

PRIZE COURSE

How Canals and Locks Work

A free prize course, generated with Claude.

8 Chapters | Philip Osztromok | Generated with Claude

Contents

Chapter 1	Why Canals	2
Chapter 2	Water Level Basics	7
Chapter 3	Inside a Lock	12
Chapter 4	Water Supply	18
Chapter 5	Flights and Staircases	25
Chapter 6	Lifts and Inclines	33
Chapter 7	Crossing and Cutting	39
Chapter 8	Canals Today	46

How Canals and Locks Work 🏆

Chapter 1 · Why Canals

A canal is a waterway dug by people, and it exists to solve one stubborn problem: **heavy things are hard to move over land.** Before engines, the power available was mostly muscle, a horse, an ox or a person, and a muscle gets tired and pulls only so hard. This chapter asks a plain question: why does water make the job so much easier, and why did people go to the huge expense of digging a channel for it?

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The two Python snippets in this chapter and the three exercise solutions were executed, and the output shown is exactly what Python printed. The numbers fed in (the horse's pull, the resistance coefficients and the payload fractions) are *round illustrative assumptions*, not measurements from a particular canal or road. They show the shape of the argument. Historical dates and figures I quote from memory are marked (check).

Why Water Wins: Resistance

Pulling a load along the ground, the force you need is roughly the load's weight multiplied by a small number, the **resistance coefficient**: the force needed for each unit of weight. A rough, rutted road soaks up energy. A boat does not roll on a surface at all. It is held up by the water (a floating boat displaces its own weight of water, which is why it floats), so the only thing to overcome is the water's drag as the boat slides through, and at walking pace that drag is small.

For a horse that can pull with a steady force `F`, the heaviest load it can move is `F / (c × g)`, where `c` is the resistance coefficient and `g` is 9.81 m/s². The code below tries three surfaces. A pull of 700 N is a plausible order of magnitude for a working horse (check), and the three coefficients are assumed.

```
g = 9.81                    # m/s^2
pull = 700                  # N, assumed steady pull of one horse (illustrative)

# rolling-resistance coefficient: force needed per unit of weight (assumed values)
surfaces = {"poor road": 0.10, "good turnpike": 0.03, "canal": 0.002}
```

```
for name, c in surfaces.items():
    mass_kg = pull / (c * g)
    print(name, round(mass_kg / 1000, 1), "tonnes moved by one horse")
# Output:
# poor road 0.7 tonnes moved by one horse
# good turnpike 2.4 tonnes moved by one horse
# canal 35.7 tonnes moved by one horse
```

The same horse that can shift 0.7 tonnes on a poor road can move about 36 tonnes on a canal. Sources commonly say a horse could pull a boat carrying around thirty tonnes (check), which is in the same range as this rough result. The important number is the ratio, about 50 to 1 between a bad road and still water. Good roads closed the gap a little, but not by anything like enough for heavy cargo.

A Day's Work: Tonne-Miles

Transport people count work in **tonne-miles**: tonnes of cargo carried, multiplied by miles. A wagon is itself heavy, and so is a boat, so only part of the load the horse moves is paying cargo. The snippet assumes a horse walks 2.5 mph for 8 hours, and uses assumed payload fractions.

```
walk_mph = 2.5
hours = 8
miles_per_day = walk_mph * hours
print("miles per horse-day:", miles_per_day)

# gross tonnes one horse moves, and the fraction that is paying cargo (assumed)
modes = {"road wagon": (0.7, 0.6), "canal boat": (35.7, 0.7)}
for name, (gross_t, payload) in modes.items():
    cargo = gross_t * payload
    print(name, "cargo", round(cargo, 1), "t, tonne-miles per day", round(cargo * miles_
# Output:
# miles per horse-day: 20.0
# road wagon cargo 0.4 t, tonne-miles per day 8
# canal boat cargo 25.0 t, tonne-miles per day 500
```

On these figures one horse-day on the canal moves about 60 times the cargo-miles of one on the road (500 against 8). The boat is slow and the route is fixed, but it carries so much for the same effort that for coal, stone, bricks, grain and clay it was a bargain. This is the whole economic reason for canals.

Canal, River or Road?

FEATURE	RIVER	CANAL	ROAD
Made by	nature	people	people
Route	wanders, may not go where you need	dug where it is wanted	goes almost anywhere
Water flow	current, floods, droughts	almost still, level in each section	none
Load per horse	large downstream, harder upstream	large either way	small
Speed	depends on the current	walking pace	walking pace or better
Weakness	unreliable, silting	needs a supply of water and much digging	wears out, costly for heavy cargo

Horse-Drawn Towing

On a canal the horse does not walk on the water. It walks on a **towpath** beside it, pulling a long rope fixed to the boat. British canal boats were commonly long and narrow so that two could pass in a channel not much wider than one (check). Where the towpath changed sides, the horse crossed on a bridge built so that the rope did not need unhooking: a "roving" or "turnover" bridge (check).

Early Canals and Where the Idea Came From

- **China:** the Grand Canal, in parts very old, was joined into a single north-south waterway around the early 600s CE (check). It carried grain to feed the cities of the north.
- **Roman Britain:** the Fossdyke in Lincolnshire is often said to be a Roman canal (check).

- **France:** the Canal du Midi, linking the Atlantic side to the Mediterranean, opened in the early 1680s (check).
- **Britain:** the Bridgewater Canal, built to carry coal from the Duke of Bridgewater's mines at Worsley to Manchester, opened in 1761 (check). Coal prices in Manchester are said to have fallen sharply, and its success helped start the great age of British canal building (check).
- **United States:** the Erie Canal, opened in 1825, joined the Hudson River to the Great Lakes (check).

The same pattern appears each time: a country with a heavy, bulky cargo to move, only muscle for power, and rivers that did not quite do the job.

COMMON MISTAKES

Thinking a boat is fast because it is easy to pull (the load was huge, but the speed stayed at walking pace); imagining the horse walked in the water (it used a towpath); and forgetting that a canal still has to be filled with water and kept level, which is the problem the next chapter takes on, in How Canals and Locks Work 2.

Challenges

EXERCISE 1

A weaker horse pulls a steady 600 N. Using resistance coefficients of 0.06 for an ordinary road and 0.003 for a canal (both assumed), find the mass it can move on each, and how many times more on the canal.

View solution

EXERCISE 2

You must move 60 tonnes of coal 15 miles. A one-horse wagon carries 0.42 t of coal (a 0.7 t gross load, 60% payload); a one-horse boat carries 20 t (assumed). Count the loads each way needs, and the horse-days if each load takes one day outward. What does the result say about why canals were built for coal?

View solution

EXERCISE 3

Short answer. (a) Give two reasons a river alone is often not good enough as a transport route, and one thing a canal can never beat a road at. (b) A canal boat moves at walking pace. Name one kind of cargo for which that hardly matters and one for which it would, and say why.

View solution

Summary

KEY POINTS

- **Resistance:** a floating boat meets far less resistance than wheels on a road, so one horse moves 20 to 50 times the load
- **Rule of thumb:** heaviest load = pull / (c × g)
- **Trade-off:** slow, fixed route, but huge loads for bulky, low-value cargo
- **Towing:** the horse walks on a towpath beside the canal
- **Early canals:** grain in China, coal in Britain, goods and grain in North America

COMING NEXT

How Canals and Locks Work 2 is about water levels: why a canal is mostly level, the long contour routes, and how builders keep the water in with puddling clay and lining.

How Canals and Locks Work

Chapter 2 · Water Level Basics

Chapter 1 showed why a boat on still water is so easy to pull: there is almost no resistance. That only works if the water really is *still*. A canal is a long thin lake built on sloping ground, and water does two awkward things: it runs downhill, and it soaks into the ground. This chapter is about how builders cope with both, by keeping canals level and by making the bed hold water.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The Python snippets and the three exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 1, the inputs (lengths, widths, lift heights, seepage rates) are *round illustrative assumptions*, not measurements from a particular canal. Historical facts quoted from memory are marked (check).

Water Finds Its Level

If you tilt a bath, the water rushes to the low end. A canal is the same: a stretch of still water has a flat surface, however the ground beneath it slopes. A stretch of canal held at one level is called a **pound**. Along its whole length the surface is level, and the depth is greater where the ground dips and smaller where it rises, so builders dig and bank up the bed to keep a steady depth for the boats.

A pound ends at a **lock** (the subject of How Canals and Locks Work 3) or at the end of the canal, where the next pound sits at a different height. So the real question for a builder is: how do I get from A to B at a set of fixed levels, with as few steps as possible?

Contour Canals: Going Round the Hill

The cheapest way to avoid steps is to follow a hillside at a constant height, a **contour**, and let the route wind to keep its level. Many early British canals were contour canals, and the Oxford Canal, for instance, is famous for its winding course (check). The price is distance. The sum below compares a straight route over a hill, which needs locks, with a longer contour route that does not. The speeds, lock lift and lock time are assumed.

```
rise = 24.0               # m, height gained crossing the hill
lock_lift = 2.4           # m per lock (assumed)
direct_km = 10.0
contour_km = 17.0
walk_kmh = 3.2            # horse and boat at walking pace (assumed)
lock_minutes = 15         # minutes per lock passage (assumed)

locks = round(rise / lock_lift)
direct_h = direct_km / walk_kmh + locks * lock_minutes / 60
contour_h = contour_km / walk_kmh
print("locks on direct route:", locks)
print("direct route hours:", round(direct_h, 1))
print("contour route hours:", round(contour_h, 1))
# Output:
# locks on direct route: 10
# direct route hours: 5.6
# contour route hours: 5.3
```

The contour route is 7 km longer but a little quicker, because ten lock passages cost two and a half hours. It also needs no lock keepers and spends no water on lockage. But the straight line is shorter to dig and build, so neither is automatically right: the sum has to be done for each hill.

