

Shadow Prices in Linear Programming

What a shadow price is, how to compute and verify it, how reduced costs and non-binding constraints fit in, and how the idea extends to nonlinear programming.

The concept

A **shadow price** tells you how much the optimal objective value would improve if you relaxed a constraint by one unit. It is the marginal value of a resource.

A concrete example

Say you run a workshop making tables and chairs to maximize profit:

- Table: \$50 profit, needs 4 hours of labor and 6 board-feet of wood
- Chair: \$30 profit, needs 2 hours of labor and 3 board-feet of wood
- Available: 100 labor hours, 140 board-feet of wood

You solve the LP and find the optimal profit. Now suppose you could get one extra labor hour. The amount your optimal profit rises is the shadow price of labor. If it's \$12, then an extra hour of labor is worth up to \$12 to you, so paying a worker \$10/hour for overtime is profitable, while paying \$15 is not.

If the wood constraint isn't binding at the optimum (you have leftover wood), its shadow price is **zero**. More of something you already have in surplus adds nothing.

A note on these numbers

In this first illustration the wood requirement is exactly 1.5× the labor requirement for both products, so the two constraints are parallel. That makes the example degenerate. The full worked example below therefore uses different numbers.

Where it comes from: duality

Every LP (the "primal") has a companion LP (the "dual"). The dual has one variable per primal constraint, and those dual variables *are* the shadow prices. Strong duality says the primal and dual optimal objective values are equal, which gives a neat interpretation:

Total profit = $\sum$ (shadow price of each resource × amount of that resource available)

In other words, the shadow prices are a consistent way of "pricing" each resource so that the total value of your resources exactly equals your maximum profit.

Key properties

- **Non-binding constraints have a shadow price of 0.** This is complementary slackness: either a constraint is tight, or its shadow price is zero.
- **Sign depends on the problem type.** In a maximization problem, relaxing a $\leq$ constraint can only help, so its shadow price is ≥ 0 . In minimization,

relaxing a constraint lowers cost, and conventions for sign vary by software.

- **Valid only over a range.** The shadow price holds as long as the change is small enough that the same set of constraints stays binding (the same optimal basis). Solvers report this as the "allowable increase/decrease" in sensitivity analysis. Beyond that range, the shadow price changes.
- **It is a marginal, one-sided value.** At a degenerate vertex, the shadow price for increasing a resource can differ from the one for decreasing it.

Why it matters

1. **Resource valuation:** it tells you the maximum you should pay for more of a scarce resource.
2. **Prioritizing investments:** if labor has a shadow price of \$12 and wood \$2, expanding labor capacity is worth far more.
3. **Pricing and economics:** in economic models, shadow prices act like equilibrium prices that reflect scarcity.
4. **Related idea:** the *reduced cost* of a variable is the shadow price's counterpart for decision variables. It says how much the objective would worsen per unit if you forced a non-optimal activity into the plan.

Quick intuition check

Think of a binding constraint as a bottleneck. The shadow price answers: "If I could widen this bottleneck a little, how much more would I earn?" A constraint that isn't a bottleneck has nothing to widen, so its shadow price is zero.

Full worked example

The problem

Let T = tables and C = chairs.

