

Patrick E. Farrell
University of Oxford

2024–2025
February 27, 2025

Computational Mathematics Projects

Contents

A	<i>2025A: Convex hulls</i>	5
	A.1 <i>Introduction</i>	5
	A.2 <i>Convex hulls</i>	6
	A.3 <i>Two-dimensional algorithms</i>	8
	A.4 <i>Graham scan</i>	8
	A.5 <i>Divide-and-conquer</i>	10
	A.6 <i>Three dimensions</i>	12
	A.7 <i>Concluding remarks</i>	12
B	<i>2025B: Orbital elements</i>	15
	B.1 <i>Introduction</i>	15
	B.2 <i>Orbital elements</i>	16
	B.3 <i>Computing orbits from orbital elements</i>	18
	B.4 <i>Computing the coordinates of the body in its orbital plane</i>	20
	B.5 <i>Rotating these coordinates to the reference plane</i>	20
	B.6 <i>Comparing to Mars' orbit</i>	20
	B.7 <i>The return of Halley's comet</i>	21
	B.8 <i>Concluding remarks</i>	22
C	<i>2025C: Pension planning</i>	23
	C.1 <i>The model</i>	23
	C.2 <i>Computing approximate solutions</i>	24

C.3 *The stock and pension we will model* 25

C.4 *Planning for retirement* 26

C.5 *Concluding remarks* 27

Bibliography 29

A 2025A: Convex hulls

A.1 Introduction

(This project relates to material in the Trinity term Prelims course Geometry.)

Convexity is a fundamental concept in geometry, optimisation, and analysis.

Definition A.1.1 (Convex set). Let $V = \mathbb{R}^d$, $d \in \mathbb{N}_+$. A set $C \subset V$ is convex if the line segment joining any two points in C is itself a subset of C :

$$\forall p, q \in C \forall t \in [0, 1] \quad tp + (1 - t)q \in C. \quad (\text{A.1.1})$$

Examples of a convex set (left) and a nonconvex set (right) are drawn in Figure A.1.

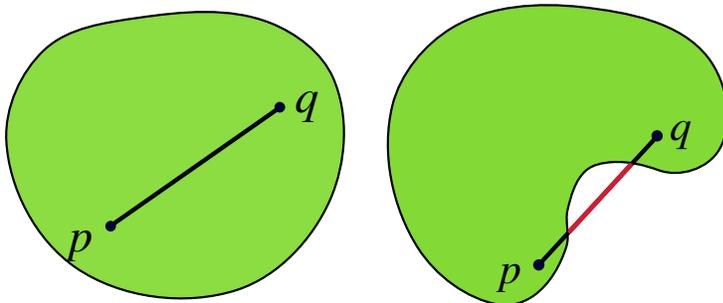


Figure A.1: Examples of a convex set (left) and a nonconvex set (right). Credit: modified from wikipedia ([convex](#), [non-convex](#)).

The concept of convexity was known to the ancient Greeks, with Archimedes giving the first formal definition of a convex curve in *On the Sphere and Cylinder* in c. 225 BC^{1,2}.

It is arguable that convexity was fundamental to the worldview of the ancient Greeks. Euclid's elements culminates in its final book, Book XIII, with Theaetetus of Athens' proof that there are only five regular³ convex three-dimensional polyhedra. In fact, some historians of mathematics argue that the construction of the five regular convex solids was the chief goal of their entire deductive system of geometry⁴. These solids are depicted in Figure A.2.

¹ Αρχιμήδης. *Περ σφαιρας κα κυλινδρου*. 225 B.C. Translation by R. Netz, Cambridge University Press

² In this book Archimedes discovered that a sphere has two-thirds the volume of its inscribing cylinder. He thought that this was his finest result; he requested a sphere and its cylinder be carved on his tombstone.

³ A regular polygon is a convex polygon with all edges equal and all corner angles equal; a regular polyhedron is one whose faces are all congruent regular polygons, with the same number around each vertex.

⁴ H. Weyl. *Symmetry*. Princeton University Press, 1952

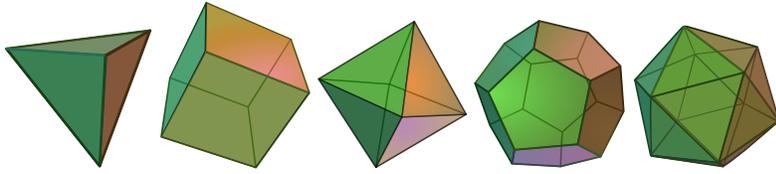


Figure A.2: The only five regular convex three-dimensional polyhedra. Credit: [wikipedia](#).

These five solids are often referred to as the Platonic solids. In his dialogue *Timaeus*, Plato associates each of the constituents of the physical universe with one of the solids⁵:

... four equilaterals form the sides of a regular solid, the tetrahedron or pyramid, which is the constituent particle of fire: eight such equilaterals are the sides of the octahedron, which is the particle of air; twenty equilaterals are the sides of the icosahedron, being the particle of water. ... six squares are the sides of a fourth regular solid called the cube, which is the particle proper to earth. A fifth regular solid still exists, namely the dodecahedron, which does not form the element of any substance; but God used it as a pattern for dividing the zodiac into its twelve signs.

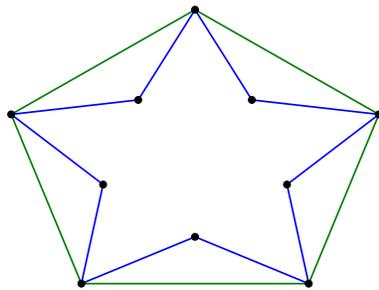
In this project we will investigate algorithms for computing the *convex hull* C of a given set S , the smallest convex set C that contains S as a subset.

⁵ Πλάτων. *Τιμαιοϛ*. Cambridge University Press, 1888. Translated by R. D. Archer-Hind, Fellow of Trinity College, Cambridge

A.2 Convex hulls

Definition A.2.1 (Convex hull). *Given $S \subset V$, its convex hull $C = \text{hull } S$ is the minimal convex set containing S .*

Convex hulls are uniquely defined. An example is depicted in Figure A.3: the convex hull of the blue shape is depicted in green. This figure was computed using `scipy`'s wrapping of the `qhull` library⁶ using the code in listing A.1⁷.



⁶ C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996

Figure A.3: The convex hull of the blue star shape is given by the green pentagon.

⁷ Feel free to use `scipy`'s implementation to verify the correctness of your own convex hull algorithms.

