Mechanising Newtonian Mechanics in Isabelle

Lars Werne

Honours Project
Artificial Intelligence and Mathematics
School of Informatics
University of Edinburgh

2022

Abstract

Despite having been experimentally falsified at the end of the 19th century, the subject of classical mechanics remains of great importance due to its predictive power in many commonly analysed scenarios. Having originated from Isaac Newton's *Principia* in 1687, the theory of Newtonian mechanics has been reformulated many times over the centuries. Consequentially, its theoretical foundations are ambiguous, and its theorems can be derived in many ways.

With the aim of exploring a certain approach to the formalisation of the theory, this report describes a mechanisation of Newtonian mechanics using the interactive theorem prover Isabelle. We chose Euclidean vector spaces as our representation of physical space and justified this choice through a formal exploration of their properties, particularly regarding angles and vector derivatives. Following modern textbooks on classical mechanics, our axiomatisation of Newtonian mechanics was based on methods from vector analysis, as formalised in the HOL-Analysis library. We formulated a set of axioms for particle motion in the context of Newtonian mechanics using Isabelle. Subsequently, we formally proved fundamental relations between the mass, position and force of point particles, and applied our framework to the study of simple harmonic oscillators in one dimension.

Acknowledgements

Throughout this project, I have had the great privilege of being supported by my supervisor Jacques Fleuriot and his PhD students Jake Palmer and Richard Schmoetten. I whole-heartedly thank them for all the advice they gave me.

Also, I thank Inka, for all the emotional support I received over the course of our undergraduate years, and in the last few weeks in particular.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Lars Werne*)

Table of Contents

Chapter 1

Introduction

Mechanics concerns itself with deriving the motion of physical objects from a set of experimentally verified laws. Attempts at formulating a theoretical framework for mechanics existed in antiquity [49], but until Newton published his *Principia* in 1687 [38], none had been as successful. Newtonian mechanics denotes the study of motion based on the law of universal gravitation and Newton's laws, which relate the acceleration of particles to the forces acting on them. Newton's formulation of these laws was the cornerstone of classical mechanics. The validity of this branch of physics has been disproven at the subatomic level and for objects with enormous mass or moving near the speed of light, giving rise to quantum mechanics and Einstein's theory of relativity in the first third of the 20th century [8, 15]. In the 18th and 19th centuries, however, mechanics was primarily based on Newton's findings [1], with the most significant reformulations of his theory arising in the form of Lagrangian [30] and Hamiltonian [22] mechanics. Even today, the theorems of classical mechanics remain of immense practical value, as they predict reality with impressive accuracy at the scale we usually observe.

In this project, we mechanised a framework for Newtonian mechanics using the interactive theorem prover Isabelle. We began with an exploration of the properties of Euclidean vector spaces, motivating their use as a formal representation of physical space. While our subsequent axiomatisation of Newtonian mechanics aimed to correspond to the fundamental assumptions made by Newton in his *Principia*, it followed common treatments of the subject in modern textbooks. In particular, this meant that we employed the rules of (vector) analysis, which had not been formally established at the time of Newton's work.

An extensive derivation of key findings from classical mechanics could be used to validate physical simulations at the macroscopic level, for example, in structural engineering or autonomous vehicles, moreover being of theoretical interest to physicists. The results of this project could inform one of many potential starting points for such a complete formalisation.

1.1 Organisation of the report

Leading on from the above introduction, we provide a brief overview of the scientific debate regarding the foundations of Newtonian mechanics and place our project in the context of this discussion (cf. chapter 2). Subsequently, we outline the primary resources that we used throughout the project. Namely, we describe the version of the Isabelle proof assistant and the source literature on which the formal proofs in this project are based (cf. section 3.1). We then introduce the concept of locales and the notation for derivatives in Isabelle, in order to ease the reader's understanding of our mechanisation (cf. section 3.2).

Next, we begin presenting our results more concretely, by discussing the considerations that led us to represent physical space generally as a Euclidean vector space (cf. section 4.1). In the ensuing sections of chapter 4, we explore some properties of these spaces in Isabelle, justifying the above representation and exemplifying the formal application of techniques from standard analysis to space curves.

We then present the core results of this project – our formalisation of Newtonian mechanics (cf. chapter 5). After discussing our axiomatisation (cf. section 5.1), we prove some fundamental results regarding the relations between the quantities present in our axioms, mainly forces, particle positions (or their time derivatives) and mass. In particular, we define gravitational forces, survey how our set of axioms may be extended by the example of point charges, and derive special properties of particle systems in the absence of external forces (cf. section 5.2). We then discuss simple harmonic motion in one dimension, ultimately deriving the harmonic oscillator equation under appropriate assumptions (cf. section 5.3). Lastly, we provide some conclusions about our results, discuss challenges that arose in this project and suggest directions for future work (cf. chapter 6).

Chapter 2

Literature Review

Newton's *Philosophiæ Naturalis Principia Mathematica*, published in 1687, is regarded as the foundation stone of Newtonian mechanics. His techniques were based on intuitive assumptions from Euclidean geometry, combined with an informal notion of infinitesimals. The historical significance of Newton's work as a paradigm shift in physics [29] is universally agreed upon. In contrast, the formal justifiability of his methods has been the topic of scientific debate. Several authors have argued that Newton's reasoning is, in places, flawed as it contains erroneous proof steps, the validity of which can only be restored via the addition of non-trivial assumptions [45, 53].

Despite their modernity, we found that standard texts on Newtonian mechanics generally do not provide an explicit and complete presentation of the assumed properties of particle motion. Moreover, the fundamental assumptions that *are* explicitly stated differ heavily between publications. For example, Kibble and Berkshire (2004) describe Newton's laws as axioms [26], whereas Gregory (2006) derives their validity from a separate set of assumptions [20]. Hence, although the results of classical mechanics appear to generally be well understood and agreed upon, the fundamentals on which the theory is based are not.

The aforementioned discussion about the validity of Newton's methods illustrates a primary cause for the existing ambiguity; the expectations on mathematical texts have changed heavily over the past centuries. During Newton's lifetime, only proofs that appealed to geometric intuition were likely to be accepted by large parts of the scientific community. By contrast, the *Principia* does not satisfy our modern expectations that mathematics should be solidly grounded on a set of precise axioms and definitions. Therefore, different conceptions about the foundations of Newtonian mechanics, which naturally resulted from numerous restatements of the theory over time, cannot be resolved by referring to any canonical source.

In 1953, Truesdell commented on an axiomatisation of classical particle mechanics which had been proposed by Suppes et al. and which was to be published in Truesdell's journal: "[I am] in complete disagreement with the view of classical mechanics expressed in this article. [I agree] however, that strict axiomatization of general mechanics [...] is urgently required." [36] He further expressed his hopes that the publication of the paper in question would arouse interest in the lack of a precise and commonly recognised theoretical characterisation of forces, and eventually lead to "a proper solution of this outstanding but neglected problem". Thus, the foundations of classical mechanics have been a topic of scientific debate for many decades.

In particular, the concept of forces has been one of the primary sources of disagreement between axiomatisations of Newtonian mechanics proposed since the start of the 20th century. Some authors, like Papachristou (2012) and Mach (1907) suggest to view the force acting on any body as a *defined* quantity, either as the product of its mass and acceleration, or as the instantaneous rate of change of its momentum [43, 33].[1] Others, including Suppes et al. (1953), and Feynman (1965) instead regard force as a primitive entity, certain properties of which are axiomatically assumed [36, 17]. Hence, the theory of Newtonian mechanics may be constructed in many ways.

Moreover, Newtonian, Lagrangian and Hamiltonian mechanics, despite usually being seen as equivalent formulations of the same theory, differ heavily in their mathematical foundations and subtly in their metaphysical assumptions [42]. Hence, if one is interested in formally deriving the theorems of classical mechanics with the help of a proof assistant, different approaches should be considered and compared in terms of their simplicity[2] and generality. Several such approaches have previously been proposed. In Isabelle, Fleuriot derived results from Newton's *Principia* using techniques from nonstandard analysis, which differ from those usually found in modern textbooks on classical mechanics [18]. In HOL Light, Guan et al. formalised results pertaining to Euler-Lagrange equations, which form the mathematical foundation of Lagrangian mechanics [21]. Both of these attempts at formalising classical mechanics differ from our approach in terms of their mathematical footing. Hence, our project can be contextualised as a distinct puzzle piece in the search for precise, practical and complete formalisations of classical mechanics.

[1] Since momentum is defined as the product of a body's mass and velocity, these two definitions of force are equivalent precisely if the mass of objects is assumed to be constant.

[2] Here, 'simplicity' is meant in two ways. Ideally, any mechanisation of classical mechanics should be intuitively understood by scholars of the field, and yield simple (partially) automated proofs.

Chapter 3

Methodology

The results of our investigations of Euclidean space, followed by our formalisation of the fundamentals of Newtonian mechanics, will be outlined in the subsequent chapters. Before this, we shall briefly describe the resources and some of the Isabelle concepts on which our results are based.

3.1 Resources

All proofs and definitions presented in this report were mechanised using the computer proof assistant Isabelle [40]. Our implementation was built on the basis of the HOL-Analysis library, as included in the most recent December 2021 Isabelle release. Our results rely on this particular version of Isabelle as it introduced the *Infinite_Sum* theory, which we employed in our discussion of forces (cf. section 5.1).

Our exploration of Euclidean spaces (cf. section 4) was partially inspired by the introductory chapter of the undergraduate textbook *Classical Mechanics* by Gregory [20]. In particular, our results on tangents, normal vectors and arc length (cf. section 4.4) were based on findings presented in this volume. We added appropriate sets of necessary assumptions for our formalisation, which had mostly been left unspecified in the book. On the other hand, our treatment of angles (cf. section 4.2) and higher-order derivatives (cf. section 4.3) did not hinge on any particular source. Instead, we followed our intuition and proved results that we considered helpful in showcasing the properties of Euclidean vector spaces.

For our mechanisation of Newtonian mechanics (cf. section 5), we considered several treatments in physics textbooks, which often contained subtle differences (cf. section 2). Ultimately, we based our axioms on the introduction to Newtonian Mechanics found in *Classical Mechanics (5th Edition)* by Kibble and Berkshire [26], mainly because its mathematical contents are consistently formulated using the language of vector analysis, not relying as much on prose as some other physics textbooks. We hoped that this choice would allow us to express the source material in a straightforward way using the notation provided by the HOL-Analysis Isabelle library. While the formulation of results in the book did indeed often carry over to Isabelle quite effortlessly, certain applications of integrals, which we discuss in section 5.3, were an exception to this rule.

3.2 Isabelle techniques

3.2.1 Filters in the context of derivatives

In Isabelle, derivatives at a point are defined in terms of neighbourhood filters [9, 24]. Because they appear throughout the report, we felt it was appropriate to briefly exemplify the relevant notation. We shall do so in the form of a small case study.

Consider the absolute value function (cf. figure 3.1), $f(x) = \begin{cases} -x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$.

The derivative of f at a point c is commonly expressed as $f'(c) = \lim_{x \longrightarrow c} \frac{f(x)-f(c)}{x-c}$. By the sequential criterion for limits, one may equivalently require $f'(c)$ to be the limit of $\frac{f(x_n)-f(c)}{x_n-c}$ for all sequences of numbers (x_n) which converge to c, with $x_n \in \mathbb{R} \setminus \{c\}$ for all n. For strictly positive or strictly negative numbers c it can be shown that this limit f' is necessarily equal to 1, or -1, respectively. This standard notion of derivatives can be captured in Isabelle using the filter `(at c)` from the HOL-Analysis library.

```
lemma assumes "c > 0" shows "(f has_vector_derivative 1) (at c)"
(...)
lemma assumes "c < 0" shows "(f has_vector_derivative -1) (at c)"
```

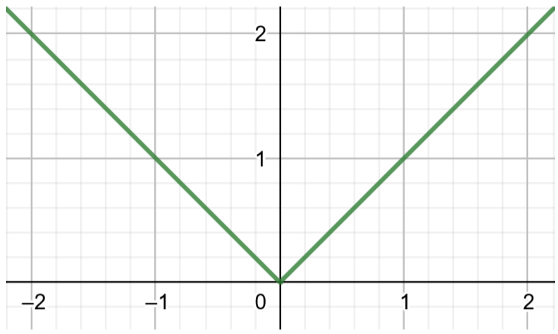

Figure 3.1: Graph of the absolute value function.