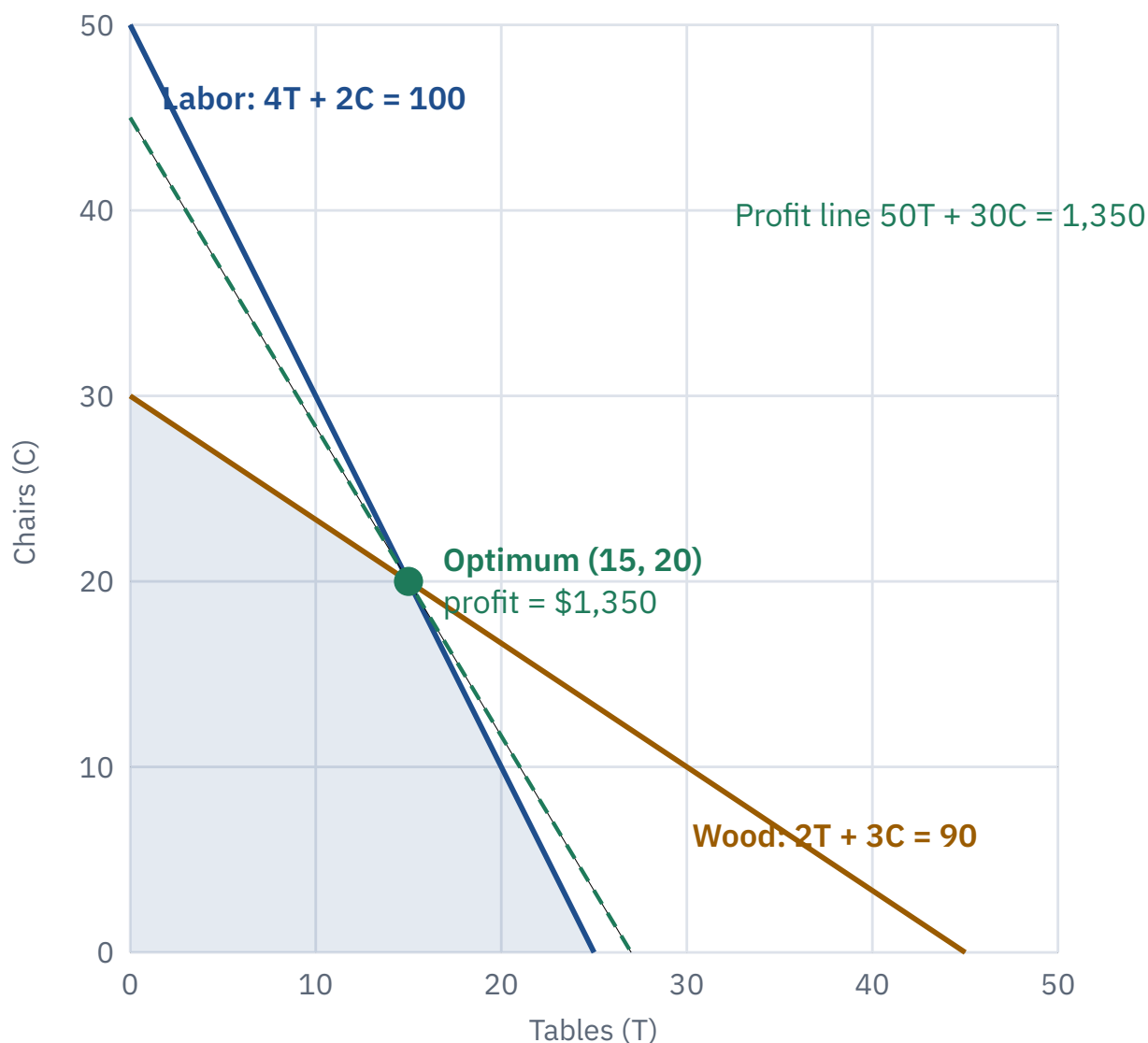
```
maximize    50T + 30C
subject to  4T + 2C ≤ 100    (labor hours)
            2T + 3C ≤ 90     (board-feet of wood)
            T, C ≥ 0
```

Step 1: Solve the primal

The feasible region has four corners:

Corner	Profit
(0, 0)	0
(25, 0)	1,250
(0, 30)	900
(15, 20)	1,350

The optimum is $T = 15$, $C = 20$, **profit = \$1,350**, where both constraints are binding (labor: $60 + 40 = 100$, wood: $30 + 60 = 90$). Neither resource has any slack.



The shaded region is feasible. The optimum sits at the corner where the labor and wood lines cross, and the dashed profit line just touches it.

Step 2: Write the dual

The dual has one variable per primal constraint: y_1 for labor and y_2 for wood. Think of them as the prices a rival would set to buy your resources.

```

minimize    100y1 + 90y2
subject to  4y1 + 2y2 ≥ 50    (a table's resources must be priced at least at its pr

```

$$2y_1 + 3y_2 \geq 30 \quad (\text{same for a chair})$$

$$y_1, y_2 \geq 0$$

The dual says: "Price your resources as cheaply as possible, but high enough that you'd never prefer to make a product over selling me the resources."

Step 3: Solve the dual

Since T and C are both positive in the primal, complementary slackness says both dual constraints must be tight:

$$4y_1 + 2y_2 = 50$$

$$2y_1 + 3y_2 = 30$$

From the first, $y_2 = 25 - 2y_1$. Substituting into the second: $2y_1 + 75 - 6y_1 = 30$, so $y_1 = 11.25$ and $y_2 = 2.5$.

Shadow prices

Labor = \$11.25 per hour. Wood = \$2.50 per board-foot.