Code block A.1. Using scipy's quickhull algorithm.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial import ConvexHull

pts = np.array([[ 0.00000000e+00,  1.00000000e+00],
                [-2.93892626e-01,  4.04508497e-01],
                [-9.51056516e-01,  3.09016994e-01],
                [-4.75528258e-01, -1.54508497e-01],
                [-5.87785252e-01, -8.09016994e-01],
                [-9.18485099e-17, -5.00000000e-01],
                [ 5.87785252e-01, -8.09016994e-01],
                [ 4.75528258e-01, -1.54508497e-01],
                [ 9.51056516e-01,  3.09016994e-01],
                [ 2.93892626e-01,  4.04508497e-01]])

hull = ConvexHull(pts)

fig = plt.figure()
ax = fig.add_subplot(111)

# Loop over triangular faces of the convex hull
for s in hull.simplices:
    ax.plot(pts[s, 0], pts[s, 1], "g-")

# Plot points in order
N = pts.shape[0]
for i in range(N):
    xx = [pts[i, 0], pts[(i + 1) % N, 0]]
    yy = [pts[i, 1], pts[(i + 1) % N, 1]]
    ax.plot(xx, yy, 'b-')

# Plot defining corner points
ax.plot(pts[:, 0], pts[:, 1], "ko")
ax.axis('off')

```

There are several equivalent definitions of convex hulls. Another defines hull S to be the intersection of all convex sets containing S .

Convex hulls have many surprising properties. For example, the Gauss–Lucas theorem states that the roots of the derivative of a polynomial lie within the convex hull of the roots of the polynomial itself⁸; convex hulls also arise in the description of the state space in quantum

⁸ M. A. Brilleslyper and B. Schaubroeck. Explorations of the Gauss–Lucas theorem. *PRIMUS*, 27(8–9):766–777, 2016

mechanics (the Schrödinger–Hughston–Jozsa–Wootters theorem). Computing convex hulls is a fundamental task in computational geometry, and many different algorithms have been proposed, with different advantages in ease of implementation, generalisability to higher dimensions, and complexity.

Question A.1. Describe an algorithm for computing the convex hull of n points in one dimension ($d = 1$). What is the runtime complexity of this algorithm? (No coding is required for this question.)

A.3 Two-dimensional algorithms

In two dimensions, the convex hull of a set of $n > 2$ points is a convex polygon whose vertices are a subset of the given points. Thus, in two dimensions, the natural way to represent a convex hull is as an ordered sequence of vertices, traversing the boundary of the convex hull in counterclockwise order⁹. Since we can represent the convex hull with $O(n)$ storage, can we *compute* it with $O(n)$ work?

⁹ Going clockwise or counterclockwise is just a convention; this is the one we will use.

Question A.2. Given a set of natural numbers $S = \{n_1, n_2, \dots, n_N\} \subset \mathbb{N}$, and any algorithm for computing the convex hull of a set of two-dimensional points $P = \{(x_j, y_j) : j = 1, \dots, N\} \subset \mathbb{R}^2$, devise a means of sorting S using the convex hull algorithm. What conclusions can we draw about the minimal runtime complexity of computing convex hulls in two dimensions? (No coding is required for this question.)

[Hint: recall from lectures that sorting algorithms have a minimal runtime complexity of $O(n \log n)$, achieved by merge sort, among others.]

A.4 Graham scan

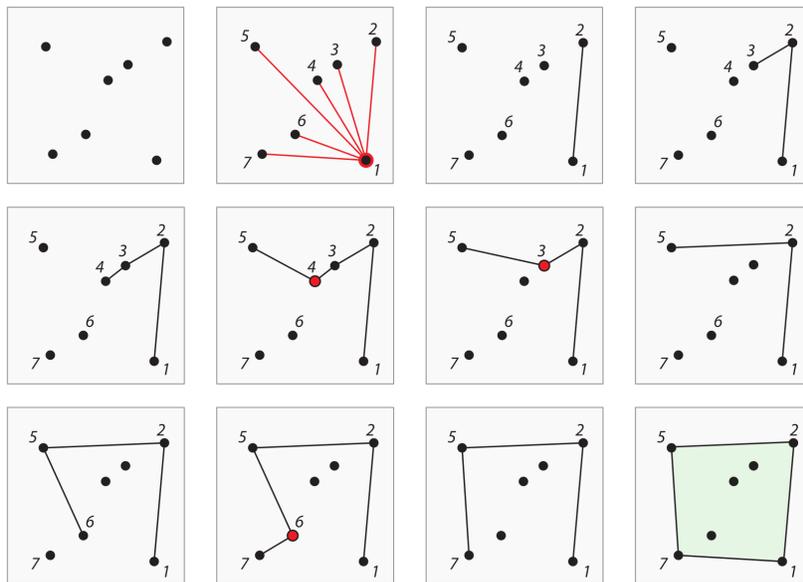
An efficient algorithm for computing convex hulls of point sets $P \subset \mathbb{R}^2$ was proposed by Graham in 1972¹⁰, and is now known as the *Graham scan*. Its steps are:

1. Find the point of P with lowest y coordinate, breaking ties by choosing the one with lowest x -coordinate. Label this point b (for base point).
2. For each other point $p \in P$, compute the angle made between the line bp and the positive x -axis. In addition, compute the squared distance $\|b - p\|^2$.

¹⁰ R.L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, 1972

3. Sort the points in P according to the angle made, from smallest to largest. Break ties by sorting the points in order of increasing distance. Henceforth the points are processed in this sorted order.
4. Initialise the convex hull with b and the first point in P .
5. Let xy denote the last constructed hull edge. For each point $c \in P$, we must determine whether xyz forms a *left turn* or a *right turn*. If xyz is a right turn, this means that y is not part of the convex hull and should be removed from further consideration. This discarding continues as long as the last three points form a right turn. If the points are colinear or form a left turn, we stop processing c .

This process is illustrated in Figure A.4, reproduced from the excellent textbook of Devadoss & O'Rourke¹¹.



¹¹ S. Devadoss and J. O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011

Figure A.4: Steps of the Graham scan. The points are sorted in order from the base point b (labelled 1). In the 6th panel, the three vertices 3-4-5 form a right turn, so 4 cannot be in the convex hull. In the 7th panel, the three vertices 2-3-5 also form a right turn, so 3 also cannot be in the convex hull.

The determination of whether xyz forms a left or right turn may be done with the cross product. Treat the vectors \vec{xy} and \vec{xc} as three-dimensional vectors with z -component zero; the z -component of $\vec{xy} \times \vec{xc}$ is zero for colinear points, positive for a left turn, and negative for a right turn.

Question A.3. Implement the Graham scan algorithm. Apply the algorithm to the points

$$(0,0), (1,0), (1,1), (0,1), (0.5,0.5), (0.25,0), (1,0.25), (0.75,1), (0,0.75).$$

Plot the given points (in black) and the computed convex hull (in blue).

[Hint: Python offers very powerful functionality for sorting lists by key functions. The key function is a callable that is applied to the list prior to making comparisons. If the key function returns a tuple, Python sorts first on the first component, and breaks ties with the second component.]

Question A.4. Conduct ten trials of the following experiment:

1. Sample 50 points in the unit square $[0, 1]^2$.
2. Compute the convex hull with Graham's scan.
3. Plot the original points (in black) and the computed convex hull (in blue).

Question A.5. Argue that the runtime complexity of Graham's scan algorithm is $O(n \log n)$. (No coding is required for this question.)

A.5 Divide-and-conquer

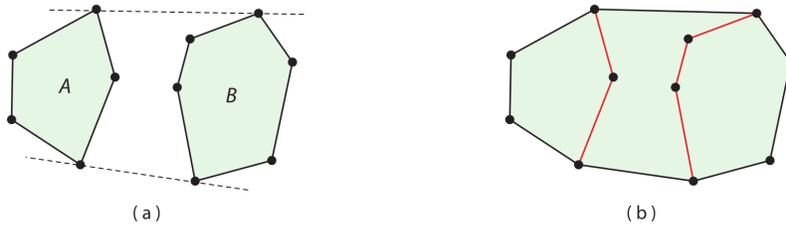