For $c = 0$ however, things are slightly more complicated. If (x_n) is an arbitrary sequence converging to 0 with values in $\mathbb{R} \setminus \{0\}$, then $\frac{f(x_n)-f(0)}{x_n-0} = \frac{f(x_n)}{x_n}$ may generally be equal to 1 or to -1 as (x_n) becomes small. Hence, no derivative is defined `at 0`. If we however restrict the values in the relevant sequence to all be non-negative, or non-positive, the resulting limits will again be guaranteed to equal 1, or -1, respectively.

In Isabelle, this restriction of values in the sequential definition of a limit to a set S can be represented by the filter `(at c within S)`. Indeed, we can prove

```
lemma shows "(f has_vector_derivative 1) (at 0 within {0..})"
(...)
lemma shows "(f has_vector_derivative -1) (at 0 within {..0})"
(* ... and therefore *)
lemma shows "¬(f differentiable (at 0))"
```

3.2.2 Locales

Locales in Isabelle allow for the declaration of variables, definitions and assumptions within a local scope. Results proved inside a locale are only visible (and considered proved) from within its context [25].

For example, the function `f` in the lemma declarations of the previous section was defined inside a locale. It would have been unwise to give this definition global scope, as we may want the letter `f` to denote something else in other contexts.

```
locale filter_deriv_example =
    fixes f :: "real ⇒ real"
  defines "f x ≡ (if x < 0 then -x else x)"
```

Moreover, locale declarations may contain assumptions. For example, we could have equivalently captured the above information in the following way.

```
locale filter_deriv_example =
    fixes f :: "real ⇒ real"
  assumes "∀x. ((x < 0) ⟶ (f x = -x))"
      and "∀x. ((x ≥ 0) ⟶ (f x = x))"
```

Note that it is also possible to fix a constant of a certain type without assuming anything further about it. Moreover, one can use type variables to leave the type of a fixed entity ambiguous. Optionally, sort constraints can be placed on type variables, which ensure that the relevant type instantiates certain type classes [24], thereby attaining certain properties. For instance, if we had not required the co-domain of `f` to be $\mathbb{R}$ but only that it contained a neutral element with respect to addition, we could have written **`fixes`** `f :: "real ⇒ ('a :: zero)"`.

The scope of a locale is initially delimited by a pair of **`begin`** and **`end`** commands following a locale header like those shown above. Locales may however be extended by other locales. In this way, results which are proven in a certain locale may carry over to more specific or extensive contexts in which further assumptions or definitions are added. In our project, we used this approach to mechanise Coulomb's law, which applies to electrically charged particles that do not move relatively to each other (cf. section 5.2). Because such point charges can be seen as constituting a subclass of point particles in the context of Newtonian mechanics, we imported our initial `newtonian_system_of_particles` locale (cf. section 5.1), fixed additional constants and added further assumptions, corresponding to the properties that differentiate them from general particles. This example is shown below:

```
locale system_of_stationary_point_charges =
   newtonian_system_of_particles
+ fixes charge :: "'a ⇒ real"
    and ε_0 :: "real" (* Electric constant *)

assumes positive_electric_constant: "ε_0 > 0"
    and stationary: "∃r. ∀t. relative_position p q t = r"
```

If one, in a theory outside the context of any locale, introduces constants which meet the type restrictions and assumptions of the variables fixed inside a locale header, those constants can be used to interpret the locale. That means, if the Isabelle user can prove that the interpretation is valid, the results proved inside the locale become available in the external theory, the locale parameters being instantiated with the provided constants [7]. Interpreting the `system_of_stationary_point_charges` locale could, for example, allow for computations of the electrostatic force between certain point charges to be formally verified, given a concrete set of inputs.

Chapter 4

Properties of space

As a physical theory, Newtonian mechanics mainly concerns itself with the motion of particles through space. Hence, for our formal implementation of such a framework, deciding which representation of physical space to employ was a fundamental decision with large implications for our mechanisation as a whole. In this section, we shall discuss the relevant decision process, before describing a selection of definitions and facts that were enabled by our choice of space in Isabelle. These results, mainly concerning angles and derivatives, shall serve as a brief exploration of the fundamental properties of space through the lens of Newtonian mechanics, and are intended to justify our choice of representing space as a Euclidean vector space.

4.1 Choosing a representation of space

Ignoring the intricacies of general relativity, a mathematical entity that, intuitively, models our spatial surroundings very accurately is $\mathbb{R}^3$, with distances being given by the Euclidean metric. While this would have been a passable choice for our implementation, it would have raised the question of whether Newton's laws may not also be applied to other notions of space. Indeed, examples that treat space as 1- or 2-dimensional are common in physics textbooks [20, 26] (cf. section 5.3). Therefore, it was important for us to decide on a notion of 'space' that allowed for a sufficiently general discussion of particle motion, while incorporating the fundamental assumptions implicit in the definitions and techniques of Newtonian mechanics.

Newton stipulated that time is a scalar property that passes at a consistent rate in a way unrelated to space [47]. A fundamental prerequisite for any viable notion of space in the context of Newtonian mechanics is that it should allow us to discuss the velocity of an object as the rate of change of its position with respect to time. From a mathematical point of view, velocity in Newtonian mechanics is naturally treated as the derivative of position with respect to real-valued time. It is worth pointing out that this interpretation of Newton's views could only be made formal in hindsight, as the real numbers were rigorously defined for the first time by Cantor [10] and Dedekind [14], two centuries after Newton completed the book 'Method of fluxions', in which he first introduced his notion of derivatives [39]. Nevertheless, to give an appropriate account of Newtonian mechanics from the perspective of modern mathematics, we had to ensure that any representation of 'space' allowed for the existence of differentiable functions from the real numbers to space.

Hence, if S is to be a set that may represent 'space' and f is a function from $\mathbb{R}$ to S, we need to be able to consider the notion of a derivative of f at any $a \in \mathbb{R}$, as $f'(a) = \lim_{h \longrightarrow 0} \frac{f(a+h)-f(a)}{h}$. This limit exists and is equal to the unique value L if

$$\forall \varepsilon > 0 \exists \delta > 0 \forall h \in \mathbb{R} : (h \neq 0 \wedge |h| < \delta) \Longrightarrow \left| \frac{f(a+h)-f(a)}{h} - L \right| < \varepsilon.$$

For this expression to make sense, three functions on S need to be defined: Subtraction $(\text{-}) : S \times S \to S$, division by non-zero scalars $(/) : S \times (\mathbb{R} \setminus \{0\}) \to S$ and a norm $|\_| : S \to \mathbb{R}$. Hence, we need S to be a normed vector space over the scalar field $\mathbb{R}$ [13]. In Isabelle, such sets are represented by the type class `real_normed_vector`.

However, this choice would not generally allow us to define the angle between two vectors using the well-known expression, $\theta(\mathbf{u}, \mathbf{v}) = \arccos\left(\frac{\mathbf{u} \bullet \mathbf{v}}{|\mathbf{u}| * |\mathbf{v}|}\right)$. To make this definition possible, there has to exist an inner product $(\bullet) : S \times S \to \mathbb{R}$. Angles arise in numerous problems in classical mechanics, which is why we require S to be a real inner product space. The norm of a vector in a real inner product space is defined as the square root of its inner product with itself. In Isabelle, the distance d between two vectors of any type of sort `real_inner` is represented by the `dist` function and assumed to be equal to the norm of the difference between the vectors. In the inner product space $\mathbb{R}^n$, where $n \in \mathbb{N}$, this norm and its associated metric are known as 'Euclidean norm/distance' and coincide with our intuitive understanding of lengths in one, two and three dimensions [32].[1] Hence, `real_inner` extends the `real_normed_vector` type class, which means that derivatives in the previously described sense can be defined for functions from $\mathbb{R}$ to any real inner product space S.

The main reason real inner product spaces are not, in general, suitable to serve as 'space' in the context of Newtonian mechanics is that they may have infinitely many dimensions [12]. The theoretical implications of an infinite-dimensional universe have been considered [50], albeit rarely, but such discussions lie far outside the scope of classical mechanics. Similarly, it is sensible to exclude the possibility of zero-dimensional space. Although the concept of a universe with zero spatial extent has famously been considered in the context of a gravitational singularity at the time of the Big Bang [23], it is clear that Newton's laws would be meaningless in such a scenario.

To avoid such spaces, we considered real inner product spaces with a positive, but finite, number of dimensions. It is widely known that each such space has an orthonormal basis [31]. In the Isabelle library, the `euclidean_space` type class is constructed by inheriting the properties of `real_inner` and fixing a finite, non-empty, orthonormal basis.[2] We can think of this 'canonical' basis as a set of coordinate axes with respect

[1] If we label points in space using Cartesian coordinates, i.e., a quadratic grid, the Euclidean distance between two points can be computed using the Pythagorean theorem [34].

[2] Furthermore, it is axiomatically assumed that no non-zero vector is orthogonal to all basis vectors at once. We did not make explicit use of this axiom in the current project, but chose to accept it as it coincides with our geometric intuition.

to which positions in space may be expressed. In Isabelle, it is straightforward to show that `euclidean_space` instantiates the type class `real_inner`, allowing for the definition of derivatives and angles, as discussed. Moreover, it can be shown that any finite-dimensional, real inner product space is complete with respect to the aforementioned Euclidean distance [46]. In other words, every Euclidean vector space is a Banach space [13] and, correspondingly, in Isabelle, `euclidean_space` extends the type class `banach`. This fact is of great interest to us as it allows us to apply the fundamental theorem of calculus [51], as well as related results about the inverse relationship between integration and differentiation, to functions from $\mathbb{R}$ to 'space'. In this project, we primarily used results from this branch of analysis in our formalisation of one-dimensional harmonic oscillations (cf. section 5.3).[3]

Thus, for our mechanisation, we chose to let 'space' be represented by an arbitrary Euclidean vector space. Since the textbooks on Newtonian mechanics that we considered for this project do not make this abstraction, we chose to further rationalise our representation, by exemplifying that the type classes `real_inner` and `real_normed_vector` do fulfil the properties we have identified as necessary in this section. We began by defining angles in any real inner product space and showed that this general notion matches our intuitions from $\mathbb{R}^2$ and $\mathbb{R}^3$.

4.2 Undirected angles

In Euclidean geometry, two rays that start at the same point give rise to two angles which, expressed in radians, add up to 2π. The undirected angle between the two rays can be defined to be the smaller one of these angles. To begin our exploration of physical space, we mechanised the definition of angles provided in section 4.1.

```
definition angle :: "('a :: real_inner) ⇒ 'a ⇒ real"
  where "angle x y = arccos ((x • y) / ((norm x)*(norm y)))"
```

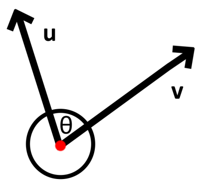

Figure 4.1: The (undirected) angle θ between two vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^2$.

Note that the inputs to the angle function are assumed to belong to the `real_inner` type class, as this suffices to guarantee the existence of a real-valued inner product, in terms of which a norm is defined. All properties of `angle` that we prove in Isabelle are automatically shown to hold for inputs of the type class `euclidean_space` as it inherits from `real_inner`.

[3] In the context of that particular discussion, we concretely instantiated the `'space` type variable with the real numbers. Nevertheless, it is reassuring that our general representation guarantees that the tools of integral calculus can be used for any concrete choice of space.

