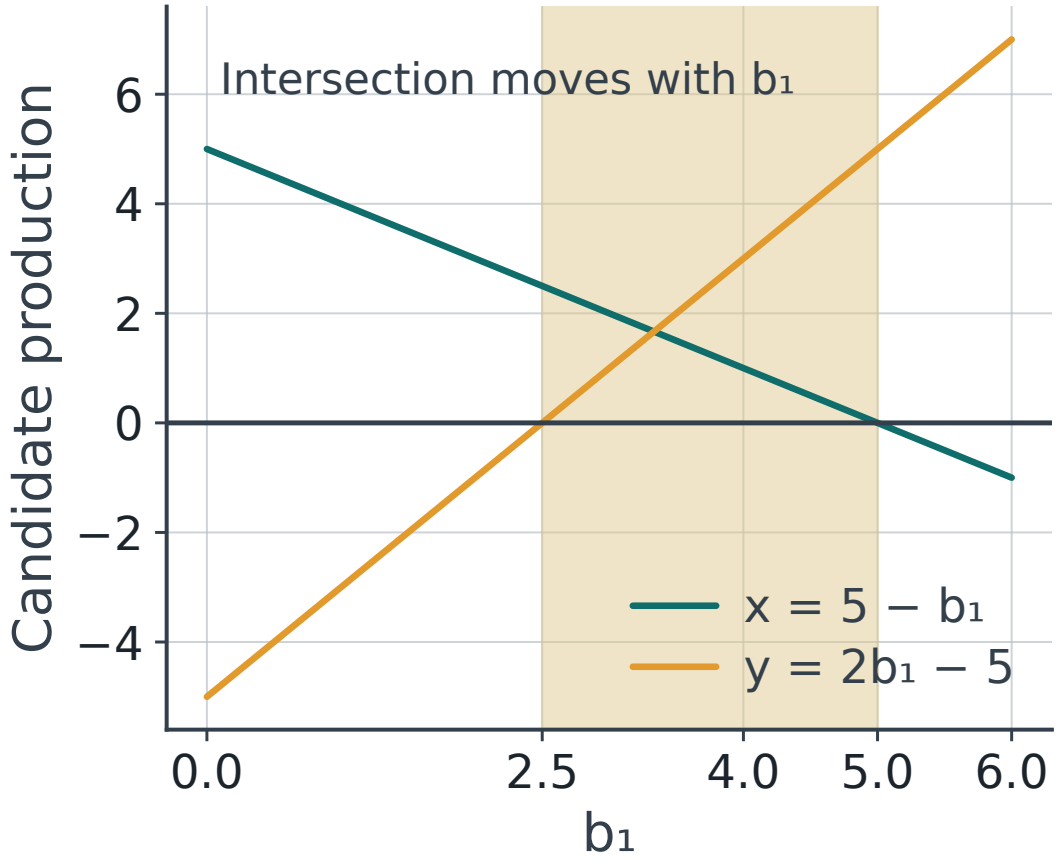


Derive the shadow-price interval

Keep $b_2 = 5$ and change b_1 only.

Solve $x + y = b_1$ and $2x + y = 5$.

$x = 5 - b_1$; $y = 2b_1 - 5$

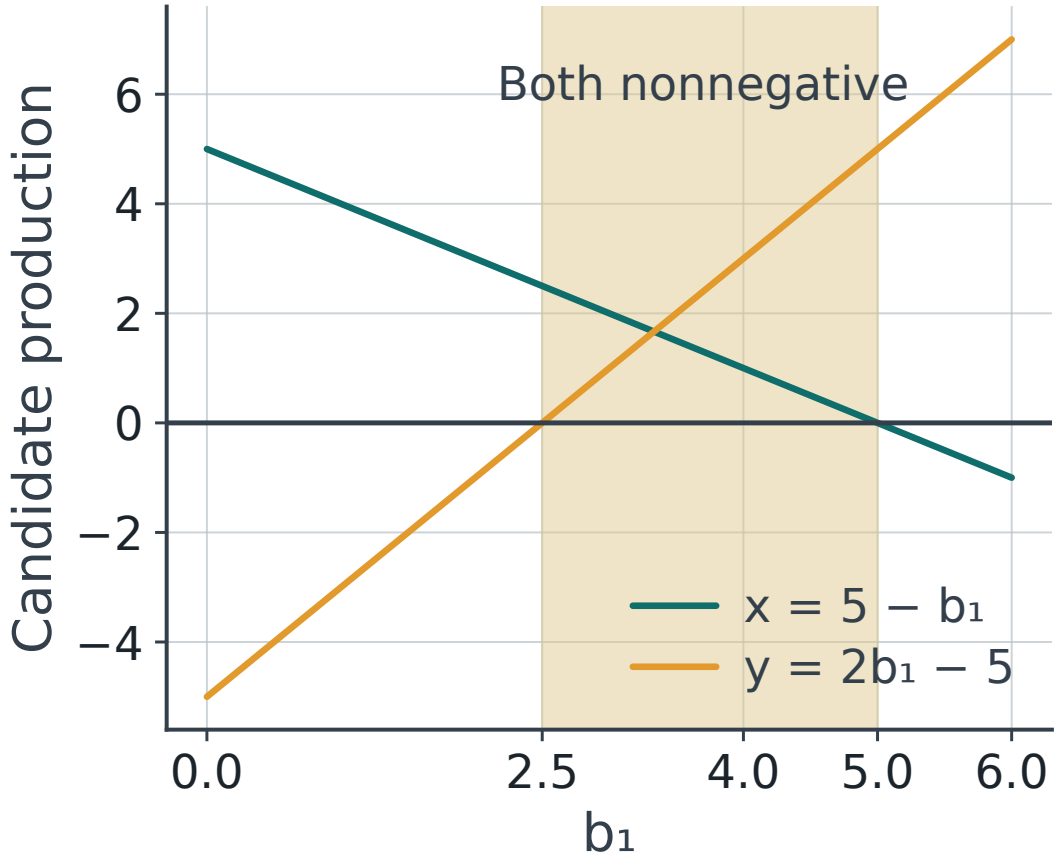


Feasibility determines the interval

$x \geq 0$ requires $b_1 \leq 5$; $y \geq 0$ requires $b_1 \geq 2.5$.

On $2.5 \leq b_1 \leq 5$: primal $z = b_1 + 5$ = dual w.

At the endpoints, use one-sided slopes.

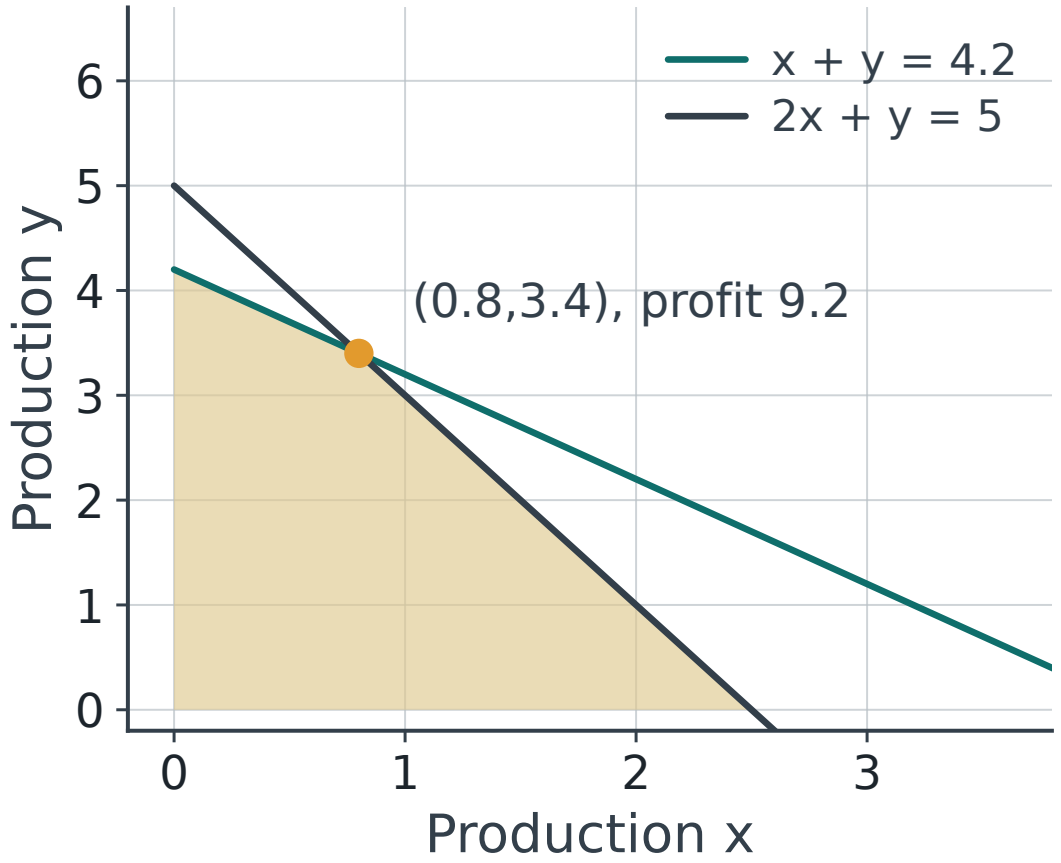


Inside the interval: $b_1 = 4.2$

$x = 0.8; y = 3.4; z = 9.2$

$\Delta b_1 = 0.2; \Delta z = 1 \times 0.2 = 0.2$

The shadow-price prediction is exact here.

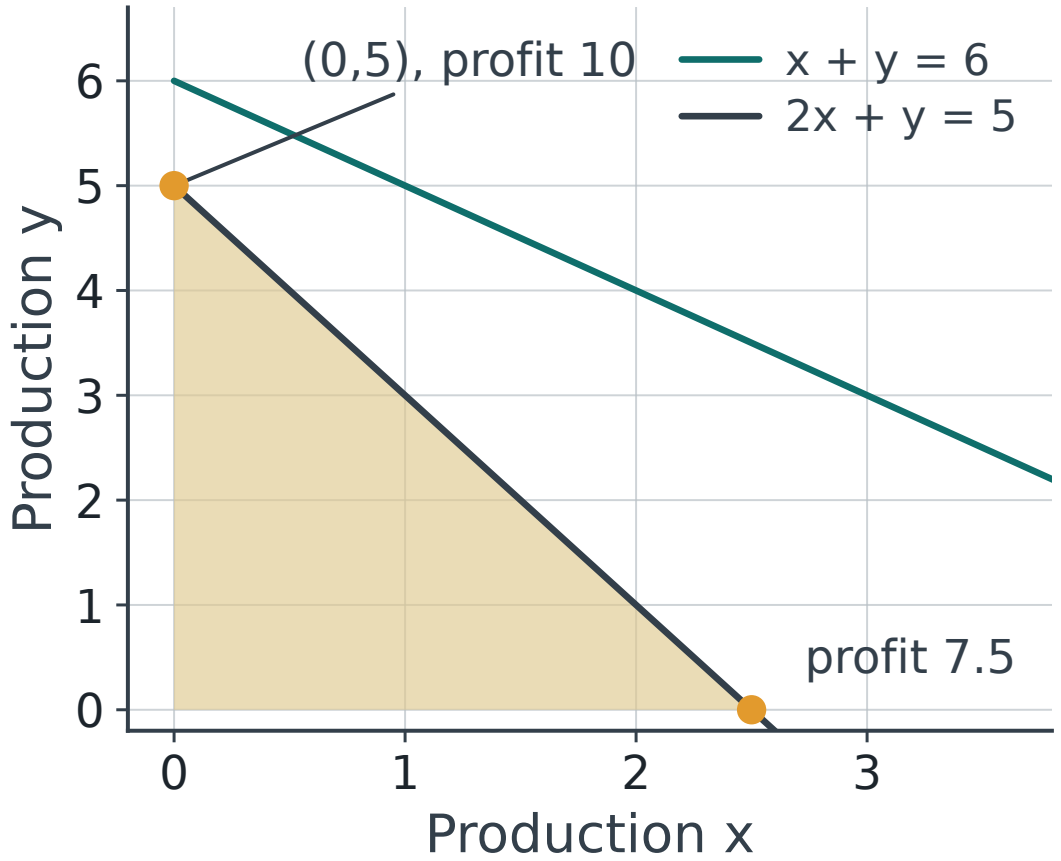


Outside the interval: $b_1 = 6$

Old intersection: $(-1, 7)$, infeasible.

Old prediction: $9 + 1(6 - 4) = 11$.

Reoptimize: $(x, y) = (0, 5)$, $z = 10$.