FEATURE	CONTOUR CANAL	CANAL WITH LOCKS
Route	long, winding	shorter, straighter
Height change	none, one level	stepped, pound by pound
Journey time	no lock delays	each lock adds minutes
Water supply	needs only to replace leaks	every passage also uses water
Staff	few	lock keepers or boaters to work locks
Weakness	a long bank on a hillside that can slip or leak	delays and water demand

Keeping Water In

Even on perfectly level ground, a canal loses water downward and sideways through the bed and banks. Sand and gravel drain freely. Clay holds water well. If the route crosses porous ground, the builders must seal the channel. How much does it matter? The code below uses a 10 km pound and assumed seepage rates (the depth of water lost per day).

```
length_m = 10_000          # a 10 km pound
width_m = 10               # average water surface width (assumed)
depth_m = 1.5              # assumed
area = length_m * width_m
volume = area * depth_m
print("pound volume (m3):", volume)

# seepage as depth of water lost per day through the bed and banks (assumed values)
seep_mm_day = {"bare earth": 20, "poorly puddled": 5, "good puddled clay": 1}
for name, mm in seep_mm_day.items():
    lost = area * mm / 1000
    print(name, "loses", round(lost), "m3 per day,", round(lost / volume * 100, 1), "% o
# Output:
# pound volume (m3): 150000.0
# bare earth loses 2000 m3 per day, 1.3 % of the pound
# poorly puddled loses 500 m3 per day, 0.3 % of the pound
# good puddled clay loses 100 m3 per day, 0.1 % of the pound
```

A percent or so a day looks small, but it is lost *every day*, without a boat in sight, and the shortfall must be fed in from a reservoir or stream (the topic of How Canals and Locks Work 4). Bare earth loses twenty times as much as good puddling. In a canal with a thin water supply, that difference decides whether boats can float all summer.

Puddled Clay

Puddling means working wet clay, usually mixed with sand or gravel, until it becomes a dense, tough, nearly watertight skin. Workers originally trampled it with their feet, and later machines helped (check). The best-known name is James Brindley, who is associated with puddled clay lining on the Bridgewater Canal (check). Typically the clay is laid in layers around the bed and banks, kept damp while it is worked, and kept covered or filled, because a dry clay lining cracks.

Stone or brick lining, and later concrete, were used where clay was not enough or where a bank needed strengthening (check).

COMMON MISTAKES

Believing the water in a canal flows along it like a river (it is almost still, and any flow comes from a lock being used or a leak); picturing the canal bed as flat in cross-section (it is shaped, usually deeper in the middle); and assuming a leak is only a problem for the water bill (a leak can also wash away a bank, which is how many canal breaches begin, check).

Challenges

EXERCISE 1

A canal must gain 18 m of height. Option A goes straight over: 8 km long, with locks that each lift 3 m and take 20 minutes. Option B is a contour canal with no locks, 13 km long. Boats travel at 3 km/h. Find the time each way and say which is quicker; then name two other things you would weigh up before choosing.

View solution

EXERCISE 2

A pound is 4 km long, 8 m wide and 1.2 m deep. Assume seepage of 10 mm of depth per day if unlined, 2 mm per day if puddled. Find the volume lost per day each way, the days for the level to fall 0.15 m with no top-up, and how many 21 m x 2.1 m x 2.4 m lock chambers the daily leak would fill.

View solution

EXERCISE 3

Short answer. (a) What is a pound, and why must the water in one stay level? (b) Give two reasons a builder might choose a contour route. (c) Name two ways to keep water in a canal bed, and say what puddling solves that bare soil does not.

View solution

Summary

KEY POINTS

- **Pound:** a level stretch of canal between locks
- **Contour canal:** follows a hillside at one height: longer, but no locks
- **Leakage:** a small percentage a day adds up, and must be replaced from a water supply
- **Puddled clay:** worked wet clay that cuts seepage many times over bare earth; it must stay damp
- **Trade-off:** distance against locks, digging against leaks

COMING NEXT

How Canals and Locks Work 3 looks inside a lock: the chamber, the gates and paddles, the fill and empty cycle, and why mitre gates close themselves tighter under pressure.

How Canals and Locks Work

Chapter 3 · Inside a Lock

Chapter 2 ended with a pound that must stay level, and with the awkward fact that the ground does not. Where the canal has to step up or down, a **lock** does the job: a watertight box with a gate at each end, in which a boat can be raised or lowered from one pound to the next. This chapter opens the box and follows one boat through it.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The Python snippets and the three exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 2, the dimensions, lift, paddle sizes and discharge coefficient are *round illustrative assumptions*, not measurements from a particular lock. Historical facts quoted from memory are marked (check).

The Parts of a Lock

A lock has four main parts. The **chamber** is the walled space the boat floats in. The **top gates** (also called head gates or upper gates) hold back the upper pound, and the **bottom gates** (tail gates) hold back the lower one. To move water without opening a gate against the full weight of the water, each end has **paddles**: sliding or hinged openings (sluices) that let water through. On many narrow canals in Britain the top paddles are built into the gates themselves, or into culverts (underground channels) beside the chamber that carry water in and out (check).

The idea is simple and needs no pump. Water always settles to one level, so opening a path between the chamber and the upper pound makes the chamber level with the upper pound, and opening a path to the lower pound makes it level with the lower one. The boat just rides up or down on the surface. The only energy used is the water itself running downhill, one chamber-full at a time.

One Trip Up the Lock

A boat going *up* arrives at a lock that is ready for it, with the chamber empty (level with the lower pound) and the bottom gates open. The steps below are in order, and the order matters: you never open a gate while there is a big difference in level across it.

```
steps = [
    ("Boat enters empty chamber", "bottom gates open", "all paddles shut"),
    ("Close bottom gates", "bottom gates shut", "all paddles shut"),
    ("Open top paddles", "top gates shut", "top paddles open"),
    ("Chamber fills; levels equalise", "top gates shut", "top paddles open"),
    ("Open top gates, boat leaves", "top gates open", "top paddles open"),
]
for n, what, gates, paddles in enumerate(steps, 1):
    print(n, what, "|", gates, "|", paddles)
# Output:
# 1 Boat enters empty chamber | bottom gates open | all paddles shut
# 2 Close bottom gates | bottom gates shut | all paddles shut
# 3 Open top paddles | top gates shut | top paddles open
# 4 Chamber fills; levels equalise | top gates shut | top paddles open
# 5 Open top gates, boat leaves | top gates open | top paddles open
```

(In the real thing a few extra steps hide in these five: the bottom paddles are checked shut before the top ones are opened, and the paddles are closed again after the boat leaves, ready for the next user. Real locks also differ in detail from place to place, check.) Going *down* is the mirror image: the chamber starts full, the top gates are closed behind the boat, the *bottom* paddles are opened to let the water out, and the bottom gates open when the levels match.

How Long Does It Take to Fill?

Water coming through a paddle is driven by the height difference, called the **head**. At the start the head is the full lift and the water rushes in; as the chamber fills, the head shrinks and the flow slows, much like a bath draining. Treating the paddles as simple holes with a discharge coefficient, the time to fill comes out as `2 × A × √H / (Cd × a × √(2g))`, where A is the chamber's floor area, H the lift, a the total paddle opening and Cd allows for friction and turbulence. It is a simplification (real culverts and paddles behave less neatly), but it gives a sensible size of answer.

```
import math
length, width, lift = 21.0, 2.1, 2.4       # m, an assumed narrow lock
chamber_area = length * width
```

```
volume = chamber_area * lift
print("chamber volume (m3):", round(volume, 1))
cd, paddle_area, g = 0.6, 0.15, 9.81    # all assumed
t = 2 * chamber_area * math.sqrt(lift) / (cd * paddle_area * math.sqrt(2 * g))
print("fill time (s):", round(t), "=", round(t / 60, 1), "minutes")
print("mean flow (m3/s):", round(volume / t, 2))
# Output:
# chamber volume (m3): 105.8
# fill time (s): 343 = 5.7 minutes
# mean flow (m3/s): 0.31
```

That chamber of about 106 cubic metres is the same one Chapter 2 used to measure leaks. About six minutes is a believable figure (check), and notice why locks use paddles in *two* stages on many canals: opening them slowly stops the boat being thrown about, and opening them fully at once would also churn the water and strain the gates. Every passage up uses one full chamber of water from the upper pound, a point How Canals and Locks Work 4 picks up when it looks at where the water comes from.

Mitre Gates: Closing Themselves Tighter

Most canal gates are not one flat door but **two leaves** that meet in a shallow V pointing *upstream*, towards the higher water. This is a **mitre gate** (the word is borrowed from the corner joint in a picture frame). It is easy to see why it works once you picture the push of the water.

When the upper pound is higher, the water pushes the V of the closed gate downstream. The two leaves lean on each other at the point and press together, so the harder the water pushes, the tighter the gate seals: the force is carried as *compression* along each leaf into the lock walls, rather than trying to bend the gate. When the levels equalise the push disappears, and the gate swings open easily. That is also why a gate can be opened only when the level on both sides is nearly the same: there is no way to push it against a head of water.

```
import math
rho, g = 1000, 9.81
h, w = 2.4, 2.1                  # water depth and lock width (m), assumed
force = 0.5 * rho * g * h**2 * w
print("total push on a closed gate pair (kN):", round(force / 1000, 1))
print("in tonnes-force:", round(force / 9810, 1))
```

```
for deg in (90, 30, 18):
    leaf = (force / 2) / math.sin(math.radians(deg)) if deg < 90 else force / 2
    print("leaf angle", deg, "deg: thrust along each leaf (kN):", round(leaf / 1000, 1))
# Output:
# total push on a closed gate pair (kN): 59.3
# in tonnes-force: 6.0
# leaf angle 90 deg: thrust along each leaf (kN): 29.7
# leaf angle 30 deg: thrust along each leaf (kN): 59.3
# leaf angle 18 deg: thrust along each leaf (kN): 96.0
```

About six tonnes of force pushes on a pair of gates holding back 2.4 m of water, from a simple pressure sum (water pressure grows with depth, so the average pressure is half the bottom value). The three angle lines compare designs. A single flat gate (90 degrees, each half simply taking half the load) has to be strong enough to resist bending. With the V, the leaves are in compression, and the shallower the V, the bigger the thrust in each leaf, so the lock walls and the gate hinges must take more. Real mitre gates are therefore a compromise, with a modest angle (the point of the V is often set a fraction of the lock width downstream, check) and a good seal at the point and at the sill, the step at the base of the gate. The sum treats the leaf as a straight strut and ignores friction and the weight of the gate, so treat it as a size guide only.