Having defined the notion of an undirected angle, we set out to prove some of its properties in order to confirm that our implementation matches our intuition.

Before all else, note that `(x • y) / ((norm x)*(norm y))` in the definition of `angle` evaluates to `0` in the case that `x` or `y` is the zero vector. That is because the indeterminate form $\frac{0}{0}$ is defined accordingly in Isabelle/HOL. Consequentially, in our formalisation, the angle between the zero vector and any other vector is $\frac{\pi}{2}$.

In some cases, this choice is convenient. Namely, it allowed us to show that two vectors are `orthogonal` if and only if the `angle` between them is $\frac{\pi}{2}$.[4] On the other hand, we proved several results pertaining to the angle between any pair of non-zero vectors which cannot be extended to all pairs of (zero or non-zero) vectors given our definition.

For instance, the angle between two *non-zero* vectors is 0 if and only if their dot product equals the product of their norms. Evidently, this does not hold for general vectors; if u or v is zero, their dot product and the product of their norms equal 0 and are thus identical, while we assign the non-zero angle $\frac{\pi}{2}$ to these inputs.

Furthermore, our discussion of angles led to a condition for the equality of *non-zero* vectors in any real inner product space; two vectors are the same if and only if they have the same magnitude and direction. The zero vector is exempt from this rule. That is because this vector is of course equal to itself despite forming a non-zero angle with itself.

```
lemma vector_equality_condition:
  assumes "x ≠ 0 ∨ y ≠ 0"
    shows "(x = y) = ((norm x = norm y) ∧ (angle x y = 0))"
```

This condition allowed us to prove the intuitive idea that two (non-zero) vectors u and v are opposite and equal exactly if u equals $-v$. This inconspicuous result will be used to justify our formalisation of Newton's third law (cf. section 5.1).

```
lemma equal_and_opposite:
  assumes "v ≠ 0"
    shows "(u = -v) ⟷ (norm u = norm v ∧ angle u v = pi)"
```

The above results could have been formalised more elegantly, that is, without assuming that the vector inputs are non-zero, if we had explicitly defined the angle between a zero vector and any other vector to be 0. The special case of division by 0 will also arise in our discussion of gravity (cf. section 5.2). More generally, since no notion of undefinedness exists in Higher Order Logic, all functions defined in Isabelle/HOL are total [28]. This caveat should be kept in mind, as it may sometimes necessitate function definitions to be split by cases. In this project however, we avoided the use of case-wise definitions in favour of simpler expressions unless there were pressing reasons against it (cf. section 5.3). Hence, we did not adjust our definition of `angle` in light of the zero-case.

On a different note, an important property of vectors in any real inner product space (or, more generally, any real vector space) is that they may be multiplied by real-valued

[4] Due to the relevant definition in the *Linear_Algebra* theory file of the HOL-Analysis library, two vectors of a real inner product space are called `orthogonal` exactly if their inner product equals 0 - which is certainly the case if one of the vectors is the zero vector.

scalars, yielding another vector within the same space. In Isabelle, the operation of scalar multiplication is represented by the function `scaleR`, denoted in infix notation as $*_R$. Intuitively, multiplying a vector by a positive scalar corresponds to 'stretching' (or 'compressing') it without changing its direction. Correspondingly, this operation does not change the angle of the vector with any other vector.

```
lemma pos_scalar_mult_angle:
  assumes scalar_pos: "(t :: real) > 0"
    shows "angle (t *R u) v = angle u v"
```

The same is not true if the scalar factor is negative. As a special case of this phenomenon, we first showed that, if the angle between vectors u and v is θ, negating either vector (corresponding to a scalar multiplication by -1) yields the 'supplementary angle' $\pi - \theta$ (cf. figure 4.2). More succinctly,

```
lemma angle_neg_right: "angle u (-v) = pi - angle u v"
(...)
lemma angle_neg_left: "angle (-u) v = pi - angle u v"
```

As multiplying a vector by any negative scalar corresponds to negating the vector before 'stretching' or 'compressing' it, and as we have shown that multiplication by a positive scalar leaves angles invariant, the following result is reached:

```
lemma neg_scalar_mult_angle:
  assumes "(t :: real) < 0"
    shows "angle (t *R u) v = pi - angle u v"
```

In particular, the vector obtained by the multiplication of non-zero v by a negative scalar points in the opposite direction as v itself:

```
lemma neg_scalar_mult_angle_self:
  assumes "t < 0"
      and "v ≠ 0"
    shows "angle (t *R v) v = pi"
```

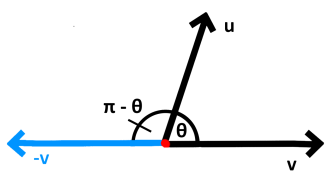

Figure 4.2: If one of two rays that form angle θ is negated, we obtain the supplementary angle of θ which, together with θ, adds up to π.

To conclude, the `real_inner` type class allowed us to define the notion of an angle that appears to match our geometric intuition. This justifies our requirement that any tenable representation of space should instantiate it. The following section acts as a brief exploration of the differentiation of vector-valued functions in Isabelle.

4.3 Vector derivatives

Analysing the rate of change of a quantity over time is the primary motivation for studying derivatives in the context of Newtonian mechanics. For instance, the velocity of a particle is the time derivative of its position. If we treat time as real-valued, we can model such derivatives in Isabelle using the function `vector_derivative` provided in the *Derivative* theory of the HOL-Analysis library. By unfolding and rewriting its definition, we managed to confirm that this library function allows us to capture the intended meaning of the derivative of function f at point t, as the unique value L (assuming its existence) such that $\left|\frac{f(t+h)-f(t)}{h} - L\right| \longrightarrow 0$ as $h \longrightarrow 0$.

```
lemma vec_deriv_limit:
    fixes f :: "real ⇒ ('a::real_normed_vector)"
  assumes "f differentiable (at t)"
    shows "vector_derivative f (at t) = (THE L.
         ((λh. norm (((f (t + h) - f t) /R h) - L)) ⟶ 0) (at 0))"
```

Here we employed the definite description operator `THE` which, in Isabelle/HOL, can be used to denote the unique constant of a type for which a certain property holds. In cases where no unique `x` such that `P x` exists, `THE x. P x` is effectively meaningless since `P (THE x. P x)` can then *not* be proved.

As shown above, we can map any point $t \in \mathbb{R}$ to the instantaneous rate of change of a given function f at t using the `vector_derivative` function. When seen as a function on $\mathbb{R}$, this map corresponds to what one would generally call 'the derivative of f' and can be defined in Isabelle as follows:

```
definition vector_derivative_fun ::
  "(real ⇒ ('a :: real_normed_vector)) ⇒ (real ⇒ 'a)"
  where "vector_derivative_fun f = (λt. vector_derivative f (at t))"
```

Using the library function `has_vector_derivative`, we confirmed that the newly defined `vector_derivative_fun`, when given a function f as input, can be evaluated at any point $t \in \mathbb{R}$ at which f is differentiable to yield the value of the derivative of f at t.

```
corollary vector_derivative_fun_works:
  assumes "f differentiable (at t)"
    shows "(f has_vector_derivative (vector_derivative_fun f t))
           (at t)"
```

For our axiomatisation of the motion of particles (cf. section 5.1), we defined the acceleration of a particle at each point in time as the second derivative of its position. Having formalised the notion of a 'derivative as a function', we could define a function f to be twice differentiable if both f itself and its derivative are differentiable.[5]

[5]For each of the following definitions and results regarding the function `differentiable`, we also implemented a corresponding version for the library function `differentiable_on`, which denotes whether a function is differentiable for the filter `at t within S` for each t in the set S.

```
definition twice_differentiable ::
  "(real ⇒ ('a :: real_normed_vector)) ⇒ real filter ⇒ bool"
  where "twice_differentiable f F = ((f differentiable F) ∧
         ((vector_derivative_fun f) differentiable F))"
```

Because we anticipated that we may also be interested in functions that are differentiable more than twice, and because we felt it would be unelegant to define distinct predicates for functions that are differentiable three times, four times, etc., we recursively defined the n-th derivative of any function f to be the result of applying the previously defined `vector_derivative_fun` to f n times. We did end up using this definition in our later lemmas regarding MacLaurin series (cf. section 4.4, 5.3). Note that the 0-th derivative of a function is simply the function itself.

```
fun nth_vector_derivative_fun where
  "nth_vector_derivative_fun 0 f = f" |
  "nth_vector_derivative_fun (Suc n) f = vector_derivative_fun
   (nth_vector_derivative_fun n f)"
```

We could then use this generalisation to define a function f to be n times differentiable at any real filter at which f and its first $n-1$ derivatives are differentiable.

```
definition n_times_differentiable ::
"nat ⇒ (real ⇒ ('a :: real_normed_vector)) ⇒ real filter ⇒ bool"
  where
  "n_times_differentiable n f F = (∀(m::nat). (m < n) ⟶
   ((nth_vector_derivative_fun m f) differentiable F))"
```

It follows from this definition that every function is (at least) '0 times differentiable' at any filter. Next, we confirmed that the library function `differentiable` and the previously defined `twice_differentiable` are equivalent to `n_times_differentiable` for inputs 1 and 2, respectively.

To conclude our current exploration, we then provided a recursive generalisation of the existing function `has_vector_derivative`. The library function takes a function f with domain $\mathbb{R}$, a value f' from its co-domain and a filter F on $\mathbb{R}$ and returns `True` if f' is the derivative of f at F. Our novel function `has_nth_vector_derivative` additionally takes a natural number n. Its output denotes whether the '(n-1)-th' derivative of f `has_vector_derivative` f' at the filter F.

The following result, which can be seen as a generalisation of the existing lemma `vector_derivative_works`, confirmed that the functions defined above combine as intended, wrapping up our formalisation of higher order derivatives.

```
lemma nth_vector_derivative_fun_works:
  assumes "n_times_differentiable n f (at t)"
    shows "has_nth_vector_derivative n f
           (nth_vector_derivative_fun n f t) (at t)"
```

4.4 Tangents, normal vectors and arc length

The previous two sections have shown that the representation of physical space as a type variable of sort `euclidean_space` allows for the intuitive notion of ‘direction’ to be formalised, and that the HOL-Analysis library allows us to apply techniques from standard differential calculus to functions from $\mathbb{R}$ to such spaces. In the following, we will consider an example for how these notions may be combined in the analysis of curves.

The tangent line to the curve of function f at point $f(x)$ can be expressed in infinitely ways. The most common way is given by $f(x) + \lambda *_R \frac{f'(x)}{|f'(x)|}$ [20]. The fraction in this formula is the unit vector pointing in the same direction as the rate of change of f at x. As we will see, this ‘unit tangent vector’ has several interesting properties.

In order to define and explore this notion in Isabelle, we started by implementing a function which takes an arbitrary vector and returns the unit vector in the same direction.

```
definition unit_vector :: "('a :: real_normed_vector) ⇒ 'a"
  where "unit_vector v = (1 / norm v) *R v"
```

This function can be rewritten slightly more concisely in the following form:

```
corollary unit_vec_div: "unit_vector v = v /R (norm v)"
```

It was trivial to confirm that ‘normalising’ any non-zero vector in this way leaves its direction unchanged, using our earlier result `pos_scalar_mult_angle_0` (cf. section 4.2). Consequentially, all vectors that point in the same direction have the same unit vector, and vice versa.

```
lemma unit_vec_represents_direction:
  assumes u_non_zero: "u ≠ 0"
      and v_non_zero: "v ≠ 0"
    shows "(angle u v = 0) ⟷ (unit_vector u = unit_vector v)"
```

Since unit vectors can thus be used to uniquely describe the direction of a vector, it is intuitively useful to employ them in the discussion of the movement of objects in classical mechanics. For instance, if function f can be interpreted as the time-dependent position of some particle and one wants to describe the direction of movement at time t, the most natural answer to this question is given by the aforementiod unit tangent vector:

```
definition unit_tangent_vector ::
  "(real ⇒ 'a) ⇒ real ⇒ S('a :: real_normed_vector)"
  where "unit_tangent_vector f t = unit_vector
                                    (vector_derivative f (at t))"
```

Conversely to this definition, the `vector_derivative` of f `at t` is given by `(norm (vector_derivative r (at t))) *R (unit_tangent_vector r t)`.