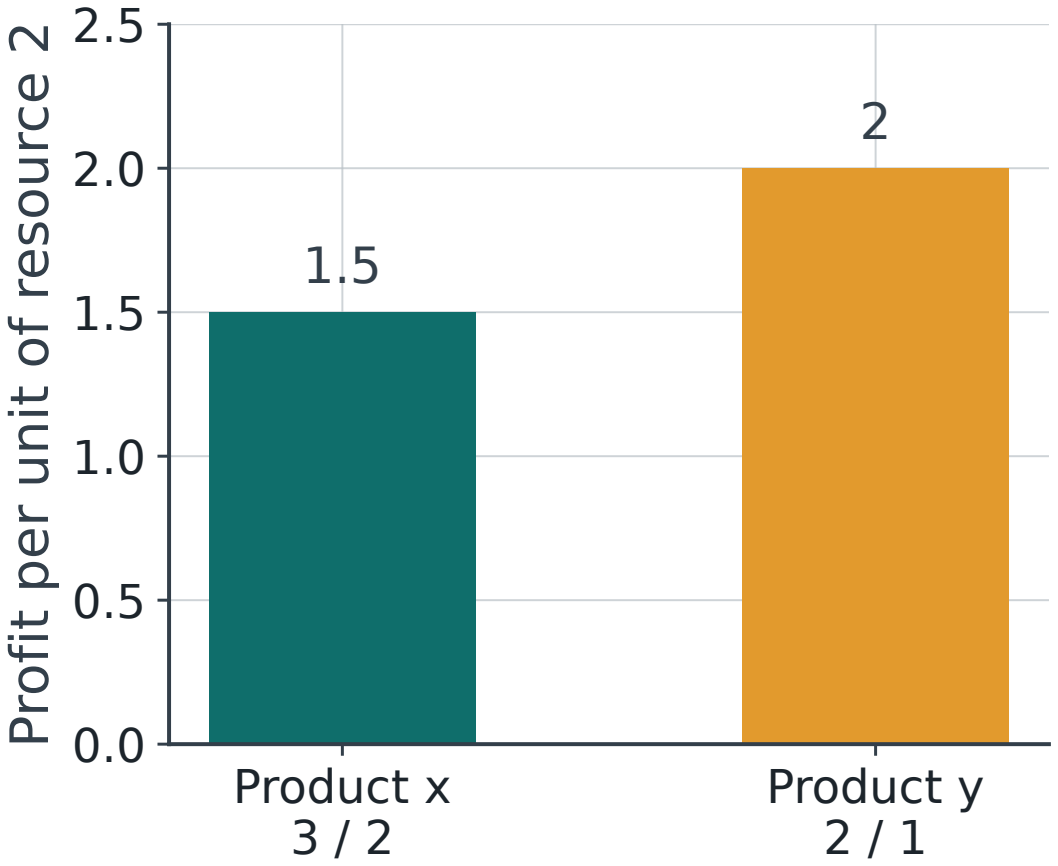


Why producing y gives the larger profit

For $b_1 = 6$: $y \le 5 - 2x$ for every feasible plan.

$z = 3x + 2y \le 10 - x \le 10$

Feasible (x,y) = (0,5) attains 10.

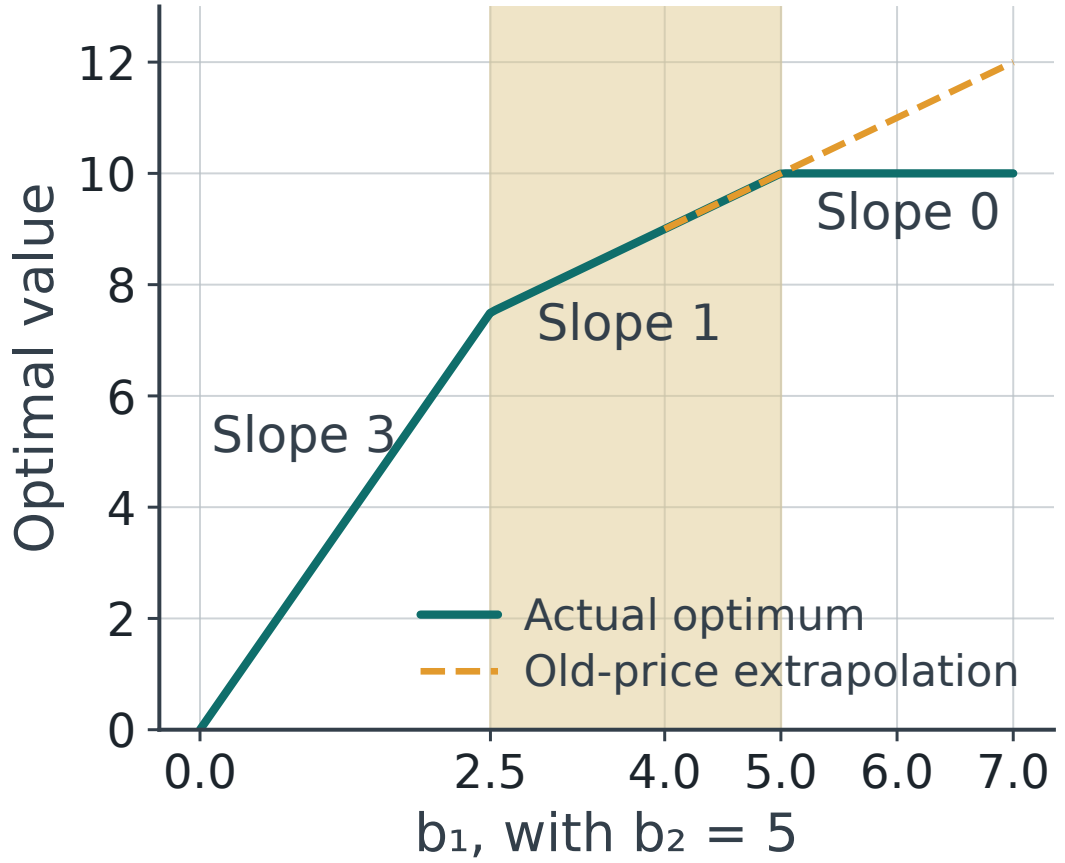


Shadow prices are slopes of the value function

For $b_1 \geq 0$: $z^* = \min(3b_1, b_1 + 5, 10)$

Slopes: 3, then 1, then 0

Use the slope only within its valid segment.

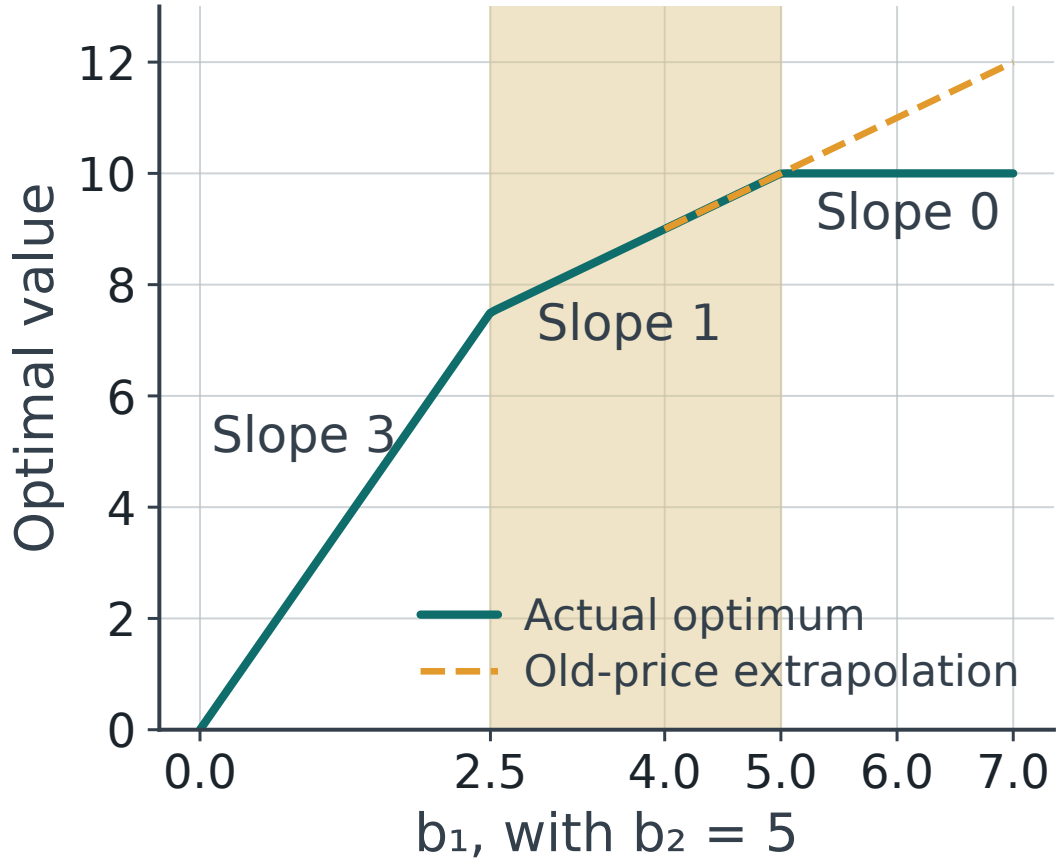


Demo 1: change resource capacity

Predict the optimum at $b_1 = 2$, 4.2 and 6.

Move the slider; compare actual profit with the old-price prediction.

Watch which resource has unused capacity.



Open demo.html → lp

LP checkpoint

At $b_1=6$, write the unused amount of each resource.

Find feasible dual prices whose value equals 10.

Does an active constraint always have a positive price?

At $b_1=6$	Answer
Primal quantities	(0,5)
Slacks / dual prices	(1,0) / (0,2)

From LP to NLP: how we check an answer

LP: dual-feasible prices with $w = z$ certify the optimum.

NLP: $\nabla f = 0$ finds candidates; curvature and convexity classify them.

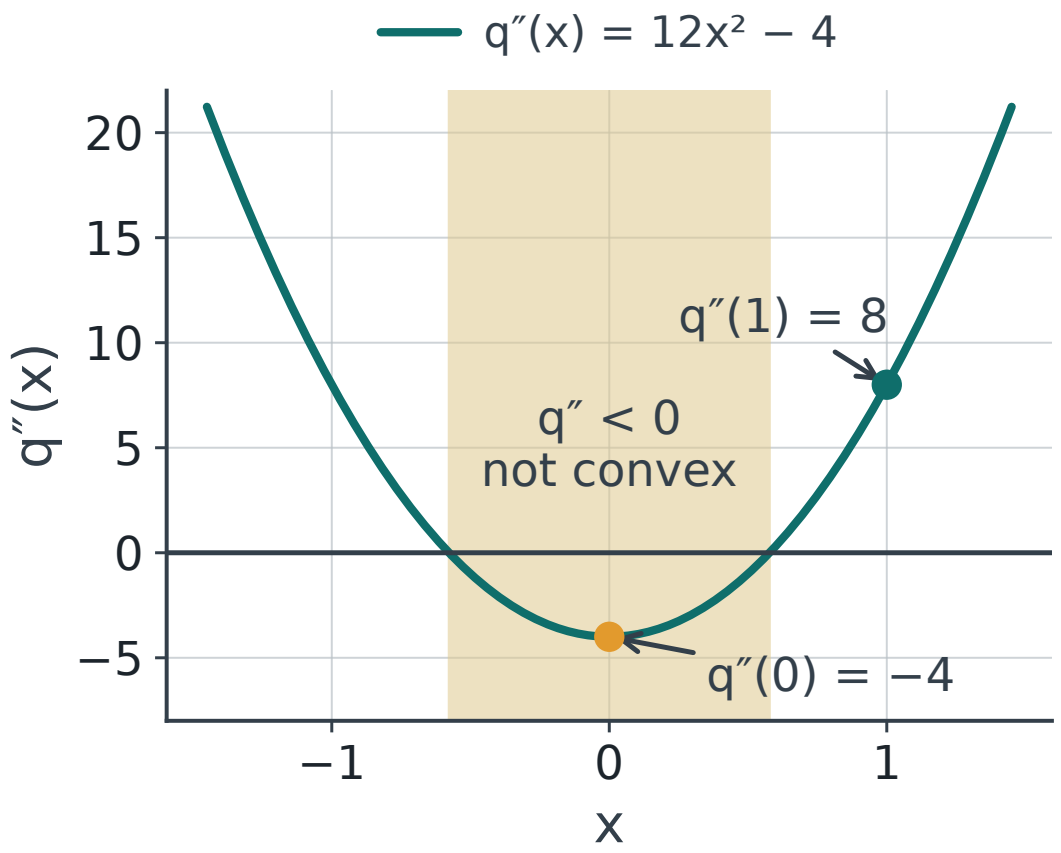
Line search: the Armijo test checks each trial step before we accept it.

Convexity requires a domain-wide check

Assume f is C^2 on an open convex domain.

Convex ⇔ Hessian is positive semidefinite everywhere.

One point is not enough: $q = (x^2-1)^2$ has $q''(1) = 8$ but $q''(0) = -4$.