FEATURE	MITRE GATES	SINGLE FLAT GATE
Closing action	two leaves, tightening under pressure	one door
Load on gate	mainly compression along the leaves	bending across the whole door
Weight and strength	each leaf lighter and shorter	one heavy, wide door
Opening	swing open when levels match	swing, lift or roll, also level only
Where seen	most British canal locks, including narrow ones (check)	some guillotine or vertical-lift gates and small lock cuts (check)

COMMON MISTAKES

Thinking the boat is lifted by a pump (it floats up as the water comes in); thinking the gates open *to* fill the chamber (the paddles fill it, and the gate stays shut until the levels match); picturing the V of a mitre gate pointing downstream (it points upstream, so the water pushes the leaves together); and forgetting that a gate left open or a paddle left up can drain a pound for the next boat to find.

Challenges

EXERCISE 1

Sketch the five steps for a boat going *down* through a lock, starting with the chamber full. For each step, say what the boat does and whether the top gates, bottom gates and paddles are open or shut. Then write the same list in Python and run it.

View solution

EXERCISE 2

A wide lock has a chamber 22 m long and 3.6 m wide with a lift of 3.0 m, and a total paddle opening of 0.20 m2 (take Cd = 0.6). Find the chamber volume and the fill time. If boats pass every fill time plus 10 minutes for 14 hours a day, how many passages are there and how much water does the lock use in a day?

View solution

EXERCISE 3

Short answer. (a) Why can a gate not be opened while the levels on either side are different? (b) In which direction does the V of a mitre gate point, and why does the gate seal better as the water pushes harder? (c) What do paddles do that the gates themselves cannot?

View solution

Summary

KEY POINTS

- **Lock:** a chamber with a gate at each end, used to move a boat between two pounds
- **Paddles:** openings that let water in or out, so the gates need not be opened under load
- **Cycle:** close one end, open the paddles at the other, wait for the levels to match, open the gates
- **Fill time:** a few minutes; flow slows as the head falls

- **Mitre gates:** a V pointing upstream, tightening as the water pushes; the force is carried as compression to the walls

COMING NEXT

How Canals and Locks Work 4 asks where all this water comes from: reservoirs, feeders and summit levels, pumping, and the side ponds that save water at every passage.

How Canals and Locks Work

Chapter 4 · Water Supply

Chapter 3 ended with a bill: every boat that climbs a lock takes one chamber-full of water from the pound above and passes it to the pound below. Do that forty times a day and the upper pound will soon be empty unless something refills it. A canal that climbs over a hill is therefore only as good as the water that can be brought to its highest point. This chapter follows that water: where it is collected, how it is delivered, how it is lifted when it will not run uphill, and how clever lock design can make the same water do the job twice.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The Python snippets and the three exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 3, the lock size, number of passages, feeder flow, evaporation and leakage figures are *round illustrative assumptions*, not measurements from a particular canal. Historical facts quoted from memory are marked (check).

The Summit Is the Problem

A canal that crosses high ground rises through locks to a **summit level**, a long pound at the top, and then falls through more locks on the far side. Water naturally runs downhill, so once it is in the summit it can flow down both sides, through the locks, to the lower pounds. But the summit is also the part of the canal that is *highest*, so no river or stream can be tapped to fill it from above. Whatever the summit loses, to locks, leaks and evaporation, has to be brought in from somewhere else, usually from streams, springs or reservoirs on higher ground nearby.

That is why early canal engineers worried about water supply before almost anything else. An elegant route that could not be kept full was useless, and several canals of the canal age were crippled or abandoned for lack of water (check). The answer was a mixture of three things: **reservoirs** to store water in wet weather, **feeders** to carry it to the canal, and, when nothing else would do, **pumps**.

Reservoirs and Feeders

A **reservoir** is a store of water, usually made by building an earth dam across a valley. It fills in winter, when rain is plentiful, and is drawn down in summer, when the canal is busy and the weather is dry. A **feeder** is a channel (sometimes a pipe or a stream diverted into a ditch) that carries water from the reservoir, or straight from a river, into the canal, often at the summit. On the Llangollen Canal in Wales, for instance, a feeder takes water from the River Dee at Horseshoe Falls near Llangollen and the water then flows along the whole canal (check). On the Grand Union Canal's climb over the Chilterns, a group of reservoirs near Tring supplied the summit (check).

Because the canal is level between locks, a canal feeder can enter at the top and that single supply serves everything below it, passing downhill through the locks. This one-way flow is a useful feature: the lower pounds only need extra water if they lose more than they receive from above. In practice a long canal often has several feeders along its length, as the lower stretches gather stream water of their own.

A Water Budget

Planning a canal's water means comparing what goes out with what comes in. The big outgoing items are the water used by locks, the leakage Chapter 2 measured, and evaporation from the open surface. The incoming items are the feeders and the streams that run into the canal. Here is a first sum with the narrow lock of Chapter 3, forty boats a day, and a feeder carrying 0.02 cubic metres per second.

```
length, width, lift = 21.0, 2.1, 2.4       # m, the narrow lock of Chapter 3 (assumed)
volume = length * width * lift
print("water per passage (m3):", round(volume, 1))
passages = 40
daily = passages * volume
print("passages per day:", passages)
print("water used per day (m3):", round(daily))
feeder = 0.02                               # m3/s, assumed
supply = feeder * 86400
print("feeder supply per day (m3):", round(supply))
print("passages the feeder can support:", int(supply // volume))
# Output:
# water per passage (m3): 105.8
# passages per day: 40
# water used per day (m3): 4234
```

```
# feeder supply per day (m3): 1728
# passages the feeder can support: 16
```

A steady trickle of 0.02 cubic metres a second (20 litres a second, about a bath every four seconds) sounds like a lot, but over a day it brings in only about 1,700 cubic metres, enough for 16 passages. Forty passages need nearly two and a half times as much, so on a busy day the summit would drop, and the gap would have to come from a reservoir. Notice that the feeder never stops working while the boats do: at night, when no boats pass, the feeder keeps topping up the canal, and the surplus runs over the **weirs** (overflow edges built into the bank) or is stored for the morning.

Sizing a Reservoir

A reservoir has to cover the whole season's shortfall, not just one day's. The next sum treats the lock water, evaporation and leakage over a 270-day boating season, and asks how large a reservoir would have to be to hold it all with a 3 m draw-down. It is a crude upper bound, since in reality rain and feeders contribute for most of the season, but it shows the scale of the problem.

```
volume_per_pass = 105.8
passes_per_day = 40
season_days = 270
need = volume_per_pass * passes_per_day * season_days
print("lock water per season (m3):", round(need))
evap_loss = 0.005 * 20000 * season_days
print("evaporation, 20000 m2 surface at 5 mm/day (m3):", round(evap_loss))
leak = 0.01 * 86400 * season_days
print("leakage at 0.01 m3/s (m3):", round(leak))
total = need + evap_loss + leak
print("total demand (m3):", round(total))
depth = 3.0
res_area = total / depth
print("reservoir area for one full draw-down of 3 m (m2):", round(res_area))
print("= hectares:", round(res_area / 10000, 1))
# Output:
# lock water per season (m3): 1142640
# evaporation, 20000 m2 surface at 5 mm/day (m3): 27000
```

```
  # leakage at 0.01 m3/s (m3): 233280
  # total demand (m3): 1402920
  # reservoir area for one full draw-down of 3 m (m2): 467640
  # = hectares: 46.8
```

The locks dominate: more than four-fifths of the total demand goes through the chambers, and evaporation is almost a rounding error. That is a useful rule of thumb on a busy lock canal (check): *save the lock water and you have saved most of the water.* A 47-hectare reservoir (roughly 65 football pitches) is large but not impossible, which fits the history: canal companies did build big reservoirs, and some of them still sit in the landscape today, now used for fishing, sailing or wildlife.

When Water Will Not Run Uphill: Pumping

Gravity feeders only work when there is a source higher than the summit. When there is not, the options are to pump water up, or to **back-pump**: recycle the water that has already gone down the locks by lifting it from a lower pound back to the summit. Many canals built or extended pumping stations for exactly this purpose, driven first by steam, beam engines, then later by electric motors (check). The Crofton pumping station on the Kennet and Avon Canal, with its steam beam engines that lift water to the summit pound, is a well-known preserved example (check). Today many canals use electric pumps that can be switched on when the water level drops.

Side Ponds: Using the Same Water Twice

Here is the neatest idea of the chapter. When a full chamber is emptied, all of its water drops into the lower pound and is, as far as the summit is concerned, gone. But that water was lifted (by earlier boats, or by the feeders) to a height that is useful, so throwing it all away seems wasteful. A **side pond** is a small reservoir beside the lock, connected to the chamber by a paddle. When the lock is emptied, the first part of the water runs sideways into the side pond instead of down to the lower pound. When the lock is filled, the side pond gives that water back first, and only the remainder has to come from the pound above.

