Convex Functions in ACL2(r)

Carl Kwan Mark R. Greenstreet
Department of Computer Science
University of British Columbia*
Vancouver, Canada
{carlkwan, mrg}@cs.ubc.ca

This paper builds upon our prior formalisation of $\mathbb{R}^n$ in ACL2(r) by presenting a set of theorems for reasoning about convex functions. This is a demonstration of the higher-dimensional analytical reasoning possible in our metric space formalisation of $\mathbb{R}^n$. Among the introduced theorems is a set of equivalent conditions for convex functions with Lipschitz continuous gradients from Yurii Nesterov's classic text on convex optimisation. To the best of our knowledge a full proof of the theorem has yet to be published in a single piece of literature. We also explore "proof engineering" issues, such as how to state Nesterov's theorem in a manner that is both clear and useful.

1 Introduction

Convex optimisation is a branch of applied mathematics that finds widespread use in financial modelling, operations research, machine learning, and many other fields. Algorithms for convex optimisation often have many parameters that can be tuned to improve performance. However, a choice of parameter values that produces good performance on a set of test cases may suffer from poor convergence or non-convergence in other cases. Hand written proofs for convergence properties often include simplifying assumptions to make the reasoning tractable. This motivates using machine generated and/or verified proofs for the convergence and performance of these algorithms. Once an initial proof has been completed, the hope is that simplifying assumptions can be incrementally relaxed or removed to justify progressively more aggressive implementations. These observations motivate our exploration of convex functions within ACL2(r).

We present example proofs of continuity, Lipschitz continuity, and convexity for some simple functions as well as some basic theorems of convex optimisation. To the best of our knowledge, there are no other formalisations of convex functions in published literature (though we were able to find some formal theorems involving convex hulls in [5]). Moreover, we also provide a proof for a set of equivalent conditions for inclusion in the class of convex functions with Lipschitz continuous gradients – a theorem that, to the best of our knowledge, has yet to be fully published in a single piece of literature with a correct proof. This characterisation is based on Yurii Nesterov's classic work on convex optimisation [11] which has applications in the convergence proofs for many gradient descent algorithms.

This paper builds on the formalisation of $\mathbb{R}^n$ as an inner product space and as a metric space in [8]. Of particular note, we make use of the Cauchy-Schwarz inequality which was a demonstration of the algebraic reasoning capable in such a formalisation. This subsequent paper demonstrates the analytical reasoning about multivariate functions $\mathbb{R}^n \to \mathbb{R}$ that is enabled by the theorems proven in the books from the previous paper. In addition to presenting some key lemmas, we also discuss some of the challenges of formalising theorems with proofs that rely heavily on informally well-established and intuitive notions.

*This work is supported in part by the National Science and Engineering Research Council of Canada (NSERC) Discovery Grant program and the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC.

Shilpi Goel and Matt Kaufmann (Eds.):
ACL2 Theorem Prover and its Applications (ACL2-2018)
EPTCS 280, 2018, pp. 128–142, doi:10.4204/EPTCS.280.10

2 Preliminaries

This section summarizes the formalisation of vector spaces, inner-product spaces, metric spaces, and the Cauchy-Schwarz inequality that are presented in our previous paper [8]. The ACL2(r) formalisation is provided in the ACL2 books that accompany these papers. This work depends greatly on the original formalisation of the reals via non-standard analysis in ACL2(r) [4]. Further background on non-standard analysis can be found in [12, 1, 10]. Background on vector, inner product, and metric spaces can be found in [15, 13, 14, 9, 6]. We also outline some theorems involving convex functions. Standard texts on convex optimisation include [2, 11].

2.1 Inner Product & Metric Spaces

An inner product space is a vector space (V,F) equipped with an inner product $\langle -,- \rangle : V \to F$. The inner product satisfies

$$a\langle u,v\rangle = \langle au,v\rangle \tag{1}$$
$$\langle u+v,w\rangle = \langle u,w\rangle + \langle v,w\rangle \tag{2}$$
$$\langle u,v\rangle = \langle v,u\rangle \text{ when } F=\mathbb{R} \tag{3}$$
$$\langle u,u\rangle \geq 0 \text{ with equality iff } u=0 \tag{4}$$

for any $u,v,w \in V$ and $a \in F$ [13, Chapter 9].