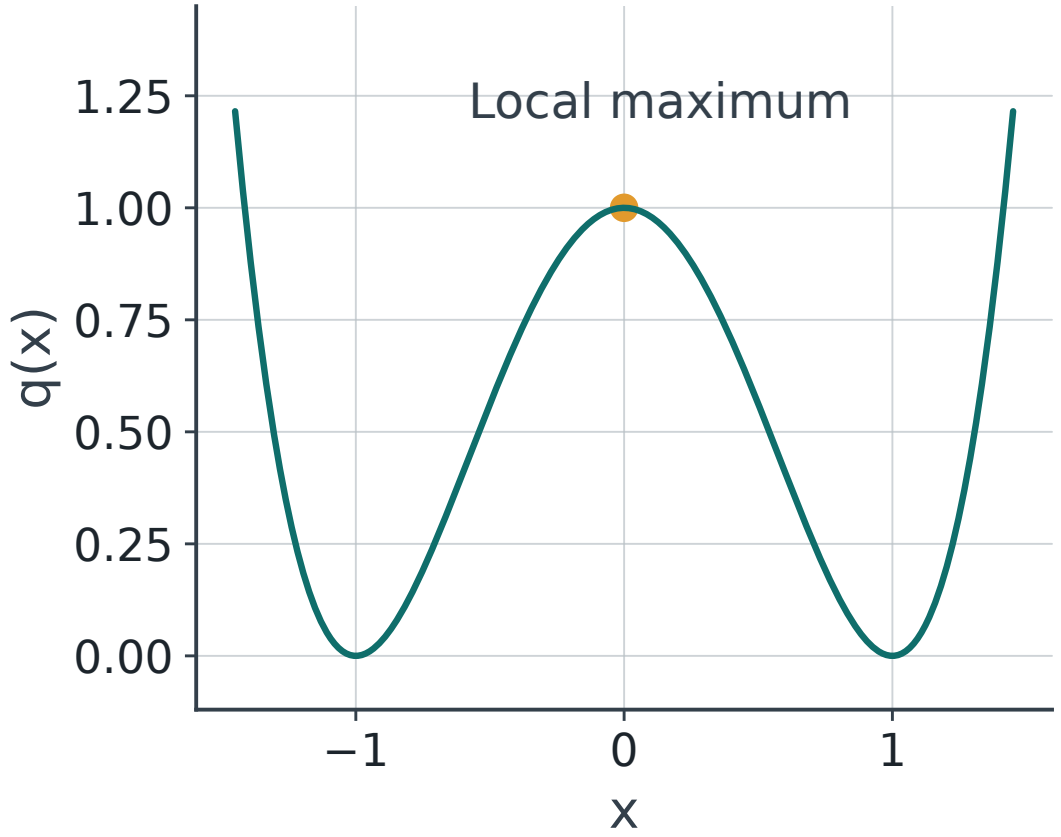


A stationary point may be a maximum

$q(x) = (x^2 - 1)^2$

$q'(x) = 4x(x^2 - 1)$; $q''(x) = 12x^2 - 4$

At $x = 0$: $q' = 0$, $q'' = -4 < 0$.

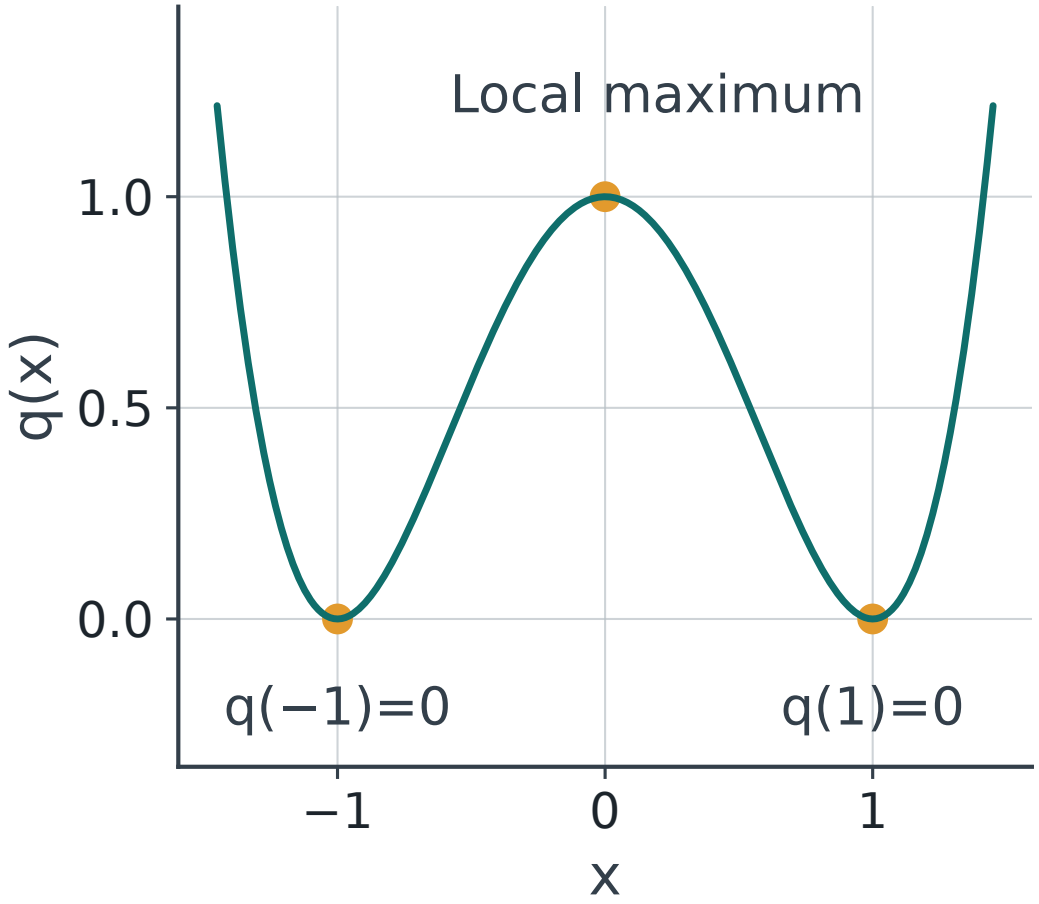


Check values and classify the point

$q(0) = 1$; $q(-1) = q(1) = 0$

$q(x) \geq 0$ for every real x.

Thus $x = \pm 1$ are global minima; $x = 0$ is a local maximum.

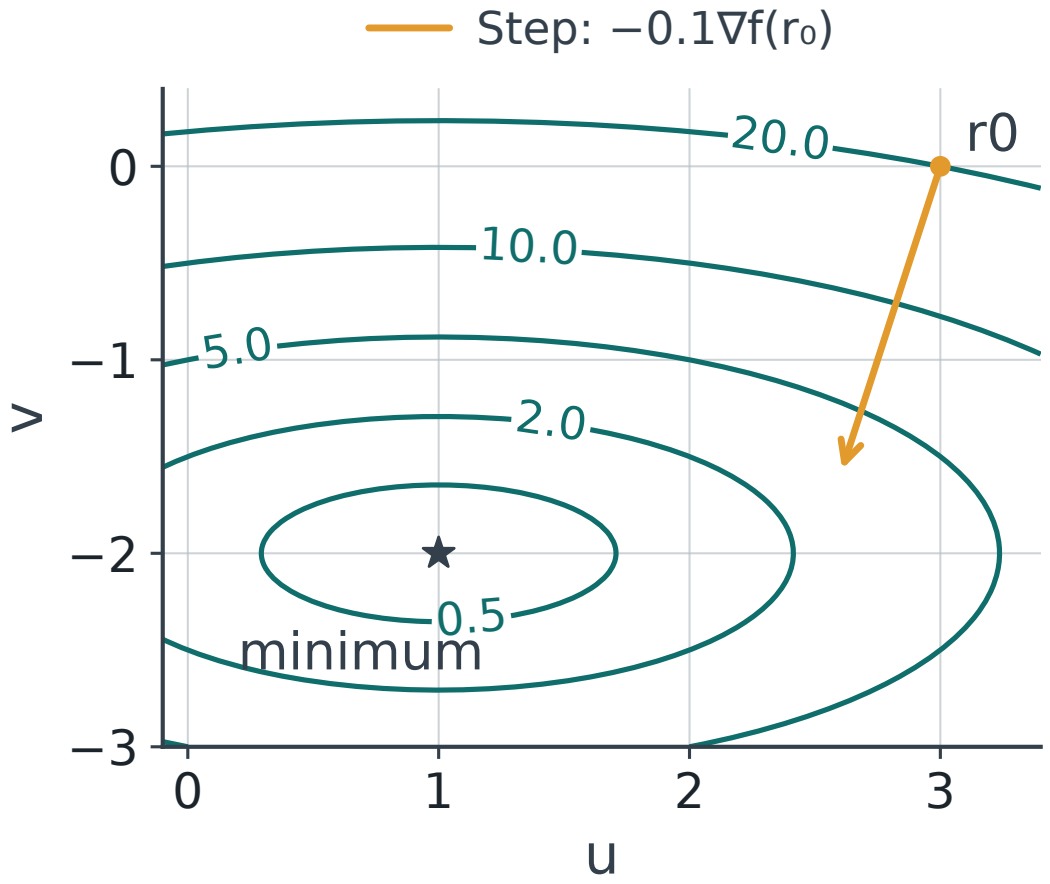


Gradient descent: compute the gradient

NLP coordinates u,v: $f = (u-1)^2 + 4(v+2)^2$

$\nabla f = (2(u-1), 8(v+2))$; Hessian diag(2, 8) is positive definite everywhere.

So f is convex and (1,−2) is its global minimum. At (3,0): $\nabla f = (4,16)$, $f = 20$.

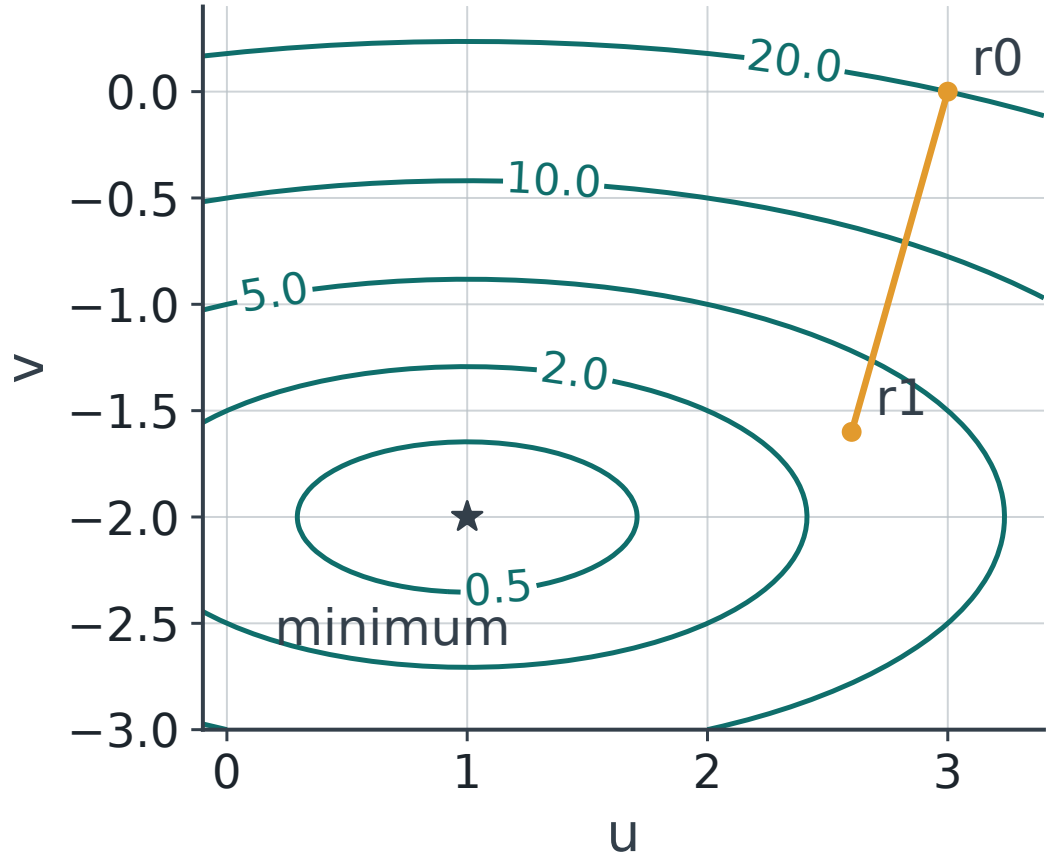


One gradient step is not the optimum

Let $r_k = (u_k, v_k)$. Use $\alpha = 0.1$: $r_1 = r_0 - \alpha \nabla f(r_0)$.