How much is saved? With one side pond with the same area as the chamber, a tidy sum shows that one third of the water is saved. With more ponds at different heights, the saving rises, but

with diminishing returns. The program below does not trust a formula: it simulates the filling and emptying cycle over and over until the levels settle, and reports the top-up needed from the upper pound.

```
def upper_draw(n, lift=2.4, cycles=200):
    # every pond has the same area as the chamber, so levels are all we need
    ponds = [lift * (n - i) / (n + 1) for i in range(n)]   # starting guess, top pond fi
    for _ in range(cycles):
        level = lift                        # chamber full: the boat goes down
        for i in range(n):                  # drain into ponds, highest first
            level = (level + ponds[i]) / 2
            ponds[i] = level
        level = 0.0                         # the rest goes to the lower pound
        for i in reversed(range(n)):        # fill from ponds, lowest first
            level = (level + ponds[i]) / 2
            ponds[i] = level
        draw = lift - level                 # topped up from the upper pound
    return draw
for n in range(4):
    d = upper_draw(n)
    print(n, "side ponds: top-up needed =", round(d, 2), "m of chamber depth, saving", r
# Output:
# 0 side ponds: top-up needed = 2.4 m of chamber depth, saving 0 %
# 1 side ponds: top-up needed = 1.6 m of chamber depth, saving 33 %
# 2 side ponds: top-up needed = 1.2 m of chamber depth, saving 50 %
# 3 side ponds: top-up needed = 0.96 m of chamber depth, saving 60 %
```

The pattern is that *n* equal ponds save *n*/(*n* + 2) of the water: a third, a half, three fifths. Each extra pond helps a little less than the one before, and each one costs land, paddles and a more complicated cycle for the lock keeper, so most locks that have side ponds have only one to three (check). On a deep lock the saving is worth having; on the shallow locks of a typical narrow canal the extra equipment seldom pays, which is why side ponds are more often found on large or busy waterways than on a rural English cut (check). The sum assumes the pond is level with the chamber when connected and ignores the small losses to friction and slow paddles, so it is an ideal figure.

METHOD	WHAT IT DOES	GOOD FOR	DRAWBACK
Reservoir	stores wet-season water for the dry season	large, steady supply	needs land, a dam and upkeep
Feeder	carries water from a river or reservoir to the canal	summits within reach of higher ground	depends on rainfall upstream
Pumping	lifts water from a lower pound back to the summit	canals with no higher source	needs energy and engineers
Side ponds	reuse part of each chamber of water	deep or busy locks	extra paddles, land and operating steps
Paired boats	two boats share one chamber of water	any lock, at no cost	needs two boats at once (check)

COMMON MISTAKES

Thinking a canal is filled from the bottom (it is filled from the top, and the water flows down); thinking evaporation is the main loss (on a busy canal it is the locks); forgetting that a boat going *down* costs a chamber of water too (the full chamber it empties into the lower pound must be refilled from above); and believing that side ponds return *all* the water (they recover a fraction, and only an ideal pond gets a third).

Challenges

EXERCISE 1

A wide lock has a chamber 22 m long and 3.6 m wide with a lift of 3.0 m. Find the water used per passage. A summit pound has 30 passages a day and a feeder carrying 0.03 m3/s. How many passages can the feeder support, how large is the daily shortfall, and how much must a reservoir supply over a 100-day dry spell?

View solution

EXERCISE 2

The same wide lock is fitted with two side ponds, each of the chamber's area. Using the simulation in this chapter, find the water taken from the summit per passage with 0, 1 and 2 ponds, then the water

saved per day at 30 passages. Why would the saving per extra pond keep getting smaller?

View solution

EXERCISE 3

Short answer. (a) Why is the summit the hardest part of a canal to keep full? (b) What does back-pumping do, and when would an engineer choose it over a new reservoir? (c) Why do locks, rather than evaporation, usually dominate the water budget of a busy canal?

View solution

Summary

KEY POINTS

- **Summit level:** the highest pound, which cannot be filled from above and so needs a supply of its own
- **Reservoirs and feeders:** store wet-season water and carry it to the summit
- **Pumping:** lifts water up when no higher source exists, or recycles water from the lower pounds
- **Lock water dominates:** a chamber-full per passage is far more than evaporation or leaks on a busy canal
- **Side ponds:** n equal ponds save n/(n + 2) of the water in the ideal case

COMING NEXT

How Canals and Locks Work 5 looks at what happens when one lock is not enough: lock flights, the staircase locks where the chambers share gates, and how a boat is worked up or down them efficiently.

How Canals and Locks Work

Chapter 5 · Flights and Staircases

Chapter 4 showed that a canal climbing over a hill pays for every metre of height with water, so engineers saved water wherever they could. But there is a second cost of height that has nothing to do with water: it is the sheer number of locks. One lock lifts a boat by two or three metres. A canal that must climb thirty metres to reach its summit needs a dozen locks, and sometimes they have to be built close together, one after another, up a steep hillside. This chapter looks at the two ways of doing that: the **lock flight**, where each lock has a short pound of its own, and the **staircase**, where the chambers are built directly on top of one another and share their gates. It also looks at how to work them efficiently, because a flight is where a good boater (and a good lock keeper) saves hours.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The Python snippets and the three exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 4, the lock size, operating times and water figures are *round illustrative assumptions*, not measurements from a particular canal. Historical facts quoted from memory are marked (check).

Flights: A Lock, a Short Pound, Another Lock

In a **lock flight** the locks follow one another closely, but between each pair there is a short stretch of water, a small pound, with its own banks and often a wide basin to the side. The boat leaves one lock, crosses the pound, and enters the next. Because each pound is a real stretch of water, a flight behaves like a staircase built from ordinary parts: every lock is a complete lock with a top gate and a bottom gate, exactly as in Chapter 3, and every pound holds its level as in Chapter 2.

The side pounds that stand next to the canal at many flights are a handy extra. They are wide areas that give the water a place to spread out when a lock is emptied into the pound below, so the level in the pound does not rise and fall by much and the lock above does not run it dry. Without them a short pound between two locks would swing up and down by a visible amount every time a neighbour was worked (check).

The great advantage of a flight is that *several boats can use it at once*. While one boat is climbing the third lock, another can already be in the second and a third in the first, each lock doing its job in turn. Boats going the other way can pass in the short pounds, which are wide enough, or in the wide basins beside them. The rule of thumb for the boater is to share locks with boats coming the other way whenever possible (check).

Staircases: Gates That Do Two Jobs

In a **staircase lock** there is no pound between the chambers at all. The top gate of the bottom chamber is also the bottom gate of the chamber above it. A boat leaving one chamber is already in the next. This is the most compact arrangement possible, which is why staircases are used on very steep ground, where there simply is not room for pounds, and where the cost of building and digging many separate locks would be high. Because a gate between two chambers carries water on both sides at different levels, the gate is rarely opened: it is only opened when the two levels are equal.

The price is that the staircase is worked in a particular order. **Each chamber is filled from the one above it, and emptied into the one below**. There is no pound to take up the slack, so a chamber that is filled when its neighbour has no water to give will not fill, and a chamber that is emptied into a neighbour that is already full will overflow. This is why a staircase usually has a notice, and often a lock keeper, to explain which paddles go first. At Foxton Locks in Leicestershire, the rule is a rhyme about the colours painted on the paddle gear: "red before white, you're all right; white before red, you're dead", the paddles being either red or white (check). The rule is a reminder that the order of operations is what keeps the water in the right place.

Simulating a Staircase

The idea can be tested with a simple model of a three-chamber staircase. Call the lifts one unit each, so that the bottom chamber is full at level 1, the middle one at level 2 and the top one at level 3, and call the chambers A (bottom), B and C (top). Each chamber is limited to its own range: A can hold between 0 and 1, B between 1 and 2, C between 2 and 3. All chambers are assumed to have the same area, so when a paddle between two of them is open the water settles at the average of their two levels. The program opens the paddles in the order for a boat going up, and flags any chamber that would overflow.

```
BAND = {"A": (0, 1), "B": (1, 2), "C": (2, 3)}   # water level range of each chamber, in
FIXED = {"bottom": 0, "top": 3}                   # the two pounds never change level
level = {"A": 0, "B": 2, "C": 3}                  # ready for a boat going up

def paddle(x, y):
    if x in FIXED:
        new = FIXED[x]
    elif y in FIXED:
        new = FIXED[y]
    else:
        new = (level[x] + level[y]) / 2
    for name in (x, y):
        if name not in FIXED:
            level[name] = new
    bad = [m for m in (x, y) if m not in FIXED and not BAND[m][0] <= new <= BAND[m][1]]
    state = "A=%s B=%s C=%s" % (level["A"], level["B"], level["C"])
    print("paddle", x, "-", y, "->", state, "OVERFLOW in " + bad[0] if bad else "")

print("a boat going up")
paddle("B", "A")        # A fills from B, boat moves up into B
paddle("C", "B")        # B fills from C, boat moves up into C
paddle("top", "C")      # C fills from the top pound, boat leaves
paddle("A", "bottom")   # empty A, ready for the next boat
print("a mistake: paddle B-A with A already full")
level["A"] = 1
paddle("B", "A")
# Output:
# a boat going up
# paddle B - A -> A=1.0 B=1.0 C=3
# paddle C - B -> A=1.0 B=2.0 C=2.0
# paddle top - C -> A=1.0 B=2.0 C=3
# paddle A - bottom -> A=0 B=2.0 C=3
# a mistake: paddle B-A with A already full
# paddle B - A -> A=1.5 B=1.5 C=3 OVERFLOW in A
```

Read it as a story. A boat enters the empty chamber A. Opening the paddle between B and A pours B's water into A: both settle at level 1, A is full and B has dropped to the bottom of its range. The boat moves up into B. Opening the paddle between C and B raises B to 2, which is B's full level, and lowers C to 2. The boat moves up into C, and a paddle from the top pound raises C back to 3. The boat leaves. A last paddle empties A into the lower pound, and notice what is

left: A empty, B full, C full, which is exactly the state in which the staircase began. A staircase worked this way is always left ready for the next boat going up.

The mistake at the end of the output is the one the Foxton rhyme warns against. If the paddle between B and A is opened while A is still full, the two chambers settle at 1.5, which is above the top of A's range: the water would run over the gate and flood the towpath (check). In real life it is a matter of knowing which paddles belong to which chamber and keeping them in order.