An inner product induces a norm $\|\cdot\|$. If $\|\cdot\| = \sqrt{\langle -,-\rangle}$, then the Cauchy-Schwarz inequality [9, Chapter 15] holds for any $x,y \in V$:

$$|\langle x,y\rangle| \leq \|x\|\|y\|. \tag{5}$$

Norms induce metrics [14, Chapter 2] $d(x,y) = \|x-y\|$ which satisfy

$$d(x,y) = 0 \iff x = y \qquad \textit{(definitness)} \tag{6}$$
$$d(x,y) = d(y,x) \qquad \textit{(symmetry)} \tag{7}$$
$$d(x,y) \leq d(x,z) + d(z,y) \qquad \textit{(triangle inequality)}. \tag{8}$$

From this it follows that

$$d(x,y) \geq 0 \tag{9}$$

because

$$0 = d(x,x) \leq d(x,y) + d(y,x) = 2d(x,y).$$

A metric space is a pair (M,d) where M is a set and d is a metric on M [14, Chapter 2].

A function $f : M \to M'$ between metric spaces is continuous [14, Chapter 4] if for every $x \in M$ and for any $\varepsilon > 0$, there is a $\delta > 0$ such that

$$d_M(x,y) < \delta \implies d_{M'}(f(x),f(y)) < \varepsilon \tag{10}$$

for any $y \in E$.

Throughout this paper, we will consider $\mathbb{R}^n$ adjoined with the dot product and Euclidean metric to be an inner product and metric space, respectively.

2.2 Continuity & Differentiability

A univariate function $f : \mathbb{R} \to \mathbb{R}$ is continuous if for any $x \in \mathbb{R}$ and $\varepsilon > 0$, there is a $\delta > 0$ such that for any $y \in \mathbb{R}$, if $|x-y| < \delta$, then $|f(x) - f(y)| < \varepsilon$ [14, Chapter 4]. Moreover, the derivative of f is defined to be

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{11}$$

if such a form exists [14, Chapter 5].

For a multivariate function $f : \mathbb{R}^n \to \mathbb{R}$, continuity is defined similarly. We call f continuous if for any $x \in \mathbb{R}^n$ and $\varepsilon > 0$, there is a $\delta > 0$ such that for any $y \in \mathbb{R}^n$, if $\|x-y\| < \delta$, then $|f(x) - f(y)| < \varepsilon$ [14, Chapter 4]. Likewise, if it exists, there is a derivative for multivariate functions defined similarly to the univariate case:

$$\langle f'(x), h \rangle = \lim_{\|h\|_2 \to 0} \frac{f(x+h) - f(x)}{\|h\|_2}. \tag{12}$$

We call $f' : \mathbb{R}^n \to \mathbb{R}^n$ the *gradient* of f and it satisfies

$$f'(x) = (f_1'(x_1), f_2'(x_2), \ldots, f_n'(x_n)) \tag{13}$$

where $f_i'(x_i)$ is the univariate derivative of f with respect to the i-th component x_i of x [14, Chapter 9].

2.3 Convex Functions & $\mathscr{F}_L^1(\mathbb{R}^n)$

A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex [2, Chapter 3] if for any $x, y \in \mathbb{R}^n$ and $\alpha \in [0,1]$,

$$\alpha f(x) + (1-\alpha) f(y) \geq f(\alpha x + (1-\alpha) y). \tag{14}$$

Equivalently [11, Def. 2.1.1], if f is differentiable once with gradient f', then it is convex if

$$f(y) \geq f(x) + \langle f'(x), y - x \rangle. \tag{15}$$

Following Nesterov, we write $\mathscr{F}(\mathbb{R}^n)$ to denote the class of convex functions from $\mathbb{R}^n$ to $\mathbb{R}$. Examples of convex functions include $f(x) = x^2$, $\|\cdot\|_2$, and $\|\cdot\|_2^2$. Moreover, the class of convex functions is closed under certain operations [2, Chapter 3].

Theorem 1. If $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ are convex with h monotonically increasing, then

1. $a \cdot f$ is convex for any real $a \geq 0$,
2. $f + g$ is convex,
3. $h \circ f$ is convex.

Informal proofs of these claims follow from the definitions and can be found in [2, Chapter 3].

