Exploring Euler's Differentials of Trigonometric Functions in Isabelle using Nonstandard Analysis

Alice Johansen

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2021

Abstract

Euler's *Foundations of differential calculus* [7] is an influential text in which Euler builds the concepts of differentials and derivatives through reasoning with infinitely small and infinitely large numbers. However, mathematical writing standards have changed since Euler's time, and modern mathematicians have criticised Euler's lack of rigour, specifically his handling of infinitesimal numbers and his occasional missing step or assumption. We investigate Euler's reasoning from *Foundations of differential calculus* [7] by reformalising Euler's proofs for the differentials of trigonometric functions in the rigorous framework of Isabelle [13] with nonstandard analysis [14]. In this exploration, we aim to formalise Euler's proofs as close as possible to their original reasoning, while also upholding modern standards of rigour through the use of nonstandard analysis.

Acknowledgements

I thank my supervisor, Dr. Jacques Fleuriot, for his endless guidance, insight, and support on this project. I have enjoyed being his student and I would choose this project again without hesitation.

I also thank Imogen Morris, who has helped me throughout this process in countless ways, from helping me understand the intricacies of Isabelle at the beginning of this year, to proofreading my report at the end of this year.

Table of Contents

Chapter 1

Introduction

This project explores findings from Euler's *Foundations of differential calculus* [7] in the proof assistant Isabelle [13] using nonstandard analysis [14]. We formalise a set of proofs of trigonometric differentials from Chapter 6, "On the Differentiation of Transcendental Functions" [7, Chapter 6]. In this text, Euler derives proofs for these differentials through reasoning with infinitely small numbers. Therefore, we use nonstandard analysis in Isabelle [9] as a framework for formalising these proofs.

Nonstandard analysis [14] is a rigorous type of analysis that includes all theorems from real analysis and also involves reasoning with the set of hyperreal numbers. The set of hyperreal numbers is rigorously constructed from the set of real numbers, and additionally includes infinitesimals (infinitely small) and infinite numbers. Nonstandard analysis is well supported by Isabelle [9], and the use of the two together allows us to transform Euler's proofs into a rigorous setting. In this project, we focus on proofs of the differentials of trigonometric equations, whose identities and properties are also well formulated in Isabelle [13].

Understanding Euler's reasoning is a main motivation for this project. Modern mathematicians have often criticised Euler's lack of rigour, specifically his treatment of infinitesimal quantities. Thus, we analyse how we must deviate from his reasoning in order to formalise his proofs rigorously. We discuss some examples of criticism in section 2.1.2. In the conclusion of this report (5), we evaluate the feasibility of formalising Euler's proofs and to what extent the criticisms against Euler's treatment of infinitesimal quantities are justified.

Additionally, we have chosen to mechanize a set of Euler's proofs which have not yet been formalised. In combination with the mechanised proofs from Frankovksa's undergraduate project [10], this work encompasses the majority of the trigonometric differential proofs that Euler presents in Chapter 6 [7], with the exceptions of the differentials of secant and cotangent and some other higher differentials.

Our goal in this project is to replicate Euler's proofs as close as possible while upholding modern standards of rigour, and to explore the following questions:

- How can we interpret Euler's meaning in his proof derivations?

- To what extent do we have to adapt Euler's proofs in Isabelle with nonstandard analysis to adhere to modern standards of rigour?

We keep these questions in mind throughout our project, and with each proof, we must make choices about how we interpret and represent Euler's reasoning in Isabelle, especially when a proof is not entirely replicable.

In the next chapter, we provide some contextual background to Euler's *Foundations of differential calculus* [7]. We also give a brief introduction to nonstandard analysis [14] in section 2.2 and further discuss our motivations for using this theory of analysis. In section 2.3, we discuss technicalities of working with Isabelle and we provide an example proof to familiarise the reader with Isabelle syntax.

In Chapter 3, we define Euler's notion of a differential in Isabelle, and use this definition to mechanise a set of first differential proofs from Euler's *Foundations of differential calculus* [7]. These proofs follow from where Frankovska's previous undergraduate project [10] stopped in Euler's text, and we use her theorems for the differential of arcsine to move forward in two of our proofs (see section 3.5 and 3.6). Our set of mechanised proofs includes:

- sine,
- cosine,
- tangent,
- arccosine,
- arctangent.

We walk through interesting parts of the mechanisation process, discuss how we can interpret Euler's reasoning, and highlight where our proofs differ. In Chapter 4, we provide two Isabelle interpretations of Euler's higher differentials of sine, which Euler presented but did not prove. To mechanise these results, we formalise a definition for higher differentials in Isabelle (section 4.1) and build a general form for the higher differentials of sine, which we prove by induction. In the conclusion, we summarise our work and evaluate how our project answers the questions given above.

Chapter 2

Background

2.1 Euler's *Foundations of differential calculus*

Published in 1755, Euler's *Foundations of differential calculus*, or *Institutiones calculi differentialis* [7] offers a basis for differential calculus through reasoning with infinitesimals.

In this text [7], Euler discusses infinitely large and small numbers, differences and differentials, and other foundational concepts which build his definition of the derivative. Euler also presents sets of algebraic and transcendental functions and proves their differentials and derivatives using ratios of vanishing increments or decrements. In this project, we focus specifically on the differentials of trigonometric equations.

Formal mathematical writing conventions have changed since Euler's time, and it is sometimes difficult to interpret Euler's meaning, both because of his prose and the fact that he often omits algebraic steps and assumptions. Occasionally, he omits an entire proof. For example, he presents the higher differentials of sine, but he is not explicit about how he proves these differentials. In Chapter 4 of this paper (see 4.2), we provide two possible interpretations and mechanisations for both, and we provide a discussion about his presentation of these differentials in section 4.2.3.3.

2.1.1 Concepts and definitions

In Chapter 4 of *Foundations of differential calculus*, "On the Nature of Differentials of Each Order" [7], Euler presents his definitions for differentials and derivatives using the notions of infinitely-small differences.

Euler defines a differential dy for a function $y = f(x)$ in paragraph 118 [7, p. 65]. Euler defines dx as the infinitely-small difference or increment by which x increases, and dy denotes the increment that y increases by when x becomes $x + dx$. He writes, "if we substitute $x + dx$ for x in the function y and we let y^I be the result, then $dy = y^I - y$, and this is understood to be the first differential" [7, p. 65]. Then,

$$dy = f(x + dx) - f(x). \tag{2.1}$$

He uses this definition of the differential to introduce the derivative, which he defines as the ratio $\frac{dy}{dx}$, or the value of the differential divided by dx [7, p. 66]:

$$\frac{dy}{dx} = \frac{f(x+dx) - f(x)}{dx}. \tag{2.2}$$

In comparison, the modern definition of the derivative uses limits. Because Euler's calculations and his definitions of the differential and derivative involve the infinitely-small number dx, we use nonstandard analysis [14] as a framework to formalise Euler's proofs, since it provides a rigorous theory for working with these numbers. We discuss nonstandard analysis in more detail in section 2.2.

2.1.2 Criticisms

Part of what makes Euler's reasoning so interesting is how strongly Euler's methods have been criticised by other mathematicians. Common criticisms include lack of rigour due to missed assumptions or unjustified proof steps, and lack of discussion of the consequences of reasoning with infinitely small numbers.

"He wants to use infinite numbers... as well as infinitesimals... to take their ratios, add, subtract and multiply them as if they matter, and then throw them away when it suits his purposes. It is exactly the behavior that Berkeley was trying to discourage and that Cauchy and Weierstrass eventually repaired" [16].

This quote from Sandifer briefly explains some of the negative sentiment towards Euler's *Foundations of differential calculus* [7]. While there are higher standards of rigour in mathematics today, some literature suggests that even before Euler's text was published, there existed certain standards that *Foundations of differential calculus* [7] would have broke. Berkeley targeted Newton and Leibniz for their methods of calculus and their use of infinitesimals [1]. Although Euler's text was not published at the time, Berkeley specifically criticised the concept of infinitesimals (or fluxions as Newton named them), calling them "ghosts of departed quantities" in his satirical book *The Analyst: A Discourse Addressed to an Infidel Mathematician* [2]. Berkeley also criticised infinitesimals by claiming mathematicians treated them in contradictory ways, "at one stage as finite and at another as zero as convenience dictated" [18]. In our project, we pay close attention to how Euler manipulates infinitesimal numbers, and how we must adapt his proofs in nonstandard analysis to maintain rigour.

Modern mathematicians have specifically criticised the way that Euler manipulated infinite numbers, infinitesimals and series. As Dunham argues in *Journey Through Genius*,

"Today, we recognize that Euler was not so precise in his use of the infinite as he should have been. His belief that finitely generated patterns and formulas automatically extend to the infinite case was more a matter of faith than science" [4].

However, some argue that Euler did indeed understand these concerns, and that it is not at all a matter of carelessness [5]. While Euler is often criticised for his treatment of infinitesimal quantities and his definition of derivative $\frac{dy}{dx}$, Edwards argues that "he makes no claim that this ratio can always be evaluated," and "we should not readily believe that he faked his way through calculus with a poor grasp of its basic concepts and a casual attitude towards mathematical reasoning" [5].

Investigating these proofs provides insight on to what extent Euler's understanding, his proofs, and his manipulation of the infinitesimal are consistent after making the correct assumptions.

2.2 Nonstandard Analysis

In this section, we discuss our motivations for using nonstandard analysis to formalise Euler's proofs, and we briefly cover some technicalities of nonstandard analysis.

2.2.1 Motivations for using nonstandard analysis

Nonstandard analysis [14] was developed by Abraham Robinson in the 1960s in order to provide a formal and rigorous framework for working with infinitesimals and infinitely-large numbers. In nonstandard analysis, infinitesimals and infinitely-large numbers are contained in the set of hyperreal numbers, which also includes real numbers.

We use nonstandard analysis [14] in this project because it is necessary to represent Euler's differentials as infinitesimals and reason with these infinitesimals, or 'vanishing quantities' as Euler calls them [7]. We choose the theory of nonstandard analysis over other theories for a few main reasons. Euler's reasoning has been investigated in nonstandard analysis before in previous works, which we discuss in the related work section (2.4). Additionally, nonstandard analysis is well supported by Isabelle, and the majority of crucial theorems that we need in this project have already been proven and are contained in the NSA library of Isabelle [9]. Lastly, nonstandard analysis relies on use of the transfer principle, a theorem that allows us to use all theorems of real analysis within nonstandard analysis. We discuss the transfer principle within this section, in 2.2.2.4.

Most importantly, nonstandard analysis provides a rigorous framework for which we can attempt to reconstruct Euler's proofs, while following his reasoning with infinitesimals as close as possible. This allows us to explore how his proofs must be adapted in order to adhere to modern standard of rigour.

2.2.2 Technical definitions

In this section, we present the nonstandard analysis definitions required to understand this report.

2.2.2.1 The set of hyperreal numbers

In nonstandard analysis, the hyperreal number set, denoted ${}^*\mathbb{R}$, is constructed from the real number set using the ultrapower construction [11, Chapter 3]. In the ultrapower construction, the set of hyperreal numbers is represented as a set of equivalence classes on $\mathbb{R}^{\mathbb{N}}$, where $\mathbb{R}^{\mathbb{N}}$ is the set of all sequences of real numbers.

If we consider a constant sequence $\mathbf{r} = \langle r, r, ... \rangle$ of the real number r, we map this sequence to its equivalence class $[\mathbf{r}] = [\langle r, r, ... \rangle]$ to obtain the hyperreal counterpart of r, denoted $^*r$. Using this construction, we can map any real number to its hyperreal counterpart in $^*\mathbb{R}$. Following from this, $\mathbb{R}$ is encompassed within $^*\mathbb{R}$, and we also identify real number 0 and 1 with their hyperreal counterparts $[\mathbf{0}]$ and $[\mathbf{1}]$. More details can be found in Goldblatt's *Lectures on the Hyperreals* [11].

Euler claims that "the infinitely small vanishes in comparison with the finite and hence can be neglected" [7, Chapter 3]. We use the nonstandard-analysis definition of infinitesimal to allow reasoning with Euler's infinitely-small numbers.

A hyperreal, nonzero number ε is *infinitesimal* if it is less than all positive numbers in $\mathbb{R}$:

$$|\varepsilon| < r \text{ for all r} \in \mathbb{R}_{>0} \tag{2.3}$$

A hyperreal number ω is an infinitely large number or *infinite* number if it is greater than all positive numbers in $\mathbb{R}$:

$$|\omega| > r \text{ for all r} \in \mathbb{R}_{>0} \tag{2.4}$$

Infinitely-large numbers are reciprocals of infinitesimals and vice-versa: $\frac{1}{\varepsilon}$, the reciprocal of ε, would be infinitely large and similarly $\frac{1}{\omega}$, the reciprocal of ω would be infinitely small. Intuitively, we can think of an infinitesimal as the equivalence class of a sequence in $\mathbb{R}$ that converges to zero, and an infinitely-large number as the equivalence class of a sequence in $\mathbb{R}$ that diverges to ∞ [11]. We denote the set of infinitesimal numbers as $^*\mathbb{R}_{\varepsilon}$ and the set of infinitely-large numbers as $^*\mathbb{R}_{\infty}$.

2.2.2.2 The infinitely-close relation

Euler uses '=' both to represent equality and to show a relation between two quantities which differ only by an infinitesimal. We distinguish between these two cases in nonstandard analysis and say two quantities are infinitely close if they differ only by an infinitesimal.

Formally, let $b, c \in {}^*\mathbb{R}$. Then, we say that b is infinitely close to c, denoted $b \approx c$, if $b - c$ is infinitesimal. Note that this means that the real parts of b and c are equal, but there is an infinitesimal difference between them.

2.2.2.3 The standard part function

Nonstandard analysis relies on using the *standard part function* to bridge a relationship between hyperreal and real numbers. The standard part function associates every finite hyperreal number x with a unique real number x_0 that x is infinitely-close to. In formal notation, we write the standard part principle as follows.

$$\forall x \in {}^*\mathbb{R} - {}^*\mathbb{R}_\infty.\ \exists x_0 \in \mathbb{R}.\ x \approx x_0 \tag{2.5}$$

2.2.2.4 The transfer principle and nonstandard extension

The transfer principle is a principle that allows us to deduce that certain statements which are known to be true over the reals have counterparts which are true over the hyperreal numbers.

To explain the transfer principle, we must first introduce the nonstandard analysis concept of a $*$-transform. Every function in $\mathbb{R}$ has a corresponding $*$-transform, which is an extension from $\mathbb{R}$ to ${}^*\mathbb{R}$ for that function, and we obtain the $*$-transform of a function f by replacing f with ${}^*f$. All relations also have a corresponding $*$-transform, for example, some relation P becomes ${}^*P$. We often drop the star signs for the most common relations (like $<$, $>$, $=$).

Properties of functions and sets in $\mathbb{R}$ are translated to properties of the extensions of these functions and sets in ${}^*\mathbb{R}$ using the transfer principle. For example, the set of natural numbers $\mathbb{N}$ is extended using the transfer principle to its corresponding nonstandard extension, the set of hypernatural numbers, denoted ${}^*\mathbb{N}$ [11]. Since $\mathbb{N}$ is closed under addition and multiplication, by the transfer principle, ${}^*\mathbb{N}$ is also closed under addition and multiplication. In this project, we often use the transfer principle to show that trigonometric identities that hold for $\mathbb{R}$ also hold for ${}^*\mathbb{R}$.

The transfer principle can translate theorems both ways. All statements about the reals can be $*$-transformed and translated to statements about the hyperreals. However, not all statements about the hyperreals can be transformed back into the reals; they must be well-formed. For example, we cannot translate an expression that uses the infinitely-close relation back into standard analysis, as there is no real equivalent. Goldblatt's *Lectures on the Hyperreals* [11] provides a complete discussion and formal presentation of the transfer principle.

Nonstandard extension from functions in $\mathbb{R}$ to ${}^*\mathbb{R}$ is written in Isabelle using `*f* f` for a function f. Also, we map a real variable x to its hyperreal counterpart in Isabelle using `star_of x`. We discuss nonstandard analysis in Isabelle in more depth in section 2.3.2.

2.3 Isabelle

2.3.1 Introduction to Isabelle/HOL and motivations

In this project, we use the generic proof assistant Isabelle/HOL (higher-order logic) [13]. Isabelle supports automated and interactive theorem proving [13], and allows the user to write mathematically-structured higher-order logic proofs in a style called Isar [17], where each proof step must be justified by explicitly named theorems or definitions.

Proof assistants, or interactive theorem provers, are concerned with helping users write formal proofs. They accomplish this by providing an interface for the user to search for and apply theorems in the construction of a proof. As the name suggests, automated theorem proving is a kind of computer-assisted theorem proving where the prover attempts to construct a proof with little user input.

