STEFFEN HAUG

Numerical Methods

Assignment 1

Problem I

Sphere *floating* in water, i. e. the sphere is still.

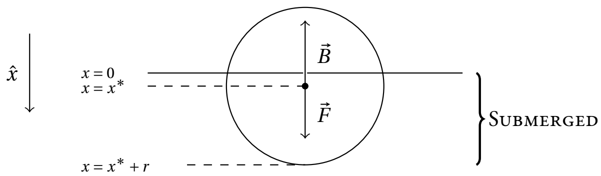

Figure 1: *Sphere in equilibrium.*

The sphere has a density of $\rho = 0.6$ times that of water. Archimedes' principle states that the buoyant force exerted og the sphere is equal (in magnitude) to the *weight* of the displaced fluid. Using density $\rho_{\text{fluid}} = 1$ for water, the weight of the displaced fluid is

$$-\vec{B} = m_{\text{fluid}} \cdot g = v_{\text{fluid}} \cdot g,$$

and we know that the weight of the sphere itself is

$$\vec{F} = m_{\text{sphere}} \cdot g = 0.6 \cdot v_{\text{sphere}} \cdot g.$$

At equilibrium, these balance out:

$$0 = \vec{F} - \vec{B} = g(0.6 v_{\text{sphere}} - v_{\text{fluid}})$$

Obviously, the volume of displaced fluid is equal to the volume of the part of the sphere that is submerged. We divide the equation by g, and thus

$$v_{\text{submerged}} = v_{\text{fluid}} = 0.6 \cdot v_{\text{sphere}} \tag{1}$$

we also know that

$$v_{\text{sphere}} = \frac{4\pi r^3}{3} \tag{2}$$

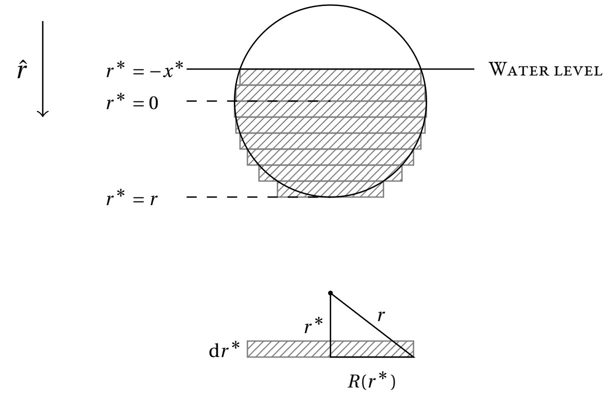

Figure 2: *Volume of the submerged part of the sphere.*

and with some analysis, we can work out the volume of the submerged part of the sphere. Our strategy is to integrate over cylinder-shaped volume elements.

Each volume element has a volume of $\pi \cdot R(r^*)^2 \cdot \mathrm{d}r^*$, and we need to integrate from water level and downward in a coordinate system on the sphere. Thus,

$$v_{\text{submerged}} = \pi \int_{r^*=-x^*}^{r^*=r} R(r^*)^2 \mathrm{d}r^* \tag{3}$$

where

$$R(r^*) = \sqrt{r^2 - (r^*)^2}.$$

We now have our mathematical model. It remains to find a solution x^* such that

$$I(x^*) := \frac{v_{\text{submerged}}}{v_{\text{sphere}}} = \frac{3}{4\pi r^3} \pi \int_{r^*=-x^*}^{r^*=r} R(r^*)^2 \mathrm{d}r^* = \rho = 0.6$$

Given an algorithm to compute definite integrals, this can probably already be solved quite easily, but we don't know any such algorithms yet, so our strategy is to evaluate the integral by hand, and try to create a root-finding problem.

$$
\begin{aligned}
I(x^*) &= \frac{3}{4r^3}\int_{r^*=-x^*}^{r^*=r} R(r^*)^2 \mathrm{d}r^* \\
&= \frac{3}{4r^3}\int_{r^*=-x^*}^{r^*=r} r^2 - (r^*)^2 \mathrm{d}r^* \\
&= \frac{3}{4r^3}\cdot \Big|_{r^*=-x^*}^{r^*=r} r^2 r^* - \frac{1}{3}(r^*)^3 \\
&= \frac{1}{4r^3}\left(3r^2(r+x^*) - r^3 - (x^*)^3\right)
\end{aligned}
$$

This is a good time to check that

$$I(-r) = 0, \quad I(0) = 1/2, \quad \text{and} \quad I(r) = 1$$

Which is our limit cases, where the sphere is either hovering above the fluid ($x^* = -r$), or is completely submerged ($x^* = r$). The $x^* = 0$ case corresponds to exactly half the sphere being submerged.