$(u_1, v_1) = (3,0) - 0.1(4,16) = (2.6, -1.6)$

$f(r_1) = 1.6^2 + 4(0.4)^2 = 3.2$

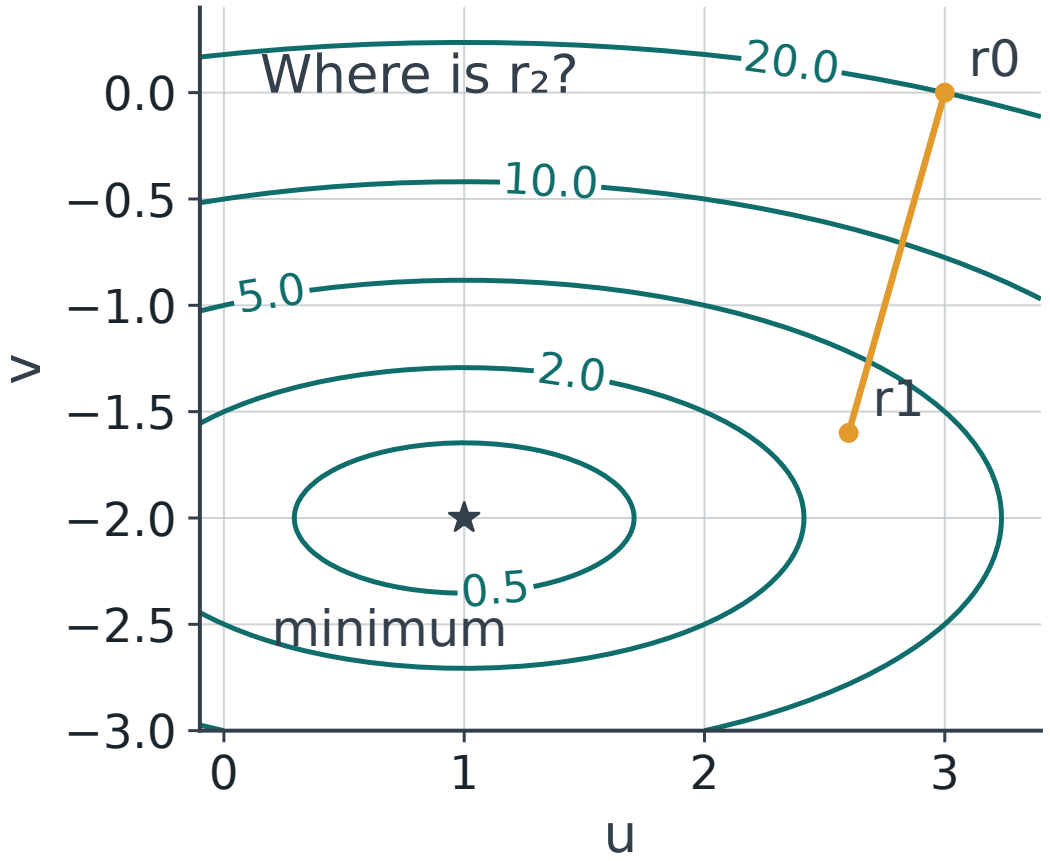


Practice: compute the second step

Start now at (2.6, −1.6); keep α = 0.1.

Compute the new gradient, point, and objective.

Does reusing the old gradient give the same method?

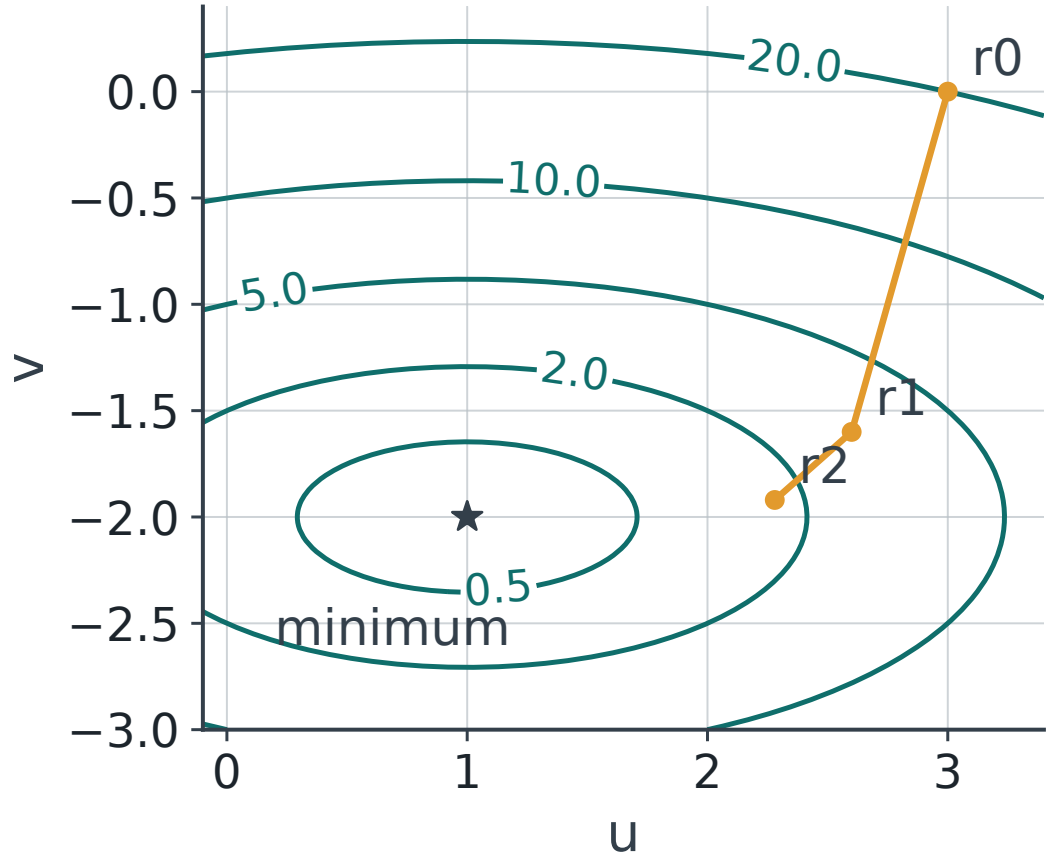


Solution: update the gradient every time

$\nabla f(2.6,-1.6) = (3.2,3.2)$

$r_2 = (2.6,-1.6) - 0.1(3.2,3.2) = (2.28,-1.92)$

$f(r_2) = 1.28^2 + 4(0.08)^2 = 1.664$

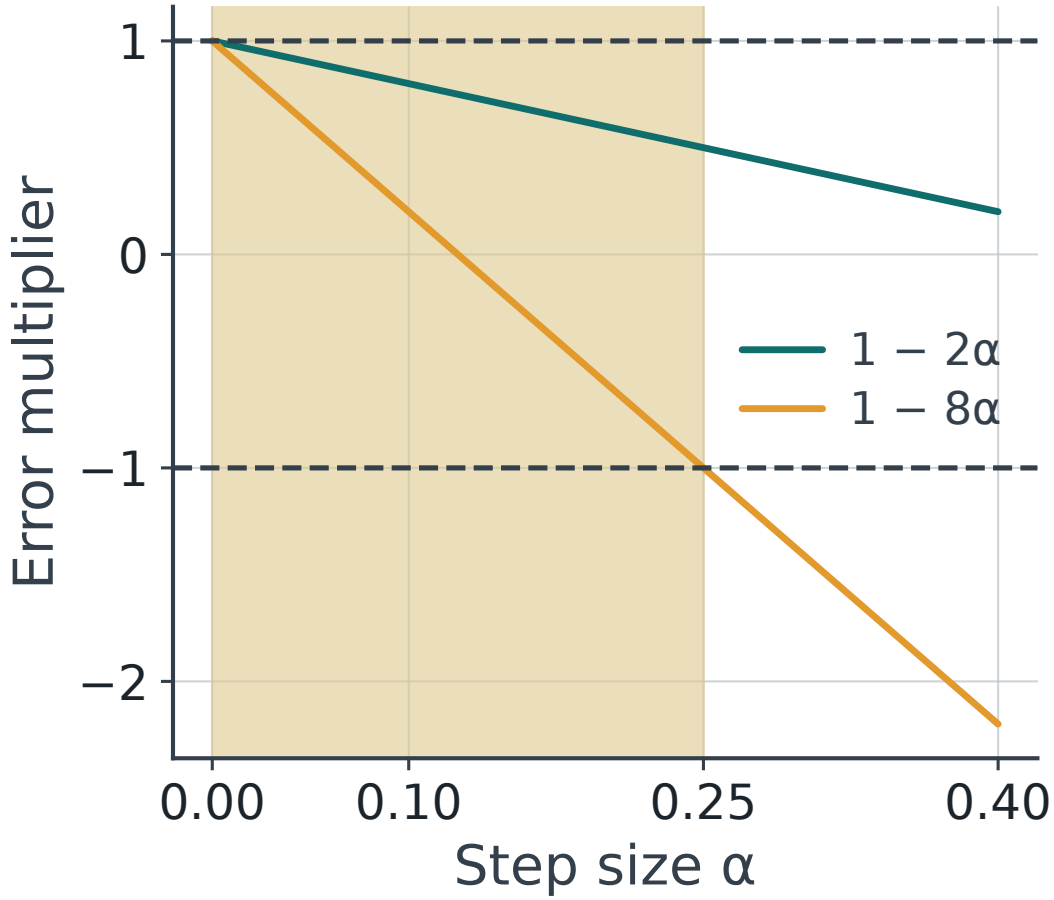


Derive the constant-step convergence range

Let $e_u = u-1$ and $e_v = v+2$.

$e_{u,\text{new}} = (1-2\alpha)e_u$; $e_{v,\text{new}} = (1-8\alpha)e_v$

Both shrink for every start when $0 < \alpha < 0.25$.

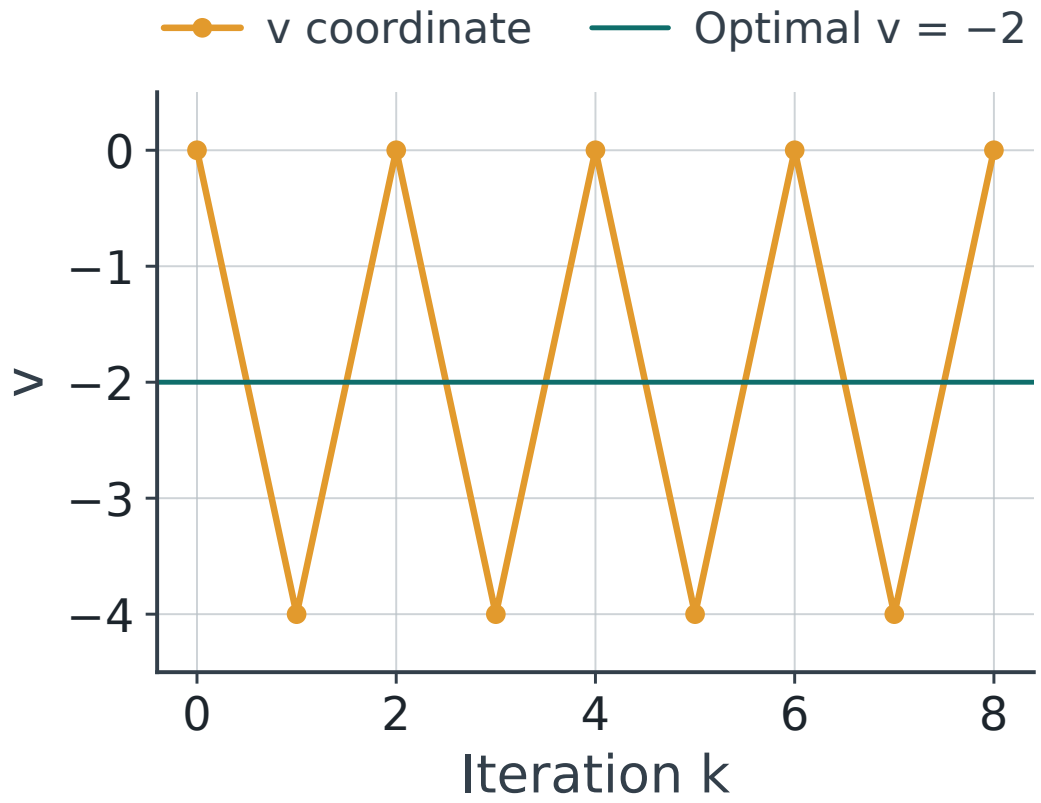


At α = 0.25, one coordinate oscillates

e_u multiplier = 0.5; e_v multiplier = −1.

From (3,0), v alternates: 0, −4, 0, −4, …

$f(r_k) = 4(1/4)^k + 16 \to 16$, not 0.

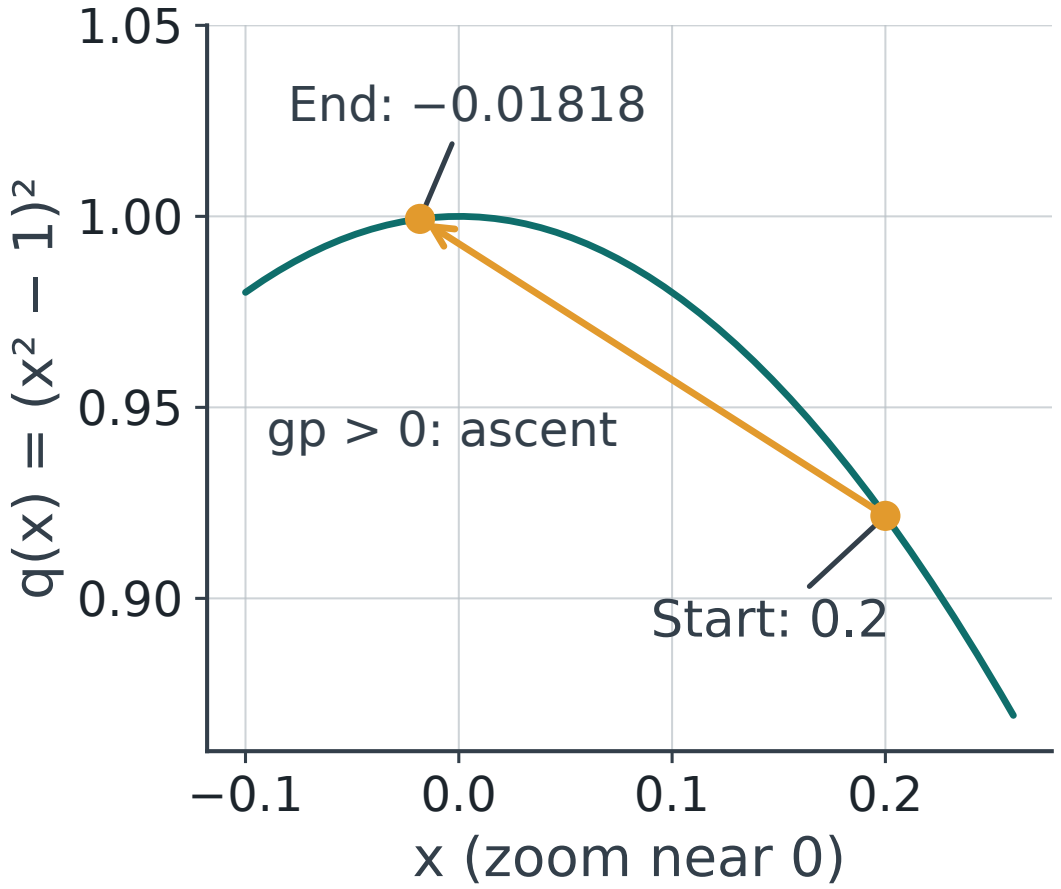


Newton directions also need a check

$p = -g/H$ in one dimension.