Two more facts follow from the same model. A boat going *down* needs the staircase in a different state: A empty, B at the bottom of its range (level 1) and C full. If B is still full from the last boat that went up, opening C into B would push B up to 2.5, again out of range, so a staircase worked for opposite directions needs one boat to go all the way through, and the chambers to be reset, before the other can start. This is the main weakness of a pure staircase, and it is why the most famous large staircases tend to have a passing place somewhere in the middle (check). And, as in a flight, each passage takes one chamber of water from the pound at the top and sends it to the pound at the bottom. The staircase does not use more water than a flight of the same height; it just uses the same water in a different arrangement.

Working Them Efficiently: Time

Which is faster, a flight or a staircase? It depends on how many boats there are. The next program compares them with assumed timings: two minutes to enter, one to close the gates, six to fill, one to open, two to leave, and six to empty the lock for the next boat. Crossing a short pound costs three minutes. A flight of three locks can have one boat per lock at once; a staircase is tied up by a single boat until it is through.

```
n = 3                                  # locks in the flight, or chambers in the stairc
enter, shut, fill, open_, leave = 2, 1, 6, 1, 2      # minutes (assumed)
between = 3                            # minutes to cross a short pound in a flight
reset = 6                              # minutes to empty a lock or chamber for the nex
lock_time = enter + shut + fill + open_ + leave
flight_one = n * lock_time + (n - 1) * between
flight_busy = lock_time + reset        # time one lock is tied up per boat
stair_one = n * lock_time              # no pounds between chambers
stair_busy = stair_one + reset         # the whole staircase is tied up per boat
print("one lock takes (min):", lock_time)
print("boats | flight (min) | staircase (min)")
for k in range(1, 5):
```

```
    flight = flight_one + (k - 1) * flight_busy
    stair = stair_one + (k - 1) * stair_busy
    print(k, "    |", flight, "         |", stair)
# Output:
# one lock takes (min): 12
# boats | flight (min) | staircase (min)
# 1     | 42          | 36
# 2     | 60          | 78
# 3     | 78          | 120
# 4     | 96          | 162
```

For a single boat the staircase wins, by 36 minutes against 42, because there are no pounds to cross. But the flight overtakes it with the second boat and runs away from it afterwards: four boats take 96 minutes up the flight and 162 up the staircase. The explanation is a pipeline, the same idea as an assembly line: in the flight, boat 2 can be working the first lock while boat 1 is in the second. In the staircase, boat 2 must wait. This is why heavily used canals prefer flights wherever the land allows, and why staircases are kept for places where a flight cannot fit (check).

Working Them Efficiently: Water

The other saving is water. Each time a lock is emptied for a boat to go down, the full chamber goes to the pound below. If a boat is coming the other way, the lock is already *full* when the first boat leaves it, and the second boat can enter at once without refilling it. The water used for the first boat's climb serves the second boat's descent. The program below counts how many chambers must be taken from the pound above for different mixes of up and down boats.

```
def fills(pattern):
    high = False                # the lock starts empty (low)
    from_above = 0
    for boat in pattern:
        if boat == "U":
            if high:
                high = False    # empty it to the lower pound first
            from_above += 1     # fill from the pound above
            high = True         # the boat leaves with the lock full
        else:
            if not high:
```

```
                from_above += 1  # fill it so the boat can enter at the top
                high = True
            high = False         # empty it to the lower pound, boat leaves
    return from_above

for p in ["UUUU", "UUDD", "UDUD"]:
    f = fills(p)
    print(p, "chambers from above:", f, "per boat:", f / len(p))
# Output:
# UUUU chambers from above: 4 per boat: 1.0
# UUDD chambers from above: 3 per boat: 0.75
# UDUD chambers from above: 2 per boat: 0.5
```

Four boats going the same way take a full chamber each, but four boats that alternate take only two chambers between them, *half a chamber per boat*. The order **UUDD** (two boats up, then two boats down) saves a quarter. The saving is the same one that Chapter 4 listed as "paired boats", now explained: it comes for free, and the lock keepers on busy flights try to arrange it. It is also why the standard advice to boaters on a canal is to leave a lock ready, with the gates closed and the paddles down, for a boat that is coming the other way (check).

Famous Flights

The flights that survive today are a tour of canal engineering:

- **Caen Hill, Devizes, Wiltshire.** A long run of 29 locks on the Kennet and Avon Canal, 16 of them in a single straight flight with large side pounds, lifting the canal about 72 metres (check).
- **Foxton Locks, Leicestershire.** Ten locks in two staircases of five, with a passing pound in the middle, on the Grand Union Canal's Leicester line, opened in 1814 (check).
- **Bingley Five Rise, West Yorkshire.** A staircase of five chambers on the Leeds and Liverpool Canal, lifting boats about 18 metres, finished in 1774 (check).
- **Hatton Locks, Warwickshire.** A flight of 21 locks on the Grand Union Canal, nicknamed the "Stairway to Heaven" (check).
- **Neptune's Staircase, Fort William, Scotland.** Eight chambers on the Caledonian Canal, opened in 1822 (check).

- **Fonserannes, Béziers, France.** A flight of locks on the Canal du Midi in the south of France, built in the 17th century (check).

FEATURE	LOCK FLIGHT	STAIRCASE LOCK
Pounds between locks	yes, short ones, often with side pounds	none: gates are shared
Boats in it at once	several, one per lock	one direction at a time
Space needed	more, for the pounds	very little
Water per passage	one chamber from the summit	one chamber from the summit
Operating risk	lower: a pound absorbs mistakes	higher: wrong order overflows a chamber
Best for	busy routes with room	steep, cramped slopes

COMMON MISTAKES

Thinking a staircase needs more water than a flight (the same one chamber per passage); thinking it is always slower (it is faster for one boat, slower for a queue); opening a staircase paddle out of order and expecting the water to find its own level (it will, but in the wrong place); and assuming that a lock must always be emptied after use (if a boat is waiting the other way, leave it full).

Challenges

EXERCISE 1

Plan the order of operations for one boat going up a flight of three narrow locks with a short pound between each pair, using the timings of this chapter. List the clock time at which the boat enters and leaves each lock, when each lock is ready again after it has been emptied, and how much water the summit pound loses for one passage.

View solution

EXERCISE 2

A busy lock is used by eight boats in one of three patterns: four up then four down, two up then two down repeated, and strict alternation. Using the narrow lock volume, find the water per boat for each pattern, and say which one a lock keeper would prefer and why.

View solution

EXERCISE 3

Short answer. (a) Name two differences between a flight and a staircase. (b) Why does a staircase usually let boats through in only one direction at a time? (c) Why can a boater save water by leaving a lock full when a boat is coming the other way?

View solution

Summary

KEY POINTS

- **Flight:** locks with short pounds between them, so several boats can be at work at once
- **Staircase:** chambers share gates and each is filled from the one above, so the order of paddles matters
- **Time:** a staircase is quicker for one boat, a flight is far quicker for a queue
- **Water:** one chamber per passage in either, but alternating up and down boats saves up to half
- **Famous flights:** Caen Hill, Foxton, Bingley Five Rise, Hatton and Neptune's Staircase (check)

COMING NEXT

How Canals and Locks Work 6 looks at what happens when even a flight is too slow or too steep: boat lifts, inclined planes and rotating lifts, and when a lift beats a lock.

How Canals and Locks Work

Chapter 6 · Lifts and Inclines

Chapter 5 ended with a flight of locks that lifts a boat up a hillside one chamber at a time, and the numbers were sobering: seven narrow locks for a rise of fifteen metres, well over an hour and a half for a single boat. Past a certain height a lock flight stops being a clever answer and becomes a chore. Engineers have always had a second idea in reserve, which is to *stop using water to do the lifting* and simply pick the boat up. This chapter looks at the three ways of doing it: the **vertical boat lift**, the **inclined plane** and the **rotating lift**, and ends by asking when a lift really does beat a lock.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The three Python snippets and the exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 5, the caisson sizes, masses and timings are *round illustrative assumptions*, not measurements from a particular lift. Historical facts quoted from memory are marked (check).

The Trick That Makes Lifts Possible

A boat lift raises a boat by carrying it inside a huge water-filled steel trough, called a **caisson** (sometimes a tank or a cradle). The boat floats in the trough as it would in any canal, the trough is lifted, and at the top the end gate is opened and the boat sails out. It sounds as if the machinery must lift the weight of the water *and* the weight of the boat, which would be enormous. It does not, and the reason is Archimedes' principle, which you met in Chapter 1: a floating boat pushes aside its own weight of water. If the trough is filled to a fixed level, then when the boat sails in, exactly its own weight of water spills out. The trough weighs the same with a boat afloat in it as without.

```
length, width, depth = 30.0, 5.0, 3.0      # water in the caisson, metres (assumed)
rho = 1000                                 # kg per m3 of water
boat = 18000                               # kg, a loaded narrowboat (assumed)
water_empty = length * width * depth * rho
displaced = boat / rho                     # m3 of water the floating boat pushes aside
water_with_boat = water_empty - displaced * rho   # level held fixed, so that water leav
```

```
print("caisson water, no boat (kg):", water_empty)
print("caisson water, boat afloat (kg):", water_with_boat)
print("total, no boat (kg):", water_empty)
print("total, boat afloat (kg):", water_with_boat + boat)
# Output:
# caisson water, no boat (kg): 450000.0
# caisson water, boat afloat (kg): 432000.0
# total, no boat (kg): 450000.0
# total, boat afloat (kg): 450000.0
```

The output shows the point. A caisson of 450 tonnes of water holds only 432 tonnes once an 18 tonne boat is afloat in it, because 18 cubic metres of water have been pushed out, and the boat itself makes up the 18 tonnes again. The total is 450 tonnes either way. That is a gift to the designer, because it means a lift can be *balanced*: two identical caissons hang from the two ends of the same system, one going up as the other comes down, and whatever boats are in them, friction and the small differences that remain.