In this project, we choose how to structure our proofs with Isar, so we primarily use interactive theorem proving. However, Isabelle also provides automated theorem proving with the tool sledgehammer [3]. When invoked on a goal statement, sledgehammer connects to external automatic theorem provers and attempts to find a proof using existing lemmas [3].

Although there exist many other kinds of theorem provers, we have decided to use Isabelle/HOL in this project for a few reasons. Since our goal is to formalise and explore Euler's reasoning, we must be able to choose how we structure and construct our proofs. Isar allows us to make these choices. Isabelle also contains a library for trigonometric definitions and equations, and using this library (while extending these lemmas to nonstandard analysis) will save the time it would otherwise take to formulate them. Lastly, Isabelle supports nonstandard analysis [14], and many essential theorems and definitions that we need have already been proven, and they are contained in the HOL/Nonstandard-Analysis library in Isabelle [9].

2.3.2 Using nonstandard analysis in Isabelle

In this section, we briefly present some of the most common syntax we use in Isabelle with nonstandard analysis.

Isabelle is simply typed, so we must be explicit with variable typing. The Isabelle types that we use in this project are `hypreal` for the set of hyperreal numbers, `real` for set of real numbers, and `nat` for the natural numbers. We assign the type of hyperreal numbers to variable a with `a::hypreal`, and we often fix the types of variables in lemma statements like so:

```
fixes dx::hypreal and x::real
```

Other sets that we use in this project are the set of infinitesimal numbers, `Infinitesimal` and `HFinite`, for the set of finite (not infinitely-large) quantites. We denote that a variable x is infinitesimal in Isabelle using `x ∈ Infinitesimal`.

Another piece of syntax that we use is `*f*`, which converts a function to its corresponding $*$-transform:

```
*f* :: ('a ⟹'b) ⟹'a star ⟹'b star
```

For example, we would use `*f*` and `star_of` to convert the function $\sin x$ to its hyperreal function counterpart with `( *f* sin)(star_of x)`, where if x is of type real, `star_of x` is of type hyperreal.

A powerful Isabelle rule for nonstandard analysis is the `transfer` rule, which implements the transfer principle that we presented in 2.2. If we have an Isabelle lemma that holds for the reals, we prove the corresponding lemma in nonstandard analysis using `apply(transfer)`. An example of this is for the hyperreal counterpart of the cosine addition formula, which we write in order to mechanise Euler's proof of the differential of *cosx* in section 3.3:

```
lemma STAR_cos_add:
  "⋀(a::hypreal) b. ( *f* cos)(a + b) =
     ( *f* cos) a *( *f* cos) b - ( *f* sin) a * ( *f* sin) b"
  apply (transfer) by (simp add: cos_add)
```

The `transfer` translates the nonstandard goal statement into its standard analysis counterpart, which it requires to be proven. Here, `simp` is a simplification proof method that adds the lemma `cos_add`, the cosine addition formula in the reals, to the set of theorems used to complete the proof.

2.3.3 Structured Isar proofs

Isar [17] allows us to construct readable proofs that look structurally similar to pen and paper proofs, where we can choose each step (and the methods to each step) of our reasoning.

To illustrate both how we use Isar and how we use the syntax presented in the previous section 2.3.2, we present an example proof, which shows that the differential of $f(x)$ is infinitely-close to zero, given that dx is infinitesimal and the function is continuous.

```
lemma differential_infinitesimal:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal" "isNSCont f x"
  shows "differential f x dx ∈Infinitesimal"
proof -
  have "differential f x dx =
      (*f* f)(star_of x + dx) - (*f* f)(star_of x)"
    using differential_def by auto
  also
  have "(*f* f)(star_of x + dx) - (*f* f)(star_of x) ≈
      (*f* f)(star_of x) - (*f* f)(star_of x)"
    by (metis approx_diff approx_sym assms(1) assms(2) ... )
  finally
  show "differential f x dx ∈ Infinitesimal"
    by (simp add: mem_infmal_iff)
qed
```

The statement consists of the title of the lemma, the `fixes` keyword where we fix the types of our variables, the `assumes` keyword that precedes the assumptions we have made, and the `shows` keyword that defines what we aim to prove.

Here, we have assumed that dx is infinitesimal, and that the function $f(x)$ is continuous in nonstandard analysis, or `isNSCont f x` in Isabelle. The equivalent statement for expressing that a function $f(x)$ is continuous in the reals is `isCont f x`.

The `proof` keyword signifies the start of a new Isar proof block. The `have` keywords precede the expression that we have at that step in the proof, and each `have` statement must be justified in order to move onto the next step. We can justify a `have` statement with the `using` keyword followed by some facts or lemmas, and then a `by` keyword followed by some proof method. An example of this is after the first `have` statement. We prove that we `have` this expression `using` the definition `differential` (we attach `_def` to a definition to use it), and `by auto` signifies that the method `auto` proves this statement.

After this line, the keyword `also` signifies that we have started a transitive chain. This proof pattern looks similar to many pen and paper proofs, where we **have** $a = b$ **[proof]**, **also have** $b = c$ **[proof]** and **finally have** $a = d$ **[proof]** [17]. There are other proof patterns and keywords aside from `also`, like `then` and `moreover` which are explained in the Isar manual [17].

Also, it is not always necessary to use the keyword `using`. In the second `have` statement, we invoked automated proof tool sledgehammer [3] which gave us keyword `by` and long string of theorems in the parenthesis, which are the theorems that prove the statement. In this sledgehammer proof, `metis` is the proof method. Lastly, the `show` statement must match the goal statement that we are trying to prove, and `qed` signifies the end of the completed proof.

This lemma shows that the differential of a function $f(x)$ is infinitesimal, given that the function is continuous and dx is infinitesimal. The proof follows from the fact that since dx is infinitesimal, $f(x+dx)$ is infinitely close to $f(x)$ by (nonstandard) continuity, therefore the definition of the differential $f(x+dx) - f(x)$ is infinitely close to $f(x) - f(x)$, and thus the differential is infinitesimal. Later, we use this result to show that two differentials which are continuous on the same interval of x are infinitely close in our mechanised proofs of the differential of $\arccos x$ and $\arctan x$ in 3.5 and 3.6.

2.4 Related work

In this section, we consider previous work in nonstandard analysis on Euler's mathematical works, specifically his pre-calculus text *Introductio in analysin infinitorum* [6] and our focus text *Foundations of differential calculus* [7].

2.4.1 Pen and paper proofs

In 'Higher Trigonometry, Hyperreal Numbers, and Euler's Analysis of Infinites' [12], McKinzie and Tuckey use nonstandard analysis to rehabilitate findings from Euler's *Introductio in analysin infinitorum* [6] in a rigourous setting, specifically the binomial series and its various extensions to the series of the sine, cosine, and logarithm. Euler's *Introductio in analysin infinitorum* [6] was published prior to our text of focus, *Foundations of differential calculus* [7], and Euler often references his results from this previous text for his proofs in the latter text. In *Introductio in analysin infinitorum* [6], Euler derives the series of sine and cosine, which he uses in *Foundations of differential calculus* [7] for his proofs of the differential of sine, cosine, and tangent.

In McKinzie and Tuckey's analysis of Euler's calculations, they investigate his use of infinitesimal and infinite quantities and explore whether "neglecting infinitely many infinitesimals leads to a negligible difference in an infinite sum" using determinacy of infinite sequences [12]. Overall, they find that Euler's treatment of infinitesimals is consistent when rehabilitated to a rigourous framework. They argue against criticisms of Euler, and explain that by focusing on the "underlying mathematics", they find that Euler's calculations are "far from reckless and nonsensical" [12].

To compare our approaches, McKinzie and Tuckey [12] use pen and paper to replicate Euler's proofs in *Introductio in analysin infinitorum* [6], whereas we use Isabelle to mechanise this process. Like McKinzie and Tuckey, we also use nonstandard analysis to frame Euler's work in a rigorous setting. On pen and paper, McKinzie and Tuckey also rigorously prove a few results from Euler's *Introductio in analysin infinitorum*, specifically, that $\sin(\theta) \approx \theta$ and $\cos(\theta) \approx 1$ when θ is infinitesimal. We use previously proven Isabelle lemmas [8] for these results in our proof mechanisations of the differentials of $\sin x$, $\cos x$, and $\tan x$. McKinzie and Tuckey's work suggests that we will arrive at similar conclusions, and ultimately find that Euler's treatment of infinitesimal differentials is consistent when reformalised rigorously.

2.4.2 Other mechanisations of Euler's work

Previous projects on *Foundations of differential calculus* [7], such as the thesis 'Exploring Euler's Foundations of Differential Calculus in Isabelle/HOL using Nonstandard Analysis' by Rockel [15] and an undergraduate project by Frankovska [10] with this same title have also investigated Euler's reasoning and formalised a few of Euler's proofs using Isabelle [13] and nonstandard analysis [14].

Rockel's exploration [15] focuses on the set of logarithmic functions and their derivatives, and Rockel succeeds for the most part in formalising Euler's arguments rigorously. She investigates paragraphs 180 to 183 [7], Euler's proofs for the differentials of the logarithm function, and she finds that these proofs can indeed be reproduced consistently in Isabelle with nonstandard analysis. In her exploration, Rockel finds many 'skipped steps' and 'hidden assumptions' in Euler's proofs [15], many of which are similar to the missing assumptions that we find in our own project, specifically assumptions of bounds on variable x to match domains of functions and ensure continuity. In her analysis of Euler's higher differentials of the logarithm, she mechanises Euler's proof for the first derivative but is unable to directly follow Euler's reasoning for the higher derivatives of this function, possibly due to time constraints. In the case of the first derivative, Rockel provides an Isabelle proof that is still within scope of methods that Euler could have used at the time [15].

Frankovska's exploration [10] focuses on Euler's proofs of geometric series and the derivative of arcsin. Like Rockel and ourselves, she uses nonstandard analysis as a rigourous framework for formalising Euler's proofs. Using Isabelle and nonstandard analysis, Frankovska provides a proof for the polynomial sum series for different case values of x, with the exception of $x < 1$ and $x \approx 1$, which she showed may not be solvable. Frankovska also attempts to mechanise Euler's proof of the derivative of arcsin, but is unable to follow Euler's approach or obtain his result as one of Euler's steps involves dividing by an infinitesimal. She instead provides two alternative interpretations in Isabelle. Additionally, Frankovska provides a proof for the higher derivatives of arcsin, which Euler presented in his text but did not prove. We find ourselves in a similar situation in this project for the higher differentials of $\sin x$ (see 4.2), and we also provide our own proof derivation. In this project, we directly build from Frankovska's exploration by importing her Isabelle proof scripts. We use her lemma for the differential of arcsin in two of our own proofs, the differentials of arccos and arctan, as Euler proves these differentials using his previous result for the differential of arcsin.

In both Rockel's thesis [15] and Frankovska's undergraduate project [10], the main problem they face in formalising Euler's reasoning occurs when Euler divides by an infinitesimal quantity, which does not necessarily preserve the infinitely-close relation. This implies that the main difficulties we might face in this project relate to preserving the infinitely-close relation, which is indeed the case.

Chapter 3

Mechanisation of first differential proofs

In this chapter, we present our Isabelle mechanisations for our first differential proofs from Euler's *Foundations of differential calculus*. Within these sections, we discuss how we can interpret Euler's prose, and what difficulties we face in doing so. We also analyse where our Isabelle proof mechanisations differ from Euler's pen and paper proofs, and why this might be the case, specifically in the context of rigour and Euler's treatment of infinitesimal quantities.

3.1 Defining differentials in Isabelle

First, we formulate the general notion of a first differential in Isabelle. As we presented in 2.1.1, Euler defines the differential for a function $y = f(x)$ as $dy = f(x+dx) - f(x)$, where dx is an infinitesimal.

In Isabelle, we have formalisation choices on the kinds of types that we give our variable x and function f, but otherwise our definition of the differential in Isabelle looks similar to Euler's definition. Due to the constraints of nonstandard analysis, the types that we use in our lemma statements can affect the steps that we must take in our proofs, since we cannot necessarily manipulate (divide and multiply) hyperreal numbers the same way that we can manipulate real numbers. We define the differential of a function in Isabelle as follows:

```
definition differential ::
    "(real ⇒ real) ⇒ real ⇒ hypreal ⇒ hypreal"
  where
     "differential f x dx =
        (*f* f)(star_of x + dx) - (*f* f)(star_of x)"
```

This definition takes three inputs, a function `f` of type `real ⇒ real`, a variable `x` of type `real`, and an infinitesimal `dx` of type `hypreal`. On the right-hand side, we transfer `f` into `*f*`, and map `x` to its hyperreal counterpart using `star_of`.

In this chapter, we use this definition to mechanise the first differentials of $\sin x$ and $\cos x$, which follow similar proof structures. We follow with a mechanisation of Euler's first proof of the differential of $\tan x$. Then, we write Isabelle mechanisations for Euler's proofs of the differential of $\arccos x$ and $\arctan x$, which both use Frankovska's proven lemma for the first differential of $\arcsin x$. Lastly, we provide our mechanisation for Euler's second proof of the differential of $\tan x$, which Euler proves using the differential of $\arctan x$.

3.2 Mechanising the first differential of $\sin x$

Euler presents a complete proof for the differential of $\sin x$ in paragraph 201 of Chapter 6, "On the Differentiation of Transcendental Functions" [7]. He writes, "let $y = \sin x$ and replace x by $x+dx$ so that y becomes $y+dx$." Then, $y+dy = \sin(x+dx)$ and

$$dy = \sin(x+dx) - \sin x. \tag{3.1}$$

Next, Euler applies the sine angle addition formula to show

$$\sin(x+dx) = \sin x \, \cos dx + \cos x \, \sin dx. \tag{3.2}$$

To simplify this expression further, Euler references his previous work, *Introductio in analysin infinitorum* [6], where he proves that $\sin dx = dx$ and $\cos dx = 1$ when we expand the series and "exclude the vanishing terms" [7]. Euler uses the symbol for equality, $=$, instead of the infinitely-close relation, $\approx$. In our Isabelle proof, we must make a distinction between the two relations, and from this point onward in our proof, we use the infinitely-close relation to represent Euler's '='. Using these results, Euler simplifies to

$$\sin(x+dx) = \sin x + dx \, \cos x. \tag{3.3}$$

Finally, Euler substitutes (3.3) into equation (3.1) to obtain the final equation for the differential of sine:

$$dy = dx \, \cos x. \tag{3.4}$$

Translating this proof into Isabelle is relatively straightforward, but there are a few additional steps we must do in our Isabelle proof to follow Euler's reasoning.

To follow obtain (3.2), we import a theory file from Frankovska's undergraduate project [10] that provides a lemma for the nonstandard sine addition formula, which Frankovksa used for her proof mechanisation of the derivatives of $\arcsin x$ [10].

Additionally, in nonstandard analysis we cannot use the equality symbol like Euler does for $\sin dx = dx$ and $\cos dx = 1$. Euler uses the equality symbol both for equality and when he wants to say that two expressions differ by an infinitesimal. The equalities only hold in these two expressions when $dx = 0$, and in our case the differential dx is a non-zero infinitesimal.

We use lemmas that prove $\sin dx \approx dx$ and $\cos dx \approx 1$, previously proven by Fleuriot [8], titled `STAR_sin_Infinitesimal` and `STAR_cos_Infinitesimal` respectively.

Once we apply these lemmas to simplify equation (3.2), we use the infinitely-close relation for the remainder of the proof.

We must make these distinctions between the equality relation and the infinitely-close relation in Isabelle, both because our goal is to follow Euler's reasoning in a rigorous setting and because Isabelle asserts correctness.

Once we have lemmas for the (nonstandard) sine addition formula and the results that $\sin dx \approx dx$ and $\cos dx \approx 1$, we can follow Euler's reasoning to build an Isabelle proof for this differential. The lemma statement for our mechanisation of the differential of sine is as follows:

```
lemma differential_sin:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows "differential sin x dx ≈ (*f* cos)(star_of x)*dx"
```

The rest of the proof can be found in appendix A.1, but we draw attention to some details in this statement. We always explicitly assume that dx is infinitesimal in any lemma statement for a differential proof. This is implicit in Euler's reasoning throughout all of his proofs, wherever he uses dx. When he defines the first differential of a function in Chapter 6 "On the Nature of Differentials of Each Order", Euler states that "from now on, dx will be the infinitely small difference by which x is understood to increase" [7, Chapter 6].

Overall, it is relatively simple to interpret Euler's reasoning in his proof here. We do not have to diverge from his reasoning at all to formalise this proof, though our Isabelle proof does look different since we explicitly use the infinitely-close relation. The use of the infinitely-close relation is implicit in Euler's reasoning. He writes, "when we exclude the vanishing terms" of the series of $\sin dx$ and $\cos dx$, "we have $\cos dx = 1$ and $\sin dx = dx$" [7, p. 116]. This implies that he is aware that both $\cos dx$ and 1, and $\sin dx$ and dx differ by an infinitesimal, or in his words, "vanishing terms", and he simply uses the same symbol for equality and the infinitely-close relation.