Listing 1: *Checking for a given radius*

```
def I(h, r):
    p = (3 * r**2 * (r + h) - r**3 - h**3)
    q = (4 * r**3)
    return p / q
```

```
>>> r = 17
>>> I(r, r)
1.0
>>> I(-r, r)
0.0
>>> I(0, r)
0.5
```

All that remains is to solve the equation

$$I(x^*) - \rho = 0; \quad 0 < \rho < 1,$$

where ρ is the density of the sphere. Our analysis reveals that $I(x^*)$ is a polynomial of degree three, which is good because its image is all of $\mathbf{R}$, so a root like this is guaranteed to exist. *Unfortunately, three roots exist.* We need to define a criteria to determine which solution is the correct one.

Obciously, the sphere will not hover in mid-air, and since we have assumed $\rho < 1$, it will not sink. Clearly then, all *physical* solutions must be in the interval $[-r, r]$. If we were to prove that the CONTRACTION MAPPING THEOREM applies to some function $g : [-r, r] \to [-r, r]$, then we would have a guarantee that the physical solution is unique, and we would have defined a procedure to approximate it in the proccess.

Applying the contraction mapping theorem

If we factor $I(x^*)$ in a particular manner, we see that

$$I(x^*) = \frac{x^*}{4r}\left(3 - \left(\frac{x^*}{r}\right)^2\right) + \frac{1}{2},$$

and that we may write the equation $I(x^*) - \rho = 0$ as

$$x^* = g(x^*) = -4r\left(\frac{1-2\rho}{2}\right)\left(3 - \left(\frac{x^*}{r}\right)^2\right)^{-1}.$$

Notice that $g(x^*)$ is continuous on $[-r, r]$. It does have singularities at $x^* = \pm r\sqrt{3}$, but that is outside the interval, hence the mean value theorem holds:

$$g(x) - g(y) = g'(m)(x - y),$$

for some $m \in [-r, r]$. To place a bound on $g'(m)$, we need to evaluate $\mathrm{d}g/\mathrm{d}x^*$ using the chain rule.

$$\frac{\mathrm{d}g}{\mathrm{d}x^*} = \underbrace{-4r\left(\frac{1-2\rho}{2}\right)}_{\text{constant}}\underbrace{\frac{\mathrm{d}}{\mathrm{d}x^*}\left(3-\left(\frac{x^*}{r}\right)^2\right)^{-1}}_{\gamma(x^*)}$$

$$\begin{aligned}\gamma &= -\left(3-\left(\frac{x^*}{r}\right)^2\right)^{-2}\frac{\mathrm{d}}{\mathrm{d}x^*} - \left(\frac{x^*}{r}\right)^2 \\ &= \left(3-\left(\frac{x^*}{r}\right)^2\right)^{-2} 2\left(\frac{x^*}{r}\right)\frac{\mathrm{d}}{\mathrm{d}x^*}\frac{x^*}{r} \\ &= \left(3-\left(\frac{x^*}{r}\right)^2\right)^{-2} 2\left(\frac{x^*}{r}\right)\frac{1}{r}\end{aligned}$$

We can rearrange this a little, and multiply by $1 = r^2 \cdot r^{-2}$:

$$\begin{aligned}\gamma &= 2x^* r^{-2}\left(3-\left(\frac{x^*}{r}\right)^2\right)^{-2} \cdot r^2 r^{-2} \\ &= 2x^* r^2\left(3r^2-\left(\frac{x^*}{r}\right)^2 r^2\right)^{-2} \\ &= 2x^* r^2\left(3r^2-\left(x^*\right)^2\right)^{-2}\end{aligned}$$

then

$$\begin{aligned}\frac{\mathrm{d}g}{\mathrm{d}x^*} &= -4r\left(\frac{1-2\rho}{2}\right)\gamma(x^*) \\ &= \underbrace{-8r^3\left(\frac{1-2\rho}{2}\right)x^*}_{\text{monotonically decreasing}}\overbrace{\left(3r^2-\left(x^*\right)^2\right)^{-2}}^{\text{strictly positive for } x^*\in[-r,r]}\end{aligned}$$

i. e., $g'(x^*)$ is itself monotonically decreasing. This is very fortunate, since it suffices to evaluate g' in the endpoints of $[-r, r]$.

$$\left.\frac{\mathrm{d}g}{\mathrm{d}x^*}\right|_{x^*=-r} = -\frac{1-2\rho}{2}\frac{8r^4}{(2r^2)^2} = 2\rho - 1$$

$$\left.\frac{\mathrm{d}g}{\mathrm{d}x^*}\right|_{x^*=r} = \frac{1-2\rho}{2}\frac{8r^4}{(2r^2)^2} = -(2\rho - 1)$$