At $x = 0.2$ for q: $g = -0.768$, $H = -3.52$.

$p \approx -0.21818$; full step $x+p \approx -0.01818$.



NLP checkpoint

For $q(x)=(x^2-1)^2$, classify $x=0$ and $x=\pm 1$.

For the quadratic, explain why $\alpha=0.25$ can fail if kept fixed.

Before line search, what sign should $g^T p$ have?

Break · 10 minutes

Keep the quadratic and the starting point in your notes.

$f(u,v) = (u-1)^2 + 4(v+2)^2$; $r_0=(3,0)$

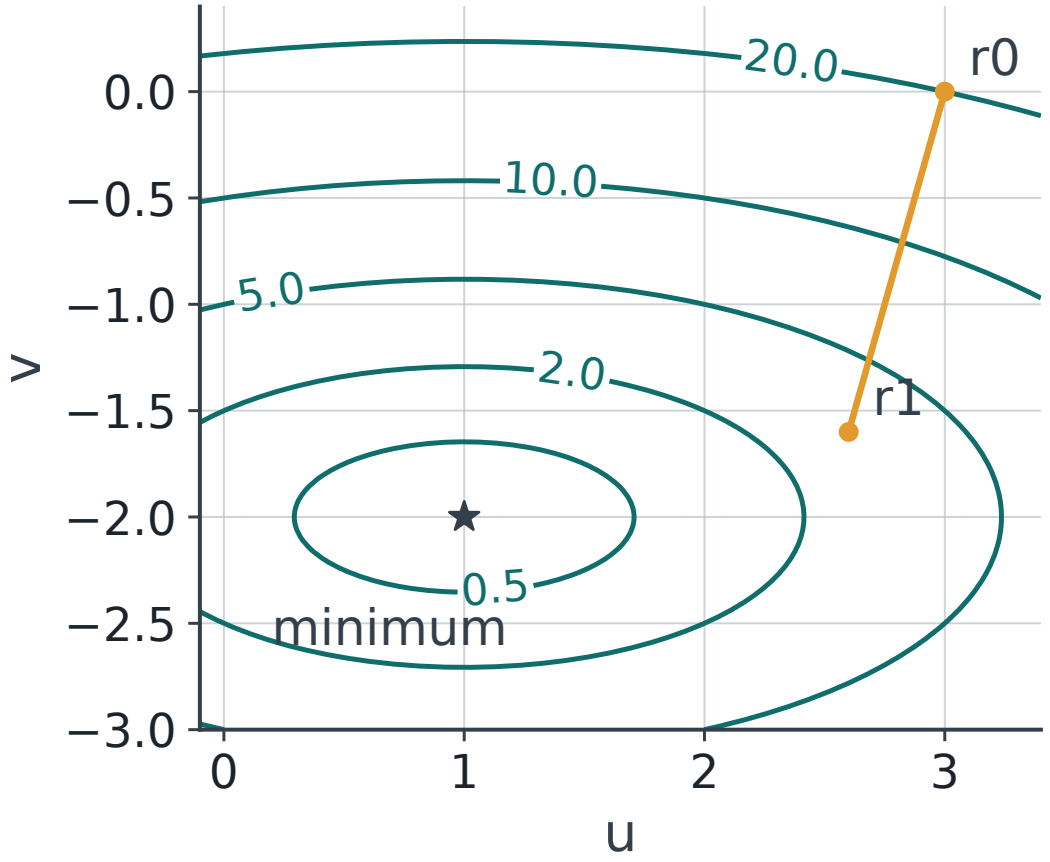
Next: choose a step size automatically.

Direction and step size

$r_{k+1} = r_k + \alpha_k p_k$, with $\alpha_k > 0$

α_k is the step size (textbooks: step length);
distance moved = $\alpha_k \|p_k\|$.

For gradient descent, $p_k = -\nabla f(r_k)$.

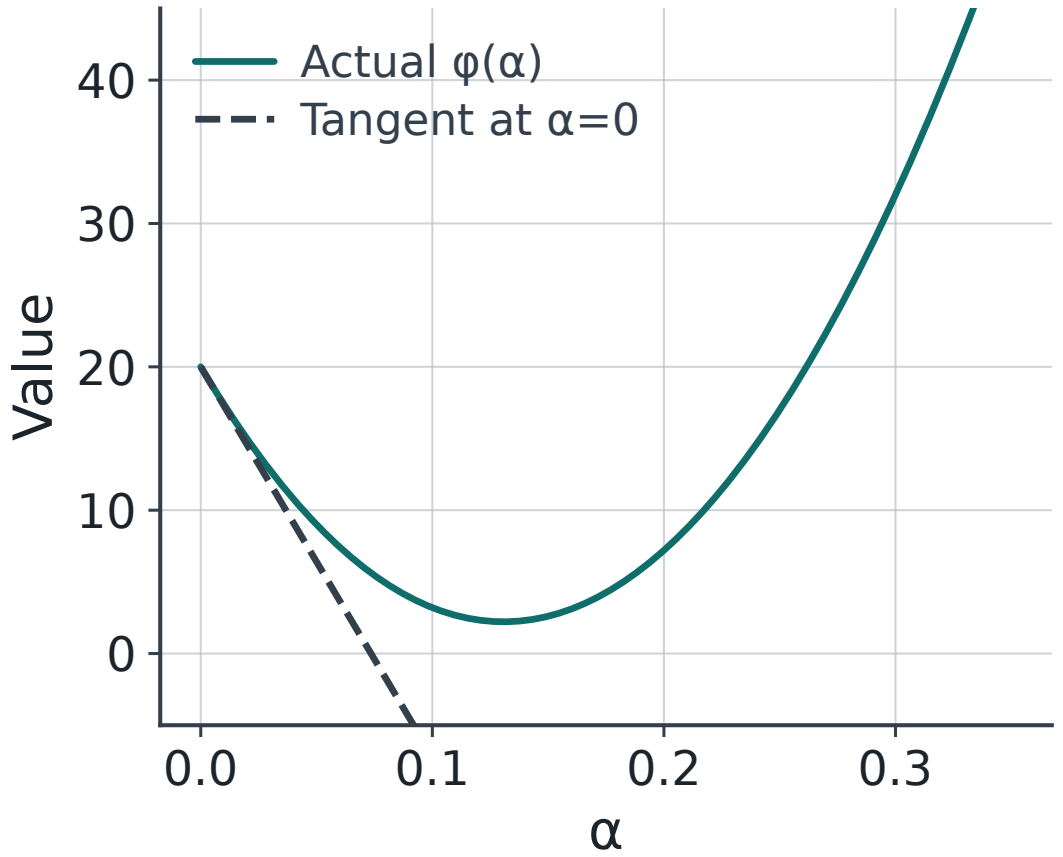


Check the direction first

With $g = \nabla f(r)$, require $g^T p < 0$ for descent.

For $p = -g$ and $g \neq 0$: $g^T p = -\|g\|^2 < 0$.

This guarantees decrease for sufficiently small positive α.

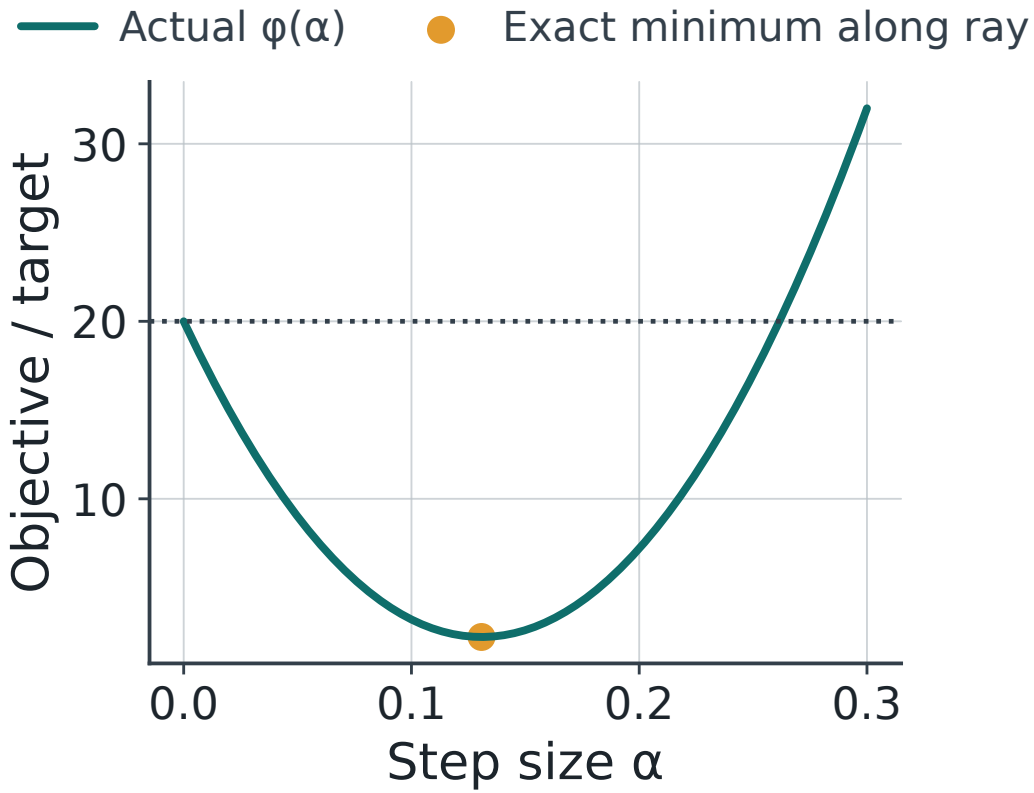


A one-dimensional function along the ray

At $r_0=(3,0)$: $g_0=(4,16)$, $p_0=(-4,-16)$.