Often, convex optimisation algorithms require f to be both convex and sufficiently "smooth". Here, we take "smooth" to be stronger than continuous but not necessarily differentiable. In particular, we say that $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous if for any $x, y \in \mathbb{R}^n$ there is some $L > 0$ such that

$$\|f(x) - f(y)\| \leq L\|x - y\|. \tag{16}$$

Informally, we have the following chain of inclusions for classes of functions:

$$\text{Differentiable} \subset \text{Lipschitz Continuous} \subset \text{Continuous}.$$

We write $\mathscr{F}_L^1(\mathbb{R}^n)$ for the class of convex differentiable functions on $\mathbb{R}^n$ with Lipschitz continuous gradient with constant L. Functions in the class $\mathscr{F}_L^1(\mathbb{R}^n)$ have many useful properties for optimisation. The main result of this paper is proving a theorem from [11, Thm. 2.1.5] that gives six "equivalent" ways of showing that a convex function is in $\mathscr{F}_L^1(\mathbb{R}^n)$:

Theorem 2 (Nesterov)**.** Let $f \in \mathscr{F}^1(\mathbb{R}^n)$, $x, y \in \mathbb{R}^n$ and $\alpha \in [0,1]$. The following conditions are equivalent to $f \in \mathscr{F}_L^1(\mathbb{R}^n)$:

$$f(y) \leq f(x) + \langle f'(x), y - x\rangle + \frac{L}{2}\|x-y\|^2 \tag{Nest. 1}$$

$$f(x) + \langle f'(x), y - x\rangle + \frac{1}{2L}\|f'(x) - f'(y)\|^2 \leq f(y) \tag{Nest. 2}$$

$$\frac{1}{L}\|f'(x) - f'(y)\|^2 \leq \langle f'(x) - f'(y), x - y\rangle \tag{Nest. 3}$$

$$\langle f'(x) - f'(y), x - y\rangle \leq L\|x-y\|^2 \tag{Nest. 4}$$

$$f(\alpha x + (1-\alpha)y) + \frac{\alpha(1-\alpha)}{2L}\|f'(x) - f'(y)\|^2 \leq \alpha f(x) + (1-\alpha)f(y) \tag{Nest. 5}$$

$$\alpha f(x) + (1-\alpha)f(y) \leq f(\alpha x + (1-\alpha)y) + \alpha(1-\alpha)\frac{L}{2}\|x-y\|^2. \tag{Nest. 6}$$

There are several motivations to prove Thm. 2 in ACL2(r). First, such a formalisation provides an unambiguous statement of the theorem. For example, the theorem requires the assumption $f \in \mathscr{F}^n$, but this hypothesis is not explicitly stated in the theorem statement in [11]. Instead, the assumption is implicit in the preceding text. On the other hand, Nest. 5 implies Eq. 14 and therefore that f is convex. Inequalities Nest. 2 through Nest. 6 implicitly have an existential quantification of L. By stating and proving the theorem in ACL2(r), these ambiguities are avoided. Furthermore, this enables the use of Nesterov's theorem for further reasoning about convex functions and optimisation algorithms.

2.4 ACL2(r) & Non-standard Analysis

The usual axioms for $\mathbb{R}^n$ as an inner product and metric space were formalised in ACL2(r) in [8] along with theorems proving their salient properties. Reals and infinitesimals are recognized by `realp` and `i-small`, respectively. Two reals are `i-close` if their difference is `i-small`. Vectors are recognized by `real-listp`. Vector addition and subtraction are `(vec-+ x y)` and `(vec-- x y)`, respectively. Scalar multiplication is `(scalar-* a x)`. The dot product is simply `(dot x y)`. The Euclidean norm and metric are `(eu-norm x)` and `(eu-metric x y)`, respectively. Sometimes it is easier to reason about the square of the norm or metric. These are called `(norm^2 x)` and `(metric^2 x y)`, respectively. More details can be found in [8].

In non-standard analysis, continuity for a function $f : \mathbb{R}^n \to \mathbb{R}$ amounts to if $\|x-y\|$ is an infinitesimal for some standard x, then so is $|f(x) - f(y)|$. The derivative of a function $f : \mathbb{R} \to \mathbb{R}$ is the standard part of $\frac{f(x+h)-f(x)}{h}$ where h is an infinitesimal. The formalisation for continuity, differentiability, integrability, and the Fundamental Theorem of Calculus already exist for univariate functions in ACL2(r). [3, 7]

3 Convexity in ACL2(r)

In this section, we provide some selected examples of formalised theorems involving convex functions. The formalised proofs follow almost directly from those of the informal proofs. For the sake of exposition, the proof for the first theorem is outlined but the rest are omitted.