Three Kinds of Lift

Vertical lifts. The caisson goes straight up and down. The Anderton Boat Lift in Cheshire, opened in 1875, raised boats about 15 metres from the River Weaver to the Trent and Mersey Canal. It was first worked by water pressure, with two caissons each sitting on a hydraulic ram connected to the other by a pipe, so that adding a little extra water to the upper one made it sink and push the other up. It was converted to electric power in 1908, closed for safety reasons in 1983 and reopened after restoration in 2002 (check). The lift at Strepy-Thieu in Belgium, opened in 2002, raises boats about 73 metres in a single movement (check). The Three Gorges ship lift in China, which lifts vessels by more than a hundred metres, is among the largest ever built (check).

Inclined planes. Here the caisson is a wheeled tank running up and down a railway-like slope, towed by cables and balanced by a second tank on the same line. The Foxton Inclined Plane in Leicestershire, opened in 1900, was built to bypass the famous Foxton staircase (Chapter 5) with a pair of tanks that lifted boats about 23 metres on rails, but it was closed in 1911, partly because it was costly to run (check). Ronquieres in Belgium, opened in 1968, lifts boats about 68 metres, and a large inclined plane at Krasnoyarsk in Russia, opened in 1983, lifts ships about 100

metres (check). The long slope makes inclined planes best for large vertical rises where a site allows a long approach.

The rotating lift. This is the strangest and most elegant. The Falkirk Wheel in Scotland, opened in 2002, joins the Forth and Clyde Canal to the Union Canal, a rise of about 24 metres. Two caissons hang at opposite ends of a giant rotating arm like the two pans of a balance. As the arm turns through half a circle, one caisson goes up as the other goes down, and a set of gear wheels keeps each caisson level throughout, so the water never slops and the boat stays flat (check). Because the arm is balanced so closely, the wheel needs only about 1.5 kilowatt-hours of electricity for a turn, about as much as boiling a few kettles of water (check).

How Much Energy Does a Balanced Lift Need?

The best way to appreciate the balance is to work out how much energy the lifting would cost if the lift were *not* balanced, and then see how that figure shrinks. Take a caisson of 300 tonnes in total, water and steel, lifted 24 metres, similar in size to the rotating lift above. The energy to raise a mass *m* through a height *h* is *m g h*, where g is about 9.81, and one kilowatt-hour is 3.6 million joules.

```
g = 9.81                                    # m/s2
mass = 300000                               # kg, one full caisson, water and steel (assu
height = 24                                 # m, lift of one rise (assumed)
joules = mass * g * height
print("energy to raise one caisson with no counterweight (kWh):", round(joules / 3.6e6,
for imbalance in (0.01, 0.02, 0.05):
    extra = imbalance * mass * g * height
    print("pair balanced to within", int(imbalance * 100), "%:", round(extra / 3.6e6, 2)
# Output:
# energy to raise one caisson with no counterweight (kWh): 19.6
# pair balanced to within 1 %: 0.2 kWh of lifting
# pair balanced to within 2 %: 0.39 kWh of lifting
# pair balanced to within 5 %: 0.98 kWh of lifting
```

Without any counterweight the lift would need about 19.6 kilowatt-hours, comparable to running a small electric heater for most of a day. Balanced to within five per cent it would need only about one kilowatt-hour, and balanced to within one per cent only about 0.2. That is why a

good lift is so cheap to run: the energy used is the *imbalance*, not the weight. Friction, the drive machinery and the gates add more, but the principle is clear.

Lift or Lock Flight?

Now the practical question. Suppose a canal has to rise 15.2 metres, roughly the Anderton height, and compare a flight of narrow locks with a lift. The lift takes ten minutes (two to enter, one to close the gate, four to travel, one to open, two to leave, all assumed) and, as already noted, takes no water from the upper canal beyond small leaks at the gates.

```
rise = 15.2                              # metres to climb (assumed)
lift_per_lock = 2.4                      # metres per narrow lock
lock_min, pound_min = 12, 3              # minutes, as in Chapter 5
chamber_m3 = round(21.0 * 2.1 * 2.4, 1)  # one narrow lock chamber
locks = -(-rise // lift_per_lock)        # round up
locks = int(locks)
flight_time = locks * lock_min + (locks - 1) * pound_min
lift_time = 2 + 1 + 4 + 1 + 2            # enter, close, travel, open, leave (assumed)
print("locks needed:", locks)
print("flight time for one boat (min):", flight_time)
print("lift time for one boat (min):", lift_time)
print("water drawn from the top per passage, flight (m3):", chamber_m3)
print("water drawn from the top per passage, lift (m3):", 0)
# Output:
# locks needed: 7
# flight time for one boat (min): 102
# lift time for one boat (min): 10
# water drawn from the top per passage, flight (m3): 105.8
# water drawn from the top per passage, lift (m3): 0
```

The same rise that takes seven locks, 102 minutes and a chamber of water from the summit takes the lift ten minutes and almost no water. The lift also moves *two boats at once* if two caissons are used, one climbing and one descending. Against that, a lift costs far more to build and maintain than a flight of lock chambers, which are masonry boxes with gates and paddles, and the moving parts of a lift (rams, cables, gears, seals) need regular inspection and eventual replacement (check). The comparison below sums up when a lift wins.

FEATURE	LOCK FLIGHT	BOAT LIFT OR INCLINE
Rise it suits	a few metres to about thirty	a large rise in one go
Time for one boat	long, many chambers	minutes
Water use	one chamber per passage from the summit	almost none
Capital cost	moderate, but grows with the rise	high
Running and repair	simple, local skills	specialist machinery and inspection
If it fails	one lock stops, others work	the whole route stops
Best where	water is plentiful and land is gentle	water is short, the rise is large, or space is cramped

COMMON MISTAKES

Thinking a lift must raise the weight of the boat as well as the water (a floating boat displaces its own weight, so it does not); thinking a lift costs more to run than a flight (balanced lifts use very little energy); assuming a lift uses no water at all (a little escapes at the gates); and assuming a lift is always the better answer (the cost, the single point of failure and the maintenance often keep a lock flight the sensible choice).

Challenges

EXERCISE 1

A canal must rise 12 metres. Using narrow locks of 2.4 metres lift and the timings of this chapter, find how many locks are needed and how long one boat takes up the flight. Then compare the number of boats per hour that the flight and a ten minute lift could pass, and say what the comparison leaves out.

View solution

EXERCISE 2

A pair of balanced caissons of 200 tonnes each lifts 12 metres, with the pair balanced to within 3 per cent. Work out the energy for a single lift of one caisson with no counterweight, and for the imbalance

case, in kilowatt-hours. Explain why the second figure is so much smaller.

View solution

EXERCISE 3

Short answer. (a) Why does a boat afloat in a caisson not make the caisson heavier? (b) Name the three kinds of lift described and give one site for each. (c) Give two reasons a canal builder might still choose a lock flight over a lift.

View solution

Summary

KEY POINTS

- **Archimedes:** a floating boat displaces its own weight, so a full caisson weighs the same with or without a boat
- **Balance:** paired caissons mean the motors handle only imbalance and friction, not the weight
- **Three kinds:** vertical lifts (Anderton), inclined planes (Foxton, Ronquieres) and rotating lifts (Falkirk Wheel) (check)
- **Against a flight:** faster and almost waterless, but costly to build and maintain, with a single point of failure
- **Rule:** choose a lift when the rise is large, water is short or land is cramped

COMING NEXT

How Canals and Locks Work 7 looks at a different obstacle: not height, but the lie of the land itself. It covers aqueducts that carry water over valleys, tunnels that carry it through hills, embankments, cuttings, stop planks and drainage.

How Canals and Locks Work

Chapter 7 · Crossing and Cutting

Chapter 6 dealt with a canal that has to go *up*. This chapter deals with a canal that has to go *across*. Because a canal must stay level, it cannot follow the ups and downs of the land the way a road can: a valley is in the way, a hill is in the way, a patch of soft ground is in the way, and a canal that stops at each obstacle is no use to anyone. The canal builder has four tools for this. An **aqueduct** carries the water over a valley, a **tunnel** carries it through a hill, an **embankment** builds up low ground, and a **cutting** digs down through a rise. A fifth, quieter set of tools (stop planks and drainage) keeps all of these working and makes them safe to repair.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The three Python snippets and the exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 6, the sizes, speeds and soil factors are *round illustrative assumptions*, not measurements from a particular canal. Historical facts quoted from memory are marked (check).

Aqueducts: A River of Water in the Air

An **aqueduct** is a bridge that carries a canal. The idea is old, since the Romans carried drinking water over valleys on arches, but a canal aqueduct has a harder job: it must hold a trough of water deep enough to float a loaded boat, and water is heavy. The first great canal aqueducts were of masonry, with the trough lined with puddled clay (Chapter 2) to keep it watertight. Brindley's aqueduct at Barton in Lancashire, which carried the Bridgewater Canal over the River Irwell from 1761, astonished people because boats now sailed above the river that had once carried their trade (check). Later engineers, especially Thomas Telford, replaced the clay-lined masonry trough with a trough of cast iron, which is both lighter and watertight, and could stand on tall, slender piers (check).

The most spectacular example is the Pontcysyllte Aqueduct in Wales, which carries the Llangollen Canal over the River Dee. It was opened in 1805, is about 307 metres long and stands about 38 metres above the river, with a cast iron trough on stone piers; it became a World Heritage Site in 2009 (check). The boat travels along a narrow trough with a towpath on one

side and nothing at all on the other, which gives boaters a view that is either thrilling or terrifying (check).