This lemma, `differential_sin`, completes our mechanisation of Euler's proof of the first differential of sine.

3.3 Mechanising the first differential of $\cos x$

In paragraph 202 of Chapter 6, "On the Differentiation of Transcendental Functions" [7], Euler proves the first differential of $y = \cos x$. He follows the same steps from his proof of the first differential of $\sin x$, where he substitutes x with $x + dx$ to obtain

$$y + dy = \cos(x + dx). \tag{3.5}$$

Next, Euler presents the cosine addition formula,

$$\cos(x + dx) = \cos x \cos dx - \sin x \sin dx. \tag{3.6}$$

Euler simplifies (3.5) using (3.6) and the results that $\cos dx \approx 1$ and $\sin dx \approx dx$ to obtain

$$y + dy = \cos x - dx\ \sin dx. \tag{3.7}$$

Following from this,

$$dy = dx\ \sin x. \tag{3.8}$$

We follow the overall structure that Euler presents here in Isabelle, with the exception that we subtract y, or $\cos x$, from $\cos(x + dx)$ at the beginning of our proof mechanisation, since our `differential` definition does this for us, whereas Euler subtracts y in the last step after equation (3.7). The process of mechanising this proof is similar to the process we used previously for the first differential of $\sin x$ in 3.2.

We construct a lemma for the nonstandard version of the cosine addition formula, which has not been previously formulated, using the `transfer` rule along with the pre-existing cosine addition formula for the reals. This is also the example lemma that we gave in 2.3.2.

The rest of the proof in Isabelle follows the same reasoning as Euler's proof, with the exception that when we simplify the cosine addition expression using Fleuriot's lemmas [8] for $\sin dx \approx dx$ and $\cos dx \approx 1$, we switch to using the infinitely-close relation.

Like with our mechanisation of Euler's proof of the differential of $\sin x$ (3.2), we do not have to overly adapt Euler's reasoning to maintain rigour. The only difference between our proof and Euler's is that we explicitly use the infinitely-close relation. This concludes our mechanisation of the first differential of cosine, which can be found in full in appendix B.

3.4 A first mechanisation of the first differential of $\tan x$

In this section, we provide an Isabelle mechanisation of Euler's first proof of the differential of $\tan x$ from paragraph 203 of Chapter 6, "On the Differentiation of Transcendental Functions" [7]. Euler begins with $y = \tan x$ and thus $dy = \tan(x + dx) - \tan x$. Then, he presents the $\tan x$ addition formula,

$$\tan(x + dx) = \frac{\tan dx + \tan dx}{1 - \tan x\ \tan dx}. \tag{3.9}$$

Since he is using the definition of the differential, $dy = \tan(x + dx) - \tan x$, Euler subtracts $\tan x$ from this expression and skips the intermediate algebraic steps to obtain,

$$dy = \frac{\tan dx(1 + \tan x\ \tan x)}{1 - \tan x\ \tan dx}. \tag{3.10}$$

To simplify this expression, Euler writes that "when dx vanishes, the tangent is equal to the arc itself, so that $\tan dx = dx$, and the denominator $1 - dx \tan x$ reduces to unity" [7]. Then, $dy = dx(1 + \tan^2(x))$, and since $\sec^2(x) = \frac{1}{\cos^2(x)}$,

$$dy = \frac{dx}{cos^2(x)}. \tag{3.11}$$

We begin our mechanisation in Isabelle with the following lemma:

```
lemma differential_tan:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal" "cos x ≠ 0"
  shows "differential tan x dx ≈ dx / ((*f* cos)(star_of x))²"
```

It is important to note here that we have added the assumption that $\cos x \neq 0$, which Euler did not mention in his proof. We must add this assumption, since $\tan x$ and thus its differential are undefined when $\cos x = 0$.

3.4.1 Preliminary lemmas

There are a few lemmas that we must prove in order to follow Euler's proof of the differential of $\tan x$. Firstly, we extend the $\tan x$ function to nonstandard analysis, which was not already defined in the HOL/Nonstandard-Analysis library [9]. We also define the (nonstandard) tangent angle addition formula in Isabelle.

```
lemma STAR_tan_def:
  "(*f* tan) = (λx. (*f* sin) x / (*f* cos) x)"

lemma STAR_tan_add:
  "⋀(x::hypreal) y. ⟦(*f* cos) x ≠ 0, (*f* cos) y ≠ 0,
        (*f* cos)(x + y) ≠ 0 ⟧ ⟹
     (*f* tan)(x + y) = ((*f* tan) x + (*f* tan) y) /
        (1 - (*f* tan) x * (*f* tan) y)"
```

In order to use our lemma `STAR_tan_add`, we must match the three assumptions that $\cos x \neq 0$, $\cos y \neq 0$, and $\cos(x+y) \neq 0$. In our subsequent proof, the y here becomes our dx.

We have already assumed that $\cos x \neq 0$, but we must also prove that $\cos dx \neq 0$ and $\cos(x+dx) \neq 0$. We know that $\cos dx \neq 0$ holds as dx is a non-zero infinitesimal, and we prove this in Isabelle using our assumption that `dx ∈ Infinitesimal` and the result that $\cos dx \approx 1$. We prove the other assumption, $\cos(x+dx) \neq 0$, using the cosine addition formula and the $\cos dx \approx 1$ and $\sin dx \approx dx$ rules.

Euler also writes that $\tan dx = dx$, though he means that they are infinitely close. We know this equality cannot hold. The equality $\tan dx = dx$ is only true when the numerator $\sin dx = dx$ and the denominator $\cos dx = 1$, and the latter two equalities only hold when $dx = 0$. Since dx is a non-zero infinitesimal, we have that $\tan dx$ is infinitely-close to dx, as $\sin dx \approx dx$ and $\cos dx \approx 1$. This fact, that $\tan dx \approx dx$, is also available in Isabelle from the lemma `STAR_tan_Infinitesimal` [8].

Additionally, we prove that the denominator $1 - \tan x \, \tan dx$ is non-zero, in order to simplify fractions when we subtract $\tan x$ from $\tan(x+dx)$. This follows from our assumption that $\cos x \neq 0$ and $\tan dx \approx dx$.

Lastly, we must prove the trigonometric identity,

$$1 + \tan^2(x) = \frac{1}{\cos^2(x)}.$$

We first prove that this holds in the reals, in a lemma titled `tan_cos_squared`. Then, we apply the `transfer` rule to obtain the corresponding nonstandard lemma. Both lemmas are included in appendix C.

Although the number of preliminary lemmas we had to write makes our mechanised proof look complicated in comparison to Euler's short proof of this differential, we argue this does not necessary imply negligence on Euler's behalf. The necessary assumption in this proof that Euler did not explicitly state is the assumption that $\cos x \neq 0$, which is implicit for continuity of $\tan x$. The other assumption, $\cos dx \neq 0$, follows from his reasoning that $\cos dx \approx 1$, and the last statement, $\cos(x + dx) \neq 0$, follows from both of these prior assumptions.

3.4.2 Continuing our first proof of the differential of $\tan x$

We return to our differential of $\tan x$ proof mechanisation, where we apply the (nonstandard) tangent angle addition formula and simplify to obtain equation (3.10) in Isabelle.

```
have "differential tan x dx =
 (*f* tan) dx * (1 + (*f* tan)(star_of x)*(*f* tan)(star_of x)) /
     (1 - (*f* tan)(star_of x)*(*f* tan) dx)"
```

At this point in his proof, Euler writes that the denominator "reduces to unity" since "$\tan dx = dx$" [7, p. 118]. Since we know that true equality does not hold here, and that actually $\tan dx \approx dx$, we instead prove that the denominator is infinitely close to 1 using Isabelle's sledgehammer tool for automatically discovering proofs:

```
have d_unity: "(1 - (*f* tan)(star_of x)*(*f* tan)(dx)) ≈1"
  by (metis ... )
```

Sledgehammer finds a proof using proof method `metis` with various NSA lemmas.

The next step in Euler's proof is to show $dy \approx dx(1 + \tan^2 x)$. Since we have proven the denominator is infinitely close to 1, we must now prove that the numerator, which is currently $\tan dx(1 + \tan^2 x)$, can be rewritten as infinitely close to $dx(1 + \tan^2 x)$. We cannot necessarily just substitute dx for $\tan dx$, as this would not preserve the infinitely-close relation. We use the NSA lemma, `approx_mult1`, which proves that if $a \approx b$ and c is finite, then we have $a * c \approx b * c$:

```
NSA.approx_mult1: a ≈b ⟹ c ∈ HFinite ⟹a * c ≈  b * c
```

We apply this lemma to prove that the old numerator $\tan dx(1 + \tan^2 x)$ is infinitely-close to the new numerator, $dx\ (1 + \tan^2 x)$, with use of the fact that $\tan dx \approx dx$ and by proving $(1 + \tan^2 x)$ is `HFinite`.

Although we have now proven that the numerator can be rewritten as $dx\ (1 + \tan^2 x)$ and that the denominator is infinitely-close to 1, we cannot directly substitute in these results like we would in a pen and paper proof. To preserve the infinitely-close relation, we use another NSA theorem, `approx_mult_HFinite`, which proves that if $a \approx b$ and $c \approx d$, and both c and d are finite, then we have $a * c \approx b * d$.

```
NSA.approx_mult_HFinite: a ≈ b ⟹ c ≈ d ⟹
    b ∈ HFinite ⟹ d ∈ HFinite ⟹ a * c ≈ b * d
```

In our case, $a \approx b$ is the proven statement that the old numerator is infinitely-close to the new numerator,

$$\tan dx(1+\tan^2 x) \approx dx\ (1+\tan^2 x).$$

Then, the $c \approx d$ statement is,

$$inverse(1+\tan x\ \tan dx) \approx inverse 1.$$

In order to use this lemma, we prove that the inverse of our new numerator is `HFinite` using sledgehammer, and the inverse of our new denominator 1, which is 1, is also `HFinite`. Therefore, we have proven that

$$\tan dx(1+\tan^2 x) * inverse(1+\tan x\ \tan dx) \approx dx\ (1+\tan^2 x).$$

From this, it follows that the differential is infinitely-close to the new expression:

```
have
  "differential tan x dx ≈
    dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))"
```

We finish the proof by rewriting the left hand side of the above statement with our (non-standard) trigonometric identity lemma from 3.4.1, that proves $1+\tan^2(x) = \frac{1}{\cos^2(x)}$. This concludes our Isabelle proof.

Overall, we find that while we had to add extra steps in this proof to preserve the infinitely-close relation, like proving that expressions are finite to apply NSA theorems for multiplication and substitution, Euler's reasoning still holds when reapplied in this rigorous setting.

3.5 Mechanising the first differential of $\arccos x$

In this section, we build from Frankovska's project [10] by using her proof of the differential of $\arcsin x$ in our proof mechanisation. In paragraph 196, Euler derives the differential of $y = \arccos x$ [7, p. 111]. His reasoning is somewhat difficult to follow due to his presentation of various trigonometric identities, and the fact that he uses abstractions in this proof for dy and dp.

Euler writes "Suppose p is any function of x and that y is the arc whose sine is p, that is, $y = \arcsin p$ [7, p. 111]. From his previous proof of the differential of arcsin, we have the differential of y,

$$dy = \frac{dp}{\sqrt{1-p^2}}. \tag{3.12}$$

Euler writes that since "$\sqrt{1-p^2}$ expresses the cosine of that same arc, we can find the differential of an arc whose cosine is given" [7, p. 195].

He continues with "if $y = \arccos x$, then the sine of this arc is equal to $\sqrt{1-x^2}$, so that $y = \arcsin(\sqrt{1-x^2})$" [7, p. 195].

Then, Euler lets $p = \sqrt{1-x^2}$, and therefore

$$dp = \frac{-x\,dx}{\sqrt{1-x^2}}. \tag{3.13}$$

Finally, from equations (3.13), (3.12), and $\sqrt{1-p^2} = x$, we have

$$dy = \frac{-dx}{\sqrt{1-x^2}}. \tag{3.14}$$

There are a few things we note in Euler's proof here. Firstly, Euler obtains equation (3.13), the differential of p, using his previous rules for differentiation of algebraic quantities. We do not provide a proof for this differential dp as it is outside the scope of this project.

Secondly, Euler uses the chain rule in equation (3.12) when he differentiates $y = \arcsin p$, and he abstracts from what this differential of p is by referring to it simply as dp, whereas, we write Euler's dp as the full expression:

```
differential (λx. sqrt(1-x²)) x dx
```

Since we do not use the same abstractions that Euler uses for dp, this also means that we write Euler's dy, the differential of $\arcsin p$ as:

```
differential arcsin (sqrt(1-x²)) (differential (λx. sqrt(1-x²)) x
   dx)
```

Compared to Euler's proof, where it is not very obvious that he is using the chain rule, we can see the chain rule clearly in our mechanised proof, since the (non-abstracted) differential dp is within the differential dy.

In our mechanised proof, it is necessary to first prove some results about continuity and the bounds of functions. Although Euler does not explicitly discuss continuity in these specific paragraphs, we assume that it is implicit in his reasoning: in order to differentiate functions, these functions must be continuous where we differentiate them. In Isabelle, we are explicit with all our assumptions. We limit the domain over which we differentiate $\arcsin\sqrt{1-x^2}$, by restricting the bounds of x to be between 0 and 1 in our lemma statement for the differential of $\arccos x$, which can be found in appendix D.1.

We note that both $\arcsin\sqrt{1-x^2}$ and $\arccos x$ are actually continuous and defined on the interval $-1 < x < 1$. However, we limit x to be positive, between 0 and 1. This is because of the relationship between x and the function p of x. In our Isabelle proof, we use the full expression $\lambda x.\sqrt{1-x^2}$ in place of p. Then, instead of substituting x for

$\sqrt{1-p^2}$ into equation (3.12) to obtain equation (3.14) as Euler does, we must show in Isabelle that $\sqrt{1-\sqrt{1-x^2}} = x$, and in order to show that this equality holds, x must be positive. The code to prove this equality is available in appendix D.3.

In order to differentiate $\arcsin\sqrt{1-x^2}$, we must prove two facts. The first is that $\sqrt{1-x^2}$ is between by -1 and 1, as this is the domain where arcsin is defined. Isabelle's sledgehammer finds a proof for $-1 < \sqrt{1-x^2} < 1$ using our assumption that $0 < x < 1$. The second fact we must prove is that $\lambda x.\sqrt{1-x^2}$ is continuous within the interval $0 < x < 1$. Because we are using nonstandard analysis, we must prove in Isabelle that the function is (nonstandard) continuous with NSA function `isNSCont`:

```
lemma arccos_sqrt_NSCont:"isNSCont (λ x. sqrt(1-x²)) x"
```

We prove this lemma by proving that the individual functions (x^2 and the square root function) of this composite function are continuous, or `isCont` in Isabelle. Then, we use the NSA theorem `isCont_isNSCont`, which proves that if a function `isCont`, then it is also continuous in nonstandard analysis.

Once we have proven these facts, we obtain equation (3.12) in Isabelle by using Frankovska's lemma [10] for the differential of arcsin. We simplify, following Euler's reasoning, and sledgehammer proves these simplification steps with use of the NSA theorems for multiplication and substitution. We obtain the Isabelle statement that the differential of $\arcsin(\sqrt{1-x^2})$ is infinitely-close to $\frac{-dx}{\sqrt{1-x^2}}$.

```
have
  "differential arcsin (sqrt(1-x²))
     (differential (λx. sqrt(1-x²)) x dx) ≈
         -dx / (*f* sqrt)(1 - (star_of x)²)"
```

On the left-hand side, we have the differential of $\arcsin(\sqrt{1-x^2})$, which includes the differential dp in full (un-abstracted).

While we have shown that we can obtain the correct resulting differential from Euler's reasoning, we now must show that this differential of $\arcsin(\sqrt{1-x^2})$ is infinitely-close to the differential of $\arccos x$. This follows from the fact that all differentials of continuous functions are infinitesimal. Then, in general, any two differentials of continuous functions must be infinitely close to one another. In this specific case, we are only working with the domain of $0 < x < 1$, and we know that both $\arctan x$ and $\arcsin\sqrt{1-x^2}$ are continuous over this domain. Therefore, we finish our proof by writing a lemma that proves these two differentials are infinitely-close to one another:

```
lemma arccos_arcsin_relationship:
  fixes dx::hypreal and x::real
  assumes "dx ∈ Infinitesimal" "0 < x ∧ x < 1"
  shows
   "differential arcsin (sqrt(1-x²))
     (differential (λx. sqrt(1-x²)) x dx) ≈
       differential arccos x dx"
```

This lemma (available in appendix D.2) follows from the fact that both differentials are continuous on the given interval of x. Therefore, they are both infinitesimal and we can prove they are infinitely close to each-other, which concludes our mechanised proof for the differential of $\arccos x$.