$$\varphi(\alpha) = f(3-4\alpha, -16\alpha)$$

$$\varphi(\alpha) = 20 - 272\alpha + 1040\alpha^2$$

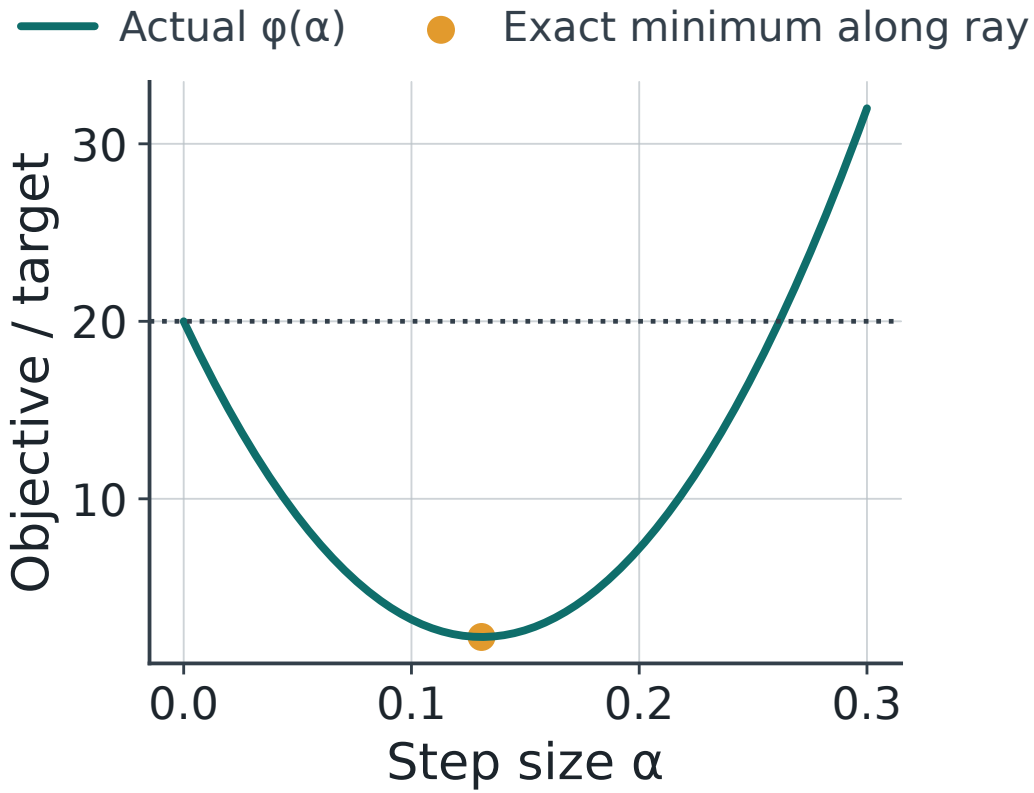


Exact line search for this quadratic

$\varphi'(\alpha) = -272 + 2080\alpha$

$\alpha^* = 272/2080 = 17/130 \approx 0.13077$

$\varphi(\alpha^*) = 144/65 \approx 2.21538$

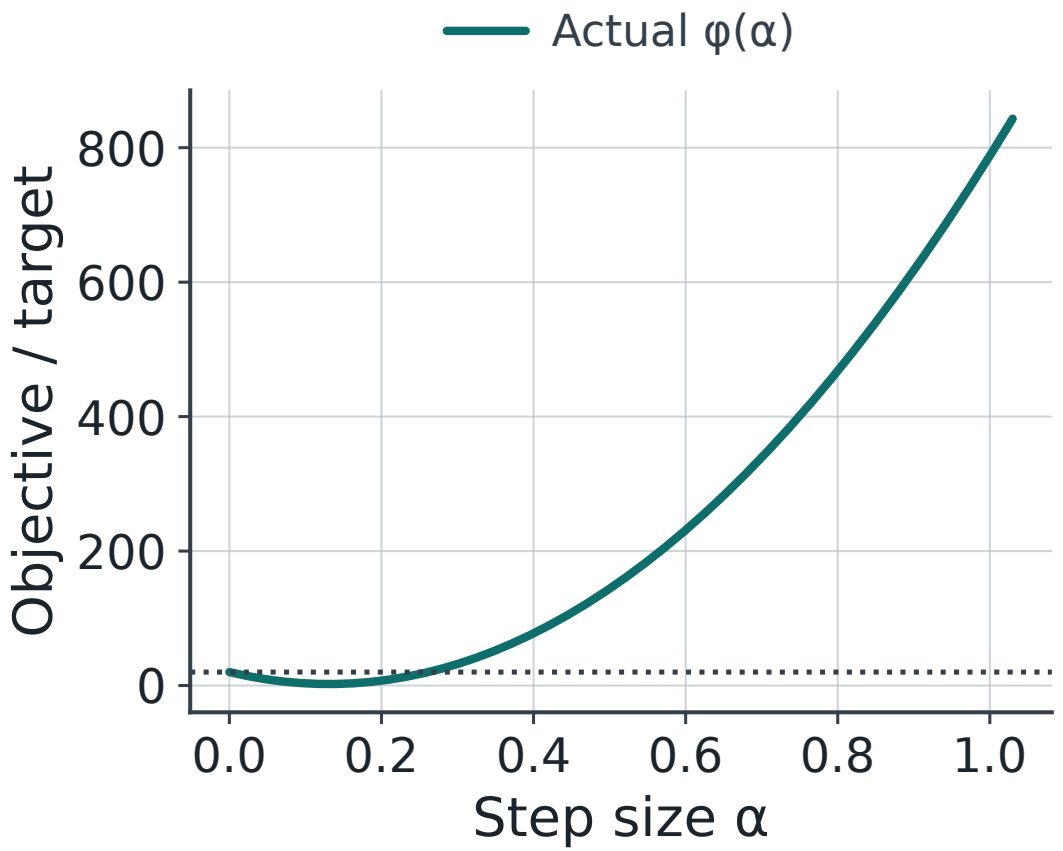


A large step can increase the objective

At $\alpha=1$: $r_1=(-1,-16)$, $f=788$.

At $\alpha=0.5$: $r_1=(1,-8)$, $f=144$.

Both exceed $f(r_0)=20$, despite $g_0^T p_0<0$.

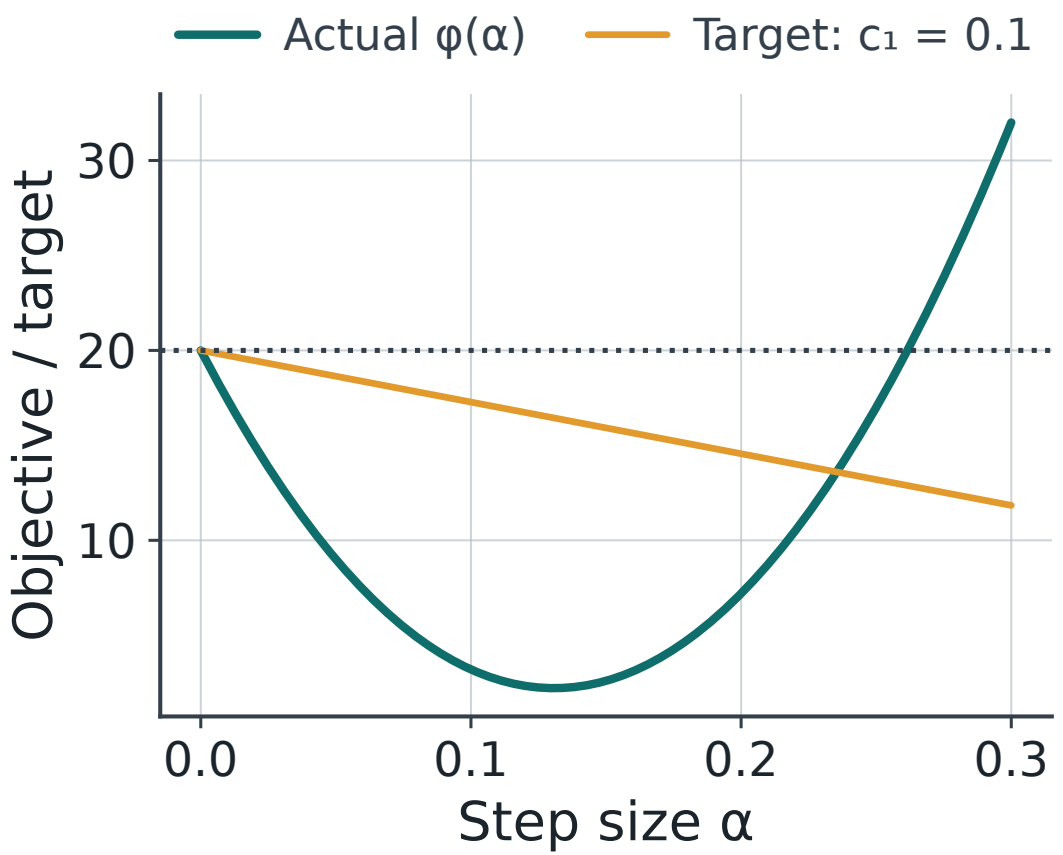


Armijo sufficient decrease

Accept α if $f(r+\alpha p) \le f(r) + c_1 \alpha g^T p$.

Choose $0<c_1<1$ and require $g^T p<0$.

The accepted value must beat a target below $f(r)$.

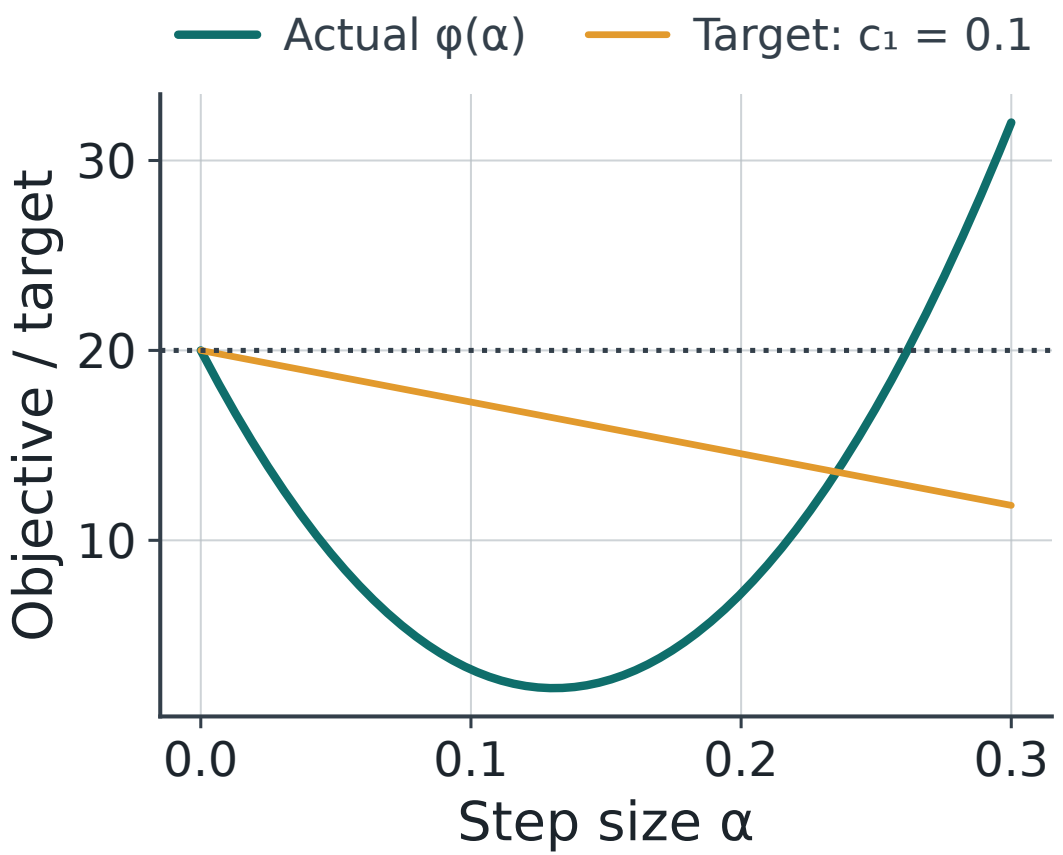


Backtracking parameters

Start each search with $\alpha_0 = 1$.

Here: $c_1 = 0.1$ and $\rho = 0.5$.

If Armijo fails, replace α by $\rho\alpha$ and test again.

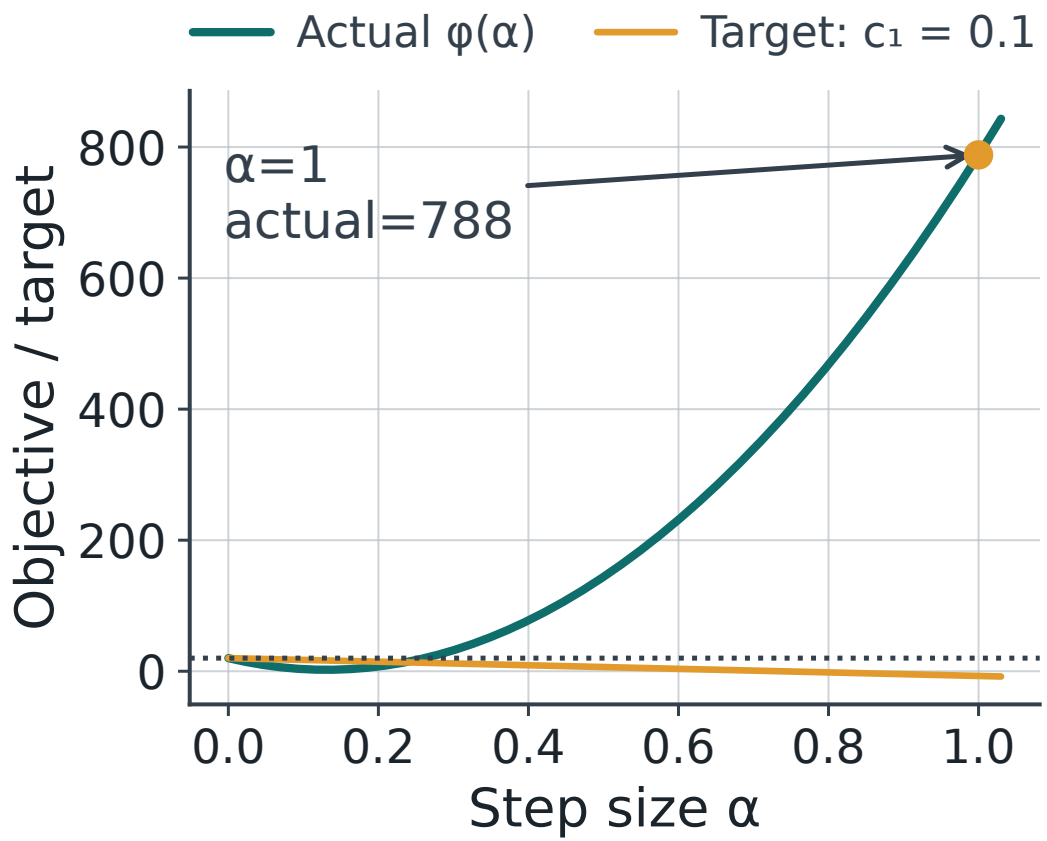


Trial 1: α=1

Actual value: φ(1)=788

Armijo target: 20 + 0.1(1)(−272) = −7.2

788 > −7.2: reject and halve α.