The first is a simple theorem positing the convexity of $f(x) = x^2$ which is true because for $0 \leq \alpha \leq 1$,

$$\alpha x^2 + (1-\alpha)y^2 - (\alpha x + (1-\alpha)y)^2 = \alpha(1-\alpha)(x^2 - 2xy + y^2) = \alpha(1-\alpha)(x-y)^2 \geq 0. \tag{17}$$

We first define a square function: `(defun square-fn (x) (* (realfix x) (realfix x)))`. The chain of equalities in Eq. 17 is immediately recognized by ACL2(r). The inequality in Eq. 17 also passes without issue. The convexity of `square-fn` then follows from a simple application of the two lemmas.

Program 3.1 The equality in Eq. 17 formalised.

```
;; ax^2 + (1-a)y^2 - (ax + (1-a)y)^2 = a(1-a)(x-y)^2
(defthm lemma-1
 (implies (and (realp x) (realp y) (realp a) (<= 0 a) (<= a 1))
          (equal (- (+ (* a (square-fn x)) (* (- 1 a) (square-fn y)))
                    (square-fn (+ (* a x) (* (- 1 a) y))))
                 (* a (- 1 a) (square-fn (- x y))))))
```

Program 3.2 A more general version of the inequality in Eq. 17 formalised.

```
;; replace a with a(1-a) and x with x-y to obtain the desired inequality
(defthm lemma-2
 (implies (and (realp a) (<= 0 a))
          (<= 0 (* a (square-fn x)))))
```

Program 3.3 The square function is convex.

```
(defthm square-fn-is-convex
 (implies (and (realp x) (realp y) (realp a) (<= 0 a) (<= a 1))
          (<= (square-fn (+ (* a x) (* (- 1 a) y)))
              (+ (* a (square-fn x)) (* (- 1 a) (square-fn y)))))
 :hints (("GOAL" :use ((:instance lemma-2 (a (* a (- 1 a))) (x (- x y)))))))
```

Also formalised is a proof of each of the statements in Thm. 1. Here we outline the proof of the convexity of $a \cdot f$ given that $a \geq 0$ and f is convex. The rest are similar. Moreover, the approach we take resembles our approach to formalising Thm. 2 albeit much simpler. In particular, to reason about functions, we use the technique of encapsulation to first prove the desired theorem for a witness function. Functional instantiation then provides a method for reasoning about functions in general. Our witness, `cvfn-1`, is a constant function and so is clearly convex. This can be seen in Prog. 3.4.

Explicitly, $a \cdot f$ is convex because

$$af(\alpha x + (1-\alpha)y) \leq a(\alpha f(x) + (1-\alpha)f(y)) = \alpha(af(x)) + (1-\alpha)(af(y)). \tag{18}$$

In particular, we invoke convexity and need to distribute a. Moreover, this line of reasoning is not dependent on the definition of f so we may disable the definition of `cvfn-1` in Prog. 3.4. This can be seen in Prog. 3.5.

Program 3.4 Encapsulating a constant function `cvnf-1` and stating its convexity.

```
(encapsulate
 (((cvfn-1 *) => *)...)

 (local (defun cvfn-1 (x) (declare (ignore x)) 1337))
 ...
 (defthm cvfn-1-convex
  (implies (and (real-listp x) (real-listp y) (= (len y) (len x))
                (realp a) (<= 0 a) (<= a 1))
           (<= (cvfn-1 (vec-+ (scalar-* a x) (scalar-* (- 1 a) y)))
               (+ (* a (cvfn-1 x)) (* (- 1 a) (cvfn-1 y)))))) ...)
```