Like `vector_derivative_fun f` (cf. section 4.3), `unit_tangent_vector f` denotes a function of type `real ⇒ 'a`. Interestingly, it is orthogonal to its own derivative at any t for which certain properties hold. Particularly, the derivative of f needs to be defined and continuous on an interval surrounding t from both sides.

```
lemma unit_tangent_vec_orthogonal_unit_tangent_vec_deriv:
  assumes "(unit_tangent_vector f) differentiable (at x)"
      and "f C1_differentiable_on S"
      (* f is ‘continuously differentiable’. *)
      and "x ∈ S"
      and "open S"
      and "vector_derivative f (at x) ≠ 0"
    shows "orthogonal (unit_tangent_vector f x)
         (vector_derivative (unit_tangent_vector f) (at x))"
```

As permitted by the above result, the derivative of the unit tangent vector function of a curve is often written in the form $\kappa *_R \mathbf{n}$, where the scalar κ is called the ‘curvature’ of f and $\mathbf{n}$ is a unit vector called the ‘principal unit normal vector’ of the curve at the given point. These entities are of fundamental importance to the field of differential geometry, and arise more commonly in Lagrangian and Hamiltonian than Newtonian mechanics [16]. Hence, although we felt it was appropriate to include their definition in our exploration of physical space, we did not delve into their geometrical interpretation more deeply. They will however reappear in our discussion of arc length towards the end of this chapter.

```
definition curvature :: "(real⇒'a) ⇒ real ⇒ real"
  where "curvature f x = norm (vector_derivative
                              (unit_tangent_vector f) (at x))"

definition principal_unit_normal_vector ::
  "(real⇒'a) ⇒ real ⇒'a"
  where "principal_unit_normal_vector f x =
         unit_vector (vector_derivative
                     (unit_tangent_vector f) (at x))"

lemma unit_tangent_vector_deriv:
  shows "vector_derivative (unit_tangent_vector f) (at x) =
        (curvature f x) *R (principal_unit_normal_vector f x)"
```

By requiring that any representation of physical space be normed, we ensured that it is possible to define a distance between any pair of points. In a mechanical context, we may further be interested in computing the distance that some particle has travelled over the course of a given time frame. Indeed, the tools of standard analysis allow us to define the arc length of any continuously differentiable curve in a Euclidean vector space. Since this ability can be said to validate the usefulness of this mathematical toolkit for the study of plane curves (and, hence, particle motion), we defined the arc length of curve f over an interval from a to b (corresponding to a finite time frame) as follows:

```
definition arc_length ::
  "(real ⇒ ('a :: real_normed_vector)) ⇒ real ⇒ real ⇒ real"
  where "arc_length f a b = integral {a..b}
                            (λt. norm ((vector_derivative f)
                            (at t)))"
```

It is worth pointing out that this expression can only be interpreted in the desired way if the function f is continuously differentiable [2]. Since the acceleration of a particle is defined to be the second derivative of its position with respect to time, its position function is generally assumed to be twice differentiable (cf. section 4.3). Since differentiability implies continuity,[6] it follows that such position functions are continuously differentiable. Hence, we can conveniently think of the argument f in the above definition as the position of some particle.

The `arc_length` function has interesting properties with respect to differentiation, which we will discuss in the following. Consider the function $f : \mathbb{R} \to S$, mapping real-valued inputs to points in some Euclidean vector space. Suppose that we are able to find a second function, $g : \mathbb{R} \to S$ that, when given the length of the curve of f, measured from some fixed starting point $f(a)$ up to any point $f(t)$, returns the value $f(t)$. This setup is illustrated in figure 4.3.

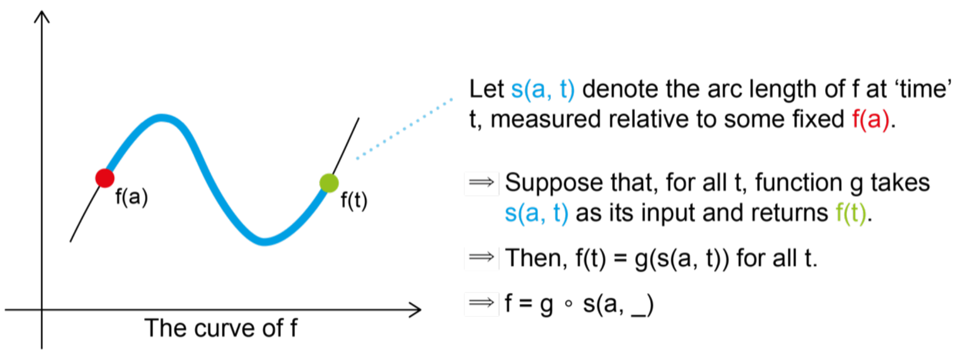

Figure 4.3: An illustration of what it means to write f as a function of its own arc length.

Suppose further that f is continuously differentiable (at least) on an interval $[a,b]$ and that x is some value such that $a \leq x < b$. Lastly, assume that the derivative of f at x is not equal to 0, and that g is differentiable at the real number that equals the length of the curve of f between $f(a)$ and $f(x)$.

When we now differentiate g at (`arc_length f a x`), the value we receive denotes the instantaneous rate of change of f with respect to its arc length, starting at the fixed point $f(a)$. By assuming that f is continuously differentiable on $[a,b]$, we are ensuring that the derivative of f with respect to time is continuous. Therefore, the direction of sufficiently small changes of f is approximated by its tangent at the value $f(x)$ with arbitrary precision. One may intuit that, since the behaviour of f at an 'infinitesimally small' region around input x is thus roughly linear, tiny changes in f are linked to an approximately equal change in its arc length. We formally proved that this intuition is correct.

```
lemma arc_length_deriv_norm_1:
  assumes "(f: real⇒('a :: real_normed_vector)) =
             g ∘ (arc_length f a)"
      and "x ∈ {a..<b}"
      and "continuously_differentiable_on reals f"
      and "(g::real⇒'a) differentiable (at (arc_length f a x))"
      and "vector_derivative f (at x) ≠ 0"
    shows "norm (vector_derivative g (at (arc_length f a x))) = 1"
```

[6] This fact is proved in the *Derivative* theory of the HOL-Analysis library.

On the one hand, this result is instructive because it suggests that differentiating with respect to the arc length of a curve may, in some cases, simplify calculations, because a 'normalised' quantity is obtained. On the other hand, and perhaps more importantly, the above elaboration exemplifies the meaning of derivatives as the ratio between infinitesimal changes.

Under the assumptions of the above lemma `arc_length_deriv_norm_1`, it trivially follows that, when f can be given as a function of its own arc length, its derivative and unit tangent vector with respect to this quantity are identical.

```
lemma tangent_vector_arc_length_deriv:
 assumes (...)
 shows "vector_derivative g (at (arc_length f a x)) =
        unit_tangent_vector g (arc_length f a x)"
```

Lastly, we used the mechanisation of MacLaurin series in the HOL-Analysis library to conclude our exploration of differentiation with respect to the arc length of a curve. Namely, suppose that $f : \mathbb{R} \to \mathbb{R}$ is continuously differentiable and can be expressed as a function g with respect to its own arc length. Without loss of generality, we can assume that the arc length is measured relative to the fixed point $(0, f(0))$. Under a set of assumptions about the differentiability of the involved functions and the behaviour of f near 0, we managed to show that, for small positive values h, $f(h)$ can be approximated by an expression involving $\mathbf{t}, \kappa$ and $\mathbf{n}$, that is, the unit tangent vector, curvature and unit normal vector of the function g.

```
lemma Maclaurin_arc_length_curve:
   fixes f :: "real ⇒ real"
 assumes "f = g ∘ (arc_length f 0)"
 (* Differentiability *)
     and "f C1_differentiable_on reals"
     and "n_times_differentiable_on 3 reals g"
     and "(unit_tangent_vector g) differentiable (at 0)"
 (* Interval {0..<h} *)
     and "h > 0"
     and "∀t∈{0..<h}. vector_derivative f (at t) ≠ 0"
     and "(at 0 within arc_length f 0 ` {0..<h}) ≠ bot"
   shows "∃s>0. s < h ∧ g h = f 0 + (unit_tangent_vector g 0) * h
          + ((curvature g 0) / 2 * h ^ 2) *R
          (principal_unit_normal_vector g 0)
          + nth_vector_derivative_fun 3 g s / fact 3 * h ^ 3"
```

Do note that the approximation we have deduced was only shown to hold for real-valued curves. Douglas claims that the approximation $g(h) = f(0) + h *_R \mathbf{t} + (\frac{1}{2}\kappa h^2) *_R \mathbf{n} + O(h^3)$ shows that curve g lies approximately in a plane spanned by $\mathbf{t}$ and $\mathbf{n}$ near 0, requiring the co-domain of g to be at least 3-dimensional. Verifying this claim in Isabelle would require an implementation of Taylor approximations for vector-valued functions [11]. This current limitation of the HOL-Analysis library would also impede the generalisation of our results regarding MacLaurin series in the context of one-dimensional simple harmonic motion (cf. section 5.3).

4.5 Conclusion

In this chapter, we have explored some general properties of functions from $\mathbb{R}$ to any real normed vector space and, hence, any Euclidean vector space. This discussion can be taken as evidence that the time-dependent position of a particle in the context of Newtonian mechanics may be meaningfully represented as a function of a corresponding type. In particular, we have seen that this representation allows for an analysis of the trajectory of particles, including their instantaneous rate of change (given by `vector_derivative`), direction (using our `angle` function), and the distance travelled in any time frame (obtained via integration as the `arc_length` of the curve).

Chapter 5

An axiomatisation of Newtonian mechanics

This chapter describes our formalisation of a set of axioms for the motion of point particles in the context of Newtonian mechanics. We begin by describing and justifying the axioms we chose. The majority of this chapter is then devoted to a summary of our exploration of the consequences of our axiomatisation, reproducing a range of basic results from the area of Newtonian mechanics.

5.1 The axioms

In this section, we present the fundamental entities and assumptions that form the basis of our formalisation of Newtonian mechanics. The scope of our discussion was delimited using a locale (cf. section 3.2.2), the header of which is shown in full below. Subsequently, we elaborate its contents.

```
locale newtonian_system_of_particles =
  fixes position :: "'particle ⇒ real ⇒ ('space::euclidean_space)"
    and mass :: "'particle ⇒ real"
    and force_on_due_to:: "'particle ⇒ 'particle ⇒ real ⇒ 'space"
    and external_force :: "'particle ⇒ real ⇒ 'space"
    and G :: "real" (* Gravitational constant *)

  fixes velocity :: "'particle ⇒ real ⇒ 'space"
defines "velocity p ≡ vector_derivative_fun (position p)"
  fixes acceleration :: "'particle ⇒ real ⇒ 'space"
defines "acceleration p ≡ vector_derivative_fun (velocity p)"

  fixes inter_particle_force :: "'particle ⇒ real ⇒ 'space"
defines "inter_particle_force p t ≡
         infsum (λq. force_on_due_to p q t) UNIV"
  fixes net_force :: "'particle ⇒ real ⇒ 'space"
defines "net_force p t ≡
         (external_force p t + inter_particle_force p t)"

assumes position_twice_diffable:
          "twice_differentiable (position p) (at t)"
```

```
    and newtons_second_law:
          "(mass p) *R (acceleration p t) = net_force p t"
    and newtons_third_law:
          "force_on_due_to p q t = -force_on_due_to q p t"
    and positive_mass: "mass p > 0"
    and positive_G: "G > 0"
```

In order to formalise the interaction of particles, we had to choose a way of representing each individual particle. We decided to implement all properties of particles as facts about functions which take them as inputs. Therefore, we declared a type variable `'particle` without assuming anything about it; when instantiating the locale, any type in Isabelle (including infinite ones like `real`, as well as finite ones like `bool`) may hypothetically be plugged in for `'particle`. Since this choice has no impact on the properties of particles, apart from their number, it is appropriate to think of the elements of the `'particle` type as indexing the system of particles considered in any particular instantiation of our locale. Each particle is assumed to have a position in space (where space is represented by the type variable `'space` of type class `euclidean_space`, as discussed in section 4.1.) at each point in time $t \in \mathbb{R}$. It would have been sensible to restrict time to be positive, by fixing the domain of all relevant functions to be $\mathbb{R}_{\geq 0}$, where time 0 would have denoted the start of our observations. However, no such restriction is necessary from a theoretical perspective, as evidenced, for example, by discussions on time-reversed systems in the context of classical mechanics [48].