Check with strong duality: $100(11.25) + 90(2.5) = 1,125 + 225 = 1,350$, matching the primal optimum.

Step 4: Verify the interpretation directly

Add one labor hour (right-hand side goes from 100 to 101) and re-solve the two binding equations:

$$4T + 2C = 101$$

$$2T + 3C = 90 \quad \rightarrow \quad T = 15.375, \quad C = 19.75$$

New profit = $50(15.375) + 30(19.75) = 1,361.25$, an increase of **\$11.25**, exactly the shadow price. The same test for wood (90 to 91) gives $T = 14.75$, $C = 20.5$ and profit 1,352.50, an increase of **\$2.50**.

Step 5: Find the valid range

Solve the binding equations with a general labor right-hand side b_1 (wood fixed at 90):

$$T = (3b_1 - 180)/8, \quad C = (360 - 2b_1)/8$$

The solution stays feasible while $T \geq 0$ ($b_1 \geq 60$) and $C \geq 0$ ($b_1 \leq 180$). So the **\$11.25 holds for labor between 60 and 180 hours**: allowable decrease 40, allowable increase 80. Outside that range, a different corner becomes optimal and the shadow price changes.

Doing the same for wood (labor fixed at 100) gives $T = (300 - 2b_2)/8$ and $C = (4b_2 - 200)/8$, so the **\$2.50 holds for wood between 50 and 150 board-feet**: allowable decrease 40, allowable increase 60.

Step 6: Reading it from solver output

Here is how you would get this in Python:

```
from scipy.optimize import linprog

# linprog minimizes, so negate the profit
res = linprog(c=[-50, -30],
              A_ub=[[4, 2], [2, 3]],
              b_ub=[100, 90],
              method="highs")
```

```
print(res.x)           # [15. 20.]
print(-res.fun)        # 1350.0
print(res.ineqlin.marginals) # [-11.25 -2.5]
```

Because we negated the objective, SciPy reports the marginals with flipped sign (-11.25 , -2.5). Flip them back to get the shadow prices. Other tools use different conventions, so always check the documentation:

Solver	What to look for
Excel Solver	"Sensitivity Report": <i>Shadow Price</i> , <i>Allowable Increase/Decrease</i>
Gurobi / CPLEX	Pi / dual values on constraints; SARHSLow / SARHSUp for ranges
R lpSolve	sensitivity.duals = TRUE

An Excel-style report for this problem would look like:

Constraint	Final Value	Shadow Price	RHS	Allowable Increase	Allowable Decrease
Labor	100	11.25	100	80	40
Wood	90	2.50	90	60	40

Decisions this supports

- **Overtime:** Paying a worker \$10/hour is profitable (up to 80 extra hours), since each hour returns \$11.25. At \$13/hour it isn't.
- **Where to invest:** Labor is worth 4.5× as much per unit as wood, but the units differ, so compare costs. If an extra labor hour costs \$8 and an extra board-foot costs \$3, buy labor (net +\$3.25) and skip wood (net −\$0.50).

- **Past the range:** If you somehow had 200 labor hours, you'd hit the $C = 0$ corner ($T = 45$, limited by wood) and labor's shadow price would drop to \$0, because wood becomes the only bottleneck.

Reduced costs: a product that isn't worth making

Both extensions build on the same base problem:

```

maximize    50T + 30C
subject to  4T + 2C ≤ 100    (labor)      shadow price  $y_1 = 11.25$ 
            2T + 3C ≤ 90     (wood)       shadow price  $y_2 = 2.50$ 
Optimum: T = 15, C = 20, profit = $1,350
  
```

Add a third product

Suppose you consider bookshelves (B): **\$40 profit, 5 labor hours, 4 board-feet of wood.**

```

maximize    50T + 30C + 40B
subject to  4T + 2C + 5B ≤ 100
            2T + 3C + 4B ≤ 90
            T, C, B ≥ 0
  
```

Price it with the shadow prices

Using the shadow prices as the "cost" of the resources a bookshelf consumes:

```

resource cost of one bookshelf = 5(11.25) + 4(2.50) = 56.25 + 10 = $66.25
  
```

A bookshelf earns only \$40 but ties up \$66.25 worth of scarce resources. The **reduced cost** is the gap:

$$\text{reduced cost} = \text{profit} - \text{resource cost} = 40 - 66.25 = -26.25$$

For a maximization problem, a negative reduced cost means the product isn't worth making, so **B = 0** stays optimal. The optimal plan and the \$1,350 profit are unchanged.