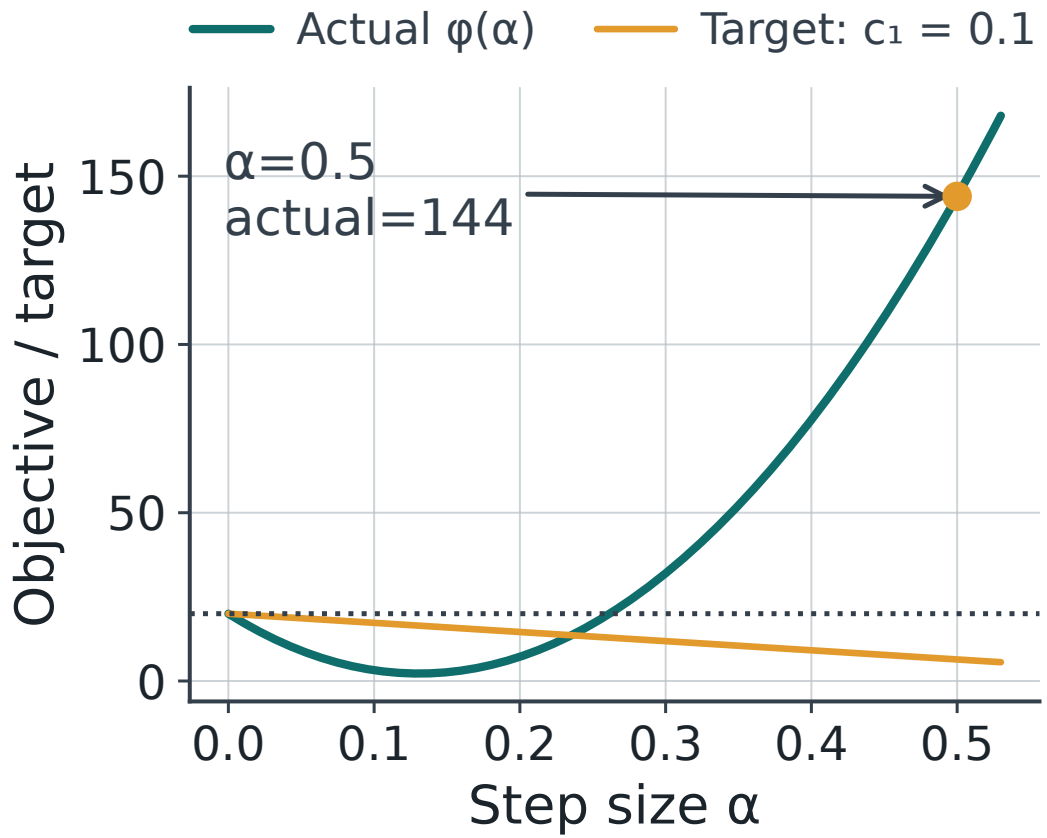


Trial 2: α=0.5

Actual value: φ(0.5)=144

Armijo target: $20 - 27.2(0.5) = 6.4$

144 > 6.4: reject and halve α.

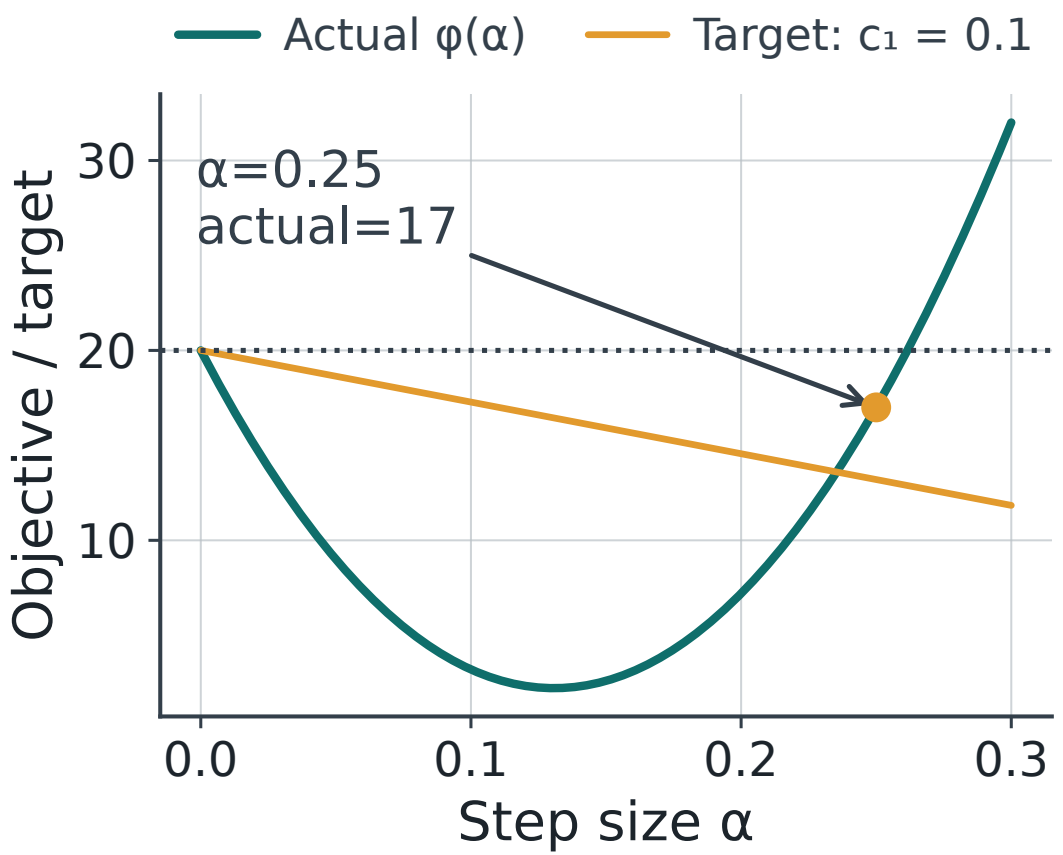


Trial 3: α=0.25

Actual value: φ(0.25)=17

Armijo target: $20 - 27.2(0.25) = 13.2$

17 > 13.2: reject, even though 17 < 20.

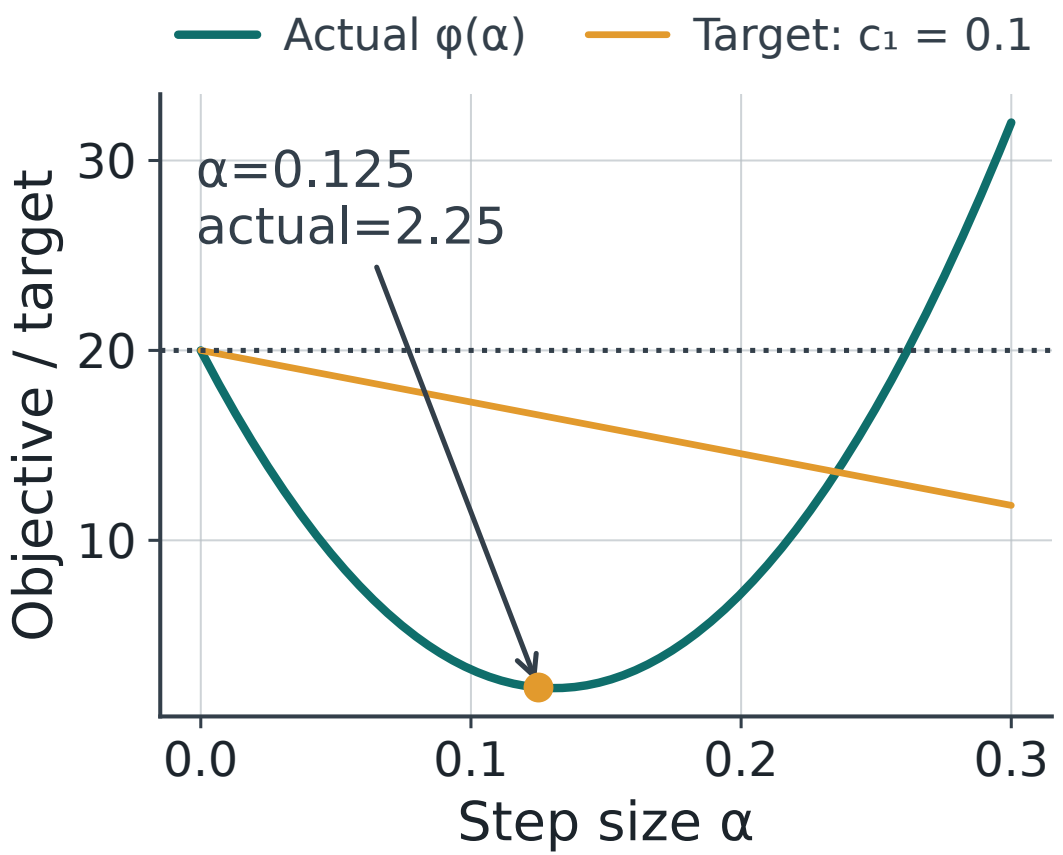


Trial 4: α=0.125

Actual value: φ(0.125)=2.25

Armijo target: 20 − 27.2(0.125) = 16.6

2.25 ≤ 16.6: accept r_1=(2.5,−2).

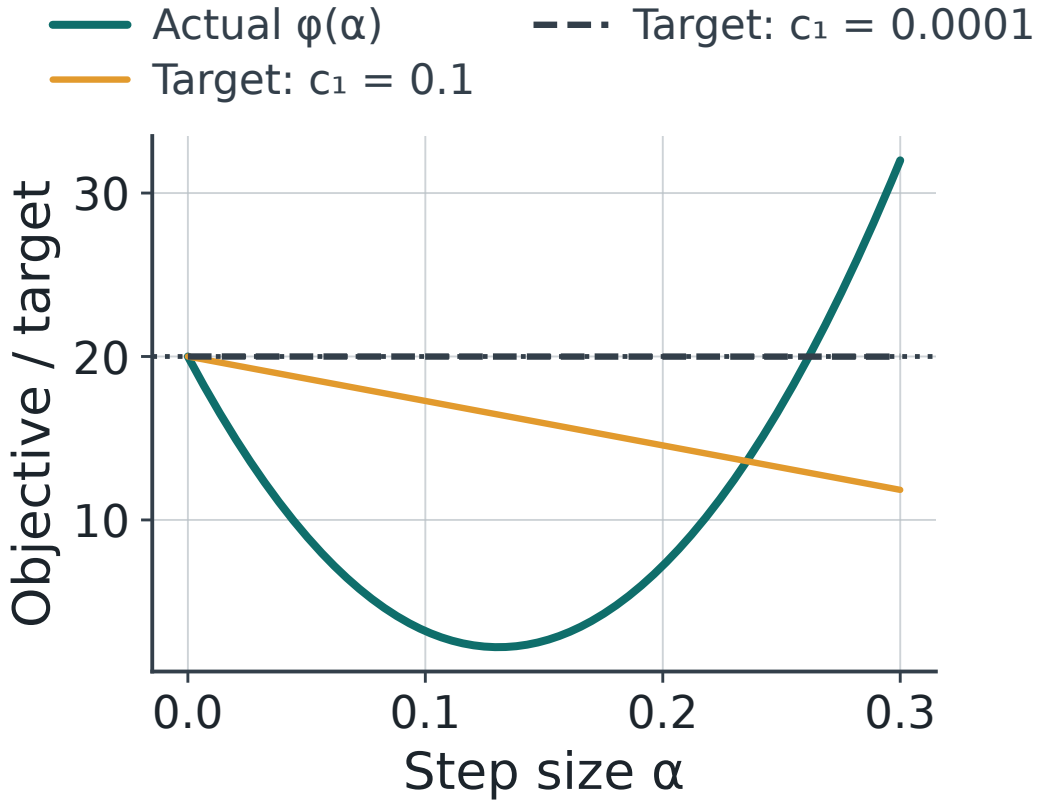


Acceptance depends on c_1

With $c_1=0.0001$, $\alpha=0.25$ passes the first search.

Its target is 19.9932, and the actual value is 17.

One accepted step does not justify keeping that α forever.



The two loops

Outer loop: compute g and choose a descent direction p.