Graham's scan algorithm is very efficient, but it is inherently limited to two dimensions: there is no analogy to sorting the points by angle in three or higher dimensions. We therefore turn to another algorithm in two dimensions that *does* extend to higher dimensions: divide-and-conquer¹². This is a classic algorithm strategy that arises again and again in different problems.

The divide-and-conquer algorithm is a *recursive* algorithm: it calls itself on smaller problem instances. This recursive process terminates when the input is sufficiently small that computing the answer is trivial (e.g. when the algorithm is to compute the convex hull of two or three points). For given input $P \subset \mathbb{R}^d$ with $|P| = n$, its steps are:

1. Sort the points by x -coordinate (i.e. along the first dimension).
2. Divide the points along the x -axis into two nearly equal groups A and B , so that A and B are separated by a vertical line.
3. Compute the convex hulls of A and B recursively, with a direct construction when A or B are sufficiently small.
4. Merge the convex hulls hull A and hull B to construct hull P .

¹² F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977

The difficulty of this algorithm clearly lies in the merge step. To merge hull A and hull B , we aim to compute two *tangent lines* between the polygons, one containing the two convex hulls above it, and the other below it (Figure A.5, reproduced from Devadoss & O'Rourke¹³).



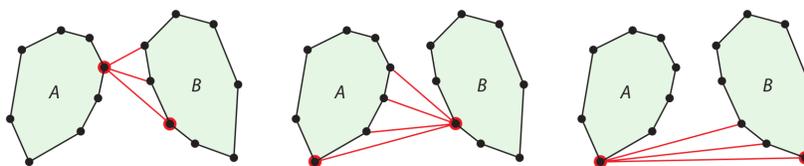
¹³ S. Devadoss and J. O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011

Figure A.5: The merge step of the divide-and-conquer algorithm. (a) Given the convex hulls of A and B , we must compute two tangent lines: one completely containing both convex hulls above it, and the other below. (b) With these tangent lines, it is straightforward to discard the interior vertices.

An efficient algorithm for this task in two and three dimensions was devised by Preparata & Hong. We discuss the two-dimensional algorithm for computing the lower tangent; the upper tangent is analogous. Its steps are:

1. Let α denote the rightmost point of A and β the leftmost point of B .
2. Fixing α , walk counterclockwise from β along the vertices of the convex hull of B . Continue doing this until all vertices on the hull of B are to one side of the line joining α to the vertex. Relabel this vertex β .
3. Fixing β , walk clockwise from α around A until a vertex is found with all vertices of hull A are to one side of the line joining β . Relabel this vertex α .
4. Repeat this process until the line joining α and β has all vertices of the hulls of A and B on the same side.

This algorithm is illustrated in Figure A.6, again reproduced from Devadoss & O'Rourke¹⁴.



¹⁴ S. Devadoss and J. O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011

Figure A.6: Computing a lower tangent. (a) Starting with α the rightmost point of A and β the leftmost point of B , walk counterclockwise along B until the line $\alpha\beta$ has all vertices of the hull of B above it. (b) Fixing this β , walk clockwise along A until the line $\alpha\beta$ has all vertices of the hull of A above it. (c) Iterate back and forth until the line has all vertices of both hulls above it.

Question A.6. Implement the divide-and-conquer algorithm for computing convex hulls in two dimensions. Care should be taken

so that the algorithm works robustly for all inputs. Demonstrate its correctness on suitably chosen inputs.

Question A.7. Analogous to question A.4, conduct ten trials of the following experiment:

1. Sample 50 points in the unit square $[0, 1]^2$.
2. Compute the convex hull with divide-and-conquer.
3. Plot the original points (in black) and the computed convex hull (in blue).

A.6 Three dimensions

While we will not implement algorithms for convex hulls in three dimensions, we will investigate their storage complexity.

In three dimensions the convex hull is a topologically two-dimensional mesh, comprising vertices, edges, and *faces* (see e.g. Figure A.2). More complicated data structures are therefore required to represent it¹⁵. In two dimensions, representing the convex hull of n vertices involved storing n edges, for $O(n)$ storage: does the same hold in three dimensions?

¹⁵ The most popular approach is to represent the convex hull as a *simplicial complex*, a powerful idea from algebraic topology.

Question A.8. The convex hull of a three-dimensional point set is a planar graph and hence satisfies Euler's famous formula

$$V - E + F = 2, \quad (\text{A.6.1})$$

where V is the number of vertices, E is the number of edges, and F is the number of faces. Using this, prove that

$$F < 2V \text{ and } E < 3V. \quad (\text{A.6.2})$$

This result means that we can represent the convex hull of a set of n three-dimensional points with $O(n)$ storage. (No coding is required for this question.)

A.7 Concluding remarks

The divide-and-conquer algorithm extends to three dimensions with optimal $O(n \log n)$ runtime complexity¹⁶, making it of substantial

¹⁶ Other algorithms have *expected* complexity $O(n \log n)$, but can take $O(n^2)$ in the worst case.

theoretical importance. However, the merge step is much more complicated than in two dimensions, and in practice it is not the pragmatic choice. Is there a *simple* algorithm for three-dimensional convex hulls with optimal complexity?

In higher dimensions the optimal possible complexity is $O(n \log n + n^{\lfloor d/2 \rfloor})$, with algorithms in general dimensions devised by Chazelle^{17,18}. These algorithms again appear to be of purely theoretical interest¹⁹.

¹⁷ B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(4):377–409, 1993

¹⁸ Chazelle's son later won fame, and an Oscar, as the director of *La La Land*.

¹⁹ R. Seidel. *Convex hull computations*, pages 361–375. CRC Press, Inc., 1997

B 2025B: Orbital elements

B.1 Introduction

(This project relates to material in Prelims Dynamics and Geometry.)

In 1609 Johannes Kepler published one of the most significant books in the history of science, the *Astronomia Nova*¹. Among his great insights was that the planets orbit in ellipses, not circles; he had initially tried to fit a circle to Mars' orbit, but observed it was about 0.13° off, and spent the next years resolving the discrepancy.

Geometers therefore paid great attention to the description of ellipses in three dimensions. Describing such an ellipse around the sun requires six parameters. These parameters are referred to as *orbital elements*. There are several different ways to describe such orbits, each with six parameters; in this project we will study the traditional 'Keplerian' orbital elements.