To reflect upon Euler's proof compared to our rigorous mechanised proof, we find that overall, the missing steps and assumptions of Euler's proof are implicit when we consider continuity over arccos and arcsin. This does not necessary mean that Euler's proof is straightforward. Actually, we find that Euler's proof looks deceptively simple due to the abstractions of dp and dy. Our formalisation more explicitly shows the mathematics behind Euler's proof, such as the chain rule and our step that shows how two differentials are infinitely close.

3.6 Mechanising the first differential of $\arctan x$

We move forward to Euler's next proof in paragraph 197 [7, p. 111], where Euler presents his proof for the differential of $\arctan x$. Euler's proof for this differential is similar to his proof of the differential of $\arccos x$ in that he also derives this result from the differential of $\arcsin p$, where p is a function of x. His reasoning in this paragraph can be confusing at points, so we have adapted his writing in the following explanation for clarity. Euler begins the proof by letting $y = \arctan x$. From trigonometric identities, we have that

$$y = \arcsin \frac{x}{\sqrt{1+x^2}}. \tag{3.15}$$

Then, we let $p = \frac{x}{\sqrt{1+x^2}}$, and therefore

$$\sqrt{1-p^2} = \frac{1}{\sqrt{1+x^2}}. \tag{3.16}$$

Since we have defined $p = \frac{x}{\sqrt{1+x^2}}$, we rewrite y as $y = \arcsin p$. Now, Euler differentiates both y and p to obtain,

$$dy = \frac{dp}{\sqrt{1-p^2}}. \tag{3.17}$$

$$dp = \frac{dx}{(1+x^2)^{3/2}}. \tag{3.18}$$

Lastly, Euler states that when we substitute (3.18) into the numerator of (3.17) and (3.16) into the denominator of that same equation, we find that the differential dy of $\arctan x$ is

$$dy = \frac{dx}{1+x^2}. \tag{3.19}$$

Though he does not explicitly mention this, Euler obtains the differential of p in equation (3.18) using his previous rules on algebraic differentials, which we do not prove in this project. Similarly to Euler's proof for the differential of $\arccos x$ (3.5), he is also using the chain rule in equation (3.17) and abstracting the differential of p to dp in this equation.

In our proof mechanisation, as with our previous $\arccos x$ proof mechanisation, we do not use any abstractions and we fully formalise dy and dp, so that we write dp as:

```
differential (λx. x / sqrt(1 + x²)) x dx
```

And following from this, the full expression for dy using the chain rule is:

```
differential arcsin (x / sqrt(1 + x²))
    (differential (λx. x / sqrt(1 + x²)) x dx)
```

Our Isabelle proof mechanisation for this differential follows the same structure as our previous mechanisation of the differential of $\arccos x$ (see 3.5). In this proof mechanisation, we also have to explicitly add assumptions, for the same reason why we had to add assumptions in our proof mechanisation of $\arccos x$: continuity.

Specifically, in order to differentiate $\arcsin p$ as Euler does in equation (3.17), we prove that $-1 < p = \frac{x}{\sqrt{1+x^2}} < 1$, as this is the domain of arcsin. Since this statement is true for all x, we do not have to restrict x as we did in the previous proof. We also prove that p is continuous in nonstandard analysis, or `isNSCont` in Isabelle. This allows us to obtain the equation (3.17) in Isabelle, without Euler's abstractions of p and dp:

```
have dy:
  "differential arcsin (x / sqrt(1 + x²))
      (differential (λx. x / sqrt(1 + x²)) x dx) ≈
    (differential (λx. x / sqrt(1 + x²)) x dx) /
       (*f* sqrt)(1 - ((*f* (λx. x / sqrt(1 + x²)))(star_of x))²)"
```

Similarly to our previous mechanisation of the differential of arccos, Euler's proof for this differential of $\arctan x$ appears misleadingly simple compared to our Isabelle mechanisation. Once again, we use a lambda function for p, which can be seen in the denominator.

In order to use dp, we assume that the differential Euler presents for p in equation (3.18) holds, as providing a proof for this differential is outside the scope of this project, which focuses on trigonometric functions.

Despite the differences in presentation and extra steps we prove due to continuity and bounds, we otherwise manage to follow Euler's reasoning to the end of the proof, as we did in the $\arccos x$ mechanisation. We have successfully proven that the differential of $\arcsin \frac{x}{\sqrt{1+x^2}}$ is infinitely close to $\frac{dx}{1+x^2}$.

Then, we use the same process from our $\arccos x$ (3.5) proof mechanisation to show that the differential of $\arcsin \frac{x}{\sqrt{1+x^2}}$ is infinitely close to the differential of $\arctan x$, where we prove they are both continuous, therefore infinitesimal and infinitely close to each-other. This proof is available in appendix E.2.

This concludes our mechanised proof of the differential of $\arctan x$.

3.7 A second mechanisation of the first differential of $\tan x$

Euler provides a second derivation of the differential of $\tan x$ in paragraph 204 of Chapter 6, "On the Differentiation of Transcendental Functions" [7]. Similarly to how Euler uses the differential of arcsin to derive the differentials of $\arccos x$ and $\arctan x$, he uses the differential of arctan to derive the differential of $\tan x$. Euler writes that if we let $y = \tan x$, then $x = \arctan(y)$, and we can write the differential dx as,

$$dx = \frac{dy}{1+y^2}. \tag{3.20}$$

The square root of the denominator $1+y^2$ can be rewritten using trigonometric identities as,

$$\sqrt{1+y^2} = \sec x = \frac{1}{\cos x}. \tag{3.21}$$

Then, Euler substitutes $1+y^2$ in the denominator of (2.18) with $(\frac{1}{\cos x})^2$ to obtain $dx = dy\ \cos^2(x)$, and he divides both sides by $\cos^2(x)$ to isolate dy:

$$dy = \frac{dx}{\cos^2(x)}. \tag{3.22}$$

Once more, Euler makes an abstraction from the differential of $x = \arctan y$ to dx, whereas we use the full expression

```
differential arctan (tan x) (differential tan x dx)
```

for Euler's dx, and Euler is using the chain rule again by differentiating $x = \arctan y$ to obtain equation (3.20) in terms of dy, the goal differential of $\tan x$. This expression, the differential of $\arctan(\tan x)$, is infinitely close to dx, which follows from the fact that both the differential of $\arctan(\tan x)$ and dx are infinitesimal, thus these expressions are infinitely close.

There are a few ways in which our Isabelle proof looks different from Euler's. Firstly, since Euler is using the differential of $\arctan(\tan x)$ to derive this differential, we restrict the domain of x to $-\pi/2 < x < -\pi/2$ in order for $\tan x$ to be defined and continuous. This also implies that the $\cos x$ is non-zero. These assumptions allow us to differentiate $\arctan(\tan x)$.

Using our lemma from section 3.6 for the differential of $\arctan x$, we obtain equation (3.20) in Isabelle, without Euler's abstraction of dy (which is `differential tan x dx`), and dx (which is `differential arctan (tan x)(differential tan x dx)`):

```
have
  "differential arctan (tan x) (differential tan x dx) ≈
      (differential tan x dx) / (1 + ((*f* tan)(star_of x))^2)"
    using arctan_function_p tan_NSCont assms by blast
```

The lemma used to prove this statement, `arctan_function_p`, is a lemma that we proved in order to apply our differential of arctan to a function p of x, and it can be found in appendix E.4.

To obtain our final differential expression in terms of dx as in Euler's equation (3.22), we prove that the differential of $\arctan(\tan x)$ is infinitely close to dx. This follows from the fact that the differential of $\arctan(\tan x)$ is continuous on our interval of x, therefore this differential is infinitesimal and infinitely close to infinitesimal dx. The proof of this relationship is included in appendix F.1.2.

Additionally, we are unable to directly follow the last step of Euler's proof in our mechanisation, where he divides infinitesimal dx by $\cos^2(x)$, since division by an infinitesimal in nonstandard analysis does not necessarily preserve the infinitely-close relation. However, we can still reach the same result as Euler by accomplishing this 'division' in a different way. In order to preserve the infinitely-close relation, we use the nonstandard analysis lemma in Isabelle called `approx_mult1`:

```
NSA.approx_mult1: a ≈ b ⟹ c ∈ HFinite ⟹  a * c ≈ b * c
```

To accomplish division by $\cos^2(x)$, we prove that the inverse of this expression, $\frac{1}{\cos^2(x)}$ is `HFinite`, and $\cos^2(x)$ multiplied by its inverse is equal to 1:

```
have inverse_is_finite:
  "inverse (((*f* cos)(star_of x))²) ∈ HFinite"
   using assms(2)
   by (simp add: Infinitesimal_inverse_HFinite power2_eq_square
        cos_zero_pi_bounds)
have inverse_is_1:
  "((*f* cos)(star_of x))² * inverse (((*f* cos)(star_of x))²) = 1"
   using assms(2) cos_zero_pi_bounds by auto
```

We prove that the inverse is `HFinite` using an NSA theorem that states if an expression is not infinitesimal, then its inverse is `HFinite`. Our expression $\cos^2(x)$ is not infinitesimal on the domain that we have defined x, therefore we can apply this theorem to prove that it is `HFinite`. Then, we prove that $\cos^2(x)$ multiplied by its inverse is equal to 1, which follows from the fact that $\cos \neq 0$ from our assumptions.

Once we apply `approx_mult1` and simplify, we obtain the statement:

```
have
  "dx * inverse (((*f* cos)(star_of x))²) ≈
     differential tan x dx"
```

To finish the proof, we simplify using the theorem `divide_inverse` that proves division of an expression can be rewritten from multiplication of the inverse, and the theorem `approx_sym` that proves the infinitely-close relation is symmetric, and we obtain the goal statement:

```
show "differential tan x dx ≈ dx / (((*f* cos)(star_of x))²)"
```

This concludes our mechanisation of Euler's second proof of differential of $\tan x$, and we have successfully reformalised Euler's proof rigorously, although we had to include multiple extra lines to accomplish the last division step.

Chapter 4

Mechanisation of higher differential proofs

4.1 A recursive definition for higher differentials

Euler introduces formulae for higher differentials in Chapter 4, "On the Nature of Differentials of Each Order" of *Foundations of differential calculus* [7]. First, Euler assumes that x increases uniformly and thus dx is constant. Then, he states that we can find the higher order differentials of a function y of x using the same process used to find first differentials, by substituting $x + dx$ into the n^{th} differential and subtracting the n^{th} differential to find the $n + 1^{th}$ differential [7].

In this section, we will follow this recursive process presented by Euler to prove the general form of the higher differentials of sine. Using this recursive process is one interpretation of how Euler might be suggesting we can find higher differentials of $\sin x$. Later in this chapter (see section 4.2.3), we present an alternative representation and approach to finding higher differentials of $\sin x$.

In Isabelle, we define a recursive function for higher order differentials as follows:

```
primrec n_dy :: "(hypreal ⇒ hypreal) ⇒ hypreal ⇒ hypreal ⇒ nat
    ⇒ hypreal"
  where
    "n_dy f x dx 0 = f x"
  | "n_dy f x dx (Suc n) = n_dy f (x + dx) dx n - n_dy f x dx n"
```

In this definition, the keyword `primrec` signifies recursion over the primitive type `nat`, which is the type of the variable `n` that determines the order of the higher differential. Due to recursion, the definition for this function `n_dy` must take a function `f` with type `hypreal ⇒ hypreal`. Thus, our variable `x` must also be of type `hypreal`. It is necessary here to give both function `f` and variable `x` types involving `hypreal` - when the definition recurs over `n`, the output `fs` and `xs` must be the same type as the input `fs` and `xs` in order to continue recursion. The base case, where $n = 0$, is the first line written under the `where` keyword and this outputs the 0^{th} differential, which is simply the input function `f x`. The second line underneath the `where` keyword defines the recursive step from `n` to `Suc n`. The expression `Suc n` is the application of the successor function to `n`, and on paper we would write this as $n+1$.

In the next section, we use this primitive recurisve definition to mechanise the higher order differentials of $\sin x$.

4.2 Mechanisation of the higher order differentials of $\sin x$

In paragraph 205 in *Foundations of differential calculus* [7, p. 119], Euler writes, "we let $y = \sin(x)$, $z = \cos(x)$ and we keep dx constant" [7]. Then, Euler presents the first four higher order differentials of $\sin x$ and $\cos x$ as follows:

$$\begin{aligned} y &= sin(x) & z &= cos(x) \\ dy &= dx * cos(x) & dz &= -dx * sin(x) \\ d^2y &= -dx^2 * sin(x) & d^2z &= -dx^2 * cos(x) \\ d^3y &= -dx^3 * cos(x) & d^3z &= dx^3 * sin(x) \\ d^4y &= dx^4 * sin(x) & d^4z &= dx^4 * cos(x) \end{aligned}$$

4.2.1 Building a general form for the higher order differentials of $\sin x$

Although Euler only provides the first four higher order differentials of $\sin x$, we assume that he suggesting that there exists some general form. We assume this since he states in paragraph 205 that "in all of the cases in which some straight line is related to a given arc, since it can always be expressed through a sine or a cosine, it can always be differentiated without difficulty" [7, Chapter 6].

Euler does not provide an expression or proof for the n^{th} differential of sine, so in the following sections we provide an expression and proof that cooperates with the reasoning that we do have from Euler. Because our goal is to prove the general form, we need to build an expression for the higher differential of $\sin x$ that refers to the order of the higher differential, represented by the variable `n` in definition `n_dy` from section 4.1. Once we have this formalisation in Isabelle, we write a proof by induction to shows that the general form holds for all n.

```
definition sin_differential_coeff
  where "sin_differential_coeff n = (-1)^((n) div 2)"

definition sin_differential_n
  where "sin_differential_n n x dx = (sin_differential_coeff n) *
         dx^n * (if even n then (*f* sin)(x) else (*f* cos)(x))"
```

The concept from the first definition, `sin_differential_coeff`, is used as a coefficient in the second definition and it returns values `-1` or `1`, determining whether the higher differential term is negative or positive. In Isabelle, the infix operator `div` will always return a term of type `nat` when its first term is also of type `nat`. In our definition `n_dy`, we restrict `n` to be of type `nat`, and so we know that we will never obtain `-1` to the power of some fractional value, which would potentially result in `sin_differential_coeff` returning an imaginary number.

The second definition, `sin_differential_n`, formulates the general term using `n`, the order of the higher differential. The resulting term consists of a positive or negative sign that comes from the coefficient definition and the infinitesimal dx^n multiplied by $\sin x$ for even n and $\cos x$ for odd n.

Next, we prove by induction that our general form holds for all higher order differentials of $\sin x$.

4.2.2 Proof by Induction

We can now write a lemma that proves by induction that our general form holds for all n using the recursive higher order differential definition `n_dy`.

```
lemma n_dy_sin:
  fixes dx::hypreal and n::nat
  assumes "dx ∈ Infinitesimal" "x ∈ HFinite"
  shows "n_dy (*f* sin) x dx n ≈ sin_differential_n n x dx"
```

One consequence of declaring `x` to be of type `hypreal` is that we now must also add the assumption that `x ∈ HFinite`, otherwise it will not necessarily always be possible to multiply and rearrange terms that include `x` due to the constraints of nonstandard analysis.

4.2.2.1 Induction step

We prove the base case $n = 0$ using our definitions for the general form from section 4.2.1. For the induction step, we need to show that the $n+1^{th}$ differential of sine is indeed the general form we have built for $n+1$, or `Suc n` in Isabelle.

Since the higher differentials of sine alternate between $\sin x$ and $\cos x$ depending on whether n is odd or even, we must split the proof into the two possibilities and prove both. When n is odd, the n^{th} differential will contain the $\cos x$ term and as $n+1$ will be even, the $n+1^{\text{th}}$ differential will contain the $\sin x$ term. Conversely, when n is even, the $n+1^{\text{th}}$ differential will contain the $\cos x$ term.

We start the proof block for the odd case as follows:

```
proof (auto simp add: sin_differential_n_def
      sin_differential_coeff_def)
    assume odd:
      "⋀x. x ∈ HFinite ⟹
        n_dy (*f* sin) x dx n ≈ (- 1)^(n div 2) * dx^n * (*f* cos) x"
      "odd n"
      "y ∈ HFinite"
```

```
show
  "n_dy (*f* sin) (y + dx) dx n - n_dy (*f* sin) y dx n ≈
      - ((- 1) ^ (n div 2) * (dx * dx ^ n) * (*f* sin) y)"
```

The assumption statement `odd n` signifies that we are working on the odd n case. We start with the assumption that the general form holds for n for any arbitrary x (denoted by $\bigwedge$ `x`). Using these assumptions, we prove that the general form holds for `Suc n` by proving this is true for the fixed `y`. In the `show` statement, we have already unfolded the expression of `n_dy` for `Suc n` using the `n_dy` recursive definition.