Verify by forcing one in

Force **B = 1** and re-solve the two binding constraints for **T** and **C**:

$$\begin{aligned} 4T + 2C &= 95 \\ 2T + 3C &= 86 \end{aligned} \rightarrow T = 14.125, C = 19.25$$

Profit = $50(14.125) + 30(19.25) + 40 = 1,323.75$, a drop of **\$26.25** from 1,350, exactly the reduced cost.

Connection to the dual

The dual gains a third constraint, one per primal variable:

$$5y_1 + 4y_2 \geq 40$$

At $y = (11.25, 2.5)$, the left side is 66.25, which is slack by 26.25. Complementary slackness says a slack dual constraint forces its primal variable to zero. The reduced cost is just the **slack in the dual constraint**. For **T** and **C**, the dual constraints are tight, so their reduced costs are 0. Basic variables always have a

reduced cost of 0.

Break-even and the allowable range

Bookshelves become attractive once their profit exceeds \$66.25. That is also the "allowable increase" for B's objective coefficient: 26.25 above \$40. Try \$70:

```
reduced cost = 70 - 66.25 = +3.75 > 0 → B enters the plan
```

Re-solving gives $B \approx 17.14$, $C \approx 7.14$, $T = 0$, profit $\approx \$1,414.29$. The product mix changes completely, and the shadow prices shift to $y_1 \approx 12.86$, $y_2 \approx 1.43$. This is why reduced costs are only valid for small changes.

In solver output

```
res = linprog(c=[-50, -30, -40],
              A_ub=[[4, 2, 5], [2, 3, 4]],
              b_ub=[100, 90], method="highs")

print(res.x)           # [15. 20.  0.]
print(res.lower.marginals) # [ 0.  0. 26.25]
```

SciPy reports 26.25 (positive, since it's minimizing $-\text{profit}$), while Excel's Sensitivity Report would show -26.25 for a maximization problem. The magnitude is the same either way.

Variable	Final Value	Reduced Cost	Objective Coeff.	Allowable Increase	Allowable Decrease
Tables	15	0	50	10	
Chairs	20	0	30	45	
Tricycles	0	40	40		40

Bookshelves	0	-26.25	40	26.25
-------------	---	--------	----	-------

Correction to the earlier version

When this table first appeared in the conversation, the Tables and Chairs ranges were flagged as approximate (25/25 and 20/15). They have now been worked out properly. The Tables coefficient can range from \$20 to \$60 before the optimal corner changes, and the Chairs coefficient from \$25 to \$75. The bookshelf row was correct as originally given.

A non-binding constraint

Add a third resource

Go back to the two-product problem and add a **finishing capacity** limit: each table and chair needs one finishing slot, and you have 50.

```

maximize    50T + 30C
subject to  4T + 2C ≤ 100    (labor)
            2T + 3C ≤ 90     (wood)
            T +  C ≤ 50      (finishing)

```

Check the old optimum

At (15, 20), finishing used = $15 + 20 = 35 \leq 50$. The constraint has **15 units of slack**, so the optimum doesn't move. It's still $T = 15$, $C = 20$, profit \$1,350.

Shadow price = 0, and why

Two ways to see it:

1. **Intuition:** you already have 15 unused finishing slots, so a 51st slot adds nothing.
2. **Complementary slackness:** the constraint has slack, so its dual variable must be 0:

$$y_3 \times (50 - 35) = 0 \rightarrow y_3 = 0$$

The dual solution is $(y_1, y_2, y_3) = (11.25, 2.5, 0)$, and the dual objective still gives $100(11.25) + 90(2.5) + 50(0) = 1,350$. The new resource contributes nothing to the total value.

The range tells you more than the zero

Constraint	Final Value	Shadow Price	RHS	Allowable Increase	Allowable Decrease
Labor	100	11.25	100	80	40
Wood	90	2.50	90	60	40
Finishing	35	0	50	∞	15

The allowable increase is infinite, since more finishing capacity never helps. The allowable decrease is exactly the slack: 15. You can cut finishing capacity to 35 for free, but below that, it starts to bind.