This finally proves that

$$\max_{x^*\in[-r,r]}\left|\frac{\mathrm{d}g}{\mathrm{d}x^*}\right| = |2\rho - 1|,$$

and moreover, $|2\rho - 1| < 1$ by the assumption $0 < \rho < 1$. This works out exactly so that the assumptions of the contraction mapping theorem holds:

$$\begin{aligned}|g(x) - g(y)| &= |g'(m)||x - y| && \text{MEAN VALUE THEOREM} \\ &\le |2\rho - 1||x - y| \\ &< |x - y|\end{aligned}$$

In other words, the contraction mapping theorem holds with $L = 1$. Thus

$$\begin{cases} \{x_n\}_{n=0}^{\infty} \\ x_n = g\left(x_{n-1}\right) \\ x_0 \in [-r, r] \end{cases}$$

converges to the unique, physical solution, no matter the choice of x_0.

Problem 2

a. Want to use the Taylor series of f around the point a, $T_a\{f\}(x)$, to show that

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{f(x+\Delta x) - f(x)}{\Delta x} + \mathcal{O}(\Delta x),$$

in other words, that the error we expect if we compute

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x},$$

is proporional to Δx, i. e. the error goes to zero as Δx approaches zero.

Fix x and Δx. We expand the Taylor series to two terms, *also including the error term*, around the point x.

$$\begin{aligned} f(x^*) &= T_x\{f\}(x^*) \\ &= f(x) + f'(x)(x^* - x) + \mathrm{error}_2(x^*) \end{aligned}$$

Which yields

$$f'(x) = \frac{f(x^*) - f(x) - \mathrm{error}_2(x^*)}{x^* - x}$$

By evaluating the series at $x^* = x + \Delta x$

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{\mathrm{error}_2(x^*)}{\Delta x}$$

To show that the error term is $\mathcal{O}(\Delta x)$ is straight forward:

$$E(\Delta x) := \frac{\mathrm{error}_2(x^*)}{\Delta x} = \frac{1}{\Delta x}\frac{f''(\widetilde{x})}{2!}(x^* - x)^2$$

for some $\widetilde{x} \in [x, x^*]$. Again, we evaluate the error term at $x^* = x + \Delta x$.

$$\begin{aligned} &= \frac{f''(\widetilde{x})}{2\Delta x}(\Delta x)^2 \\ &= \frac{f''(\widetilde{x})}{2}\Delta x \end{aligned}$$

as long as f'' does not blow up on $I = [x, x + \Delta x]$. Because of the fact that I is bounded and closed, f'' attains its maximum and minimum on I. Thus we can let

$$L = \mathrm{diam}\, f(I) = \max_{x,y \in I} \left| f''(x) - f''(y) \right|$$

Which makes

$$|E(\Delta x)| \leq L\Delta x.$$

Testing the approximation

We want to compute the error commited by our formula for some function which we know how to differentiate symbolically. We want to compute

$$\left.\frac{\mathrm{d}}{\mathrm{d}x}\cos x\right|_{x=\pi/4}.$$

We know that the symbolic derivative of $\cos x$ is $-\sin x$.

Listing 2: *Approximate differentiation program*

```python
# src/alg.py
def ddx(f, dx):
    return lambda x: (f(x + dx) - f(x)) / dx
```

`ddx` returns a function of type `float -> float` that closes over `f` and `dx`. This new function in turn evaluates the derivative at a point `x`.

Listing 3: *Using the differentiation program*

```python
import src.alg as alg
import numpy as np
from math import sin, cos, pi

def errors(deltas):
    for dx in deltas:
        dcosdx = alg.ddx(cos, dx)
        approx = dcosdx(pi/4)
        exact  = -sin(pi/4)
        yield (abs(approx - exact), dx)

dxs = (1/2**k for k in range(30))
data = list(errors(dxs))
error, delta = zip(*data)
```

Plotting `error` against `delta` in a logarithmic coordinate system confirms that the error is indeed linear in Δx.

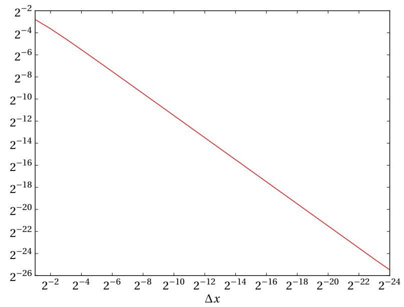

FIGURE 3: *The absolute error of the approximation as a function of Δx for $f(x) = \cos x$ and $x = \pi/4$*

b. We want to observe the propagation of numerical error when iterating with a transformation using floating-point arithmetic.