The expression in the `show` statement was originally the expression for the general formula for $n+1$ from `show` statement in the previous block:

```
"n_dy (*f* sin) y dx (Suc n) ≈ sin_differential_n (Suc n) y dx"
```

Unfolding the expression with `simp` and our general form definitions from 4.2.1 with `n_dy` gives us:

```
"n_dy ( *f* sin) (y + dx) dx n - n_dy ( *f* sin) y dx n ≈
     - ((- 1) ^ (n div 2) * (dx * dx ^ n) * (*f* sin) y)"
```

Our goal is to manipulate the left hand side of this term to ultimately show that it is infinitely close to the general form for the odd case, which is on the right hand side. Note that the right hand side of the expression includes the $\sin x$ term, as we are proving the odd n case and hence the $n+1$ is even.

We present our proof reasoning as follows. We start with the left hand expression, which is the unfolded definition of the $n+1^{\text{th}}$ higher differential of $\sin x$.

$$\mathbf{n\_dy}\ \sin\ (y+dx)\ dx\ n - \mathbf{n\_dy}\ \sin\ y\ dx\ n \tag{4.1}$$

Consider the left term in this expression, $\mathbf{n\_dy}\ \sin\ (y+dx)\ dx\ n$. Since we have assumed that the general form holds for n for arbitrary x in the inductive step, we can rewrite this left term as follows:

$$\mathbf{n\_dy}\ \sin\ (y+dx)\ dx\ n \approx (-1)^{n\ div\ 2} dx^n \cos(y+dx) \tag{4.2}$$

It is important to note that rewriting this term relies on the assumption that the expression $y+dx$ finite, since this is one of the assumptions from our inductive step (assume for n). This follows from the fact that y is finite and the infinitesimal dx is also finite. We can rewrite the right-side term in the equation (4.1) from earlier, and substitute to obtain:

$$(-1)^{n\ div\ 2} dx^n \cos(y+dx) - (-1)^{n\ div\ 2} dx^n \cos y$$

From this point, we apply the cosine angle addition rule to the left-side term.

$$(-1)^{n\ div\ 2} dx^n [\cos y \cos dx - \sin y \sin dx]\ - (-1)^{n\ div\ 2} dx^n \cos y$$

Then, we can simplify using the results, discussed in 3.1, that $\sin dx \approx dx$ and $\cos dx \approx 1$, cancel out the terms, and regroup the dxs.

$$
\begin{aligned}
&(-1)^{n\,div\,2}dx^n[\cos y - \sin y\, dx] \;-(-1)^{n\,div\,2}dx^n \cos y \\
\Longrightarrow\; &(-1)^{n\,div\,2}dx^n \cos y - (-1)^{n\,div\,2}dx^n \sin y\, dx - (-1)^{n\,div\,2}dx^n \cos y \\
\Longrightarrow\; &-(-1)^{n\,div\,2}dx^{n+1} \sin y
\end{aligned}
$$

□

Thus, we have shown what we were trying to prove, that the expression (4.1) is infinitely close to the general form expression for $n+1$ (or `Suc n`).

We translate this proof into Isabelle, which can be found in full in appendix G.1.3. First, we apply our assumption that the general form holds for n on the first term and substitute in $y+dx$ for the arbitrary x.

```
have "y + dx ∈ HFinite"
   using HFinite_add Infinitesimal_subset_HFinite assms odd(3)
   by blast
have odd_dx: "n_dy (*f* sin) (y + dx) dx n ≈
        (- 1) ^ (n div 2) * dx ^ n * (*f* cos)(y + dx)"
   using "y + dx ∈ HFinite" odd
   by presburger
then
have "n_dy (*f* sin) (y + dx) dx n ≈
  (- 1) ^ (n div 2) * dx ^ n
   * ((*f* cos) y * (*f* cos) dx - (*f* sin) y *(*f* sin) dx)"
   using STAR_cos_add
   by simp
```

In order to substitute $y+dx$ into the assumption for n, we need to make sure we match the required assumption, that $y+dx$ is finite. We use theorems from the NSA theory files to prove that this is finite, which follows from the fact that the sum of two finite quantities is finite. Once we have this missing assumption, we have the statement titled `odd_dx`. From here, we can apply our (nonstandard) cosine angle addition formula to right hand side term $\cos(y+dx)$.

Like with the pen and paper proof, at this point in the Isabelle proof we simplify the resulting term from the cosine angle addition formula using the results that $\cos dx \approx 1$ and $\sin dx \approx dx$. In order to do this, we apply Fleuriot's theorems [8] for these results as we did previously in our first differential proof mechanisations (see 3.2).

We also write two additional lemmas that prove when x is a finite hyperreal number, its sine and cosine are also finite, which are available in appendix G.1.4. We prove these lemmas by applying the transfer principle to the known fact that $\sin x$ and $\cos x$ are both bounded by 1 and -1.

We also write another lemma to prove that the coefficient term, $(-1)^{n\,div\,2}dx^n$, is also finite (see appendix G.1.5). We must do this to preserve the infinitely-close relation.

One we have proved these lemmas for finiteness, we complete the rest of our proof mechanisation following our pen and paper proof that we presented previously.

The induction step for the even n case follows the same reasoning as for the odd n case. There are a few slight differences. Our corresponding `assume` and `show` statements appear as follows:

```
assume even:
  "⋀x. x ∈ HFinite ⟹
     n_dy (*f* sin) x dx n ≈(- 1)^(n div 2) * dx^n * (*f* sin) x"
  "even n"
  "y ∈ HFinite"
show
  "n_dy (*f* sin) (y + dx) dx n - n_dy (*f* sin) y dx n ≈
    (- 1)^(n div 2) * (dx * dx^n) * (*f* cos) y "
```

Our inductive assumption in the `assume even` statement is slightly different as it includes the $\sin x$ term instead of the $\cos x$ term for the odd n case. Additionally, $n+1$ will be odd, and so the general form for $n+1$, or `Suc n`, will contain $\cos y$. Therefore, when we prove this using the same reasoning as we have done in the previous case, we use the (nonstandard) sine addition formula instead. The code for the entirety of this proof is contained within the appendix G.1.3.

We note that the difficulties we have encountered with multiplying terms together in these proofs relate to the infinitely-close relation. Although we assumed that x is finite in our lemma statement for this proof by induction, we still must show at many steps that terms involving this x are `HFinite`, in order to preserve the infinitely-close relation when we multiply and make substitutions. This is a consequence of letting `x` be of type `hypreal` in our `n_dy` definition from before. This problem is not quite as prevalent in our first differentials mechanisations of Chapter (3), since we chose to let variable x be of type `real` in our first differential mechanisations.

This concludes our mechanised proof by induction that our general form for the higher differentials of $\sin x$ holds.

4.2.2.2 Analysis of our mechanised proof of the higher differentials of sine

Although Euler did not directly provide a proof by induction for these higher differentials of sine, we have managed to use his reasoning to provide our own proof of the general form of higher differentials of $\sin x$. From examination of this process, we can see that with the exception of the parts of the proof that are specific to working with nonstandard analysis, it would be entirely possible to argue Euler could have provided a similar proof himself. The methods we used, specifically the sine and cosine angle addition formulas and the simplifications of $\sin dx$ to dx and $\cos dx$ to 1, are all methods that Euler used in his proofs for the first differentials of $\sin x$ and $\cos x$ from previous paragraphs.

We know from our analysis of the first differential proofs in our Chapter (3) that Euler often skips steps or chooses not to further explain particular methods that he uses. From this, we could argue that the steps we have taken in building this proof are steps that Euler could have taken himself, and chose not to, instead leaving it up to the reader as many modern mathematics textbooks do. We could also argue that when Euler presented these first four higher differentials of $\sin x$ without any notion of a general form, he was simply making a statement that one *could* find these higher differentials.

Moreover, if we revisit what Euler has presented in paragraph 205 with the belief that Euler was not making a statement about any general form of the n^{th} differential of sine, we can find a slightly different way to approach this higher differentiation. We will cover this next, in section 4.2.3.

4.2.3 A different representation of higher differentiation of $\sin x$

In our previous proof mechanisation, we proved the general form for higher order differentials of $\sin x$ using our primitive recursive function `n_dy`, which follows Euler's formula for higher differentials from Chapter 4, "On the Nature of Differentials of Each Order" [7]. However, we will show in this section that we can find any higher order differential of sin only by using the first differentials of $\sin x$ and $\cos x$.

Euler states, "in all of the cases in which some straight line is related to a given arc, since it can always be expressed through a sine or a cosine, it can always be differentiated without difficulty," suggesting that the differentials of sin can be used to find the differentials of cos and vice versa [7, p. 119]. A clue towards the interpretation, where we only use the first differentials of $\sin x$ and $\cos x$ to find higher order differentials and without any notion of a general form, might be in his phrase, "we keep dx constant". Using this assumption, we can differentiate continuously to find higher differentials. If we alter our first differential proofs to handle such constant, we can apply these lemmas to any sine higher order differential and find the successor higher order differential.

We show this is true on paper. We start with the function $y = \sin x$. Then, we take the differential to obtain $dy \approx \cos x\, dx$. Now, if we treat dx as a constant and differentiate as normal, we obtain $d^2y \approx -\sin x\, dx\, dx$, since we know that the differential of $\cos x$ is $-\sin x\, dx$. We differentiate once more using the first differential of $\sin x$, treating the term $dx\, dx$ as a constant, and obtain the third differential, $d^3y \approx -\cos x\, dx\, dx\, dx$. Thus, we have shown that we do not need any extra definition or function to find higher differentials, as having a first differential representation suffices.

It is important to note here that when we use the definition of the first differential in this manner, it performs the exact same function as `n_dy` from section 4.1. Applying the first differential n times to an expression is the same as applying `n_dy` to the expression with value n. Additionally, they must be equivalent, otherwise one of them would not suffice as a correct method for finding a higher differential.

4.2.3.1 Altering our first differential of $\sin x$ and $\cos x$

We alter our lemma for the first differential of $\sin x$ to accommodate a constant, specifically a finite constant of the type hyperreal, in order to treat dx as a constant when we differentiate. This also involves altering the types of our definition from 3.1 for the first differential of a function. Our new definition of the first differential is as follows:

```
definition hypreal_differential ::
    "(hypreal ⇒ hypreal) ⇒ hypreal ⇒ hypreal ⇒ hypreal"
  where "hypreal_differential f x dx = f(x + dx) - f x "
```

The only difference between this definition and the definition `differential` from 3.1 are the types that we give to input function `f` and input variable `x`. In this altered definition, their input types are `hypreal ⇒ hypreal` and `hypreal` respectively, whereas in the definition from 3.1, these terms only involve type `real`. We change these types to `hypreal` in order to multiply a constant of type hyperreal with function `f`, as otherwise Isabelle would prevent us from placing dx as a multiplying constant in front of a real typed function `f`. Note that this definition is equivalent to applying `n_dy` with a value of 1 for `n`.

We build a lemma for the differential of $a \sin x$, where a is some constant, as follows:

```
lemma sin_differential_constant:
  fixes dx::hypreal and a::hypreal
  assumes "dx ∈ Infinitesimal" "a ∈ HFinite"
  shows "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx ≈
    a*(*f* cos)(star_of x)*dx "
```

In this lemma, we also add the assumption that constant a is finite in order to preserve the infinitely-close relation. The goal is to apply this lemma to an expression with dx in place of constant a. On paper, we could write the show statement as $d(a\ sin(x)) \approx a\ cos(x)\ dx$, where d represents taking the differential.

Proving this lemma is similar to how we proved the original first differential of $\sin x$ lemma from section 3.2. The proofs differ slightly. In this lemma, we must unpack the lambda expression and we must also prove some intermediate statements about finiteness due to types. We present the full mechanised proof in appendix G.2.2.

We also write and prove an altered lemma for the first differential of $\cos x$, which can be found in appendix G.2.3.

```
lemma cos_differential_constant:
  fixes dx::hypreal and a::hypreal
  assumes "dx ∈ Infinitesimal" "a ∈HFinite"
  shows "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx ≈
    -a*(*f* sin)(star_of x)*dx "
```

Although we use the infinitely-close relation, overall, the mechanisations of these lemmas follow Euler's reasoning from his own proofs of the first differentials of $\sin x$ and $\cos x$ [7, Chapter 6]. Therefore, we argue that it might be reasonable to assume that the intuition behind these two lemmas corresponds to what Euler might have intended when presented the higher differentials of $\sin x$ and $\cos x$ without any notion of a general form.

In the next section, we show how we can use these two lemmas, titled `sin_differential_constant` and `cos_differential_constant`, to find higher differentials of $\sin x$.

4.2.3.2 Finding higher differentials of $\sin x$ with this approach

In this section, we show examples of how we can apply lemmas `sin_differential_constant` and `cos_differential_constant` with `dx` as constant `a` to find higher differentials of $\sin x$.

We consider the second differential of sine, $d^2y \approx -dx^2 \sin x$. We write a lemma to show how we can apply our altered lemma for the first differential of $\cos x$, `cos_differential_constant`, to obtain the succeeding differential of $\sin x$. We apply this lemma to the first differential of $\sin x$, which is $dy \approx -dx \cos x$.

```
lemma differential_sine_2:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows "hypreal_differential (λy. dx*(*f* cos) y) (star_of x) dx ≈
     -(*f* sin)(star_of x)*dx²"
```

We use a lambda expression to construct input function f as $dx \cos x$. Here, we can see that `dx` takes place of constant `a`. The rest of the proof is only a few lines:

```
have "hypreal_differential (λy. dx*(*f* cos) y) (star_of x) dx ≈
      -dx*(*f* sin)(star_of x)*dx"
    using assms cos_differential_constant dx_x_HFinite
    by blast
then
show "hypreal_differential (λy. dx*(*f* cos) y) (star_of x) dx ≈
      -(*f* sin)(star_of x)*dx²"
    by (simp add: mult.assoc mult.commute power2_eq_square)
qed
```

Note that we must have the fact that infinitesimal dx is finite, as we must match the assumption that `a ∈ HFinite`. We prove this in the lemma `dx_x_HFinite`, which we apply in the first step, along with `cos_differential_constant`. Then, we arrange the dx terms and we have shown how we can find the successor higher differential of $dy \approx -dx \cos x$ using only this first differential mechanisation, `cos_differential_constant`.

As an example, we provide the lemma statement for one order higher:

```
lemma differential_sine_3:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows
    "hypreal_differential (λy. -(dx²)*(*f* sin) y) (star_of x) dx ≈
      -(*f* cos)(star_of x)*dx³"
```

```
proof -
  have
    "hypreal_differential (λy. -(dx²)*(*f* sin) y) (star_of x) dx ≈
           -(dx²)*(*f* cos)(star_of x)*dx"
      using assms sin_differential_constant [where a="(-(dx²))"]
      by (simp add: dx_x_HFinite HFinite_minus_iff hrealpow_HFinite)
```

When we apply the lemma `sin_differential_constant`, we specify that the constant `a` is `-(dx²)` within the `[where]` clause. The rest of the proof only includes rearranging the dx terms into dx^3.

We do not provide any more proofs for other higher differentials of $\sin x$ in this report, but these two lemmas `sin_constant_differential` and `cos_constant_differential` could be used to find any higher differential of $\sin x$ or $\cos x$ with repeated applications of `hypreal_differential` to $\cos x$ or $\sin x$.

4.2.3.3 Analysis of this representation of Euler's higher differentials of sine

This method and representation requires fewer lemmas than the previous mechanisation from 4.2.2. Although we did not provide any proof for the higher differentials of $\cos x$, we could use the definitions we presented in this section to find any higher differential of $\cos x$ without needing to formalise anything extra.

We revisit Euler's prose in paragraph 205, where he presents the first four differentials of both $\sin x$ and $\cos x$, and he states that a given arc of cosine or sine "can always be differentiated without difficulty... not only of the first differentials, but also of the second and succeeding differentials by the given rules" [7, p. 119].

We cannot be certain what "given rules" he is referring to [7, p. 119]. Euler may be referencing the recursive formula for higher differentials that he presents in Chapter 4, "On the Nature of Differentials of Each Order", as when he presents his formula for higher order differentials in general he also assumes that x increases uniformly and thus dx is constant [7]. Or, he may be referencing only the first differential proofs he provides for $\sin x$ and $\cos x$ in Chapter 6, "On the Differentiation of Transcendental Functions" [7] since he does not provide any notion of a general form for the higher differentials of $\sin x$, which our proof by induction requires. In this project, we have provided a mechanisation for both interpretations.