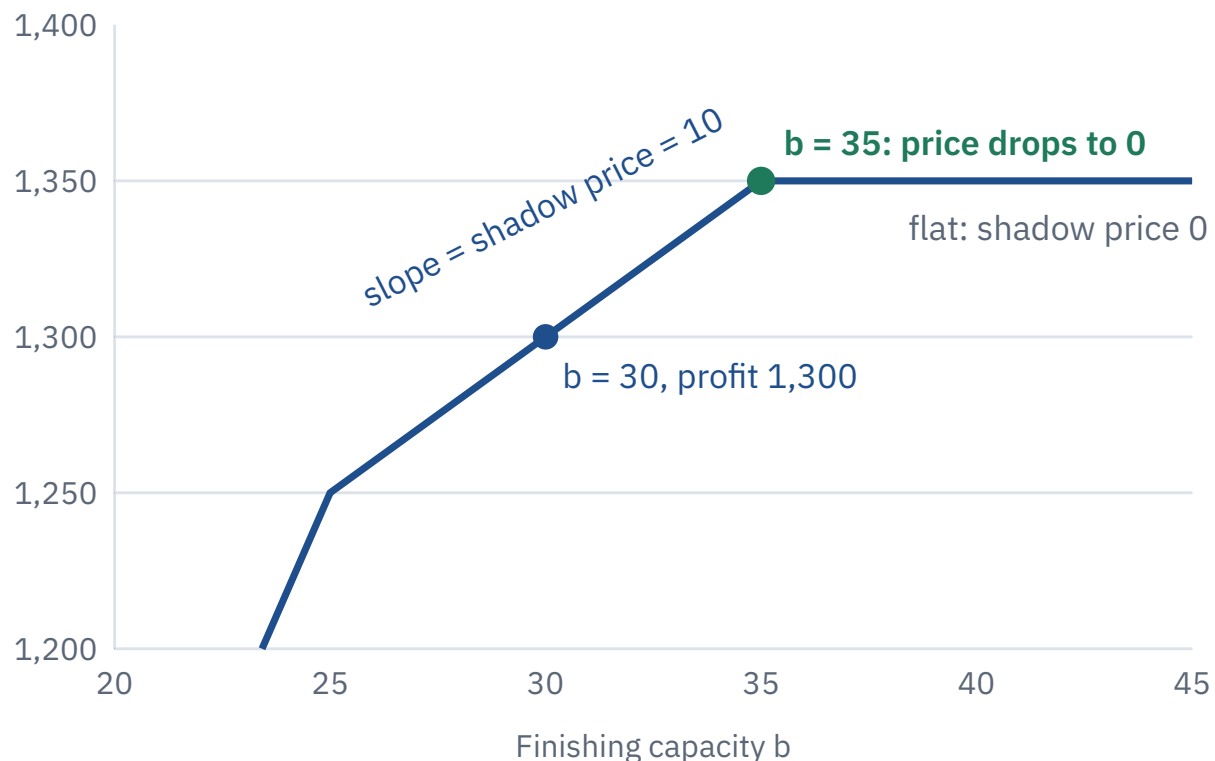
What happens when it starts to bind

Cut finishing to 30. Now finishing and labor are the binding pair (wood has slack: $40 + 30 = 70 \leq 90$), giving $T = 20$, $C = 10$:

$$\text{profit} = 50(20) + 30(10) = \$1,300$$

The new shadow prices come from solving the binding pair's dual: $4y_1 + y_3 = 50$ and $2y_1 + y_3 = 30$, so $y_1 = 10$, $y_3 = 10$, $y_2 = 0$. Now finishing is a bottleneck worth \$10 per slot, and wood is the non-binding one. The check: $100(10) + 30(10) = 1,300$ ✓.

In general, for a finishing limit b between 25 and 35, $\text{profit} = 1000 + 10b$, a straight line with slope equal to the shadow price. At $b = 35$ it hits the old \$1,350 and flattens out, which is the shadow price dropping to 0 for any larger b .



Optimal profit as the finishing limit changes. Between 25 and 35 the line has slope 10, the shadow price. Past 35 the constraint is slack and the curve is flat. Below 25 a different corner takes over, where the slope is steeper (50 per slot).

```

res = linprog(c=[-50, -30],
              A_ub=[[4, 2], [2, 3], [1, 1]],
              b_ub=[100, 90, 50], method="highs")

print(res.ineqlin.marginals)  # [-11.25 -2.5  0.]
print(res.slack)             # [ 0.  0. 15.]