The velocity and acceleration of a particle at time t are given by the first and second derivative, respectively, of its `position` function. In order to ensure that these quantities are well-defined, we required the position of any particle to be twice differentiable (cf. section 4.3).

Note that, although types with infinitely many elements can be used to serve as an instantiated version of `'particle`, providing, e.g., a meaningful position function for each of infinitely many particles would be challenging. This is a practical issue, but we saw no reason to prohibit infinite systems of particles in principle.

Moreover, it is worth noting that we implicitly assume that particles have no spatial extent; this simplifying assumption is a common one in physics and, more specifically, in classical mechanics [41, 26].

Apart from a position, each particle has a mass, which is assumed to be constant. Of course, there are scenarios in which it may be necessary to analyse the change in mass of an object over time (e.g. to consider the changing momentum of a rocket as it gets lighter due to fuel consumption) but we were not concerned with such cases in this project.

Furthermore, we fixed a real-valued gravitational constant, `G`, which plays a crucial role in Newton's law of universal gravitation (cf. section 5.2). We deemed it inappropriate to assign a precise numerical value to this quantity. A sufficient reason for this was that the gravitational constant can only be approximated through measurements in the real world [3], so that knowing its precise value is clearly not needed to develop a theory of Newtonian mechanics. It would be the responsibility of any Isabelle user instantiating our locale to plug in an appropriate value for `G`. However, we did assume `G > 0` which, as we will see, corresponds to the established empirical fact that gravity is an attractive, rather than repulsive force.

The remainder of the `newtonian_system_of_particles` locale header corresponds to the axioms that can be seen as defining Newtonian mechanics - Newton's laws of motion. In the first English translation of Newton's *Principia* by Motte [37], published in 1729, the laws are formulated as follows:

1. "Every body perseveres in its state of rest, or of uniform motion in a right line, unless it is compelled to change that state by forces impressed thereon."
2. "The alteration of motion is ever proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed."
3. "To every action there is always opposed an equal reaction; or the mutual actions of two bodies upon each other are always equal, and directed to contrary parts."

Evidently, in order to mechanise these axioms in Isabelle, we had to implement the notion of forces that may act on any particle. In particular, according to Newton's third law, bodies (or, in our case, particles) may exert a force on each other. We formalised this notion as a function `force_on_due_to` which assigns to each pair of particles p, q, and each $t \in \mathbb{R}$ the force exerted on p by q at time t. In our representation, forces are given as vectors of type `'space` which, as we have seen in section 4.2, are characterised by a magnitude and direction. There we also showed that two non-zero vectors are "equal and directed to contrary parts" exactly if they are additive inverses of each other. This justifies our formulation of `newtons_third_law`.

The "alteration of motion" mentioned in Newton's second law can be understood as the rate of change of a body's velocity, i.e. its acceleration. Here, 'motive force' refers to the total, or net force acting on a particle at a certain time, where actions from different sources combine additively [26]. In our locale, the sum of the forces exerted on p by particles in the system at time t is called `inter_particle_force`. Because this may generally be a summation over infinitely many values, we made use of the `infsum` library function. In the case that there are only finitely many particles, this definition can easily be shown to be equivalent to a corresponding definition in terms of the more conventional `sum` function[1]. As alluded to above, we further accounted for the possible existence of a time-dependent external force that may act on each particle without being caused by another particle in the system.[2] In turn, the net force acting on a particle is the sum of the inter particle force and external force. In its original form, Newton's second law only states that the relationship between a particle's acceleration and the net force acting on it is linear. Denoting this net force as F and the acceleration as a for an arbitrary particle, we thus have $F = ma$, where m is some constant. In the way that Newton's second law is taught in modern textbooks, this constant is identified as the body's (inertial) mass [26].

[1]The *Infinite_Sum* theory in which `infsum` is defined was added to the Isabelle library for the December 2021 release, which was published in the midst of this project. Prior to this, we had assumed the number of particles to be finite.

[2]In physical experiments performed on the surface of Earth, the gravitational pull of the planet acting on each observed object is one obvious cause for such external forces.

Note the use of `UNIV` in the definition of `inter_particle_force`. In Isabelle/HOL, this abbreviation denotes the set of all elements of a certain type (or type variable), in this case `'particle`.

Moreover, note that if the mass of any particle p was 0, its acceleration could no longer be said to be proportional to its net force given our above formalisation of Newton's second law. Furthermore, if this mass was negative, the resulting alteration of motion would be made *opposite* the direction of the impressed force, contradicting Newton's formulation of the second law (cf. section 5.1). Hence, to ensure that `newtons_second_law` captures the intended meaning, we had to assume that masses are strictly positive.

In fact, this concludes our locale header. To see why it contains no mention of Newton's first law, consider the following:

At least on the surface, the first law states that the velocity of any particle remains unchanged unless a (net) force acts on that particle. In other words, if the net force acting on a particle is 0, the particle is not accelerated. Given our aforementioned interpretation of Newton's second law ($F = ma$), the content of Newton's first law, interpreted in the above manner, follows easily. We could even prove the converse of the above formulation of the law: If the net force acting on a particle is 0, it is not accelerated. This finding resulted in the following equivalence:

```
lemma newtons_first_law:
  shows "(acceleration p t = 0) ⟷ (net_force p t = 0)"
```

It should be noted that some authors have deemed all interpretations that view Newton's first law as a mere consequence of the second law overly simplistic. E.g., Galili and Tseitlin (2003) argue that the first law contains subtle implications on the temporal nature of the relation between changes in force and velocity, which are not fully captured by Motte's famous translation of Newton's *Principia* [19]. However, most physicists over the past century have adopted the view that Newton's first law is indeed a consequence of the second [52, 27].

Moreover, Gregory (2006) states that Newton's first law proclaims the existence of an observer from whose perspective measurements are taken, and who is itself unaccelerated with respect to the observed particles [20]. Similarly, our choice of representing space as a Euclidean vector space in Isabelle assumes the existence of an origin (the 0 vector, corresponding to the observer's position) and a set of coordinate axes, in the form of an orthonormal `Basis`, with respect to which the position of any particle may be expressed. If Gregory's interpretation is accepted, Newton's first law reassures us that it is indeed possible to fix such a reference frame, in which the observed acceleration of any particle can be understood as a property of the particle itself, and is hence caused by forces acting on it, rather than being brought about by a relative acceleration of the utilised reference frame. Then, it can be argued that Newton's first law can only be proved in our locale because our representation of the position of particles implicitly assumes its validity.

Ultimately, we do not view the apparent redundancy of Newton's first law in the context of our formalisation as worrisome.

5.2 Corollaries and forces

Having decided on a set of axioms, we explored their immediate consequences before moving on to defining new entities based on the ones appearing in the locale header.

First, note that we had a lot of freedom when deciding how to formulate the assumption that the `position` of any particle may be differentiated twice at any point. To confirm that the `position_twice_diffable` axiom, which we had expressed in terms of our own function `twice_differentiable`, captured the intended meaning, and to make it easier to employ it in proofs, we proved the validity of several alternative ways of expressing the axiom's content. Namely, we phrased the differentiability of `position` and `velocity`, as well as the relations between them and the `acceleration` function in terms of the library functions `differentiable`, `differentiable_on` and `has_vector_derivative`.

Next, it is worth pointing out that Newton's second law is often given in a form other than $F = ma$ in the literature. Commonly, it is instead presumed that the force acting on a particle is equal to the time derivative of its momentum, where the latter is given by the product of the particle's mass and acceleration [17]. We showed that this version of the law arises as a consequence of our axioms.

```
definition momentum :: "'particle ⇒ real ⇒ 'space"
  where "momentum p t = (mass p) *R (velocity p t)"

lemma force_is_momentum_deriv:
  "((momentum p) has_vector_derivative (net_force p t)) (at t)"
```

It is crucial to note that the two aforementioned formulations of Newton's second law are only equivalent if the mass of the relevant particle is assumed to be constant. Indeed, only the version $F = \frac{d(mv)}{dt}$ is considered appropriate in the context of variable mass systems [44]. Because we chose not to consider such systems in our formalisation, we did not change our formulation of `newtons_second_law` upon discovering this ambiguity. Hence, care would have to be taken if particle masses were to be allowed to change over time in a future version of the locale.

As an immediate consequence of Newton's third law, it is effectively impossible for any particle to inflict any force on itself because any such action would be paired with an equal reaction in the opposite direction, so that the two would cancel out. Correspondingly, the following fact could easily be shown:

```
lemma no_force_on_itself: "force_on_due_to p p t = 0"
```

Moreover, because the two forces between each pair of particles in our system cancel out, it follows that, at each point in time, the forces invoked on each particle by other particles in the system add up to 0, provided that this sum is well-defined, i.e., attains a value. In Isabelle, this condition on the summation over particle pairs is represented by the `summable_on` function from the *Infinite_Sum* theory file. In particular, the lemma is guaranteed to hold if there are only finitely many particles in our system, because any sum of finitely many vectors is defined.

```
lemma summable_inter_p_force_0:
  assumes "(λ(p, q). force_on_due_to p q t) summable_on (UNIV×UNIV)"
  shows "infsum (λp. inter_particle_force p t) UNIV = 0"
(...)
 lemma finite_inter_p_force_0:
  assumes "finite (UNIV :: 'particle set)"
  shows "infsum (λp. inter_particle_force p t) UNIV = 0"
```

Although our axioms relate the net force acting on any particle to its mass and acceleration, they do not generally allow us to derive the value of the force between any pair of particles, even if the latter two quantities are given. Precise laws to that effect can however be formulated for certain types of forces between pairs of particles, such as gravity. Kibble and Berkshire identify central, conservative forces as constituting the most important class of forces [26].

Any appropriately typed function F embodies a central, conservative force on particle p caused by particle q if, at each point in time, the force (of this type) acting on p due to q is directed along the line joining them, such that the magnitude of the force depends solely on the distance of the particles. In such cases, there is a function f which maps the distance of the two particles to the magnitude of F at any time t (perhaps multiplied by -1), such that `F p q t` is given by the scalar product of the value of f and the unit vector in the direction from particle q to particle p. Moreover, to qualify as a force in the first place, we demanded that the relevant function needs to satisfy Newton's third law.

```
definition is_force ::
  "('particle ⇒ 'particle ⇒ real ⇒ 'space) ⇒ bool"
  where "is_force F = (∀p q t. F p q t = -F q p t)"

(* At any point in time t, 'relative_position p q t' is the vector
   pointing from the position of q to the position of p. *)
definition relative_position ::
  "'particle ⇒ 'particle ⇒ real ⇒ 'space"
  where "relative_position p q t = position p t - position q t"

definition central_conservative_with_func ::
  "('particle ⇒ 'particle ⇒ real ⇒ 'space) ⇒
    'particle ⇒ 'particle ⇒ (real ⇒ real) ⇒ bool"
  where "central_conservative_with_func F p q f ≡ is_force F ∧
         (∀t. (f(particle_distance p q t) *R
              unit_vector(relative_position p q t)) = (F p q t))"
```

Above and in the following, `particle_distance p q t` denotes the Euclidean distance between the positions of particles `p` and `q` at time `t`.