Working a century later, in *A Synopsis of the Astronomy of Comets*, Halley computed the orbital elements for several different observations of comets. His aim was to determine which observations were in fact the same comet. As he wrote²,

The principal Use therefore of this Table of the Elements of their Motions, and that which induced me to construct it, is, That whenever a new Comet shall appear, we may be able to know, by comparing together the Elements, whether it be any of those which has appear'd before, and consequently to determine its Period, and the Axis of its Orbit, and to foretell its Return. And, indeed, there are many Things which make me believe that the Comet which Apian observ'd in the Year 1531, was the same with that which Kepler and Longomontanus took Notice of and describ'd in the Year 1607, and which I my self have seen return, and observ'd in the Year 1682. All the Elements agree, and nothing seems to contradict this my Opinion, besides the Inequality of the Periodick Revolutions.

He had realised that the comet he had personally observed in 1682 was the same seen by Kepler in 1607 and by Petrus Apianus in 1531. Another observation of the comet in 1066 was recorded in the Bayeux tapestry.

In this question we shall use the orbital elements of Halley's comet

¹ J. Kepler. *Astronomia Nova: seu physica coelestis, tradita commentariis de motibus stellae Martis ex observationibus G.V. Tychois Brahe*. 1609. Translated by W. H. Donohue. Green Lion Press, 2015.

² E. Halley, Savilian Professor of Geometry at Oxford; And Fellow of the Royal Society. *A Synopsis of the Astronomy of Comets. Translated from the original, printed at Oxford*. Printed for John Senex, next to the Fleece-Tavern, in Cornhill, 1705

to ‘foretell its Return’.

B.2 Orbital elements

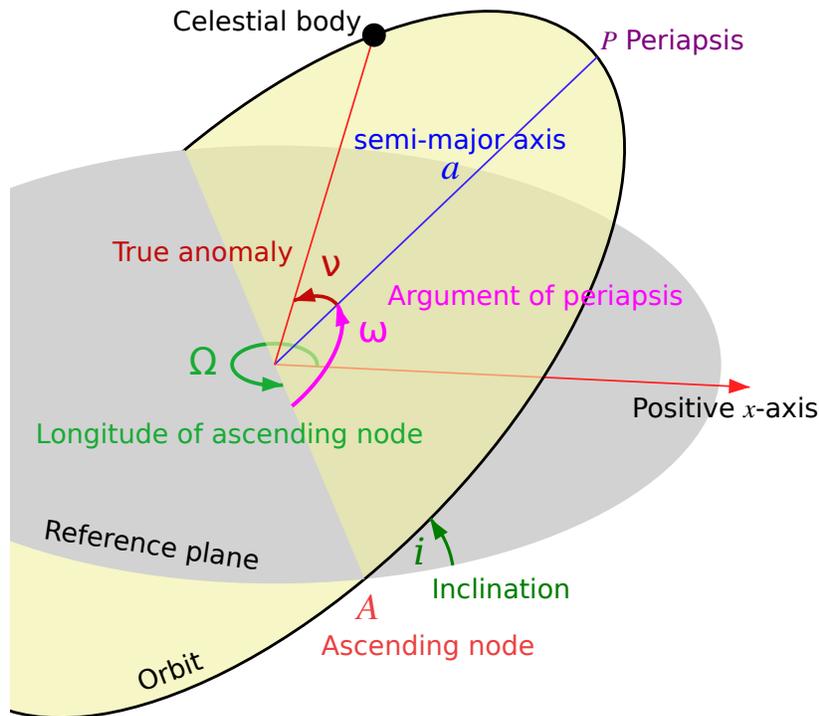


Figure B.1: The Keplerian orbital elements. Credit: modified from wikipedia.

Consider the orbit of a celestial body (such as a planet or a comet, the black disk in Figure B.1) orbiting in an ellipse (the yellow ellipse). Its orbital elements (a , e , i , Ω , ω , T) are defined with respect to a reference plane (the grey ellipse). We shall choose as our reference plane the *ecliptic plane*, the plane defined by the Earth’s movement around the sun³. The z -axis is normal to this ecliptic plane. The x -axis lies in the ecliptic plane and points towards the March equinox⁴. The y -axis is defined so that the coordinate system is right-handed. The origin of our coordinate system lies at the centre of mass of the solar system.

³ Gravitational perturbations from other bodies cause the plane to oscillate slightly, so astronomers choose a plane associated with a specific fixed time, referred to as “Julian epoch J2000.0”.

⁴ The equinoxes are the two points where the ecliptic plane of a body intersects its *equatorial plane*, the plane normal to its spin axis.

B.2.1 Size and shape

Our first two elements, a and e , describe the size shape of the ellipse.

The major axis of the ellipse is its longest diameter, the straight line of maximal distance between two opposite points on the ellipse. The first Keplerian element is the *semi-major axis* a : it is half of the length of the major axis. In Figure B.1 the semi-major axis is labelled in blue.

The eccentricity e describes how circular or elliptical the orbit is. The minor axis of the ellipse is its shortest diameter, the straight line of minimal distance between two opposite points on the ellipse; the semi-minor axis b is half its length. The eccentricity describes the relationship between a and b :

$$e = \sqrt{1 - \frac{b^2}{a^2}}. \quad (\text{B.2.1})$$

If the orbit is circular, then $b = a$ and $e = 0$; an ellipse has $e \in (0, 1)$. The eccentricity is not labelled in Figure B.1.

B.2.2 Orientation of the orbital plane

The next two elements, i and Ω , describe how the orbital plane is oriented in space.

The *inclination* i measures the angle between the plane of the orbiting ellipse and the reference plane. In Figure B.1 this is labelled in dark green.

There are two points on the orbit where the celestial body intersects the reference plane (see Figure B.1). Define the *ascending node* to be the intersection point where the body's z -coordinate is increasing (i.e. going from negative to positive). This is labelled as A on Figure B.1. The *longitude of the ascending node* Ω measures the angle in the reference plane between the positive x -axis and A . In Figure B.1 this is labelled in light green.

B.2.3 Orientation within the orbital plane

With the plane of the orbit now defined, we can describe the ellipse within it.

Let L be the line of intersection between the reference plane and the orbital plane. This line contains the ascending node A , and its counterpart, the descending node. It is therefore referred to as the line of nodes. Define the *periapsis* P to be the point on the orbit where the body is closest to the origin⁵. The periapsis is labelled in purple on Figure B.1.

The *argument of periapsis* ω measures the angle between the line of nodes L and the periapsis P , in the orbital plane. In Figure B.1 this is labelled in pink.

⁵ If the celestial body orbits the sun, the periapsis is also referred to as the perihelion.

B.2.4 Orbital period

Our final element, T , describes the period of orbit, the time taken to complete one orbit. Since this is related to the mass \mathcal{M} and semi-

major axis a by

$$T = \frac{2\pi}{\sqrt{GM/a^3}} \quad (\text{B.2.2})$$

where G is the gravitational constant, we could also equivalently record the mass of the body.

B.3 Computing orbits from orbital elements

Computing the position of the body at a given time proceeds in four steps:

1. Compute the mean anomaly M for the given time. This is simple, as M varies linearly with time.
2. Compute the eccentric anomaly E from the mean anomaly M by numerically solving Kepler's equation.
3. Compute the coordinates of the body in its orbital plane.
4. Rotate these coordinates to the reference plane.

B.3.1 Computing the mean anomaly