Fundamentally, both interpretations can be seen as the same method. If we simplify `n_dy`, it accomplishes same task as repeatedly applying `hypreal_differential`, which is to substitute $x + dx$ into the expression and subtract the original expression.

This concludes our exploratory Chapter on Euler's presentation of the higher differentials of $\sin x$.

Chapter 5

Conclusions

5.1 Summary

We successfully mechanised all of the first differential proofs that we investigated from Euler's *Foundations of differential calculus* [7]:

- the differential of sine (paragraph 201) in 3.2,
- the differential of cosine (paragraph 202) in 3.3,
- a first derivation of the differential of tan (paragraph 203) in 3.4,
- the differential of arccosine (paragraph 196) in 3.5,
- the differential of arctan (paragraph 197) in 3.6,
- a second derivation of the differential of tan (paragraph 204) in 3.7.

We also provided a mechanised proof of the higher differentials of sine, which Euler presented but did not prove, and this mechanisation process included deriving a general form for the higher differentials of sine in 4.2.1, providing a pen and paper proof which we discussed in 4.2.2, and proving this by induction. We presented an alternative interpretation of Euler's reasoning for these differentials in 4.2.3, and we provided examples of how this mechanisation can be used to find higher differentials of sine.

5.2 Reflections and future work

In this section, we return to our questions from the Introduction (1) to reflect upon our work and discuss how it may be extended upon in the future.

5.2.1 Interpreting Euler's meaning

Firstly, how far were we able to interpret Euler's meaning in his proof derivations?

Interpreting Euler's reasoning in *Foundations of differential calculus* [7] ranged from straightforward to complex, depending on the proof. His proofs for the differential of sine, cosine and tangent are relatively straightforward to interpret, especially since he proves these differentials using his definition of the first differential, $dy = f(x+dx) + f(x)$ (see Euler's definitions in 2.1.1).

It is more difficult to follow and understand his proofs for the differentials of arccos (section 3.5) and arctan (section 3.6). Euler makes these proofs look deceptively simple due to his abstraction of dp and dy, and they involve many intermediate, unpresented steps. But overall, his reasoning in these proofs can still be interpreted with consideration.

We found it most difficult to interpret Euler's reasoning for the higher differentials of sine in paragraph 205 [7, p. 119], since he does not provide a proof for these differentials at all. We are uncertain whether Euler meant that we could find these higher differentials using his recursive definition for orders of differentials (section 4.1), or whether he meant that we could find higher differentials by using just the first differentials of sine and cosine (section 4.2.3). Regardless, although he has omitted the proofs for these higher differentials of sine, we are still able to find two interpretations through analysis of his prose, and we formalise mechanisations for these interpretations using only the definitions that he provided in previous paragraphs.

If we consider the specific criticisms presented against Euler (which we discussed in section 2.1.2) that Euler skips steps, or fails to mention assumptions, we argue in response that we have found it is entirely possible to interpret Euler's meaning and move forward in our proofs from what he has given us. The only missing assumptions we faced involved the domains of functions, or where functions are continuous, and these are implicit from our knowledge of the functions themselves. Additionally, we have shown that even though Euler omits the proofs of the higher differentials of sine, we can still prove his results by using only the reasoning he provides elsewhere in the text.

5.2.2 Euler's intuition and modern standards of rigour

Secondly, to what extent do we have to adapt Euler's proofs in Isabelle with nonstandard analysis to adhere to modern standards of rigour?

Overall, we find that Euler's reasoning in his proofs is consistent with our rigorous mechanised proofs in Isabelle. The steps of Euler's proofs that we were not able to directly follow relate to division and multiplication with infinitesimals. For multiplication of infinitesimals, we address this problem of preserving the infinitely-close relation by proving expressions are finite, or `HFinite` in Isabelle, and using the nonstandard analysis theorems to 'multiply' and rewrite equations.

We did not face same problem that Rockel [15] and Frankovska [10] faced in their projects, where Euler divides by an infinitesimal, but we did discover an example of division of an infinitesimal, when Euler divides infinitesimal dx by $\cos^2(x)$ in 3.4. In this case, we also found a workaround by using the NSA multiplication theorems and 'manipulating' division into multiplication of the inverse.

To address criticisms against Euler's treatment of infinitesimals, we argue in response that although we have to adapt Euler's reasoning to adhere to the rules of nonstandard analysis (and thus, adhere to modern standards of rigour), Euler's intuition is still consistent, since we arrive at the same result as Euler when rehabilitating his steps with rigour. Moreover, although we cannot necessarily follow the exact step (division of an infinitesimal) as Euler in our rigorous framework, since we must preserve the infinitely-close relation, we have still shown that this given step can be reformalised rigorously to provide the same result. Our workaround can be applied to the general case of division of an infinitesimal, but this also requires finiteness of the expression that we are dividing the infinitesimal by.

5.2.3 Future work

In our section for our mechanisations of the higher differentials of $\sin x$ (section 4.2), we provided two different interpretations of how Euler might have intended the reader to find these higher differentials. Both interpretations only use Euler's reasoning from previous chapters, and we showed how both mechanisations can find a higher differential of $\sin x$. However, we only produced and proved a general form for the higher differentials of $\sin x$, whereas Euler presents higher differentials for both $\sin x$ and $\cos x$. A possible continuation of the work in this section (section 4.2) might be to formalise a general form for the higher order differentials of $\cos x$, and prove this by induction.

Other future work might include completing proof mechanisations for the other trigonometric differentials in Euler's *Foundations of differential calculus* [7, Chapter 6], such as the differential of cotangent in paragraph 198, the differential of secant in paragraph 205, and the higher differentials of cosine and tangent, presented in paragraph 207. This would complete the set of mechanisations of trigonometric differentials from Chapter 6 of Euler's *Foundations of differential calculus* [7, Chapter 6].

5.2.4 Concluding remarks

In conclusion, our exploration shows that we can indeed rigourously reformulate Euler's proofs of trigonometric functions using our rigorous framework of Isabelle with nonstandard analysis. Although we found that Euler did occasionally omit steps and assumptions, we argue that overall, this does not prevent us from following his proofs. Likewise, we find that Euler's reasoning is consistent when adapted to this rigorous framework. Our exploration has provided useful insights on understanding Euler's reasoning and his implicit assumptions, and our mechanisations provide groundwork for future research on this influential mathematician and his *Foundations of differential calculus* [7].

Bibliography

[1] Kirsti Andersen. Berkeley's arguments on compensating errors in the calculus. Historia Mathematica, 38:219–231, 5 2011.

[2] George Berkeley. The Analyst: A Discourse Addressed to an Infidel Mathematician. Wilkins, David R., 1734.

[3] Jasmin Blanchette. Hammering Away A User's Guide to Sledgehammer for Isabelle/HOL, 2021.

[4] William Dunham. Journey through Genius: Great Theorems of Mathematics. John Wiley & Sons, 3 1990.

[5] Harold M. Edwards. Euler's definition of the derivative. Bulletin of the American Mathematical Society, 44:575–580, 10 2007.

[6] Leohnard Euler. Introductio in Analysin Infinitorum. Springer-Verlag, 1988.

[7] Leohnard Euler. Foundations of Differential Calculus. Springer-Verlag, 2000.

[8] Jacques D. Fleuriot. Theorem proving in infinitesimal geometry, 2001.

[9] Jacques D. Fleuriot and Lawrence C. Paulson. Mechanizing nonstandard real analysis. LMS Journal of Computation and Mathematics, 3:140–190, 2000.

[10] Zuzana Frankovska. Exploring exploring euler's foundations of differential calculus in isabelle/hol using nonstandard analysis, 2019.

[11] Robert Goldblatt. Lectures on the Hyperreals. Springer-Verlag New York, 1998.

[12] Mark McKinzie and Curtis Tuckey. Higher trigonometry, hypperreal numbers, and euler's analysis of infinites. 74:339–368, 12 2001.

[13] L. C. Paulson. Isabelle: A Generic Theorem Prover. Springer-Verlag Berlin, 1 edition, 1994.

[14] Abraham Robinson. Non-standard Analysis. Princeton University Press, 1974.

[15] Jessika Rockel. Exploring exploring euler's foundations of differential calculus in isabelle/hol using nonstandard analysis, 2019.

[16] Ed Sandifer. How euler did it, 9 2006.

[17] Makarius Wenzel. The Isabelle/Isar Reference Manual, 2020.

[18] J. O. Wisdom. Berkeley's criticism of the infinitesimal. The British Journal for the Philosophy of Science, 4(13):22–25, 1953.

Appendix A

Paragraph 201 - $\sin x$

A.1 The first differential of $\sin x$

```
lemma differential_sin:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows "differential sin x dx ≈(*f* cos)(star_of x)*dx"
proof -
  have sindx: "(*f* cos)(star_of x)*(*f* sin)(dx) ≈
      (*f* cos)(star_of x)*(dx)"
    using STAR_sin_infinitesimal
    by (simp add: approx_mult2 assms)
  have cosdx: "(*f* sin)(star_of x)*(*f* cos)(dx) ≈
      (*f* sin)(star_of x)"
    using STAR_cos_infinitesimal
    by (metis HFinite_star_of mult.right_neutral starfun_star_of
    approx_mult2 assms)
  have "differential sin x dx = (*f* sin)(star_of x + dx) -
      (*f* sin)(star_of x)"
    using differential_def by auto
  also
  have "... = (*f* sin)(star_of x)*(*f* cos)(dx) +
      (*f* cos)(star_of x)*(*f* sin)(dx)
           - (*f* sin)(star_of x)"
    using STAR_sin_add by simp
  also
  have "... ≈ (*f* sin)(star_of x) + (*f* cos)(star_of x)*(dx) -
       (*f* sin)(star_of x)"
    using sindx cosdx approx_diff approx_add by blast
  finally
  show "differential sin x dx ≈(*f* cos)(star_of x)*dx"
    by simp
qed
```

Appendix B

Paragraph 202 - $\cos x$

B.1 The first differential of $\cos x$

```
lemma differential_cos:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows "differential cos x dx ≈-dx*(*f* sin)(star_of x)"
proof -
  have sindx: "(*f* sin) (star_of x) * (*f* sin) dx ≈
       dx*(*f* sin)(star_of x)"
    using STAR_sin_Infinitesimal
    by (metis (full_types) approx_mult_subst_star_of approx_refl
    assms mult.commute starfun_eq)
  have cosdx: "(*f* cos) (star_of x) * (*f* cos) dx ≈
      (*f* cos)(star_of x)"
    using STAR_cos_Infinitesimal
    by (metis HFinite_star_of mult.right_neutral starfun_star_of
    approx_mult2 assms)
  have "differential cos x dx = (*f* cos)(star_of x + dx) -
       (*f* cos)(star_of x)"
    using differential_def by auto
  also
  have "... = (*f* cos)(star_of x)*(*f* cos)(dx) -
       (*f* sin)(star_of x)*(*f* sin)(dx) - (*f* cos)(star_of x)"
    using STAR_cos_add by simp
  also
  have "... ≈ (*f* cos)(star_of x) - dx*(*f* sin)(star_of x) -
       (*f* cos)(star_of x)"
    using sindx cosdx approx_diff by blast
  finally
  show "differential cos x dx ≈-dx*(*f* sin)(star_of x)"
    by auto
qed
```

Appendix C

Paragraph 203 - $\tan x$ v1

C.1 Tan and cos identity

C.1.1 Tan and cos identity in the reals

```
lemma tan_cos_squared:
  assumes "cos x ≠ 0"
  shows "1 + (tan x)² = 1 / (cos x)²"
proof -
  have sin_cos_squared: "(sin x)² + (cos x)² = 1"
    by simp
  also
  have "(sin x)² / (cos x)² + (cos x)² / (cos x)² = 1 / (cos x)²"
    by (metis add_divide_distrib calculation)
  then
  have "((sin x)/(cos x))² + 1 = 1 / (cos x)²"
    by (simp add: assms power_divide)
  then
  show "1 + (tan x)² = 1 / (cos x)²"
    by (simp add: add.commute tan_def)
qed
```

C.1.2 Tan and cos identity in nonstandard analysis

```
lemma STAR_tan_cos_squared:
 "⋀(x::hypreal). (*f* cos)(x) ≠ 0 ⟹
     1 + ((*f* tan)(x))² = 1 / ((*f* cos)(x))²"
  apply (transfer) by (simp add: tan_cos_squared)
```

C.2 The first differential of $\tan x$

```
lemma differential_tan:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal" "cos x ≠ 0"
  shows "differential tan x dx ≈dx / ((*f* cos)(star_of x))²"
proof -
  have tan_add_assumptionA: "(*f* cos)(star_of x) ≠ 0"
    by (simp add: assms(2))
  have tan_add_assumptionB: "(*f* cos)(dx) ≠ 0"
    using STAR_cos_Infinitesimal assms(1) by fastforce
  have tan_add_assumptionC: "(*f* cos)(star_of x + dx) ≠ 0"
  proof -
    have cos_dx: "(*f* cos)(star_of x)*(*f* cos)(dx) ≈
        (*f* cos)(star_of x)"
      using STAR_cos_infinitesimal
      by (metis HFinite_star_of approx_mult2 assms(1)
          mult.right_neutral starfun_star_of)
    have sin_dx: "(*f* sin)(star_of x)*(*f* sin)(dx) ≈
        (*f* sin)(star_of x)*(dx)"
      by (simp add: approx_mult2 assms(1))
    have "(*f* cos)(star_of x + dx) =
          (*f* cos)(star_of x)*(*f* cos)(dx) -
             (*f* sin)(star_of x)*(*f* sin)(dx)"
      using STAR_cos_add by auto
    also
    have "... ≈ (*f* cos)(star_of x) - (*f* sin)(star_of x)*dx"
      using cos_dx sin_dx approx_diff by blast
    also
    have "... ≈ (*f* cos)(star_of x)"
      using Infinitesimal_approx_minus Infinitesimal_star_of_mult2
         assms(1) by fastforce
    finally
    show "(*f* cos)(star_of x + dx) ≠ 0"
      using assms(2) by auto
  qed
  have tan_dx:"(*f* tan)(dx) ≈dx"
    by (simp add: assms(1))
  have tan_dx_zero: "(1 - (*f* tan)(star_of x)*(*f* tan)(dx)) ≠ 0"
    by (metis Infinitesimal_star_of_mult approx_trans assms(1)
       mem_infmal_iff mult.commute one_not_Infinitesimal
       right_minus_eq starfun_star_of tan_dx)
  have "differential tan x dx = (*f* tan)(star_of x + dx) -
      (*f* tan)(star_of x)"
    using differential_def by simp
  also
```

```
have "... = (((*f* tan)(star_of x) + (*f* tan)(dx)) /
     (1 - (*f* tan)(star_of x)*(*f* tan)(dx))) -
       (*f* tan)(star_of x)"
  using STAR_tan_add tan_add_assumptionA tan_add_assumptionB
    tan_add_assumptionC
  by simp
finally
have A: "differential tan x dx =
   (*f* tan)(dx)*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x)) /
          (1 - (*f* tan)(star_of x)*(*f* tan)(dx))"
  using tan_dx_zero by (simp add: field_simps)
have d_unity: "(1 - (*f* tan)(star_of x)*(*f* tan)(dx)) ≈1"
    by (metis Infinitesimal_HFinite_mult HFinite_star_of
        Infinitesimal_add_approx_self approx_trans
        assms(1) diff_add_cancel mem_infmal_iff
        mult.commute starfun_star_of tan_dx)
have numerator:
  "(*f* tan)(dx)*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))
    ≈ dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))"
proof -
  have "(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x)) ∈HFinite"
    by (simp add: HFinite_add)
  then
  show
   "(*f* tan)(dx)*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))
      ≈ dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))"
    using approx_mult1 tan_dx by blast
qed
have "differential tan x dx ≈
  dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))"
proof -
  have d2_unity:
    "inverse (1 - (*f* tan)(star_of x)*(*f* tan)(dx)) ≈1"
    using d_unity
    by (metis hypreal_of_real_approx_inverse inverse_1
      one_neq_zero star_one_def)
  have n_finite:
    "dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x)) ∈HFinite"
    by (metis HFinite_1 HFinite_add HFinite_mult HFinite_star_of
         approx_star_of_HFinite assms(1)
         mem_infmal_iff star_zero_def starfun_star_of)
  have
    "(*f* tan)(dx)*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))
  *
        inverse (1 - (*f* tan)(star_of x)*(*f* tan)(dx)) ≈
        dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))*1"
```

```
      using numerator n_finite d2_unity HFinite_1
    approx_mult_HFinite
      by blast
    then
    show
      "differential tan x dx ≈
        dx*(1 + (*f* tan)(star_of x)*(*f* tan)(star_of x))"
      using A
      by (simp add: divide_inverse)
  qed
  then
  show
    "differential tan x dx ≈
      dx / ((*f* cos)(star_of x))²"
    using STAR_tan_cos_squared tan_add_assumptionA
    by (metis divide_inverse inverse_eq_divide power2_eq_square)
qed
```