Any central, conservative force on particle p due to particle q is called 'attractive' at a given time if it acts in the direction from p to q. Since `relative_position p q t` (and, hence, its unit vector, cf. section 4.4) points in the direction from q to p, this is the case if the corresponding scalar value of f in the definition of `central_conservative_with_func` is strictly negative. Similarly, if this scalar value is strictly positive, the relevant force is called 'repulsive', as p is then pushed away from q (cf. figure 5.1).

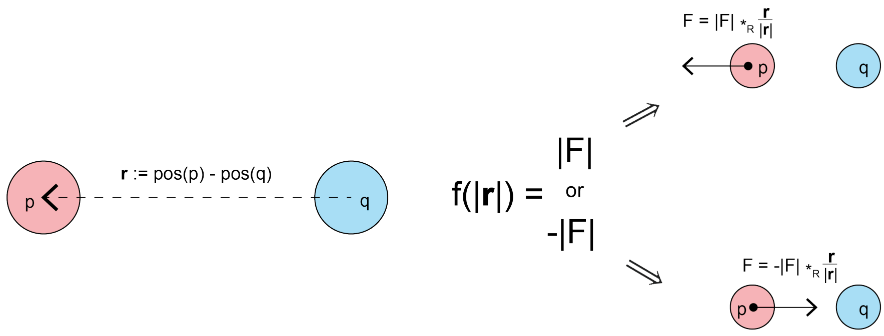

Figure 5.1: Let $\mathbf{r}$ denote the vector pointing from particle q to particle p. A central conservative force acting on p due to q can, by definition, be expressed in the form $F = f(|\mathbf{r}|) *_R \frac{\mathbf{r}}{|\mathbf{r}|}$ for some function f. The sign of $f(|\mathbf{r}|)$ determines the direction of the force.

By Newton's third law, an attractive force on p due to q is always paired with an attractive force on q due to p. The same holds for repulsive forces (cf. figure 5.2). For example, for attractive forces, we wrote the following:

```
definition attractive_cent_cons_force_with_func
  where "attractive_cent_cons_force_with_func F p q f t =
         (central_conservative_with_func F p q f ∧
         f(particle_distance p q t) < 0)"

definition attractive_cent_cons_force
  where "attractive_cent_cons_force F p q t =
         (∃f. attractive_cent_cons_force_with_func F p q f t)"

lemma attractive_symmetric: "attractive_cent_cons_force F p q t =
                             attractive_cent_cons_force F q p t"
```

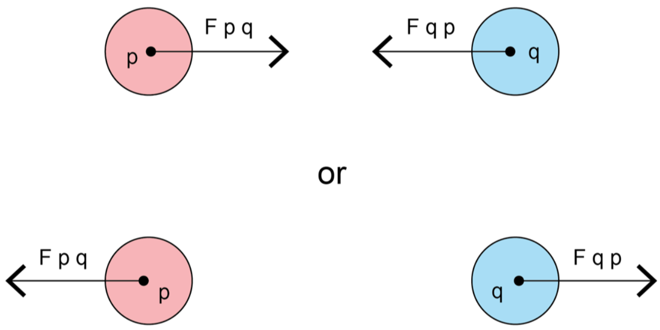

Figure 5.2: By Newton's third law, attractive forces always come in pairs. The same is true for repulsive forces. Here, $F\,pq$ denotes the force acting on particle p due to q.

Having defined the above notions, we were ready to formalise Newton's law of universal gravitation, which states that the gravitational force between two particles acts inwards with a magnitude of $G\frac{m_1 m_2}{r^2}$, where G is the gravitational constant, m_1, m_2 are the masses of the particles, and r is their distance. We proceeded by defining the function `gravitational_force_on_due_to`, before confirming that it has the expected magnitude and is indeed an attractive, central, conservative force.

```
definition gravitational_force_on_due_to ::
    "'particle ⇒ 'particle ⇒ real ⇒ 'space"
  where "(gravitational_force_on_due_to p q t) =
          -(G * (mass p) * (mass q) /
          ((particle_distance p q t) powr 2)) *R
          (unit_vector(relative_position p q t))"

lemma gravity_magnitude:
  shows "norm (gravitational_force_on_due_to p q t) =
          G * (mass p) * (mass q) /
          ((particle_distance p q t) powr 2)"
(...)
lemma gravity_is_attractive_cent_cons:
  assumes "position p t ≠ position q t"
    shows "attractive_cent_cons_force
           gravitational_force_on_due_to p q t"
```

A subtlety arises in the case that two particles share the same `position`. Then, because division by 0 yields 0 in Isabelle (cf. section 4.2), the magnitude of the gravity between the two particles is 0. Given that we had to assign some real number to this quantity, we argue that this option is the most sensible, since it is unclear in which direction the force would otherwise act. Our choice explains the presence of the `position_not_equal` assumption in the `gravity_is_attractive_cent_cons` lemma above.

Note that we have still not seen a way of deducing the value of `force_on_due_to` for any pair of particles. And indeed, this is impossible, unless we assume that we can account for all types of force acting between the pair. For instance, one may choose to equate the fundamental `force_on_due_to p q t` and the defined `gravitational_force_on_due_to p q t` if one has reason to believe that all types of force other than gravity are negligible. Generally however, additional types of force may be at play. For example, Coulomb's law quantifies the force between particles that do not move relative to each other, and that carry an electrical charge. To provide an example of how our `newtonian_system_of_particles` locale may be extended to incorporate additional entities or assumptions, and to apply our definitions on central conservative forces to a kind of force other than gravity, we formalised Coulomb's law in the following locale:

```
locale system_of_stationary_point_charges =
           newtonian_system_of_particles +
    fixes charge :: "'a ⇒ real" (* 'a ≡ 'particle.*)
      and ε_0 :: "real" (* electric constant, for Coulomb's Law. *)

  assumes positive_electric_constant: "ε_0 > 0"
      and stationary: "∃r. ∀t. relative_position p q t = r"
```

According to Coulomb's Law, the electric force between two stationary, point particles with charges q_1, q_2 and a distance of r has a magnitude equal to $\frac{1}{4\pi\varepsilon_0}\frac{|q_1q_2|}{r^2}$, where ε_0 refers to the electric constant fixed above. Like the gravitational constant, ε_0 has been measured with great, but not perfect, precision, which is part of the reason we merely assume it to be positive rather than fixing an exact value for it [4]. Due to the clear similarities between Newton's previously discussed law of universal gravitation and Coulomb's law, it was straightforward to formalise the latter and prove that it is `central_conservative`.

```
definition K :: real
  where "K = 1 / (4 * pi * ε_0)" (* Coulomb's constant. *)

definition coulomb_force_on_due_to :: "'a ⇒ 'a ⇒ real ⇒ 'b"
  where "coulomb_force_on_due_to p q t =
         (K * (charge p) * (charge q) /
         ((particle_distance p q t) powr 2)) *R
         unit_vector(relative_position p q t)"

lemma coulomb_cent_cons:
  shows "central_conservative coulomb_force_on_due_to p q"
```

Note that the definition of `coulomb_force_on_due_to` was chosen such that the force acting on particle p due to q points in the direction from q to p if the product of the charges is positive (i.e., if the charges have the same sign) and in the opposite direction if the product is negative. This corresponds to the well-known fact that same-sign charges repel, while opposite-sign charges attract.

```
lemma same_sign_charge_coulomb_repulsive_cent_cons:
  assumes "position p t ≠ position q t"
      and "(charge p) * (charge q) > 0"
    shows "repulsive_cent_cons_force coulomb_force_on_due_to p q t"

lemma opposite_sign_charge_coulomb_attractive_cent_cons:
  assumes "position p t ≠ position q t"
      and "(charge p) * (charge q) < 0"
    shows "attractive_cent_cons_force coulomb_force_on_due_to p q t"
```

Notably, the `external_force` that may act on any particle in our implementation has not appeared in any of our proofs or definitions. And indeed, few interesting things can be said about external forces, as their origins are, by definition, unknown within the context of our `newtonian_system_of_particles` locale. In an ideal scenario, experimentalists would be able to isolate a system of particles, in order to study its properties in the absence of any force not accounted for by the particles themselves. We modelled this situation using the assumption `∀ p. external_force p t = 0`. Note that the time variable t is not bound in this expression, so that we are implicitly fixing a particular point in time at which the assumption holds. This corresponds to the fact that scientists do not have to isolate a system of particles for all eternity in order to make measurements that depend on the system being in an isolated state. Whenever no external forces act on a particle, the entire `net_force` acting on it is accounted for by its `inter_particle_force`.

```
corollary isolated_net_force_inter_force:
  assumes "external_force p t = 0"
    shows "net_force p t = inter_particle_force p t"
```

The reason this is useful is that one can unfold the definition of `external_force` in order to express it as the sum of values of the `force_on_due_to` function, which are in turn related to the mass and acceleration of the particles in our system according to `newtons_second_law`. For instance, this relationship yields a formal procedure for measuring the mass ratio between two particles, given that they can be isolated and that there is some non-zero force between the two.[3]

```
lemma isolated_pair_force_due_to:
  assumes "∀p. external_force p t = 0"
      and "(UNIV :: 'particle set)  = {p, q}"
    shows "net_force p t = force_on_due_to p q t"

lemma isolated_pair_mass_measurable:
 assumes "∀p. external_force p t = 0"
     and "(UNIV :: 'particle set)  = {p, q}"
     and "force_on_due_to p q t ≠ 0"
   shows "(mass p) / (mass q) = (norm (acceleration q t)) /
                                (norm (acceleration p t))"
```

Moreover, the above intermediary result `isolated_pair_force_due_to`, in combination with `newtons_third_law`, entails that when two isolated particles interact with each other, the net forces acting on the two point in opposite directions. And since the acceleration of a particle is a positive scalar multiple of its net force according to Newton's second law, any two isolated particles are necessarily accelerated in opposite directions.

```
lemma isolated_pair_acc_angle_pi:
  assumes "∀p. external_force p t = 0"
      and "(UNIV :: 'particle set)  = {p, q}"
      and "force_on_due_to p q t ≠ 0"
    shows "angle (acceleration p t) (acceleration q t) = pi"
```

As our final notable result on isolated systems, we proved the law of conservation of momentum. Because the derivative of the momentum (mv) of any particle with constant mass is given by the net force ($F = ma$) acting on it at any point in time, the time derivative of the sum of momenta over all particles in the system is given by the corresponding sum of net forces. At least this is true if the number of particles is finite.[4] Then, if a system of particles is isolated over some period of time, the sum of momenta is constant within that time frame.

```
lemma conservation_of_momentum_isolated:
assumes isolated_system: "∀p.∀t∈{a<..<b}. external_force p t = 0"
  shows "∃c. ∀t∈{a<..<b}. (∑p∈UNIV. momentum p t) = c"
```

[3]If we fixed the `mass` of a particle of our choice to be equal to an arbitrary value, we would hence be able to assign a value to the `mass` of the second particle in the isolated system.

[4]There are cases in which infinite sums may be differentiated [6]. However, this does not hold for general `infsums` in Isabelle.

Note that we did not have to assume that the number of particles is finite in the above lemma, because any `sum` (∑) over infinitely many values is `0` in Isabelle.

5.3 Energy and simple harmonic motion

In this section, we briefly present our formalisation of one-dimensional simple harmonic motion. We introduce the concepts of kinetic and potential energy, formalise the notion of a particle being in equilibrium and derive an equation of motion for such particles. Moreover, we discuss certain notational difficulties we encountered regarding integrals in Isabelle.