The *mean anomaly* M is a fictitious angle defining where on the orbit the body currently lies, as measured from periapsis. Its value at periapsis is $M = 0$; its value at the point on the orbit furthest away from the origin⁶ is $M = \pi$. The mean anomaly varies linearly with time and does not actually describe the real geometric angle of the position of the orbiting body at any given time, since the orbiting body does not sweep out equal angles in equal times⁷. The real geometric angle is referred to as the *true anomaly* ν and is labelled in dark red in Figure B.1.

Let τ denote a time at which the body is at periapsis. For a given time t , the mean anomaly is given by

$$M = \frac{2\pi}{T}(t - \tau), \quad (\text{B.3.1})$$

which is then adjusted to satisfy $M \in [0, 2\pi)$.

B.3.2 Computing the eccentric anomaly

In chapter 60 of *Astronomia Nova*, Kepler discussed the computation of the position of the orbiting body from the mean anomaly M . He derived a fundamental equation of elliptical orbits, now known as *Kepler's equation*. For given mean anomaly M and eccentricity e , we compute the *eccentric anomaly* E . The eccentric anomaly measures the angle from the body to the centre of its ellipse (whereas the true

⁶ The point furthest away from the origin is known as the *apoapsis*.

⁷ It sweeps out equal *areas* in equal times, by Kepler's second law.

anomaly measures the angle with respect to the body it is orbiting at the origin, e.g. the sun)⁸ by solving Kepler's equation,

$$M = E - e \sin E. \quad (\text{B.3.4})$$

You can find a modern derivation of this equation in Orlando et al.⁹ Kepler's equation is a transcendental equation: E cannot be solved for algebraically. As Kepler wrote,

It is enough for me to believe that I could not solve this a priori, owing to the heterogeneity of the arc and the sine. Anyone who shows me my error and points the way will be for me the great Apollonius.

This motivated a great deal of research into numerical rootfinding algorithms^{10,11}.

Question B.1. Prove that the residual

$$f(E) = E - e \sin E - M \quad (\text{B.3.5})$$

has a root in some interval $[E_-, E_+]$, where you should determine E_- and E_+ . (No coding is required for this question.)

Question B.2. Devise a simple fixed point iteration to compute E . Prove that your iteration converges with the Banach contraction mapping theorem (and hence that the solution is unique).

Question B.3. Write a function `eccentric_anomaly(M, e, verbose=False)` which computes $E(M)$ as a function of $M \in [0, 2\pi)$ for a fixed eccentricity e . Your code should use Halley's method to solve Kepler's equation. The code should employ a sensible initial guess, a sensible termination criterion, and return $E(M) \in [0, 2\pi)$.

If the optional flag `verbose=True`, the code should print out the iterations of Halley's method, as well as the residual, so that its convergence may be studied.

[Hint: it is convenient to use `sympy` to calculate the necessary derivatives, but this is not necessary.]

Question B.4. Apply your code to compute the eccentric anomaly for $e = 0.5$ and $M \in \{0.1\pi, 0.3\pi, 0.7\pi, \pi\}$, passing `verbose=True`.

⁸ The eccentric anomaly can be used to compute the true anomaly via

$$\sin v = \frac{\sqrt{1-e^2} \sin E}{1-e \cos E}, \quad (\text{B.3.2})$$

$$\cos v = \frac{\cos E - e}{1 - e \cos E}, \quad (\text{B.3.3})$$

and then applying the arctangent function. We will not need these formulae.

⁹ F. G. M. Orlando, C. Farina, C. A. D. Zarro, and P. Terra. Kepler's equation and some of its pearls. *American Journal of Physics*, 86(11):849–858, 2018

¹⁰ P. Colwell. *Solving Kepler's equation over three centuries*. Atlantic Books, 1993

¹¹ For example, it was Kepler's equation that Taylor was trying to solve with Halley's method when he discovered Taylor's theorem; see

T. R. Scavo and J. B. Thoo. On the geometry of Halley's method. *The American Mathematical Monthly*, 102(5):417–426, 1995

Question B.5. For $e = 0, 0.1, 0.5, 0.9, 0.99$, plot $E(M)$ as a function of M . Draw each curve on the same plot, clearly labelled.

B.4 *Computing the coordinates of the body in its orbital plane*

Now that we know the eccentric anomaly E , we can calculate the coordinates of the body in its orbital plane, in a coordinate system (x', y', z') with the x' -axis pointing to periapsis, the z' -axis normal to the plane with $z \cdot z' \geq 0$, and the y' -axis chosen to make a right-handed coordinate system.

The coordinates in the orbital plane are given by

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a(\cos E - e) \\ a\sqrt{1 - e^2} \sin E \\ 0 \end{pmatrix}. \quad (\text{B.4.1})$$

B.5 *Rotating these coordinates to the reference plane*

Once we have the coordinates in the orbital plane, a sequence of three rotations transforms them into coordinates on the reference plane:

1. First, a rotation by the argument of periapsis ω around the z' -axis. Before this, the point of periapsis lies on the positive x' -axis; after this, the ascending node lies on the positive x' -axis, and the descending node on the negative x' -axis.
2. Second, a rotation by i around the (transformed) x' -axis; since the x' -axis now represents the line of nodes L , this brings points above the line of nodes (with positive y') to have positive z' -coordinates.
3. Third, a rotation by Ω around the (transformed) z' -axis; this rotates the line of nodes to have the correct longitude, as measured in the reference coordinate system.

Question B.6. Write out the three matrices representing these three rotations, and (by symbolic or manual computation) derive the formula for their composition.

[Hint: each matrix will have one row and column of the identity matrix.]

B.6 *Comparing to Mars' orbit*

These formulae approximate the true orbit with an ellipse. How accurate is this approximation? We shall compare it to the orbit of Mars,

the planet which motivated Kepler to discover his first law. Mars' orbital elements are given in Table B.1.

Description	orbital element	value
semi-major axis	a	1.523679 [AU]
eccentricity	e	0.0934
inclination	i	0.0323 [rad]
longitude of ascending node	Ω	0.8656 [rad]
argument of periapsis	ω	5.0006 [rad]
orbital period	T	687 [days]
time of periapsis	τ	2024-05-08 11:08:00

Table B.1: Orbital elements for Mars.

Question B.7. Over the time interval

$$I = [2020-08-03, 2024-05-08](\approx [\tau - 2T, \tau]),$$

use the orbital elements to predict the position of Mars at 11:08 each day.

Plot your predicted positions in a three-dimensional plot¹². On the same figure, plot the observed positions of Earth and Mars as recorded in the NASA/JPL Horizons database¹³.