LISTING 4: *Iterating with* `sqrt`

```
from math import sqrt
import random as rand

def iter_sqrt(N):
    x  = rand.random() * 100
    xh = x

    for l in range(1, N):
        for k in range(1, l):
            xh = sqrt(xh)
        for k in range(1, l):
            xh = xh**2

        e  = abs(x - xh)
        er = abs(x - xh) / abs(x)

        yield (e, er, l)

data = list(iter_sqrt(N = 100))
es, ers, ls = zip(*data)
```

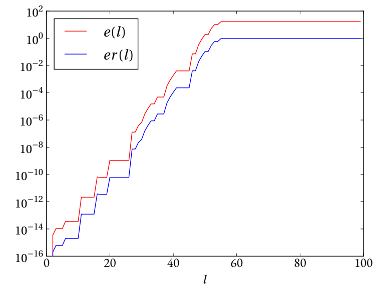

FIGURE 4: *The absolute and relative error committed by the algorithm.*

PROBLEM 3

LISTING 5: *Bisection and Newtons method*

```
MAX_ITER = 100

def bisect(f, I):
    a, b = I

    assert a < b
    assert f(a) * f(b) < 0

    for _ in range(MAX_ITER):
        m = 0.5 * (a + b)
        yield m
        if f(a) * f(m) < 0:
            b = m
        elif f(b) * f(m) < 0:
            a = m
        else: continue # then f(m) == 0


def fpi(g, x0, tol=10E-6):
    # g contraction, finds x = g(x) by iteration
    xn    = g(x0)  # x1
    xn_m1 = x0     # x0
    for _ in range(MAX_ITER):
        yield xn
        xn, xn_m1 = g(xn), xn


def newton(f, dfdx, x0, tol=10E-6):
    g = lambda x: x - f(x) / dfdx(x)
```

```
    yield from fpi(g, x0)

def pairs(it):
    # this technique is deprecated, but whatever
    # used to get pairwise consecutive elements from
    # an iterator
    x = next(it)
    y = next(it)
    while True:
        yield(x, y)
        x, y = y, next(it)
```

As you can see, we can implement Newtons method as a generic fixed-point iteration. I have opted for differentiating f symbolically and providing it to `newton`.

The functions are defined in a way, such that they yield consecutively better approximations, instead of returning the best approximation in the end. This is just because we want to illustrate the convergence. Moreover, we are interested in two types of convergence, which the following figures show.

Plotting the errors versus `i` and `j`, which are the iteration count, we see clearly that Newtons method converges quadratically, while bisection converges linearly.

LISTING 6: *Comparing the two methods*

```
def g(x): return x**5 - 4*x + 2
def dgdx(x): return 5*x**4 - 4

n = list(alg.newton(g, dgdx, 1))
b = list(alg.bisect(g, (1, 5)))

def abs_err(data):
    for xn, xn_p1 in alg.pairs(iter(data)):
        yield abs(xn - xn_p1)

i, err_abs_n = zip(*list(enumerate(abs_err(n))))
_, err_abs_b = zip(*list(enumerate(abs_err(b))))

def f_err(f, data):
    for xn in data:
        yield abs(f(xn))

j, err_f_n = zip(*list(enumerate(f_err(g, n))))
_, err_f_b = zip(*list(enumerate(f_err(g, b))))
```

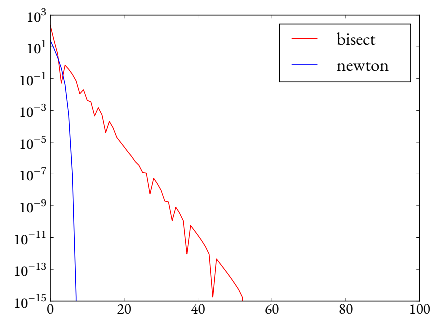

FIGURE 5: *Speed of convergence $|f(x_k)|$. This clearly illustrates that the midpoint moves back and forth, as the interval shrinks from different sides.*

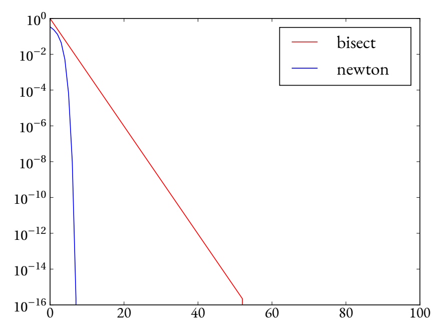

FIGURE 6: *Speed of convergence $|x_{k+1} - x_k|$.*