In the textbook this section is based on [26], Kibble and Berkshire consider particles moving in one dimension, under a force that is given as a function of its position. Correspondingly, we instantiated [7] our `newtonian_system_of_point_particles` locale, plugging in the type `real` for the type variable `'space`, and fixing a function `F` that maps the position of a particle at any time t to the net force acting on it.[5]

```
locale linear_motion = newtonian_system_of_particles position mass
                       external_force force_on_due_to G
  for position :: "'a ⇒ real ⇒ real"
  and  mass external_force force_on_due_to G
   +
    fixes F :: "'a ⇒ real ⇒ real"
  assumes force_is_fun_of_pos: "F p (position p t) = net_force p t"
```

Note that extending the initial locale in this way is possible because the type `real` instantiates the type class `euclidean_space`.

In our initial locale, we had defined the kinetic energy of a particle `p` at time `t` as `(1/2)*(mass p)*(speed p t)`, where `speed p t = norm (velocity p t)`. Since `velocity p` is a real-valued function in the context of the `linear_motion` locale, the squared speed of a particle is the same as its squared velocity. Hence, kinetic energy can be differentiated with ease.

```
corollary kinetic_energy_vector_deriv:
  "((kinetic_energy p) has_vector_derivative
   (mass p * acceleration p t * velocity p t)) (at t)"
```

Thanks to Newton's second law and the fundamental theorem of calculus, the following result holds likewise:

```
lemma kinetic_energy_integral_of_power:
  assumes "a ≤ b"
    shows "kinetic_energy p b - kinetic_energy p a =
           integral {a..b} (λt. net_force p t * velocity p t)"
```

Here, integration is performed with respect to time over the interval from `a` to `b`. Crucially, the above lemma contains the assumption that b is no smaller than a. The commonly taught convention that swapping the bounds of an integral negates its value would remove the need for this assumption, as one can easily confirm. However, this

[5]Note the similarity of this setup to the results regarding arc length in section 4.4.

convention cannot easily be realised in Isabelle because the interval {`a..b`} is effectively only meaningful if `b ≥ a`. This notational restriction may seem insignificant at first but turned out to be quite inconvenient in the context of our current discussion, as it repeatedly compelled us to provide separate lemmas accounting for both possible orderings of the arguments, as exemplified by the following.

Remember that we assumed that function `F` can be used to denote the force acting on any particle as a function of its position. Under the additional assumption that `F p` is continuous, we managed to prove that its antiderivative (with respect to the particle's position) is given by the kinetic energy of `p`.

```
lemma kinetic_energy_integral_of_force_forward:
  assumes "a ≤ b"
      and "position p a ≤ position p b"
      and "continuous_on UNIV (F p)"
    shows "kinetic_energy p b - kinetic_energy p a =
            integral {position p a..position p b} (F p)"
```

Importantly, note the assumption `pos_a_to_pos_b`. Via integration by substitution, we managed to show that, if one replaces the above assumption by `position p a > position p b`, it follows that `kinetic_energy p a - kinetic_energy p b` is equal to `integral {position p b..position p a} (F p)`. This identity would match the above lemma if integral bounds could be swapped in Isabelle.

Moreover, the Isabelle restrictions on the ordering of interval bounds meant that we had to write the definition of potential energy, provided by Kibble and Berkshire as $V(x) = \int_{x_0}^{x} F(x')dx'$, in a piece-wise way:

```
definition potential_energy :: "'a ⇒ real ⇒ real ⇒ real"
  where "potential_energy p s t =
         (if position p t ≥ position p s then
             -integral {position p s..position p t} (F p) else
             integral {position p t..position p s} (F p))"
```

In our formalisation, s corresponds to an arbitrary point in time with respect to which the potential energy is measured. We managed to show that, for any particle p and any fixed choice of s, the kinetic and potential energy of p add up to the same constant, denoted `total_energy p s`, for all points in time *after* s. Next, we expressed the potential energy of a particle as a function of its position, rather than time.

```
definition potential_energy_pos :: "'a ⇒ real ⇒ real ⇒ real"
  where "potential_energy_pos p s x = (if x ≤ position p s then
         -integral {position p s..x} (F p) else
         integral {x..position p s} (F p))"
```

Given the two above definitions of potential energy, Isabelle's automated provers confirm with ease that they are equivalent, in that `potential_energy p s t` equals `potential_energy_pos p s (position p t)`.

Having thus explored 1-dimensional kinetic and potential energy in sufficient detail, we began our formalisation of harmonic oscillators. The most common example of harmonic motion in one dimension is given by a body attached to a spring that is swinging back and forth - in the absence of friction or similar sources of energy loss, such oscillators are called *simple* and the motion continues indefinitely.

One characteristic of the trajectory of particles in simple harmonic motion is that they include a point of equilibrium, at which the net force acting on the particle is 0. Assuming that the function `F p` is assumed to be continuous (a prerequisite which, by the way, is never spelled out explicitly by Kibble and Berkshire [26]), the derivative of `potential_energy_pos p a` at any point on the trajectory of the particle `p` *to the right* of `position p a` was shown to equal `F p`. Hence, conveniently taking the potential energy of a particle to be measured relative to the point `0 = position p 0`, the first order term in the MacLaurin expansion of the potential energy of `p` vanishes if `0` happens to be an equilibrium point. We proved the following:

```
lemma Maclaurin_V_approx:
 assumes "0 < h"
     and "position p 0 = 0"
     and "in_equilibrium p 0"
     and "continuous_on reals (F p)"
     and "∀x. (0 ≤ x ∧ x ≤ h) ⟶ n_times_differentiable
          3 (potential_energy_pos p 0) (at x)"
   shows "∃x>0. x < h  ∧ potential_energy_pos p 0 h =
          (nth_vector_derivative_fun 2
          (potential_energy_pos p 0) 0) / fact 2 * h^2 +
          (nth_vector_derivative_fun 3 (potential_energy_pos p 0) x)
          / fact 3 * h^3"
```

Next, by further making the simplifying assumption that the third derivative of the potential energy of `p` vanishes at all points between 0 and h, we obtained a precise expression for its potential energy $V(x)$ at any sufficiently small, positive position x; namely, for all $0 \leq x \leq h$, $V(x) = \frac{V''(0)}{2}x^2$.

Using our earlier result that, under some of the assumptions of `Maclaurin_V_approx`, `F p` is equal to the derivative of `potential_energy_pos p`, we differentiated both sides of the above identity to prove that `F p x` is equal to the product of `x` and the second derivative of `potential_energy_pos` at 0.

Lastly, we simply needed to apply Newton's second law to formally prove, under the aforementioned simplifying assumptions, an equation of motion for particles moving in one dimension near a point of equilibrium at the origin; $ma + V''(0)x = 0$.

5.4 Conclusion

In this chapter, we have proposed an axiomatisation of particle motion in the context of Newtonian mechanics, using the tools of standard analysis. We have proved several simple relationships between the fundamental entities of our framework, defined attractive and repulsive forces, and used Coulomb's law to exemplify how our basic theory can be expanded through additional axioms. Lastly, we explored how the theoretical foundations we formalised may be applied to the mechanical study of harmonic oscillators in one dimension. By doing so, we have demonstrated that our axioms, in combination with the tools of the HOL-Analysis library, are generally well-suited to emulate discussions on classical mechanics in modern textbooks. However, we have also discussed certain challenges that may arise in the formal study of integrals in Isabelle.

Chapter 6

Discussion

The theory of Newtonian mechanics has been applied to a large variety of problems, of which we were only able to scratch the surface in our implementation. Moreover, in the centuries after the publishing of Newton's *Principia* in 1687 [38], several laws extending the scope of Newtonian mechanics were published by different authors. For the most part, such extensions are absent from our formalisation. For instance, we did not discuss Euler's laws of motion, which allow for the treatment of rigid bodies with non-zero spatial extent, in addition to the point particles considered in our implementation [35]. Naturally, it was never going to be possible to formalise *'everything'*. Nevertheless, we could have potentially covered more ground, both in terms of applications and extensions of Newton's theory, had it not been for certain challenges that arose during the mechanisation of Newtonian mechanics in Isabelle. In the following section of this report, we shall discuss several such obstacles, before reviewing the results of our investigations in this project.

6.1 Challenges

Formulating rigorous mathematical proofs is not an easy task. Mathematicians commonly assume that their readers will remember relatively basic rules that justify some steps of a proof. For instance, applications of the chain and product rule of differentiation, or changes of variables, are rarely seen to be deserving of an explicit justification in modern mathematical texts because such techniques are well known to most mathematicians.

Indeed, these techniques are also 'known' to the Isabelle proof assistant, as relevant results have been proved in the HOL-Analysis library; but any Isabelle user who wishes to employ them in a proof needs to refer to the corresponding lemmas.[1] The Sledgehammer tool available in Isabelle can be helpful in this regard, as it may automatically find a set of proved facts that justify an attempted proof step. Nevertheless, a solid knowledge of the results on analysis in the Isabelle library was crucial for this project,

[1] It should be noted that some basic routines, such as addition, are understood by some automated provers built into Isabelle, like *simp* and *auto*, without the need of referring to specific rules.

in particular because it allowed us to express statements in proofs so that they matched the relevant library results as closely as possible, thereby facilitating the proofs.

Although it became easier as we gained practical experience over the course of the project, acquiring this knowledge was time-consuming and challenging, because the HOL-Analysis library is vast and it is not always clear where a specific result can be found. For instance, results about derivatives used in this project were distributed over the theory files *Green.Derivs*, *HOL-Analysis.Derivative*, *HOL.Deriv* and *Smooth_Manifolds.Analysis_More*. Although existing tools such as FindFacts [5] can be immensely useful when looking for a certain result, the primary way in which knowledge of the Isabelle library can be gained is by reading the theory files. To make this knowledge more accessible to users, we believe it is beneficial to investigate potential ways of providing a structured overview of the facts within the Isabelle library.

On a related note, this project has taught us the importance of getting a solid understanding of the definitions and results on which a project in Isabelle is to be based *early*. Rather than commencing our practical work in Isabelle as soon as possible, it may have been advantageous to spend a few weeks primarily studying the contents of the HOL/Analysis library. We would have saved time later on, by avoiding subtle confusions about the behaviour or meaning of certain entities[2], and by effectively having access to a wider range of library results, which would have allowed for simpler proofs.

Let us now consider challenges that arose during our mechanisation of Newtonian mechanics and which are not, or less strongly, related to the Isabelle proof assistant.

As a qualitative assessment of the proof techniques we employed in this project, it has to be said that 'the more analysis' a proof contained, that is, the more strongly it depended on the differentiation or integration of functions, the more difficult it felt. In particular, this became evident during our discussion of potential energy (cf. section 5.3). Certainly, this sentiment is at least partially subjective, and was lessened as we gained a better knowledge of the HOL-Analysis library over the course of the project. Nevertheless, it is striking that, for example, the entire contents of our discussion on angles (cf. section 4.2) were proved in a fraction of a work day, whereas some more analytical proofs, such as the finding that derivatives with respect to arc length are normalised (cf. section 4.3) required significantly more time and several revisions. Perhaps this is an unfair comparison, as the latter results were certainly more intricate. Nevertheless, our experience could be taken as light evidence in support of previous claims that the use of methods from nonstandard analysis for the discussion of curves allows for relatively simple and intuitive proofs compared to standard methods [18].

In particular, our proofs that involved MacLaurin approximations (cf. section 4.4, 5.3) turned out to be more laborious and time-consuming than their conclusions proved insightful. In hindsight, it may have been wise to shift our focus to other areas we left unexplored (such as the discussion of angular momentum, force fields or rigid bodies) upon realising this, rather than continue our attempts at refining these proofs. We believe that this, too, is an important takeaway from our work on this project.