[Hint: you can apply `np.linspace` to `astropy.Time` objects.]

¹² This can be done with `matplotlib`; please consult its documentation.

¹³ This database can be conveniently queried with the `astroquery` package, as discussed in the handbook.

Question B.8. What is the maximum distance between the true and predicted position over the time interval I , measured in AU? When does this maximum distance occur?

B.7 The return of Halley's comet

Now let us compute the orbit of Halley's comet. Its orbital parameters are given in Table B.2.

Question B.9. Use the orbital elements to predict the position of Halley's comet at midnight each day over the interval

$$I = [2025-01-01, 2075-01-01].$$

Plot your predicted positions in a three-dimensional plot. On the same figure, plot the position of Earth as predicted in the NASA/JPL Horizons database.

Description	orbital element	value
semi-major axis	a	17.93 [AU]
eccentricity	e	0.9679
inclination	i	2.8308 [rad]
longitude of ascending node	Ω	1.031 [rad]
argument of periapsis	ω	1.958 [rad]
orbital period	T	27731.29 [days]
time of periapsis	τ	1986-08-06 08:06:00

Table B.2: Orbital elements for Halley's comet.

Question B.10. Plot the distance between Earth and Halley's comet (as predicted by the orbital elements) as a function of time over the interval I . On what day will Halley's comet come closest to Earth? How close will that minimal distance be?

B.8 Concluding remarks

It is one matter to predict the position of a celestial body from its orbital elements; *estimating* these orbital elements from observational data is harder still. Halley estimated his orbital parameters by fitting a parabola to three observations. Gauss' famous breakthrough in 1801 allowed him to estimate orbital parameters for any conic section (including ellipses) from three observations, which he then refined by least-squares; he published his methods in 1809¹⁴. A modern description of Gauss' methods is given in the bachelor's thesis of Bed'atš¹⁵.

We leave the last word to Halley:

But *Seneca the Philosopher*, having consider'd the *Phænomena* of Two remarkable Comets of his Time ... foretells that there should be Ages sometime hereafter, to whom Time and Diligence shou'd unfold all these Mysteries, and who shou'd wonder that the Ancients cou'd be ignorant of them, after some lucky Interpreter of Nature had shewn, *in what Parts of the Heavens the Comets wander'd, and how great they were.*

¹⁴ C. F. Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*. Perthes & Besser, 1809

¹⁵ D. Bed'atš. Gauss' calculation of Ceres' orbit. Bachelor's thesis, Charles University, Prague, Czechia, 2021

C 2025C: Pension planning

(This project relates to material in Prelims and Part A courses on Probability and Statistics, and Part A Simulation and Statistical Programming.)

In 1827 a Scottish botanist named Robert Brown was studying grains of pollen suspended in water under a microscope. The pollen appeared to undergo a random, jittery motion, with the particle dancing around unpredictably. The explanation was provided by Einstein in his *annus mirabilis* of 1905¹: the pollen grain is constantly being hit by tiny, invisible molecules of the fluid. Since at different times the pollen is hit more on one side than the other, it moves in random directions².

The same mathematical description as a stochastic process had been introduced in a different context five years earlier, in the PhD thesis of Louis Bachelier³. Bachelier, a PhD student of Poincaré, introduced the same idea to model stock prices on the Paris stock exchange. His thesis is now understood as the foundation of mathematical finance⁴.

In this project we will employ a stochastic differential equation (SDE) to model a stock market index, and use it to estimate the probability of comfortable retirement under different investment strategies.

C.1 The model

In modern notation, Bachelier considered the model

$$dS = \mu dt + \sigma dW, \quad (\text{C.1.1})$$

where $t \geq 0$ is time (measured in years), $S(t) \in \mathbb{R}$ is the value of the stock market index, μ is the expected rate of return⁵, σ is the volatility, and $W(t)$ is a Wiener process, a one-dimensional Brownian motion, formalised by Norbert Wiener in 1923⁶. The rigorous mathematical description of what SDEs mean and the Wiener process is beyond Prelims⁷, but for us it is enough to know that $W(t)$ satisfies the following properties:

1. $W(0) = 0$;

¹ In the same year, he published four famous papers on the photoelectric effect (which won him the Nobel prize), Brownian motion, special relativity, and deriving his equation $E = mc^2$ relating mass and energy.

² A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 17:549–560, 1905

³ Louis Bachelier. Théorie de la spéculation. *Annales scientifiques de l'École Normale Supérieure*, 17:21–86, 1900

⁴ J.-M. Courtault, Y. Kabanov, B. Bru, P. Crépel, I. Lebon, and A. Le Marchand. Louis Bachelier on the centenary of Théorie de la spéculation. *Mathematical Finance*, 10(3):339–353, 2000

⁵ In the mathematical finance literature this is referred to as the 'drift'. Throughout this project we ignore inflation, so that μ encodes the real rate of return after inflation.

⁶ N. Wiener. Differential space. *Journal of Mathematics and Physics*, 2(1):131–174, 1923

⁷ They are defined in B8.2 Continuous Martingales and Stochastic Processes.

2. For $0 = t_0 < t_1 < \dots < t_m$, the increments

$$W(t_1) - W(t_0), W(t_2) - W(t_1), \dots, W(t_m) - W(t_{m-1}) \quad (\text{C.1.2})$$

are independent, and each are normally distributed with mean and variance

$$\mathbb{E}[W(t_{i+1}) - W(t_i)] = 0, \quad (\text{C.1.3})$$

$$\text{Var}[W(t_{i+1}) - W(t_i)] = t_{i+1} - t_i. \quad (\text{C.1.4})$$

SDEs do not define a single trajectory in the way an ordinary differential equation might. Instead, SDEs define a stochastic process, a family of random variables (indexed in this case by time). We therefore aim to sample paths from the distribution implied by the SDE.

Bachelier's model (C.1.1) has a major limitation: it allows for the price $S(t)$ to become negative, which is not realistic, as stocks possess limited liability. This is remedied with a *geometric Brownian motion* model, as proposed by Samuelson and others in the 1950s and 1960s⁸:

$$dS = \mu S dt + \sigma S dW. \quad (\text{C.1.5})$$