Program 3.5 Supressing the definition of `cvfn-1`, stating a special case of distributivity, and stating the convexity of $a \cdot f$.

```
(local (in-theory (disable (:d cvfn-1) (:e cvfn-1) (:t cvfn-1))))

(encapsulate ...
  ;;  factor out alpha
  (local (defthm lemma-1
   (implies (and (real-listp x) (real-listp y) (= (len y) (len x))
                 (realp a) (<= 0 a) (<= a 1)
                 (realp alpha) (<= 0 alpha))
            (= (+ (* a (* alpha (cvfn-1 x)))
                  (* (- 1 a) (* alpha (cvfn-1 y))))
               (* alpha
                  (+ (* a (cvfn-1 x))
                     (* (- 1 a) (cvfn-1 y))))))))

  (defthm a-*-cvfn-1-convex
   (implies (and (real-listp x) (real-listp y) (= (len y) (len x))
                 (realp a) (<= 0 a) (<= a 1)
                 (realp alpha) (<= 0 alpha))
            (<= (* alpha (cvfn-1 (vec-+ (scalar-* a x) (scalar-* (- 1 a) y))))
                (+ (* a (* alpha (cvfn-1 x)))
                   (* (- 1 a) (* alpha (cvfn-1 y))))))
   :hints (("GOAL" :in-theory (disable distributivity)
                   :use ((:instance cvfn-1-convex)
                         (:instance lemma-1))))))
```

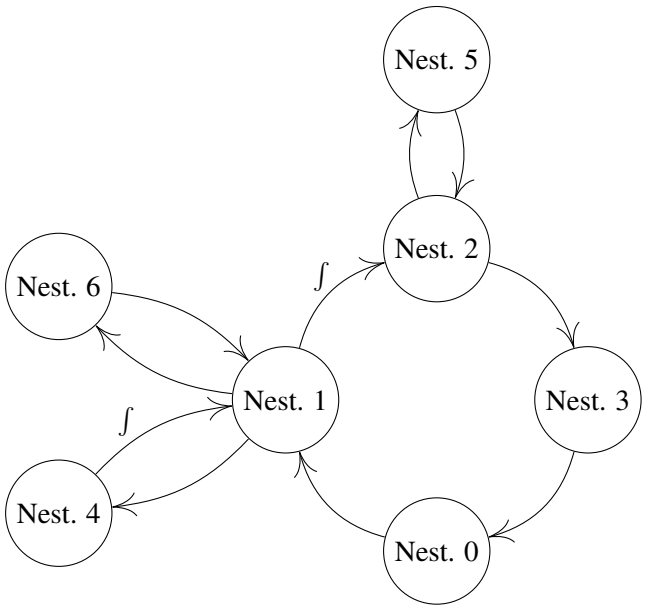

Figure 1: Nesterov's proof of Thm. 2. Here Nest. 0 is Lipschitz continuity. Integration is denoted by $\int$.

We omit the formal proofs for the other claims of Thm. 1 as well as the proofs of convexity for the Euclidean norm $\|\cdot\|_2$ and its square $\|\cdot\|_2^2$ and the inequality [1]

$$\alpha(1-\alpha)\|x-y\|^2 \leq \alpha\|x\|^2 + (1-\alpha)\|y\|^2. \tag{19}$$

4 Nesterov's Theorem in ACL2(r)

4.1 Approach

In [11], Nesterov provided a proof that followed the structure visualised by Fig. 1. Nesterov's proof, however, uses techniques that are not amenable to proofs in ACL2(r). In particular, integration is used multiple times to show some inequalities. However, integration in ACL2(r) is dependent on the function that is being integrated [7]. This places extra obligations on the user. The alternate approach shown in Fig. 2 requires fewer instances of integration than Fig. 1. Moreover, Fig. 2 has fewer implications to prove in general. The primary difference in our approach is that we prove Nest. 4 from a straightforward application of Cauchy-Schwarz and omit Nest. 1 implies Nest. 4.

Stating a theorem about functions in ACL2 is an unnatural endeavour because ACL2 is a theorem prover for first-order logic so we cannot predicate over sets in general. The natural solution is to leverage encapsulation and functional instantiation to obtain pseudo-higher order behaviour. However, this means that the desired function for which the user wishes to apply Thm. 2 must pass the theorems within the encapsulation. To formalise the theorem in its greatest generality, it is necessary to suppress the definition of the witness function in the encapsulation and instead prove the theorems based on the properties of the function. To use functional instantiation, these properties must be proven for the desired function. Thus, we aim to minimize the number of properties that the user must show, and derive as much as possible

[1] These proofs can be found in `convex.lisp`.