Now here is a calculation that ties this chapter to the previous one. How much does the trough weigh? Use round figures close to Pontcysyllte: a trough 300 metres long, 3.6 metres wide, holding water 1.5 metres deep, carried on 18 spans. One cubic metre of water weighs one tonne.

```
length, width, depth = 300.0, 3.6, 1.5     # trough, metres (assumed round figures)
spans = 18                                  # number of spans carried on piers (assumed)
water_t = length * width * depth            # tonnes (1 m3 of water is 1 tonne)
boat_t = 18                                 # tonnes, a loaded narrowboat (assumed)
print("water in the trough (tonnes):", water_t)
print("per metre of trough (tonnes):", water_t / length)
print("per span (tonnes):", round(water_t / spans, 1))
print("boat on top of it (tonnes added):", 0)
print("a boat's weight as a share of one span's water (%):", round(100 * boat_t / (water
# Output:
# water in the trough (tonnes): 1620.0
# per metre of trough (tonnes): 5.4
# per span (tonnes): 90.0
# boat on top of it (tonnes added): 0
# a boat's weight as a share of one span's water (%): 20.0
```

The water alone is 1,620 tonnes, or 5.4 tonnes for every metre of trough and 90 tonnes for each span. A narrowboat weighing 18 tonnes would add only a fifth to the water in one span if it were simply laid on top, and yet the output says **boat on top of it (tonnes added): 0**. This is Chapter 6's Archimedes again: a floating boat pushes aside its own weight of water, and the aqueduct is always full to the brim, so as the boat enters, its weight of water leaves. A trough weighs the same with a crowd of boats in it as with none. This is why the engineers could design an aqueduct for the weight of the water alone, and why a canal aqueduct can be as light and elegant as a road bridge (check).

Embankments and Cuttings: Cut and Fill

Where a valley is too low or too wide for an aqueduct, the canal can be carried on an **embankment**, a long raised bank of earth with the canal sitting along its top. Where the land rises, the canal can be dug through it as a **cutting**. Both are earthworks, and a skilled canal engineer tries to *balance* them: the soil dug out of a cutting is carried a short distance and used to build the nearby embankment, so that nothing has to be brought in or taken away. This is called **cut and fill**.

The sums are simple but they have a trap. When soil is dug it loosens and takes up more space than it did in the ground, by perhaps a quarter (a figure called the *swell factor*, assumed here). The program below takes a 400 metre cutting 3 metres deep, with a flat bottom 8 metres wide and sloping sides, and an embankment of 2 metres built over the same distance, and compares the soil they need.

```
def section(bottom, depth, slope):          # trapezoid: slope is horizontal metres per m
    return depth * (bottom + slope * depth)

length = 400                                # metres of canal along the hillside
cut = section(8, 3, 1.5) * length           # a cutting 3 m deep (assumed)
swell = 1.25                                # dug soil takes up about 25 % more space (a
loose = cut * swell
fill_section = section(8, 2, 2.0)           # an embankment 2 m high, built over the sam
fill = fill_section * length
print("cutting, in place (m3):", cut)
print("cutting, once dug and loose (m3):", loose)
print("embankment needs, loose (m3):", round(fill * swell))
print("spare soil (m3, negative = shortfall):", round(loose - fill * swell))
# Output:
# cutting, in place (m3): 15000.0
# cutting, once dug and loose (m3): 18750.0
# embankment needs, loose (m3): 12000
# spare soil (m3, negative = shortfall): 6750
```

The cutting yields 15,000 cubic metres of soil in the ground and 18,750 once it has been dug and loosened. The embankment needs 12,000 once loosened. There is a surplus of 6,750 cubic metres, which can be spread over the surrounding land or used to widen the embankment. Had the numbers come out the other way, soil would have had to be carried in, a costly and slow job, and the engineer would have changed the line or the levels to bring the figures back into

balance (check). The most dangerous part of an embankment is the water it holds. A small leak can wash out the soil, and a breach lets the whole length of canal run down the bank. The remedy is the puddled clay core described in Chapter 2, along with careful monitoring of the bank (check).

Tunnels: Through the Hill

When the hill is too high for a cutting, the canal goes through it. A **tunnel** is expensive and slow to build, because it was dug by hand with picks, gunpowder and a great many ventilation shafts, so a canal builder would use one only when going round or over was impossible. Early tunnels were narrow, with no towpath, because every extra metre of width multiplied the digging. A horse could not pull the boat, so the horse was led over the top while the crew pushed the boat through by **legging**: lying on their backs on boards on the side of the boat and walking the tunnel walls with their feet. Later tunnels were wider and carried towpaths, and from the nineteenth century steam tugs (and then diesel tugs) towed boats through the long tunnels (check).

The Standedge Tunnel on the Huddersfield Narrow Canal in Yorkshire is the longest, highest and deepest canal tunnel in Britain: about 5,000 metres long, opened in 1811 (check). Take a tunnel of exactly 5 kilometres with no passing places, and suppose boats are legged through at 1.5 kilometres an hour. A single boat takes 200 minutes. But if boats go in as a convoy, following each other ten minutes apart, the cost per boat drops quickly:

```
tunnel_km = 5.0                          # one narrow tunnel with no passing places (
speed = 1.5                              # km per hour by legging (assumed)
gap = 10                                 # minutes between boats in the same convoy
passage = tunnel_km / speed * 60         # minutes for one boat to get through
print("one boat takes (min):", round(passage))
print("convoy size | minutes for one convoy | minutes per boat")
for n in (1, 2, 4, 6):
    t = passage + (n - 1) * gap          # last boat leaves this long after the first
    print(n, "          |", round(t), "                    |", round(t / n))
# Output:
# one boat takes (min): 200
# convoy size | minutes for one convoy | minutes per boat
# 1           | 200                    | 200
# 2           | 210                    | 105
# 4           | 230                    | 58
```

```
# 6          | 250                  | 42
```

A convoy of six boats takes 250 minutes in all, which is only 42 minutes per boat, against the 200 minutes that a lone boat has to spend. This is the reason that long narrow tunnels usually have a timetable, with boats sent through in groups, one direction at a time (check). The tunnel keeper's job is to make sure that two boats never meet head on in the dark.

Stop Planks and Drainage

The last tools are the unglamorous ones. A canal that cannot be emptied cannot be repaired, and a canal that cannot shed floodwater will eventually overflow. **Stop planks** are heavy timber planks that slide into vertical grooves cut in the walls of a bridge hole or a narrow part of the canal, built up one on another like a wall. With the planks in place, a length of canal can be sealed off and drained for repair, and if an embankment bursts, the planks seal the break on either side to limit the loss (check). Many canals have grooves for stop planks at regular intervals, and stacks of planks kept beside them in a hut (check).

Drainage works the other way: it gets water out. A heavy rain can pour a lot of water into a canal from the surrounding land, and a canal bank is not designed to overflow. **Overflow weirs** are low points built into the side of a pound, where any water above the normal level spills away into a stream or drain. A **sluice** or **culvert** carries a stream under the canal so that it does not flood the canal, or discharges unwanted water below it. And if a stretch has to be emptied on purpose, a **drain plug** or a **drain valve** in the bed of the canal does the job (check). These parts, like the paddles in Chapter 3, are simple, but they are the reason a canal stays safe in a storm.

FEATURE	AQUEDUCT	EMBANKMENT	TUNNEL	CUTTING
Obstacle	valley or river	low ground	high ground	moderate rise
What carries the water	a trough on piers	a raised bank of earth	a lined bore	the natural ground, dug down
Main cost	stone, iron, tall piers	earth moved and clay	digging, lining, ventilation	digging and spoil
Main risk	leaks in the trough	breach of the bank	collapse, meeting boats in the dark	slips of the sides

FEATURE	AQUEDUCT	EMBANKMENT	TUNNEL	CUTTING
Typical aid	clay or cast iron trough	puddled clay core	legging or tug, timetable	drainage ditches

COMMON MISTAKES

Thinking an aqueduct has to carry the weight of the boats as well as the water (a floating boat displaces its own weight, so the trough weighs the same with or without it); forgetting the swell factor when balancing cut and fill (loose soil takes up more room than it did in the ground); thinking a tunnel always needs a towpath (many narrow ones had none, and boats were legged through); and assuming stop planks hold back flowing water forever (they are a temporary seal for repair, not a permanent gate).

Challenges

EXERCISE 1

An aqueduct trough is 150 metres long, 3.0 metres wide, holds water 1.2 metres deep, and is carried on 10 spans. Work out the weight of the water in the whole trough, per metre and per span, and say how much weight a loaded 20 tonne boat adds when it floats in.

View solution

EXERCISE 2

A cutting 250 metres long and 2.5 metres deep has a flat bottom 6 metres wide and sides sloping 1.5 metres across for every metre of depth. The nearby embankment is 150 metres long, 1.5 metres high, with a 6 metre top and sides sloping 2 metres across for every metre of height. With a swell factor of 1.2, find the surplus or shortfall of soil.

View solution

EXERCISE 3

Short answer. (a) Why can an aqueduct be built for the weight of the water alone? (b) Why do long narrow tunnels have timetables? (c) What is the job of a stop plank, and of an overflow weir?

View solution

Summary

KEY POINTS

- **Aqueduct:** a trough on piers; it weighs the same with a boat afloat in it as without, by Archimedes
- **Embankment and cutting:** balanced as cut and fill, remembering that dug soil swells
- **Tunnel:** narrow and costly, often legged, with timetables so that convoys run one way at a time
- **Stop planks:** slide into grooves to seal a section for repair or to limit a breach
- **Drainage:** overflow weirs, culverts and drain valves keep floodwater under control

COMING NEXT

How Canals and Locks Work 8 brings the course up to date: ship canals and sea locks, leisure boating, restoration, maintenance and the future of waterways.

How Canals and Locks Work

Chapter 8 · Canals Today

Chapter 7 showed how a canal crosses the land: over valleys, through hills, along banks and cuttings. This last chapter asks what has become of all that work. Most of the working canals of the eighteenth and nineteenth centuries lost their cargoes to railways and lorries, and many were abandoned. Yet the story did not end there. Some canals grew into vast **ship canals** that carry the world's trade, some were rescued by volunteers and reborn as places for **leisure boating**, and all of them depend on steady **maintenance**. We finish by looking at what a waterway is good for in the twenty-first century.