This is the most widely used SDE to model stock price behaviour and is used e.g. in the Black–Scholes model popular in quantitative finance. It is this model that we will employ in our subsequent calculations. It has analytical solution⁹

$$S(t) = S(0) \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right) \quad (\text{C.1.6})$$

which still requires simulation to use because of the (random) $W(t)$ on the right-hand side.

Geometric Brownian motion is popular because it fits real time series data for stocks reasonably well. It is not perfect; real data appears to have a greater probability of large rises and falls than afforded in the model (C.1.5). However, it is a sensible first choice and can form a basis for more complicated models¹⁰.

C.2 Computing approximate solutions

The simplest numerical algorithm for computing approximate solutions of SDEs is the Euler–Maruyama method¹¹, the stochastic extension of the simplest algorithm for solving ordinary differential equations. We partition our time interval of interest into $[0, T]$ into N intervals with time step $\Delta t = T/N$, and compute approximations

⁸ P. A. Samuelson. Rational theory of warrant pricing. In *Henry P. McKean Jr. Selecta*, pages 195–232. Springer International Publishing, 1965

⁹ See, for example, the discussion in §3.2.1 of

P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer New York, 2003

¹⁰ P. Wilmott, S. Howison, and J. Dewynne. *The Mathematics of Financial Derivatives*. Cambridge University Press, 1995

¹¹ G. Maruyama. Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4:48–90, 1955

$S_i, i = 0, \dots, N$, to $S(i \cdot \Delta t)$. Given S_n , the Euler–Maruyama method computes S_{n+1} for (C.1.5) by

$$S_{n+1} = S_n + \mu S_n \Delta t + \sigma S_n \Delta W_n, \quad (\text{C.2.1})$$

where ΔW_n is a random variable drawn from a normal distribution with zero mean and variance Δt . Of course, in order to get a sense of the range of possibilities encoded in the SDE, we must sample many different trajectories and compute statistics from them.

The Euler–Maruyama scheme can be applied to any SDE. In our special case, however, we know the analytical solution (C.1.6). This motivates the specialised formula

$$S_{n+1} = S_n \exp\left((\mu - \sigma^2/2)\Delta t + \sigma \Delta W_n\right) \quad (\text{C.2.2})$$

where ΔW_n is the same as in (C.2.1)¹².

Question C.1. Reflect on both approximations. Why should we prefer (C.2.2) over (C.2.1)? (No coding is required for this question.)

¹² P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer New York, 2003

C.3 The stock and pension we will model

We will apply our model to describe the behaviour of a stock that is designed to follow the performance of a global index tracker¹³.

Let $S(t)$ denote the price of the stock, measured in pounds, and ignoring the effects of inflation. The stock price starts at $S(0) = 1$ at time $t = 0$ when our hypothetical investor is 25 years old. We assume the stock price follows geometric Brownian motion (C.1.5) with drift $\mu > 0$ and volatility $\sigma = 0.15$.

Each month the investor invests $M \geq 0$ pounds in his or her pension, and uses it to purchase stocks¹⁴. Let $H(t)$ denote the amount of stocks held at time t ; $H(t)$ increases by $M/S(t)$ at times $t = 0, 1/12, 2/12, \dots$, i.e. when $t \in (1/12)\mathbb{N}$. The investor retires at age 65, so $T = 40$.

The question we would like to address is: what is the probability of the final value of our pension

$$V = H(T) \cdot S(T) \quad (\text{C.3.1})$$

being enough to live on for a comfortable retirement, as a function of investment horizon T , drift μ , and monthly savings M ?

¹³ Most retail investors (i.e. you and me) have no particular inside information or advantage in predicting the behaviour of stocks: we do not have the skill to reliably outperform the market. A rational response to this is *passive investment*, where our investments are designed to track an externally specified index, like the S&P 500 or the FTSE 100. This allows us to achieve the average return of the market as a whole, with very low management fees. By choosing a suitable index to track we can achieve good diversification across industries and geographical regions. Passive investments like this generally yield higher returns than actively-managed funds, since they achieve the same average return but at lower cost. For example, the actively-managed Universities Superannuation Scheme (the pension scheme for university academics) has achieved annualised returns of 6.4% over the past five years, but a global equities tracker has achieved annualised returns of 10.5%: see

Universities Superannuation Scheme. Quarterly Investment Report as of 30 September 2024, 2024

¹⁴ For employees, both the employer and employee contribute to the total M . For example, as of writing, in the Universities Superannuation Scheme the employee contributes 6.1% of his or her gross salary, and the employer contributes 14.5%.

C.4 *Planning for retirement*

Question C.2. Implement a Python function `simulate_gbm(mu, P, T, N)` that computes P different sample paths for the SDE (C.1.5) over $[0, T]$ with the numerical scheme (C.2.2) using drift parameter μ and a timestep $\Delta t = T/N$. The function should return a matrix of size $P \times (N + 1)$. Your function can hardcode the values of σ and $S(0)$, for simplicity.

[Hint: your code should compute the update for all P paths in a vectorised manner.]

Question C.3. Apply your code to simulate $P = 100,000$ paths with $\mu = 0.05$ (5% annual real growth), $T = 40$, and $N = 12T$, i.e. timesteps of one month.

Plot the first five paths, the mean path, and the 5th and 95th percentiles, as a function of time. (The 5th percentile is the function $c(t)$ so that at any time t , 5% of the sample paths $S(t)$ satisfy $S(t) < c(t)$; the 95th percentile is defined analogously.)

[Hint: choose sensible colours and line styles for each curve so that the plot is easy to understand.]

Question C.4. Write a function `pension_value(paths, T, M)` that computes the final value of our investment at time T for a given set of sample paths and monthly investment M ¹⁵.

Verify that your function works correctly by applying it to suitable examples.

¹⁵ Here we assume our purchases do not alter the stock price, which is reasonable for retail investors.

Question C.5. Compute the final value V of the pension for the paths computed in Question C.3 for $M = \text{£}1,000$. Plot the result on a histogram with bin size $\text{£}50,000$. Print out the bin with the highest count of paths.