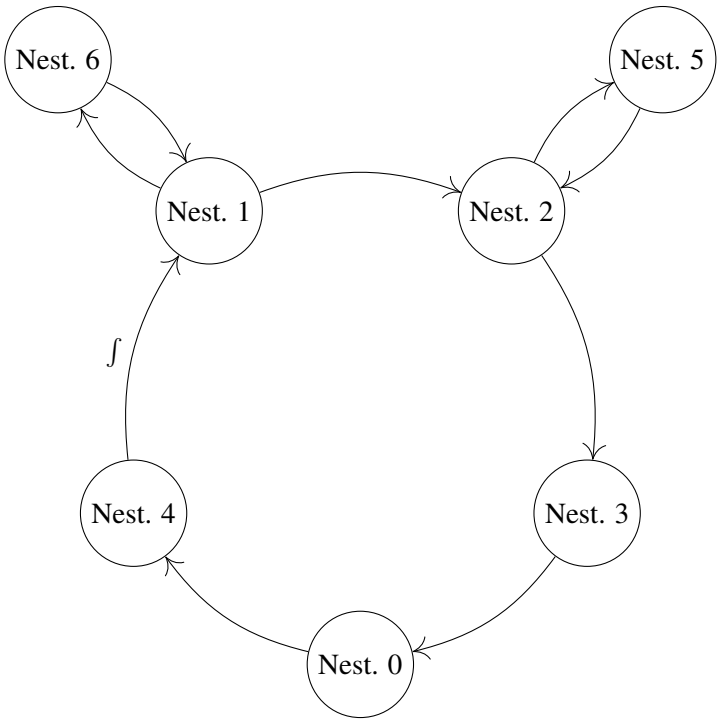

Figure 2: Another proof of Thm. 2. Here Nest. 0 is Lipschitz continuity. Integration is denoted by $\int$.

within the encapsulation. In our case, the user obligations are the theorems and identities involving the continuity, derivative, and integral of the encapsulated functions as well as any forms explicitly involving the dimension of the space.

The encapsulated functions include the multivariate function of interest `mvfn` and its derivative `nabla-mvfn`, the function that evaluates to the Lipschitz constant `L`, a helper function `phi` based on `mvfn` and its derivative `nabla-phi`, the recognizer for vectors with standard real entries `standard-vecp`, and the function, `DIM`, that evaluates to the dimension n of the vector space.

The fundamental challenges of formalising $\mathbb{R}^n$ are explored in [8]. In the context of this paper, however, the particular difficulties to note are the recognizer for vectors with standard entries, exhibiting the relationship between vectors with infinitesimal entries and the norm, and invoking two copies of certain inequalities. More details will be discussed as we proceed.

4.2 Basic Definitions & Lemmas

As an example of the formalisation, the definition of continuity for multivariate functions can be seen in Prog. 4.1. The hypotheses ensure the entries are both real vectors of the same dimension and `i-small` is the recognizer for infinitesimals. This form is consistent with the non-standard analysis definition of continuity.

The prohibition of non-classical recursive functions and the necessity of a recognizer for vectors with standard entries forces the recognizer to be defined with a specific n within the encapsulation. An encapsulation requires nevertheless a witness for the convex function of interest (which in this case happens to be the function $f(x) = 42$) so we set $n = 2$. The recognizer `standard-vecp` simply checks

Program 4.1 Continuity for a multivariate function `mvfn`.

```
(defthm mvfn-is-continuous-3
 (implies (and (real-listp vec1) (real-listp vec2)
               (= (len vec2) (len vec1)) (= (len vec1) (DIM))
               (i-small (eu-metric vec1 vec2)))
          (i-small (abs (- (mvfn vec1) (mvfn vec2))))))
```

whether a vector of dimension 2 has standards in both its coordinates.[2]

Another recurring and particularly useful theorem is the Cauchy-Schwarz inequality. It was formalised in [8] and has applications in proving the triangle inequality and the properties of $\mathbb{R}^n$ as a metric space. We use a specific version of the inequality that can be seen in Prog. 4.2.