> **WHAT WAS RUN, AND WHAT TO TRUST**
>
> The two Python snippets and the exercise solutions were executed and the output shown is exactly what Python printed. As in How Canals and Locks Work 7, sizes, costs and cargo weights are *round illustrative assumptions*, not measurements of a particular canal. Historical and current facts quoted from memory are marked (check).

Ship Canals and Sea Locks

A **ship canal** is a canal built for seagoing vessels instead of barges. The principles are exactly those of Chapters 1 to 7, scaled up: a level pound, a lock at each change of level, gates, valves and a reliable supply of water. The Suez Canal in Egypt, opened in 1869, joins the Mediterranean to the Red Sea; because the two seas are at nearly the same level it needs no locks at all (check). The Panama Canal, opened in 1914, has to climb over the isthmus, and so uses a series of enormous locks that lift ships up to an artificial lake and down again (check). In Britain, the Manchester Ship Canal, opened in 1894, carries sea-going ships some 36 miles inland to Manchester through a series of locks (check).

A **sea lock** is a different animal again. It sits where a canal meets the tidal sea, and it has to cope with a water level on one side that rises and falls twice a day. The gates may have to hold back water from either direction, so many sea locks have pairs of gates facing both ways (check). Compare the water a sea lock uses with a narrow lock. Here, a sea lock chamber 200 metres by 25 metres is lifted by 6 metres, against a narrow lock 21 metres by 2.1 metres lifting 2.4 metres:

```
narrow = 21.0 * 2.1 * 2.4          # narrow lock: length x width x lift, m3 (assumed)
sea = 200.0 * 25.0 * 6.0           # sea lock chamber: length x width x tide difference,
print("narrow lock fill (m3):", round(narrow, 1))
print("sea lock fill (m3):", round(sea))
print("sea lock = this many narrow locks:", round(sea / narrow))
tons_per_ship = 20000              # tonnes of cargo in one ship (assumed)
tons_per_boat = 0.5                # tonnes carried by a leisure boat (assumed)
print("sea lock water per tonne of cargo (m3):", round(sea / tons_per_ship, 2))
print("narrow lock water per tonne of cargo (m3):", round(narrow / tons_per_boat))
# Output:
# narrow lock fill (m3): 105.8
# sea lock fill (m3): 30000
# sea lock = this many narrow locks: 283
# sea lock water per tonne of cargo (m3): 1.5
# narrow lock water per tonne of cargo (m3): 212
```

The sea lock needs about 30,000 cubic metres of water, as much as the fill of roughly 283 narrow locks. At first glance that looks wasteful, but the last two lines tell the real story. Per tonne of cargo, the sea lock uses about 1.5 cubic metres, whereas the narrow lock uses over 200. A big ship moving a great deal of cargo makes much better use of its water, which is one reason the size of vessels has grown whenever canals have allowed it (check). It also connects back to Chapter 4: the water supply, not the lock itself, sets the limit on traffic.

Leisure Boating and Restoration

By the middle of the twentieth century many British canals had fallen out of use, with locks rotting, channels silted up and bridges replaced by culverts. The turning point came when enthusiasts and volunteers began campaigning to save them. The Inland Waterways Association was founded in 1946, and over the following decades volunteers and local councils reopened one canal after another for pleasure boats (check). Today the system carries holidaymakers, anglers, walkers, cyclists, and people who live on their boats, and the towpath that once carried a horse is now a green path through towns (check). A restored canal is also a valuable asset: it attracts visitors, supports small businesses, and is a corridor for wildlife (check).

Restoration is never cheap, and never all at once. A canal trust with a fixed budget must decide which jobs to do first. A sensible method is to rank jobs by how much benefit each gives per

pound spent, and to give safety work, like a leaking embankment, a high benefit score, since a breach is both dangerous and costly. The program below works through a fictional canal. The costs, in thousands of pounds, and the benefit scores out of 10 are all assumed.

```
jobs = [                          # (name, cost in thousands of pounds, benefit score 1
    ("repair leaking embankment", 400, 9),
    ("rebuild three locks", 900, 8),
    ("clear towpath and dredge", 150, 6),
    ("restore a visitor basin", 300, 5),
    ("replace a tunnel lining", 1200, 7),
]
budget = 1000                     # thousands of pounds available this year (assumed)
ranked = sorted(jobs, key=lambda j: j[2] / j[1], reverse=True)
spent = 0
for name, cost, benefit in ranked:
    ok = spent + cost <= budget
    if ok:
        spent += cost
    print(name, "| benefit per 100k:", round(100 * benefit / cost, 2), "|", "funded" if 
print("spent (thousands):", spent)
# Output:
# clear towpath and dredge | benefit per 100k: 4.0 | funded
# repair leaking embankment | benefit per 100k: 2.25 | funded
# restore a visitor basin | benefit per 100k: 1.67 | funded
# rebuild three locks | benefit per 100k: 0.89 | waits
# replace a tunnel lining | benefit per 100k: 0.58 | waits
# spent (thousands): 850
```

The cheap towpath and dredging job ranks first, the embankment repair second and the visitor basin third. Together they use 850 of the 1,000 available; the three locks would need 900, which does not fit in what is left, so they wait for next year, and so does the tunnel lining. Notice that the ranking does *not* say the locks are unimportant, only that, given this budget, spending it on cheaper jobs gives more total benefit. A real trust would also weigh what must be done to keep the whole route connected, since a canal with one blocked lock is a canal that cannot be used end to end (check).

Maintenance: The Work That Never Ends

A canal is never finished. Gates swell, shrink and rot, and are replaced on a cycle of a couple of decades (check). Silt settles in the channel and has to be dredged so that boats do not run aground. Banks erode, leaks open up, lock walls bulge, and every one of the structures of Chapters 3 to 7 needs inspection. Reservoirs are checked under reservoir safety law (check). Vegetation grows into the water, and invasive plants need control. Much of this is done in the winter, when boat traffic is low, during a programme of **winter stoppages**, when a lock or a length of canal is drained behind stop planks (Chapter 7) and the crews can work in the dry (check).

There is an old rule of thumb that small repairs done early are far cheaper than large repairs done late, which is why inspection matters as much as repair: a damp patch on an embankment is a cheap job today and an expensive emergency after a breach (check).

The Future of Waterways

What will canals be used for next? Several answers are already visible. They still carry some freight: barges move bulk goods on larger waterways and, in places, into cities, and a loaded barge uses far less fuel per tonne than a lorry (check). They offer **water transfer**, moving water from where there is plenty to where there is a shortage (check). They provide **green corridors** for wildlife and for walkers and cyclists. Canal water can also be used for heat: heat pumps can draw warmth from the water to heat nearby buildings (check). Boats are moving to electric and hybrid engines, which are quiet and clean, and charging points are appearing at moorings (check). And as the climate changes, engineers are asking how to manage more extreme rain and drought, with the same tools as before: reservoirs, feeders, overflow weirs and careful control of the water.

FEATURE	SHIP CANAL	WORKING NARROW CANAL (PAST)	LEISURE CANAL (TODAY)
Main traffic	sea-going ships	horse-drawn cargo boats	pleasure boats, walkers, anglers
Locks	huge chambers, or none at all	narrow, one boat at a time	the same narrow locks, restored
Water per tonne of cargo	low	high	not a cargo question

FEATURE	SHIP CANAL	WORKING NARROW CANAL (PAST)	LEISURE CANAL (TODAY)
Main challenge	keeping a big channel open and supplied	competing with road and rail	paying for maintenance and restoration
Typical aid	sea locks, dredging	towpath, horse, stables	volunteers, trusts, grants

COMMON MISTAKES

Thinking a ship canal always needs locks (a canal linking two seas of the same level needs none); judging a lock by the water it uses rather than the water per tonne of cargo; assuming a restored canal stays restored without regular maintenance; and assuming that a low rank in a benefit-per-pound list means a job is unimportant (it may only be too expensive for this year's budget).

Challenges

EXERCISE 1

A sea lock chamber is 150 metres long, 15 metres wide and lifts by 4 metres. How many cubic metres of water does one fill take, and how many narrow locks (21 metres by 2.1 metres, lift 2.4 metres) is that? If the ship carries 5,000 tonnes of cargo, how many cubic metres of water is that per tonne?

View solution

EXERCISE 2

A trust has 600 thousand pounds for a fictional disused canal. The jobs are A (cost 200, benefit 8), B (cost 500, benefit 9), C (cost 100, benefit 3) and D (cost 250, benefit 6). Rank them by benefit per cost, fund them in that order while they fit, and say what is funded and what is left over.

View solution

EXERCISE 3

Short answer. (a) Why does a sea lock often have gates that face both ways? (b) Why is a lock for a big ship more water-efficient per tonne of cargo than a narrow lock? (c) Name two things a canal can be used for today other than carrying boats.

View solution

Summary

KEY POINTS

- **Ship canals:** the same ideas as small canals, scaled up; some need locks, some none
- **Sea locks:** cope with tides, so gates may need to hold back water from both sides
- **Leisure and restoration:** volunteers and trusts brought canals back; money is limited, so jobs are ranked
- **Maintenance:** never finished; inspect early, repair in the winter stoppage
- **Future:** freight, water transfer, wildlife corridors, heat and electric boats

COURSE COMPLETE

Over eight chapters you have followed a canal from its first reason for existing to its modern uses: why water carries heavy loads (Chapter 1), how a level is held and kept watertight (Chapter 2), what happens inside a lock (Chapter 3), where the water comes from (Chapter 4), how locks are grouped into flights and staircases (Chapter 5), how lifts and inclines replace locks (Chapter 6), how canals cross and cut through the land (Chapter 7), and how they are used and cared for today. The best next step is practical: walk a stretch of towpath, watch a boat work through a lock, and name each part you see. Follow-on courses on related engineering topics may be added in future.