Question C.6. How does the final value of the pension depend on $S(0)$, the initial stock price? (No coding is required for this question.)

Question C.7. For the paths computed in Question C.3, what is the observed probability of making a loss on the investment (i.e. V is less than the amount invested)? What is the probability of the value of our investment doubling? What is the probability of the value of the investment being at least £2,000,000?

Question C.8. We decide that we would like an investment value of at least $V_{\min} = \text{£}1,000,000$ for a comfortable retirement¹⁶.

For $T = 40$ and $\mu \in \{3\%, 5\%, 7\%\}$, compute the observed probability of comfortable retirement

$$R(M) := \mathbb{P}[V(M) \geq V_{\min}]$$

as a function of monthly investment rate M , for $M \in [0, 4000]$. Use $P = 100,000$ paths for each simulation. Plot $R(M)$ for each μ on a single plot. What M is necessary to achieve 95% probability of comfortable retirement?

[Hint: you will need to choose a suitable spacing to discretise M .]

Question C.9. Our investor decides that he or she would like to retire early to study mathematics full-time. Repeat Question C.8 but with $T = 20$, with M chosen from a suitable interval. Comment on how much larger M must be in this case for each μ , relative to the $T = 40$ case.

C.5 Concluding remarks

While this is a reasonable first model for the self-invested personal pension of a retail investor, there are many improvements that could be made. Instead of geometric Brownian motion, we could employ a more sophisticated SDE, such as the Merton jump-diffusion model¹⁷. Another facet to model is that investors are advised to rebalance from riskier assets (like equities) to safer assets (like bonds) as they approach retirement.

In this project we have employed Monte Carlo sampling to approximate our paths. Monte Carlo is simple and effective, but can be very expensive when the cost of each sample is high, e.g. when the discretisation parameter Δt must be small. In 2008 Mike Giles, a professor in the Mathematical Institute, proposed a revolutionary new algorithm that in certain cases dramatically reduces the cost of Monte Carlo simulations. His *multilevel Monte Carlo* algorithm achieves high accuracy

¹⁶ The usual rule of thumb is to withdraw no more than 4% of your pension each year of retirement, so this amounts to an annual gross income of £40,000, in addition to the state pension. Generally your pension income is lower than your salary, since (i) you no longer need to save for retirement (ii) major expenses like house purchases and childrearing have concluded.

¹⁷ R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125-144, 1976

by performing the majority of simulations at low accuracy (and thus low cost), combined with few simulations performed at high cost¹⁸. This is now used in the financial industry for pricing complex derivatives, and in modelling physical systems with uncertainty, such as groundwater flow or biochemical reactions; for a review, see Giles (2015)¹⁹.

¹⁸ M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008

¹⁹ M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 5 2015

Bibliography

- [1] Louis Bachelier. Théorie de la spéculation. *Annales scientifiques de l'École Normale Supérieure*, 17:21–86, 1900.
- [2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.
- [3] D. Bed'atš. Gauss' calculation of Ceres' orbit. Bachelor's thesis, Charles University, Prague, Czechia, 2021.
- [4] M. A. Brilleslyper and B. Schaubroeck. Explorations of the Gauss–Lucas theorem. *PRIMUS*, 27(8–9):766–777, 2016.
- [5] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(4):377–409, 1993.
- [6] P. Colwell. *Solving Kepler's equation over three centuries*. Atlantic Books, 1993.
- [7] J.-M. Courtault, Y. Kabanov, B. Bru, P. Crépel, I. Lebon, and A. Le Marchand. Louis Bachelier on the centenary of Théorie de la spéculation. *Mathematical Finance*, 10(3):339–353, 2000.
- [8] S. Devadoss and J. O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.
- [9] E. Halley, Savilian Professor of Geometry at Oxford; And Fellow of the Royal Society. *A Synopsis of the Astronomy of Comets. Translated from the original, printed at Oxford*. Printed for John Senex, next to the Fleece-Tavern, in Cornhill, 1705.
- [10] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 17:549–560, 1905.
- [11] C. F. Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*. Perthes & Besser, 1809.

- [12] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [13] M. B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 5 2015.
- [14] P. Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer New York, 2003.
- [15] R.L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1(4):132–133, 1972.
- [16] J. Kepler. *Astronomia Nova: seu physica coelestis, tradita commentariis de motibus stellae Martis ex observationibus G.V. Tychoonis Brahe*. 1609. Translated by W. H. Donohue. Green Lion Press, 2015.
- [17] G. Maruyama. Continuous Markov processes and stochastic equations. *Rendiconti del Circolo Matematico di Palermo*, 4:48–90, 1955.
- [18] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1–2):125–144, 1976.
- [19] F. G. M. Orlando, C. Farina, C. A. D. Zarro, and P. Terra. Kepler’s equation and some of its pearls. *American Journal of Physics*, 86(11):849–858, 2018.
- [20] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977.
- [21] P. A. Samuelson. Rational theory of warrant pricing. In *Henry P. McKean Jr. Selecta*, pages 195–232. Springer International Publishing, 1965.
- [22] T. R. Scavo and J. B. Thoo. On the geometry of Halley’s method. *The American Mathematical Monthly*, 102(5):417–426, 1995.
- [23] Universities Superannuation Scheme. Quarterly Investment Report as of 30 September 2024, 2024.
- [24] R. Seidel. *Convex hull computations*, pages 361–375. CRC Press, Inc., 1997.
- [25] H. Weyl. *Symmetry*. Princeton University Press, 1952.

- [26] N. Wiener. Differential space. *Journal of Mathematics and Physics*, 2(1):131–174, 1923.
- [27] P. Wilmott, S. Howison, and J. Dewynne. *The Mathematics of Financial Derivatives*. Cambridge University Press, 1995.
- [28] Αρχιμήδης. *Περ σφαιρας κα κυλινδρου*. 225 B.C. Translation by R. Netz, Cambridge University Press.
- [29] Πλάτων. *Τμαιος*. Cambridge University Press, 1888. Translated by R. D. Archer-Hind, Fellow of Trinity College, Cambridge.