```

Putting it together

Concept	Applies to	Zero when...	Tells you
Shadow price	A constraint	the constraint has slack	the value of loosening that resource
Reduced cost	A variable	the variable is in the plan	the cost of forcing it in (or how much its profit must rise to enter)

Both are two faces of complementary slackness: for every primal-dual pair, either the primal side is "tight" or the dual side is zero. When a number comes back as zero in sensitivity output, it's telling you which side of that pair is loose.

The nonlinear analog: Lagrange and KKT multipliers

Yes, there is one. The nonlinear analog is the **Lagrange multiplier**, or in the inequality-constrained case, the **KKT multiplier**. In fact, the LP shadow price

is just a special case of it. The dual variables in LP duality are the KKT multipliers of a linear problem.

The core result

For a problem like

$$\begin{array}{ll}\text{maximize} & f(x) \\ \text{subject to} & g(x) \leq b\end{array}$$

with Lagrangian $L = f(x) - \lambda(g(x) - b)$, the multiplier λ at the optimum satisfies

$$dV/db = \lambda$$

where $V(b)$ is the optimal objective value as a function of the right-hand side. This is a version of the **envelope theorem**. So λ is the marginal value of relaxing the constraint, exactly as with a shadow price. The other LP features carry over too:

- **Complementary slackness:** $\lambda(g(x) - b) = 0$, so a slack constraint has $\lambda = 0$.
- **Sign:** for a maximization with a $\leq$ constraint, $\lambda \geq 0$.
- **Reduced costs:** these are the multipliers on bound constraints like $x \geq 0$.

A quick example

$$\begin{array}{ll}\text{maximize} & xy \\ \text{subject to} & x + y \leq b\end{array}$$

Setting $\nabla f = \lambda \nabla g$ gives $y = \lambda$ and $x = \lambda$, so $x = y = b/2$ and $\lambda = b/2$. The optimal

value is $V(b) = b^2/4$.

Check: $dV/db = b/2 = \lambda$ ✓.

At $b = 10$, $V = 25$ and $\lambda = 5$. Raising b to 11 gives $V = 30.25$, a gain of 5.25, close to the multiplier's prediction of 5 (the gap is the curvature).

What's different from LP

- **It's local, and it changes continuously.** In an LP, $V(b)$ is piecewise linear, so the shadow price is constant across a whole range. In the example above, $V(b)$ is a curve, so $\lambda = b/2$ changes with every unit of b . It's only the instantaneous rate at the current point, which is why the 5 slightly underpredicted the 5.25. There's no "allowable increase" range, because there are no flat stretches.
- **It depends on the optimum you found.** In a nonconvex problem, the multiplier describes the local optimum your solver landed on, and a different local optimum would have different multipliers.
- **Some conditions are required.** The KKT conditions, and the interpretation of λ as dV/db , need a constraint qualification (such as linear independence of the active constraint gradients) and differentiability. In degenerate cases the multipliers may not be unique, or $V(b)$ may have kinks, giving the same one-sided behavior as degenerate LPs.
- **Duality can fail to be exact.** For convex problems with a strictly feasible point (Slater's condition), strong duality holds and the multipliers solve a genuine dual problem, just as in LP. For nonconvex problems there can be a duality gap, so the multiplier is only a local sensitivity, not part of a clean primal-dual pair.
- **Units depend on how you write the constraint.** If you rescale $g(x) \leq b$ by

multiplying through by 1,000, λ shrinks by 1,000. This is true in LP too, but it matters more when constraints are complicated nonlinear functions.

	Linear programming	Nonlinear programming
Name	Shadow price (dual variable)	Lagrange / KKT multiplier
Meaning	dV/db	dV/db (envelope theorem)
Constant over a range?	Yes, until the basis changes	No, varies continuously
Slack constraint	Price is 0	Multiplier is 0
Strong duality	Always (if feasible and bounded)	Convex case with Slater's condition; may fail otherwise
Sensitivity ranges	Allowable increase/decrease	None; the value is local

Where you'll see it

In economics, the multiplier on a budget constraint in utility maximization is the **marginal utility of income**, which is the same idea as a shadow price. In solvers, nonlinear tools like IPOPT and Gurobi's nonconvex mode report multipliers, and Excel's GRG Nonlinear engine produces a Sensitivity Report with a **Lagrange Multiplier** column instead of "Shadow Price."

The short version

Shadow price is a Lagrange multiplier on a linear problem. In the nonlinear case you keep the interpretation but lose the constant, range-wide validity.

Worked examples use a table, chair, and bookshelf workshop with labor and wood constraints. All figures were checked by re-solving the binding equations.