[2] For instance, we had been long been unaware of the fact that any `sum` ($\sum$) over an infinite set evaluates to 0 in Isabelle. Hence, we retroactively changed the definition of `inter_particle_force` to use `infsum` instead, entailing necessary changes to several proofs, which would have best been avoided.

Overall, we feel that most of the challenges that arose over the course of the project can be overcome by gaining more experience as an Isabelle user. Indeed, formalising proofs became significantly easier the more accustomed we became to the results of the HOL-Analysis library. Moreover, it took time to gain an intuition for how to write relatively complex expressions in ways easily handled by automated provers. The Isabelle learning curve is steep, but we learned a lot as we progressed and are certain that our gained experience will help us in future projects.

6.2 Conclusion

In this project, we investigated some of the primary mathematical tools that are available in the discussion of motion in Euclidean space. We then formalised Newton's laws of motion and universal gravitation, which can be seen as forming the basis of the field of Newtonian mechanics, and proved basic relationships between the entities present in these laws. Furthermore, we explored how this fundamental framework may be extended to allow for the discussion of additional concepts, such as point charges. Lastly, we saw an example of how the mathematical tools of vector analysis may combine with Newton's laws to enable the formal study of specific types of motion, in the form of simple harmonic oscillations.

While we have not provided a holistic formalisation of a wide range of key results in Newtonian mechanics, we believe that the approach we chose for our formalisation, which was enabled by the Isabelle type class mechanism, is interesting. Making use of the `euclidean_space` type class for our representation of physical space encapsulated many implicit assumptions. Therefore, our `newtonian_system_of_particles` locale managed to subsist on a small number of simple, explicit axioms. Despite having encountered some practical difficulties when formalising proofs using the methods of standard analysis (cf. sections 5.3, 6.1), we believe that our axiomatisation was achieved well, as it is notationally simple and conceptually clear.

To assess how the practicality of our approach compares to alternative formalisations of classical mechanics, more work is needed. As a future research direction, one could attempt to solve a range of practical problems from the area of classical mechanics using our, and comparable, frameworks in Isabelle. Such experiments would allow for an informed comparison of the Newtonian, Lagrangian and Hamiltonian formalisms in terms of their ease of use in the context of interactive theorem proving.

If our approach is deemed promising, it may be interesting to extend the axioms we have chosen, to account for subsequent additions to the fundamental assumptions made by Newton in the *Principia*. In particular, Euler's laws, which allow for the discussion of rigid bodies, rather than solely point particles, could be considered. In this project, we have seen how a fundamental set of axioms may be expanded upon using the locale import [7] mechanism in Isabelle (cf. sections 3.2.2, 5.2). We conclude that this feature may be used to model the development of mathematical or physical theories over time, incrementally expanding their scope. The sequences of locales that would result from such research efforts could be interesting, not least because of their pedagogical value. If, in the future, the knowledge contained in the Isabelle library is made more accessible (cf. section 6.1), we believe it can potentially be a valuable source of precise formal insight to a larger scientific audience. Our work on this project certainly made us more aware of the mathematical rigour needed to prove basic results from classical mechanics.

Ultimately, we believe that, despite certain challenges, our efforts of formalising Newtonian mechanics on the basis of the HOL-Analysis library have shown that this is generally a promising approach that should be considered for future research.

Bibliography

[1] 2007. URL: https://plato.stanford.edu/archives/win2008/entries/newton-principia/.

[2] 2022. URL: https://en.wikibooks.org/wiki/Differential_Geometry/Arc_Length.

[3] 2022. URL: https://physics.nist.gov/cgi-bin/cuu/Value?bg.

[4] 2022. URL: https://physics.nist.gov/cgi-bin/cuu/Value?ep0.

[5] 2022. URL: https://search.isabelle.in.tum.de/.

[6] R. W. Bagley. "Differentiation of Infinite Series of Functions". In: *The American Mathematical Monthly* 66.10 (1959), p. 890. DOI: 10.2307/2309793.

[7] Clemens Ballarin. "Tutorial to locales and locale interpretation". In: *Contribuciones científicas en honor de Mirian Andrés Gómez*. Universidad de La Rioja. 2010, pp. 123–140.

[8] Mara Beller. "The conceptual and the anecdotal history of quantum mechanics". In: *Foundations of Physics* 26.4 (1996), pp. 545–557.

[9] N. Bourbaki. *General Topology: Chapters 1-4*. Elements de mathematique.lEnglish. Springer-Verlag, 1989. ISBN: 9780387193748. URL: https://books.google.co.uk/books?id=VotUAQAACAAJ.

[10] G. Cantor. "Ueber die Ausdehnung eines Satzes aus der Theorie der trigonometrischen Reihen". In: *Mathematische Annalen* 5.1 (1872), pp. 123–132. DOI: 10.1007/bf01446327.

[11] José E Chacón and Tarn Duong. "Higher order differential analysis with vectorized derivatives". In: *arXiv preprint arXiv:2011.01833* (2020).

[12] Giuseppe Da Prato. *An introduction to infinite-dimensional analysis*. Springer, 2006.

[13] E. Brian Davies. *Linear operators and their spectra*. Cambridge University Press, 2010.

[14] Richard Dedekind. *Continuity and irrational numbers*. Friedrich Vieweg und Sohn, 1872.

[15] Jean Eisenstaedt. *The curious history of relativity*. Princeton University Press, 2018.

[16] M. Fecko. *Differential Geometry and Lie Groups for Physicists*. Cambridge University Press, 2006. ISBN: 9781139458030. URL: https://books.google.co.uk/books?id=vQR0mN1dgUEC.

[17] Richard P Feynman, Robert B Leighton, and Matthew Sands. "The feynman lectures on physics; vol. i". In: *American Journal of Physics* 33.9 (1965), pp. 750–752.

[18] Jacques Fleuriot. *A Combination of Geometry Theorem Proving and Nonstandard Analysis with Application to Newton's Principia*. 1st ed. Springer-Verlag London, 2001.

[19] Igal Galili and Michael Tseitlin. "Newton's First Law: Text, Translations, Interpretations and Physics Education". In: *Science & Education* 12 (2003), pp. 45–73.

[20] R. Douglas Gregory. *Classical mechanics*. Cambridge University Press, 2006.

[21] Yong Guan et al. "Formalization of Euler–Lagrange Equation Set Based on Variational Calculus in HOL Light". In: *Journal of Automated Reasoning* 65 (2020), pp. 1–29.

[22] William Rowan Hamilton. *On a General Method of Expressing the Paths of Light, & of the Planets, by the Coefficients of a Characteristic Function*. PD Hardy, 1833.

[23] Stephen Hawking. *A brief history of time*. Bantam Dell Publishing Group, 1988.

[24] Johannes Hölzl, Fabian Immler, and Brian Huffman. "Type classes and filters for mathematical analysis in Isabelle/HOL". In: *International Conference on Interactive Theorem Proving*. Springer. 2013, pp. 279–294.

[25] Florian Kammüller, Markus Wenzel, and Lawrence C. Paulson. "Locales A Sectioning Concept for Isabelle". In: *Theorem Proving in Higher Order Logics*. Ed. by Yves Bertot et al. Springer Berlin Heidelberg, 1999, pp. 149–165.

[26] T. W. B Kibble and FH Berkshire. *Classical mechanics*. Imperial College Press, 2004.

[27] Charles Kittel et al. *Mechanics*. McGraw-Hill, 1973.

[28] Alexander Krauss. *Defining recursive functions in Isabelle/HOL*. 2008.

[29] Thomas S Kuhn et al. *Criticism and the growth of knowledge: Volume 4: Proceedings of the International Colloquium in the Philosophy of Science, London, 1965*. Vol. 4. Cambridge University Press, 1970.

[30] Joseph-Louis Lagrange. *Mécanique analytique*. 1st ed. 1788.

[31] Steven J. Leon, Åke Björck, and Walter Gander. "Gram-Schmidt orthogonalization: 100 years and more". In: *Numerical Linear Algebra with Applications* 20.3 (2012), pp. 492–532. DOI: 10.1002/nla.1839.

[32] Leo Liberti and Carlile Lavor. *Euclidean distance geometry: An introduction*. Springer International, 2018.

[33] Ernst Mach. *The science of mechanics*. Prabhat Prakashan, 1907.

[34] Eli Maor. *The Pythagorean Theorem: A 4,000-Year History*. 1st ed. Princeton University Press, 2007, p. 133.

[35] David McGill and Wilton King. *Engineering Mechanics: An Introduction to Dynamics*. 3rd ed. PWS Publishing Company, 1995.

[36] J. C. C. McKINSEY, A. C. SUGAR, and PATRICK SUPPES. "Axiomatic Foundations of Classical Particle Mechanics". In: *Journal of Rational Mechanics and Analysis* 2 (1953), pp. 253–272. ISSN: 19435282, 19435290. URL: http://www.jstor.org/stable/24900331.

[37] Isaac Newton. *Mathematical principles of Natural Philosophy*. Trans. by Andrew Motte. 1728.

[38] Isaac Newton. *Philosophiæ Naturalis Principia Mathematica*. 1st ed. 1687.

[39] Isaac Newton. *The method of fluxions and infinite series; with its application to the geometry of curve-lines*. Henry Woodfall, completed 1671, published 1736.

[40] Tobias Nipkow, Markus Wenzel, and Lawrence C Paulson. *Isabelle/HOL: a proof assistant for higher-order logic*. Springer, 2002.

[41] Silas Boye Nissen. *Point Particles to Capture Polarized Embryonic Cells & Cold Pools in the Atmosphere*. 2020.

[42] Jill North. “Formulations of Classical Mechanics”. In: *The Routledge Companion to Philosophy of Physics* (2021).

[43] CJ Papachristou. “Foundations of Newtonian dynamics: an axiomatic approach for the thinking student”. In: *arXiv preprint arXiv:1205.2326* (2012).

[44] Angel R. Plastino and Juan C. Muzzio. “On the use and abuse of Newton’s second law for variable mass problems”. In: *Celestial Mechanics and Dynamical Astronomy* 53.3 (1992), pp. 227–232. DOI: 10.1007/bf00052611.

[45] Bruce Pourciau. “Newton’s Argument for Proposition 1 of the Principia”. In: *Archive for History of Exact Sciences* 57.4 (2003), pp. 267–311. DOI: 10.1007/s00407-002-0062-x.

[46] Walter Rudin. *Principles of mathematical analysis*. 1st ed. McGraw-Hill Inc., 1953.

[47] Robert Rynasiewicz. “Newton’s Views on Space, Time, and Motion”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2014. Metaphysics Research Lab, Stanford University, 2014.

[48] Steven F. Savitt. “Is Classical Mechanics Time Reversal Invariant?” In: *The British Journal for the Philosophy of Science* 45.3 (1994), pp. 907–913. ISSN: 00070882, 14643537. URL: http://www.jstor.org/stable/687800 (visited on 04/05/2022).

[49] Joachim Schummer. “Aristotelian physics”. In: *KL Lerner & BW Lerner (Hg.), Scientific Thought in Context* 2 (2008), pp. 759–768.

[50] David Sloan and Pedro G. Ferreira. “Cosmology of an infinite dimensional universe”. In: *Physical Review D* 96.4 (2017). DOI: 10.1103/physrevd.96.043527.

[51] Garret Sobczyk and Omar León Sánchez. “Fundamental Theorem of Calculus”. In: *Advances in Applied Clifford Algebras* 21.1 (2010), pp. 221–231. DOI: 10.1007/s00006-010-0242-8.

[52] Lloyd William Taylor. *Physics, the pioneer science*. 1st ed. Houghton Mifflin Company, 1941.

[53] Robert Weinstock. “Newton’s Principia and inverse-square orbits: The flaw reexamined”. In: *Historia Mathematica* 19.1 (1992), pp. 60–70. ISSN: 0315-0860. DOI: https://doi.org/10.1016/0315-0860(92)90055-G.