Appendix D

Paragraph 196 - $\arccos x$

D.1 The first differential of $\arccos x$

```
lemma differential_arccos:
  fixes dx::hypreal and x::real
  assumes "dx ∈ Infinitesimal" "0 < x ∧  x < 1"
  shows "differential arccos x dx ≈
     - dx / (*f* sqrt) (1 - (star_of x)²)"
proof -
    that is, y = arcsin p.'... Let p = sqrt(1-x²) *)
  let ?y = "arcsin(sqrt(1-x²))"
  have assumpt1: "-1 < (sqrt(1-x²)) ∧(sqrt(1-x²)) < 1"
    by (smt assms(2) one_min_sq_positive real_sqrt_ge_0_iff
    real_sqrt_lt_1_iff zero_less_power2)
  have assumpt2: "isNSCont (λx. sqrt(1-x²)) x"
    using arccos_sqrt_NSCont by simp
  from assumpt1 assms assumpt2
  have "differential arcsin (sqrt(1-x²))
      (differential (λx. sqrt(1-x²)) x dx) ≈
         (differential (λx. sqrt(1-x²)) x dx) /
           (*f* sqrt)((1 - ((*f* (λx. sqrt(1-x²))) (star_of x))²))"
    by (rule_tac arcsin_function_p)
  moreover
  have p:
    "((*f* sqrt)((1 - ((*f* (λx. sqrt(1-x²))) (star_of x))²))) =
        ((*f* sqrt)(1 - ((*f* sqrt)(1 - (star_of x)²))²))"
    by (metis (no_types, lifting) star_of_diff star_of_power
    star_one_def starfun_eq)
  ultimately
  have dy:
    "differential arcsin (sqrt(1-x²))
      (differential (λx. sqrt(1-x²)) x dx) ≈
```

```
      (differential (λx. sqrt(1-x²)) x dx) /
      (*f* sqrt)(1 - ((*f* sqrt)(1 - (star_of x)²))²)"
    by simp
  have dp: "differential (λx. sqrt(1-x²)) x dx ≈
    -(star_of x)*dx / (*f* sqrt)(1 - (star_of x)²)"
    by (smt Infinitesimal_approx Infinitesimal_star_of_mult2
       assms(1) assumpt2 differential_infinitesimal
       divide_inverse inverse_eq_divide mem_infmal_iff
       mult.commute star_of_diff star_of_divide
       star_of_minus star_of_power star_one_def starfun_star_of)
  have p_x: "(*f* sqrt)(1 - ((*f* sqrt)(1 - (star_of x)²))²) =
      star_of x"
    using arccos_sqrt_simplify assms(2) by blast
  have "differential arcsin (sqrt(1-x²))
       (differential (λx. sqrt(1-x²)) x dx) ≈
         -(star_of x)*dx / (*f* sqrt)(1 - (star_of x)²) /
         (*f* sqrt)(1 - ((*f* sqrt)(1 - (star_of x)²))²)"
    by (smt Infinitesimal_HFinite_mult dy dp p approx_sym
        approx_trans arcsin_deriv_finite assms(1)
        assumpt1 assumpt2 differential_infinitesimal
        divide_inverse mem_infmal_iff starfun_star_of)
  then
  have "differential arcsin (sqrt(1-x²))
       (differential (λx. sqrt(1-x²)) x dx) ≈
       -dx / (*f* sqrt)(1 - (star_of x)²)"
    using assumpt1 p_x by auto
  then
  show "differential arccos x dx ≈
       - dx / (*f* sqrt) (1 - (star_of x)²)"
    using approx_trans3 arccos_arcsin_relationship
        assms(1) assms(2) by blast
qed
```

D.2 The differential of $\arcsin(\sqrt{1-x^2})$ is infinitely-close to the differential of $\arccos x$

```
lemma arccos_arcsin_relationship:
  fixes dx::hypreal and x::real
  assumes "dx ∈ Infinitesimal" "0 < x ∧ x < 1"
  shows
    "differential arcsin (sqrt(1-x²))
     (differential (λx. sqrt(1-x²)) x dx) ≈
     differential arccos x dx"
proof -
  have A: "differential arcsin (sqrt(1-x²))
      (differential (λx. sqrt(1-x²)) x dx) ≈0"
    by (smt arccos_sqrt_NSCont assms(1) assms(2)
        differential_infinitesimal isNSCont_arcsin mem_infmal_iff
        one_min_sq_positive real_sqrt_le_0_iff
        real_sqrt_lt_1_iff zero_less_power2)
  have "isCont (λx. arccos x) x"
    using assms(2) isCont_arccos by auto
  then
  have "isNSCont (λx. arccos x) x"
    using isCont_isNSCont by blast
  then
  have B: "differential arccos x dx ≈  0"
    using assms(1) differential_infinitesimal mem_infmal_iff
    by blast
  then
  show "differential arcsin (sqrt(1-x²))
     (differential (λx. sqrt(1-x²)) x dx) ≈
     differential arccos x dx"
    using A approx_trans2
    by blast
qed
```

D.3 $\sqrt{1-p^2} = x$ when $p = \sqrt{1-x^2}$

```
lemma arccos_sqrt_simplify:
  fixes x::real
  assumes "0 < x ∧   x < 1"
  shows
    "(*f* sqrt)(1 - ((*f* sqrt)(1 - (star_of x)²))²) = star_of x"
  proof -
    have "(*f* sqrt)(1 - (star_of x)²)² = (1 - (star_of x)²)"
      by (smt assms diff_ge_0_iff_ge hypreal_sqrt_pow2_iff
```

```
          one_power2 power2_le_imp_le
          power2_minus star_of_le_1 star_of_power)
    then
    have "1 - (*f* sqrt)(1 - (star_of x)²)² = (star_of x)²"
      by simp
    then
    have "(1 - (*f* sqrt)(1 - (star_of x)²)²) ≥0"
      using assms by auto
    then
    show
      "(*f* sqrt)(1 - ((*f* sqrt)(1 - (star_of x)²))²) = star_of x"
      using <1 - (( *f* sqrt) (1 - (hypreal_of_real x)²))² =
         (hypreal_of_real x)²> assms
      by auto
qed
```

Appendix E

Paragraph 197 - $\arctan x$

E.1 The first differential of $\arctan x$

```
lemma differential_arctan:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows "differential arctan x dx ≈dx / (1 + (star_of x)²)"
proof -
  let ?y = "arctan(x)"
  have sin_y: "?y = arcsin(x / sqrt(1 + x²))"
    by (simp add: arctan_arcsin)
  have cos_y: "cos(?y) = 1 / sqrt(1 + x²)"
    by (metis cos_arctan)
  let ?p = "x / sqrt(1 + x²)"
  have sqrt_p_value: "sqrt(1 - ?p²) = 1 / sqrt(1 + x²)"
    using arctan_p_sqrt by blast
  have assm1: "-1 < ?p ∧?p < 1"
    using p_bounds by blast
  have assm2: "isNSCont (λx. x / sqrt (1 + x²)) x"
    using arcsin_arctan_sqrt_NSCont by simp
  have dy: "differential arcsin (x / sqrt(1 + x²))
     (differential (λx. x / sqrt(1 + x²)) x dx) ≈
     (differential (λx. x / sqrt(1 + x²)) x dx) /
     (*f* sqrt)(1 - ((*f* (λx. x / sqrt(1 + x²)))(star_of x))²)"
    using assm1 assm2 assms arcsin_function_p
    by meson
  have dp: "differential (λx. x / sqrt(1 + x²)) x dx =
    dx / ((*f* sqrt)(1 + (star_of x)²))³"
    sorry
  have STAR_sqrt_p_value:
    "(*f* sqrt)(1 - ((*f* (λx. ?p))(star_of x))²) =
      1 / (*f* sqrt)(1 + (star_of x)²)"
```

```
    by (metis STAR_cos_arctan cos_y sqrt_p_value star_of_diff
       star_of_power star_one_def starfun_star_of)
  have "differential arcsin (x / sqrt(1 + x²))
    (differential (λx. x / sqrt(1 + x²)) x dx) ≈
    dx / ((*f* sqrt)(1 + (star_of x)²))³ /
    (*f* sqrt)(1 - ((*f* (λx. x / sqrt(1 + x²)))(star_of x))²)"
    using "dy" dp by auto
  then
  have "differential arcsin (x / sqrt(1 + x²))
     (differential (λx. x / sqrt(1 + x²)) x dx) ≈
     dx / ((*f* sqrt)(1 + (star_of x)²))³ /
     (1 / (*f* sqrt)(1 + (star_of x)²))"
    using STAR_sqrt_p_value by auto
  then
  have "differential arcsin (x / sqrt(1 + x²))
     (differential (λx. x / sqrt(1 + x²)) x dx) ≈
     ((*f* sqrt)(1 + (star_of x)²) * dx) /
     ((*f* sqrt)(1 + (star_of x)²))³"
    by (simp add: mult.commute)
  then
  have "differential arcsin (x / sqrt(1 + x²))
    (differential (λx. x / sqrt(1 + x²)) x dx) ≈
    dx / ((*f* sqrt)(1 + (star_of x)²))²"
    by (smt Infinitesimal_approx Infinitesimal_star_of_mult2
        STAR_cos_arctan assm1 assm2 assms differential_infinitesimal
        divide_inverse inverse_eq_divide isNSCont_arcsin
        mem_infmal_iff mult.commute power_divide power_one
        star_of_power starfun_star_of)
  then
  have final: "differential arcsin (x / sqrt(1 + x²))
      (differential (λx. x / sqrt(1 + x²)) x dx) ≈
     dx / (1 + (star_of x)²)"
    by (smt hypreal_sqrt_gt_zero_pow2 star_of_add star_of_less
        star_of_power star_one_def star_zero_def zero_le_power2)
  have "differential arcsin (x / sqrt(1 + x²))
      (differential (λx. x / sqrt(1 + x²)) x dx) ≈
     differential arctan x dx"
    using approx_sym assm1 assms differential_arcsin_arctan by
   blast
  then show "differential arctan x dx ≈dx / (1 + (star_of x)²)"
    using final approx_trans3 by blast
qed
```

E.2 The differential of $\arcsin \frac{x}{\sqrt{1+x^2}}$ is infinitely-close to the differential of $\arctan x$

```
lemma differential_arcsin_arctan:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal"
  shows "differential arctan x dx ≈
    differential arcsin (x / sqrt(1 + x²))
      (differential (λx. x / sqrt(1 + x²)) x dx)"
proof -
  have A: "differential arcsin (x / sqrt(1 + x²))
     (differential (λx. x / sqrt(1 + x²)) x dx) ≈0"
    by (metis (no_types, lifting) arcsin_arctan_sqrt_NSCont
       assms(1) p_bounds differential_infinitesimal
       isNSCont_arcsin mem_infmal_iff)
  have B: "isNSCont (λx. arctan x) x"
  proof -
    have "isCont (λx. arctan x) x"
      by (simp add: isCont_arctan)
    then
    show "isNSCont (λx. arctan x) x"
      by (simp add: isCont_isNSCont)
  qed
  have B': "differential arctan x dx ≈0"
    by (simp add: B Infinitesimal_approx assms(1)
       differential_infinitesimal)
  show "differential arctan x dx ≈
     differential arcsin (x / sqrt(1 + x²))
       (differential (λx. x / sqrt(1 + x²)) x dx)"
    using A B' approx_trans2
    by blast
qed
```

E.3 The relationship between p and x: $\sqrt{1-p^2} = \frac{1}{\sqrt{1+x^2}}$ when $p = \frac{x}{\sqrt{1+x^2}}$

```
lemma arctan_p_sqrt:
  "sqrt(1 - (x / sqrt(1 + x²))²) = 1 / sqrt(1 + x²)"
proof -
    have sin_y: "sin(arctan(x)) = x / sqrt(1 + x²)"
      by (metis sin_arctan)
    have cos_y: "cos(arctan(x)) = 1 / sqrt(1 + x²)"
      by (metis cos_arctan)
```

```
    have cos_zero: "cos(arctan(x)) ≥0"
      using cos_y hypreal_sqrt_ge_zero by auto
    have "x / sqrt(1 + x²) = sin(arctan(x))"
      by (metis sin_y)
    then
    have "(cos(arctan(x)))² = 1 - (x / sqrt(1 + x²))²"
      by (simp add: cos_squared_eq)
    then
    have "cos(arctan(x)) = sqrt(1 - (x / sqrt(1 + x²))²)"
      using cos_sin_sqrt cos_zero
      by (metis sin_y)
    then
    show "sqrt(1 - (x / sqrt(1 + x²))²) = 1 / sqrt(1 + x²)"
      using cos_y
      by metis
  qed
```

E.4 The differential of arctan of a function p of x

```
lemma arctan_function_p:
  fixes dx::hypreal and x::real
  assumes "dx ∈ Infinitesimal" "isNSCont p x"
  shows "differential arctan (p x) (differential p x dx) ≈
    (differential p x dx) / (1 + ((*f* p)(star_of x))²)"
  by (simp add: assms(1) assms(2) differential_arctan
    differential_infinitesimal)
```

E.5 Helper lemmas: bounds and continuity

E.5.1 (Nonstandard) Continuity of p, where $p = \frac{x}{\sqrt{1+x^2}}$

```
lemma arcsin_arctan_sqrt_NSCont:
  "isNSCont (λx. x / sqrt (1 + x²)) x"
proof -
  have "isCont (λx. x²) x"
    using isCont_power [where f="λx. x" and n=2]
       continuous_ident
    by blast
  then
  have "isCont (λx. 1 + x²) x"
    by auto
  then
```

```
  have denominator: "isCont (λx. sqrt(1 + x²)) x"
    using continuous_real_sqrt by blast
  have "sqrt(1+x²) ≠ 0"
    by (metis numeral_One power_one real_sqrt_eq_zero_cancel_iff
        sum_power2_eq_zero_iff zero_neq_numeral)
  then
  have "isCont (λx. x / sqrt(1 + x²)) x"
    using isCont_divide [where f="λx. x" and g="λx. sqrt(1+x²)"]
          continuous_ident denominator by blast
  moreover
  show "isNSCont (λx. x / sqrt(1 + x²)) x"
    by (simp add: calculation isCont_isNSCont)
qed
```

E.5.2 Bounds of p, or $\frac{x}{\sqrt{1+x^2}}$

```
lemma p_bounds: "-1 < x / sqrt(1 + x²) ∧x / sqrt(1 + x²) < 1"
  by (smt arsinh_real_aux divide_less_eq_1_pos divide_minus_left
    real_less_rsqrt)
```

Appendix F

Paragraph 204 - $\tan x$ v2

F.1 Helper lemmas

F.1.1 Bounds implies cosine non-zero

```
lemma cos_zero_pi_bounds: "x < pi/2 ∧x > -(pi/2) ⟹cos x ≠ 0"
  using cos_gt_zero_pi by fastforce
```

F.1.2 The differential of $\arctan(\tan x)$ is infinitely-close to dx

```
lemma differential_arctan_of_tan:
  assumes "dx ∈ Infinitesimal" "x < pi/2 ∧ x > -(pi/2)"
  shows
    "differential arctan (tan x) (differential tan x dx) ≈ dx"
proof -
  have
   "differential arctan (tan x) (differential tan x dx) ∈
      Infinitesimal"
    using differential_infinitesimal
    by (metis assms(1) assms(2) cos_gt_zero_pi isCont_arctan
       isCont_isNSCont isCont_tanless_irrefl)
  then
  show
   "differential arctan (tan x) (differential tan x dx) ≈ dx"
    using assms
    by (simp add: Infinitesimal_approx)
qed
```

F.1.3 Tan and cos identity in nonstandard analysis adapted

```
lemma STAR_tan_cos_squared2:
  "x < pi/2 ∧  x > -(pi/2) ⟹
```

```
    (1 + ((*f* tan)(star_of x))^2) = 1 / ((*f* cos)(star_of x))^2"
proof -
  assume "x < pi / 2 ∧  - (pi / 2) < x"
  then have "(*f* cos) (hypreal_of_real x) ≠ 0"
    using cos_gt_zero_pi by force
  then show ?thesis
    by (metis (no_types) STAR_tan_cos_squared)
qed
```