Inner loop: keep r,p fixed; test α, $\rho\alpha$, $\rho^2\alpha$, …

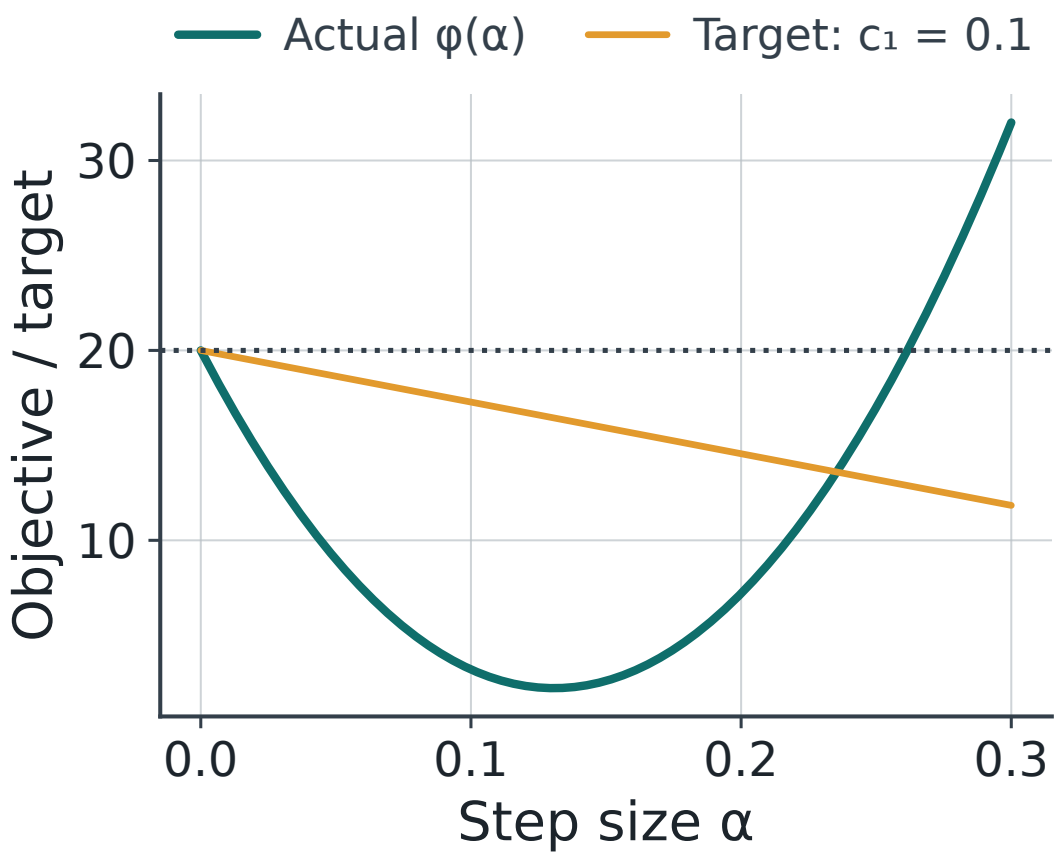
After acceptance: update r, then recompute g.

Demo 2: inspect each Armijo trial

Reset to (3,0), c_1=0.1, ρ=0.5.

Test 1 → 0.5 → 0.25 → 0.125; explain every decision.

Accept the step, then inspect the NEW gradient.



Open demo.html → armijo

Practice: one-dimensional backtracking

Minimize $f(x)=3x^2$. Start at $x=1$, so $p=-6$.

Use $\alpha_0=1$, $c_1=0.1$ and $\rho=0.5$.

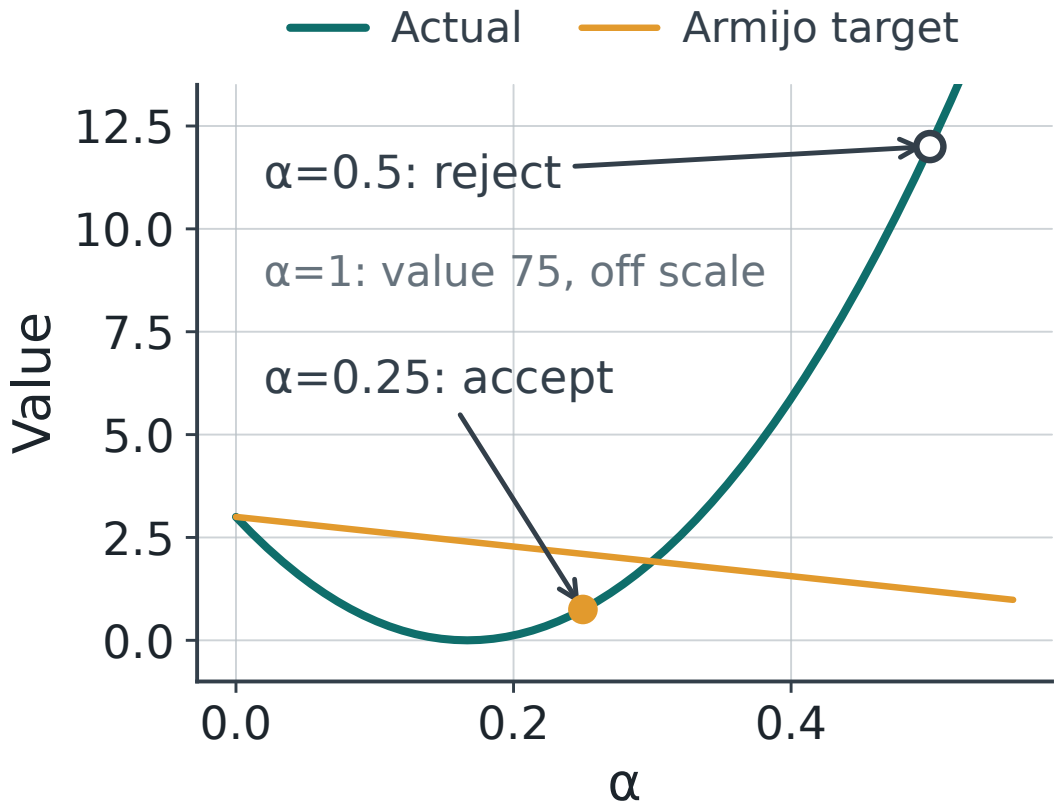
Find the first accepted α and the new point.

Practice solution

α=1: 75 > −0.6 and α=0.5: 12 > 1.2 → reject both.

α=0.25: actual 0.75 ≤ target 2.1 → accept.

New point x=−0.5: f drops from 3 to 0.75, but x* = 0.



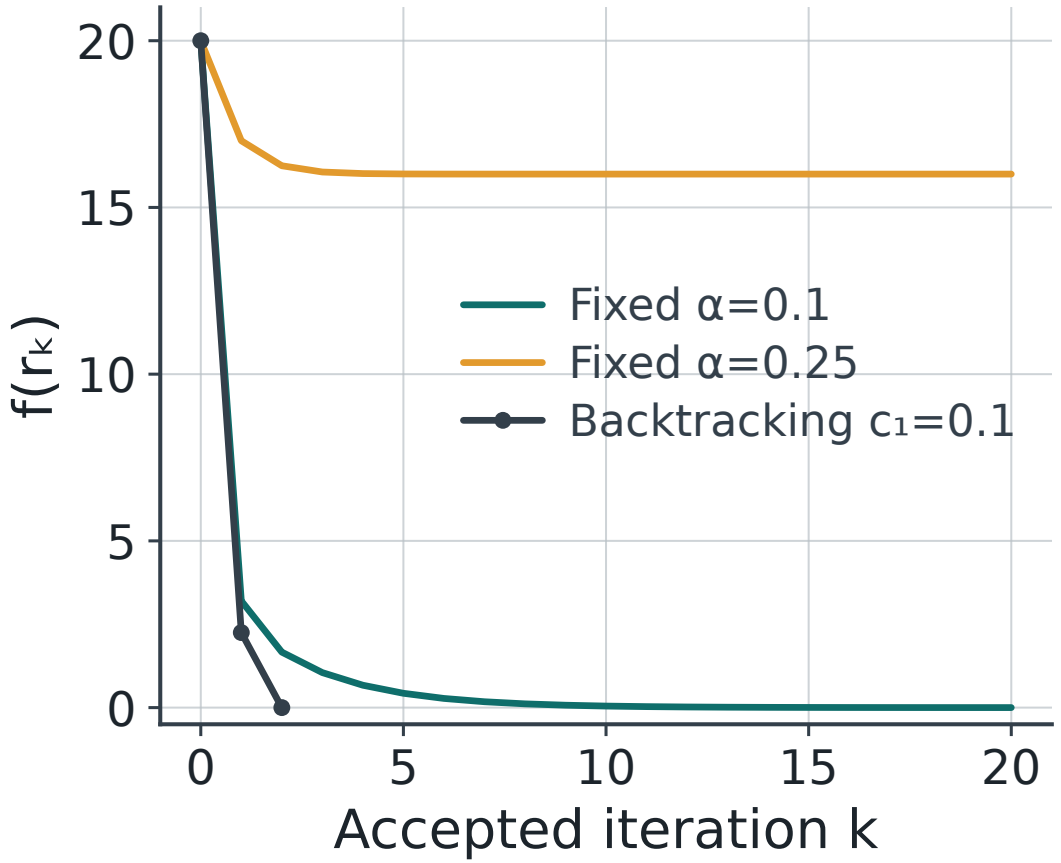
Compare complete runs

Use the same start (3,0) for every method.

Fixed α=0.1 converges; fixed α=0.25 stalls above 0.

Backtracking (c_1=0.1) reaches (1,−2) in two steps; exact arrival is special to this example.

Open demo.html → gd



Python demonstration

Run line_search_lab.py or open the notebook.

Print α, actual value, target and accept/reject.

Change the start or c_1, then explain the new trace.

Stopping and reporting

Check $\|g\|$, objective, iteration count and finite values.

A tiny step or small change can mean stagnation.

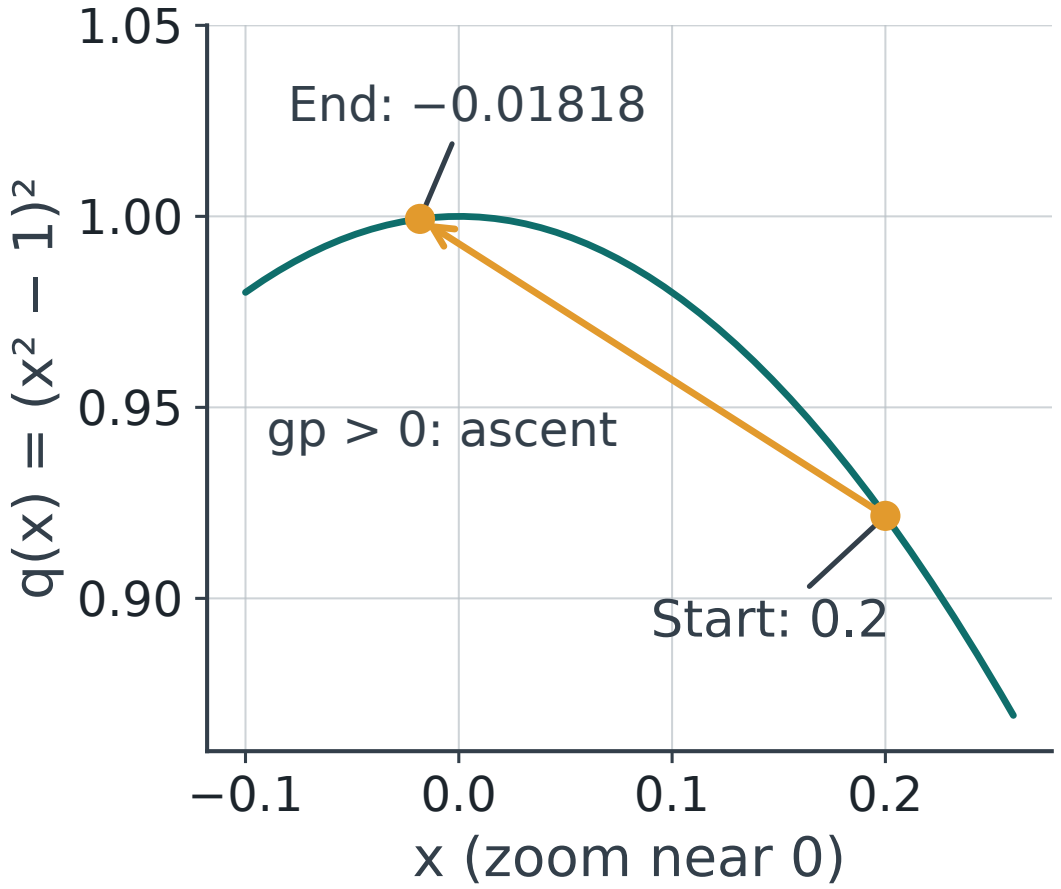
For nonconvex problems, also inspect curvature and multiple starts.

Line search cannot repair every direction

For q at x=0.2: Newton gives p≈−0.21818 and gp>0.

Small positive steps in this direction increase q.

Replace the direction, for example with p=−g, before descent search.



Open demo.html → newton

BFGS: a preview

Gradient descent uses $p = -g$.

Newton solves $Hp = -g$ using the Hessian.

BFGS updates a curvature approximation using changes in r and g.

Opening questions: answers

Prices bound profit only if dual feasible: (1,0) gives $w = 4 < 9$.

$\nabla f = 0$ is not enough: $x = 0$ is a local maximum of q.

A descent direction can still fail: $\alpha = 1$ and 0.5 increase f.

Exit check

Why was $\alpha=0.25$ rejected when $c_1=0.1$?

What stays fixed during the inner backtracking loop?

What must be checked before a descent line search?

Exit check: answers

17 > 13.2: the decrease is insufficient for this target.

Keep the current point r and direction p fixed.

Check $g^T p < 0$. If $g = 0$, stop and classify the point.

Next: constrained nonlinear optimization

Feasible directions and active constraints

Lagrange multipliers and KKT as optimality conditions

How practical constrained solvers use local models