F.2 The first differential of $\tan x$

```
lemma differential_tan_2:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal" "x < pi/2 ∧  x > -(pi/2)"
  shows
    "differential tan x dx ≈    dx / ((*f* cos)(star_of x))^2"
proof -
  have tan_NSCont: "isNSCont (λx. tan x) x"
    using isCont_isNSCont assms(2) isCont_tan cos_zero_pi_bounds
    by blast
  have y: "arctan(tan x) = x"
    using arctan_tan cos_zero_pi_bounds assms
    by blast
  have inverse_is_1: "((*f* cos)(star_of x))^2 *
      inverse (((*f* cos)(star_of x))^2) = 1"
    using assms(2) cos_zero_pi_bounds by auto
  have "sqrt(1 + (tan x)^2) = 1 / cos x"
    using y by (metis cos_arctan div_by_1 inverse_divide)
  have inverse_is_finite:
      "inverse (((*f* cos)(star_of x))^2) ∈HFinite"
    using assms(2)
    by (simp add: Infinitesimal_inverse_HFinite power2_eq_square
       cos_zero_pi_bounds)
  have "differential arctan (tan x) (differential tan x dx) ≈
       (differential tan x dx) / (1 + ((*f* tan)(star_of x))^2)"
    using arctan_function_p tan_NSCont assms by blast
  also
  have "differential arctan (tan x) (differential tan x dx) ≈
       (differential tan x dx) * ((*f* cos)(star_of x))^2"
    using STAR_tan_cos_squared2 assms calculation by auto
  then
  have "dx ≈ (differential tan x dx) * ((*f* cos)(star_of x))^2"
    using differential_arctan_of_tan approx_trans3 assms(1)
       assms(2)
```

```
    by blast
  then
  have "dx * inverse (((*f* cos)(star_of x))²) ≈
      differential tan x dx * ((*f* cos)(star_of x))² *
      inverse (((*f* cos)(star_of x))²)"
    using inverse_is_finite approx_mult1
    by blast
  then
  have "dx * inverse (((*f* cos)(star_of x))²) ≈
      differential tan x dx"
    using inverse_is_1
    by (simp add: mult.assoc)
  then
  have "dx / (((*f* cos)(star_of x))²) ≈
       differential tan x dx"
    by (simp add: divide_inverse)
  then
  show "differential tan x dx ≈
      dx / (((*f* cos)(star_of x))²)"
    using approx_sym by blast
qed
```

Appendix G

Paragraph 205 - Higher differentials of $\sin x$

G.1 First Approach

G.1.1 Recursive definition of higher differentials

```
primrec n_dy ::
  "(hypreal ⇒ hypreal) ⇒ hypreal ⇒ hypreal ⇒ nat ⇒ hypreal"
  where
    "n_dy f x dx 0 = f x"
  | "n_dy f x dx (Suc n) = n_dy f (x + dx) dx n - n_dy f x dx n"
```

G.1.2 General forms for higher differentials of $\sin x$

G.1.2.1 The $\sin x$ differential coefficient

```
definition sin_differential_coeff
  where "sin_differential_coeff n = (-1)^((n) div 2)"
```

G.1.2.2 The $\sin x$ differential general form

```
definition sin_differential_n
  where "sin_differential_n n x dx = (sin_differential_coeff n)*dx^n
   *
    (if even n then (*f* sin)(x) else (*f* cos)(x))"
```

G.1.3 Proof by Induction

```
lemma n_dy_sin:
  fixes dx::hypreal and n::nat
  assumes "dx ∈ Infinitesimal" "x ∈HFinite"
  shows "n_dy (*f* sin) x dx n ≈    sin_differential_n n x dx"
proof (insert assms(2), induct n arbitrary: x)
  case 0
  then show ?case by (simp add: sin_differential_n_def
    sin_differential_coeff_def)
next
  fix n
  fix y::hypreal
  assume odd: "⋀x. x ∈HFinite ⟹n_dy (*f* sin) x dx n ≈
    sin_differential_n n x dx" "y ∈HFinite"
  then show "n_dy (*f* sin) y dx (Suc n) ≈
     sin_differential_n (Suc n) y dx"
  proof (auto simp add: sin_differential_n_def
      sin_differential_coeff_def)
    assume odd: "⋀x. x ∈HFinite ⟹  n_dy (*f* sin) x dx n ≈
         (- 1) ^ (n div 2) * dx ^ n * (*f* cos)(x)"
       "odd n"
       "y  ∈  HFinite"
    have cos_addition:
       "(*f* cos)(y)*(*f* cos)(dx) - (*f* sin)(y)*(*f* sin)(dx) ≈
           (*f* cos)(y) - (*f* sin)(y)*dx"
      using STAR_sin_infinitesimal STAR_cos_infinitesimal
        sine_HFinite cosine_HFinite
      by (metis approx_diff approx_mult2 assms(1) mult.right_neutral
        odd(3))
    have "y + dx ∈ HFinite"
      using HFinite_add Infinitesimal_subset_HFinite assms odd(3)
      by blast
    have odd_dx:
     "n_dy (*f* sin) (y + dx) dx n ≈(- 1) ^ (n div 2) * dxⁿ *
        (*f* cos)(y + dx)"
      using \<open>y + dx ∈HFinite\<close> odd
      by presburger
    then
    have
      "n_dy (*f* sin) (y + dx) dx n ≈(- 1) ^ (n div 2) * dxⁿ *
        ((*f* cos)(y)*(*f* cos)(dx) - (*f* sin)(y)*(*f* sin)(dx))"
      using STAR_cos_add
      by simp
    also
    have "... ≈ (- 1) ^ (n div 2) * dx ^ n
               * ((*f* cos)(y) - (*f* sin)(y)*dx)"
```

```
      using cos_addition approx_mult_subst assms(1)
        sine_coefficient_HFinite by blast
    also
    have "... ≈ (- 1) ^ (n div 2) * dx ^ n
                  * (*f* cos)(y) - ((- 1) ^ (n div 2) * dx ^ n
                  * (*f* sin)(y)*dx)"
      by (simp add: algebra_simps)
    then
    have "n_dy (*f* sin) (y + dx) dx n - n_dy (*f* sin) (y) dx n ≈
          ((- 1) ^ (n div 2) * dx ^ n
            * (*f* cos)(y)) - ((- 1) ^ (n div 2) * dx ^ n
            *(*f* sin)(y)*dx) - ((- 1) ^ (n div 2) * dx ^ n) *
              (*f* cos)(y)"
      using approx_diff approx_trans calculation odd(1) odd(3)
      by blast
    then
    show "n_dy (*f* sin) (y + dx) dx n - n_dy (*f* sin) (y) dx n ≈
            - ((- 1) ^ (n div 2) * (dx * dx ^ n) * (*f* sin)(y))"
      by (smt add_diff_cancel_left' diff_add_cancel minus_diff_eq
          mult.commute mult.left_commute)
next
    assume even:
      "⋀x. x ∈ HFinite ⟹  n_dy (*f* sin) x dx n ≈
        (- 1) ^ (n div 2) * dx^n * (*f* sin) (x) "
      "even n"
      "y ∈ HFinite"
    have sin_addition:
      "(*f* sin)(y)*(*f* cos)(dx) + (*f* cos)(y)*(*f* sin)(dx) ≈
         (*f* sin)(y) + (*f* cos)(y)*dx"
      using STAR_sin_Infinitesimal STAR_cos_Infinitesimal
        sine_HFinite cosine_HFinite
      by (metis approx_add approx_mult2 assms(1) mult.right_neutral
        odd(2))
    have "n_dy (*f* sin) (y + dx) dx n ≈
        (- 1) ^ (n div 2) * dx ^ n * (*f* sin) (y + dx)"
      using HFinite_add Infinitesimal_subset_HFinite
        assms(1) even(1) even(3) by blast
    also
    have "... ≈ (- 1) ^ (n div 2) * dx ^ n *
        ((*f* sin)(y)*(*f* cos)(dx) + (*f* cos)(y)*(*f* sin)(dx))"
      using STAR_sin_add by simp
    also
    have "... ≈ (- 1) ^ (n div 2) * dx ^ n *
        ((*f* sin)(y) + (*f* cos)(y)*dx) "
      using approx_mult2 assms(1) sin_addition
        sine_coefficient_HFinite by blast
```

```
    finally
    have "n_dy (*f* sin) (y + dx) dx n ≈
        (- 1) ^ (n div 2) * dx ^ n * (*f* sin)(y) +
        (- 1) ^ (n div 2) * dx ^ n * (*f* cos)(y)*(dx)"
      by (simp add: algebra_simps)
    then
    have "n_dy (*f* sin) (y + dx) dx n - n_dy (*f* sin) (y) dx n ≈
        (- 1) ^ (n div 2) * dx ^ n * (*f* sin)(y) +
        (- 1) ^ (n div 2) * dx ^ n * (*f* cos)(y)*(dx) -
        (- 1) ^ (n div 2) * dx ^ n * (*f* sin) (y)"
      using approx_diff even(1) even(3) by blast
    then
    show "n_dy (*f* sin) (y + dx) dx n - n_dy (*f* sin) y dx n ≈
        (- 1) ^ (n div 2) * (dx * dx ^ n) * (*f* cos) (y) "
      by (metis (no_types, lifting) add_diff_cancel_left'
        mult.commute mult.left_commute)
  qed
qed
```

G.1.4 Finiteness: $\sin x$ and $\cos x$ are finite

G.1.4.1 $\sin x$ is finite

```
lemma sine_HFinite:
  assumes "x ∈ HFinite"
  shows "((*f* sin)(x)::hypreal) ∈  HFinite"
proof -
  have "⋀x. (*f* sin)(x) ≤(1::hypreal) ∧
      (*f* sin)(x) ≥(-1::hypreal)"
    apply (transfer)
    by auto
  then
  show "((*f* sin)(x)::hypreal) ∈   HFinite"
    by (meson HFinite_1 HFinite_HInfinite_iff
      HInfinite_ge_HInfinite HInfinite_minus_iff linear
      minus_le_iff neg_0_le_iff_le)
qed
```

G.1.4.2 $\cos x$ is finite

```
lemma cosine_HFinite:
  fixes x::hypreal
  shows "((*f* cos)(x)::hypreal) ∈   HFinite"
proof -
  have "⋀x. (*f* cos)(x) ≤(1::hypreal) ∧
       (*f* cos)(x) ≥(-1::hypreal)"
    apply (transfer)
```

```
    by auto
  then
  show "((*f* cos)(x)::hypreal) ∈    HFinite"
    by (meson HFinite_1 HFinite_HInfinite_iff
       HInfinite_ge_HInfinite HInfinite_minus_iff linear
       minus_le_iff neg_0_le_iff_le)
qed
```

G.1.5 Finiteness: the coefficient term is finite

```
lemma sine_coefficient_HFinite:
  fixes dx::hypreal and n::nat
  assumes "dx ∈ Infinitesimal"
  shows "(- 1) ^ (n div 2) * dx ^ n ∈    HFinite"
proof -
  have "-1 ∈ HFinite"
    by (simp add: HFinite_minus_iff)
  then
  have "((- 1) ^ (n div 2)::hypreal) ∈   HFinite"
    using hrealpow_HFinite by blast
  also
  have "dx ^ n ∈ HFinite"
    by (metis hrealpow_HFinite approx_star_of_HFinite
      assms mem_infmal_iff star_of_simps(9))
  then
  show "(- 1) ^ (n div 2)* dx ^ n ∈     HFinite"
    using HFinite_mult calculation by blast
qed
```

G.2 Second Approach

G.2.1 Altered definition of the differential

```
definition hypreal_differential ::
      "(hypreal ⇒ hypreal) ⇒ hypreal ⇒ hypreal ⇒ hypreal"
  where "hypreal_differential f x dx = f(x + dx) - f x "
```

G.2.2 The hyperreal definition for the first differential of $\sin x$

```
lemma sin_differential_constant:
  fixes dx::hypreal and a::hypreal
  assumes "dx ∈ Infinitesimal" "a ∈HFinite"
```

```
  shows "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx ≈
     a*(*f* cos)(star_of x)*dx "
proof-
  have substitution: "a*((*f* sin)(star_of x)*(*f* cos)(dx) +
      (*f* sin)(dx)*(*f* cos)(star_of x)) -
       a*(*f* sin)(star_of x) ≈
         a*((*f* sin)(star_of x) + dx*(*f* cos)(star_of x)) -
         a*(*f* sin)(star_of x)"
    proof -
      have "(*f* sin)(star_of x) +
          dx*(*f* cos)(star_of x) ∈    HFinite"
        by (metis HFinite_add HFinite_mult HFinite_star_of
          Infinitesimal_subset_HFinite assms(1)
          basic_trans_rules(31) starfun_star_of)
      then
      have "(*f* sin)(star_of x)*(*f* cos)(dx) +
            (*f* sin)(dx)*(*f* cos)(star_of x) ≈
              (*f* sin)(star_of x) + dx*(*f* cos)(star_of x) "
        using STAR_sin_Infinitesimal STAR_cos_Infinitesimal
        by (metis approx_add approx_mult2 approx_star_of_HFinite
           assms(1) mult.commute mult.right_neutral starfun_approx)
      then
      show "a*((*f* sin)(star_of x)*(*f* cos)(dx) +
         (*f* sin)(dx)*(*f* cos)(star_of x)) -
        a*(*f* sin)(star_of x) ≈
           a*((*f* sin)(star_of x) + dx*(*f* cos)(star_of x)) -
           a*(*f* sin)(star_of x)"
        using assms approx_diff approx_mult2
        by blast
    qed
  have "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx =
      (λy. a*(*f* sin) y) (star_of x + dx) -
        (λy. a*(*f* sin) y) (star_of x)"
    using hypreal_differential_def
    by blast
  then
  have "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx =
       a*(*f* sin)(star_of x + dx) - a*(*f* sin)(star_of x)"
    by blast
  then
  have "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx =
       a*((*f* sin)(star_of x)*(*f* cos)(dx) +
       (*f* sin)(dx)*(*f* cos)(star_of x)) - a*(*f* sin)(star_of x)"
    by (simp add: STAR_sin_add)
  then
  have "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx ≈
```

```
      a*((*f* sin)(star_of x) +
      dx*(*f* cos)(star_of x)) - a*(*f* sin)(star_of x)"
    using substitution by auto
  then
  show "hypreal_differential (λy. a*(*f* sin) y) (star_of x) dx ≈
     a*(*f* cos)(star_of x)*dx"
    by (simp add: algebra_simps)
qed
```

G.2.3 The hyperreal definition for the first differential of $\cos x$

```
lemma cos_differential_constant:
  fixes dx::hypreal
  assumes "dx ∈ Infinitesimal" "a ∈HFinite"
  shows "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx ≈
     -a*(*f* sin)(star_of x)*dx "
proof -
  have substitution: "a*((*f* cos)(star_of x)*(*f* cos)(dx) -
       (*f* sin)(dx)*(*f* sin)(star_of x)) -
      a*(*f* cos)(star_of x) ≈
         a*((*f* cos)(star_of x) - dx*(*f* sin)(star_of x)) -
         a*(*f* cos)(star_of x)"
    proof -
      have "(*f* cos)(star_of x) -
           dx*(*f* sin)(star_of x) ∈    HFinite"
        by (metis HFinite_add HFinite_minus_iff HFinite_mult
          HFinite_star_of Infinitesimal_subset_HFinite
          assms(1) cosine_HFinite diff_conv_add_uminus
          sine_HFinite subset_eq)
      then
      have "(*f* cos)(star_of x)*(*f* cos)(dx) -
          (*f* sin)(dx)*(*f* sin)(star_of x) ≈
             (*f* cos)(star_of x) - dx*(*f* sin)(star_of x)"
        using STAR_sin_Infinitesimal STAR_cos_Infinitesimal
        by (metis approx_diff approx_mult2 approx_star_of_HFinite
           assms(1) mult.commute mult.right_neutral starfun_approx)
      then
      show "a*((*f* cos)(star_of x)*(*f* cos)(dx) -
           (*f* sin)(dx)*(*f* sin)(star_of x)) -
           a*(*f* cos)(star_of x) ≈
              a*((*f* cos)(star_of x) - dx*(*f* sin)(star_of x)) -
               a*(*f* cos)(star_of x)"
        using assms approx_diff approx_mult2
        by blast
    qed
```

```
  have "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx =
       (λy. a*(*f* cos) y) (star_of x + dx) -
        (λy. a*(*f* cos) y) (star_of x)"
    using hypreal_differential_def
    by blast
  then
  have "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx =
       a*(*f* cos)(star_of x + dx) - a*(*f* cos)(star_of x)"
    by blast
  then
  have "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx =
       a*((*f* cos)(star_of x)*(*f* cos)(dx) -
        (*f* sin)(dx)*(*f* sin)(star_of x)) -
         a*(*f* cos)(star_of x)"
    by (simp add: STAR_cos_add)
  then
  have "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx ≈
       a*((*f* cos)(star_of x) - dx*(*f* sin)(star_of x)) -
         a*(*f* cos)(star_of x)"
    using substitution
    by auto
  then
  show "hypreal_differential (λy. a*(*f* cos) y) (star_of x) dx ≈
       -a*(*f* sin)(star_of x)*dx"
    by (simp add: algebra_simps)
qed
```