Program 4.2 A version of the Cauchy-Schwarz inequality used in our proof of Thm. 2.

```
(defthm cauchy-schwarz-2
 (implies (and (real-listp u) (real-listp v) (= (len u) (len v)))
          (<= (abs (dot u v))
              (* (eu-norm u) (eu-norm v))))...)
```

There are several proofs of implications in Fig. 2 that follow almost immediately from the mentioned definitions and lemmas. We outline one of them: Nest. 0 implies Nest. 4. The proof mimics the chain of inequalities

$$L\|x-y\|^2 \geq \|f'(x)-f'(y)\| \cdot \|x-y\| \geq \langle f'(x)-f'(y), x-y\rangle \tag{20}$$

where the first inequality follows from Lipschitz continuity and the second inequality follows from Cauchy-Schwarz. We begin with the first inequality of Eq. 20 in Prog. 4.3. Applying the Cauchy-

Program 4.3 The first inequality follows from Lipschitz continuity.

```
;; ||f'(x) - f'(y)|| <= L||x - y|| implies
;; ||f'(x) - f'(y)|| ||x - y|| <= L||x - y||^2
(local (defthm lemma-2
 (implies (and (real-listp x) (real-listp y)
               (= (len y) (len x)) (= (len x) (DIM))
               (<= (eu-norm (vec-- (nabla-mvfn x) (nabla-mvfn y)))
                   (* (L) (eu-norm (vec-- x y)))))
          (<= (* (eu-norm (vec-- (nabla-mvfn x) (nabla-mvfn y)))
                 (eu-norm (vec-- x y)))
              (* (L) (eu-norm (vec-- x y))
                     (eu-norm (vec-- x y)))))...))
```

Schwarz inequality from Prog. 4.2 gives the second inequality of Eq. 20 under absolute values as seen in Prog. 4.4. Eliminating absolute values gives the desired inequality (in an "expanded" form) as seen in Prog. 4.5. To state the implication in its full generality and for reasons to appear in the next section, we

[2]The rest of the basic definitions and theorems can be found in `nesterov-1.lisp`.

Program 4.4 The second inequality follows from Cauchy-Schwarz.

```
;; |<f'(x) - f'(y), x - y>| <= L||x - y||^2
(local (defthm lemma-3
 (implies (and (real-listp x) (real-listp y)
               (= (len y) (len x)) (= (len x) (DIM))
               (<= (eu-norm (vec-- (nabla-mvfn x) (nabla-mvfn y)))
                   (* (L) (eu-norm (vec-- x y)))))
          (<= (abs (dot (vec-- (nabla-mvfn x) (nabla-mvfn y))
                        (vec-- x y)))
              (* (L) (eu-norm (vec-- x y))
                     (eu-norm (vec-- x y)))))
 :hints (("GOAL" :use ((:instance lemma-2)
                       ...
                       (:instance cauchy-schwarz-2
                                  (u (vec-- (nabla-mvfn x)
                                            (nabla-mvfn y)))
                                  (v (vec-- x y))))))))
```

Program 4.5 The desired implication in an expanded form.

```
;; <f'(x) - f'(y), x - y> <= L ||x - y||^2
(defthm ineq-0-implies-ineq-4-expanded
 (implies (and (real-listp x) (real-listp y)
               (= (len y) (len x)) (= (len x) (DIM))
               (<= (eu-metric (nabla-mvfn x) (nabla-mvfn y))
                   (* (L) (eu-metric x y))))
          (<= (dot (vec-- (nabla-mvfn x) (nabla-mvfn y)) (vec-- x y))
              (* (L) (metric^2 x y))))
 :hints (("GOAL" :use (...(:instance lemma-3)))))
```

use Skolem functions to replace the inequalities in the theorem. The function definitions can be seen in Prog. 4.6. The theorem then becomes of the form seen in Prog. 4.7 where the hypotheses ensure that we are dealing with real vectors. All the implications, whether they follow straight from the definitions or otherwise, are of this form.[3]

The other implications that follow mainly from the definitions are Nest. 4 implies Nest. 1 and Nest. 1 implies Nest. 2.

Program 4.6 Skolem function definitions that allow us to invoke the `forall` quantifier.

```
;; Lipschitz continuity ||f'(x) - f'(y)|| <= L ||x - y||
(defun-sk ineq-0 (L)
 (forall (x y)
  (<= (eu-metric (nabla-mvfn x) (nabla-mvfn y))
      (* L (eu-metric x y))))...)
...
;; <f'(x) - f'(y), x - y> <= L ||x - y||^2
(defun-sk ineq-4 (L)
 (forall (x y)
  (<= (dot (vec-- (nabla-mvfn x) (nabla-mvfn y)) (vec-- x y))
      (* L (metric^2 x y))))...)
```

Program 4.7 The desired implication.

```
(defthm ineq-0-implies-ineq-4
  (implies (and (hypotheses (ineq-4-witness (L)) (DIM))
                (ineq-0 (L)))
           (ineq-4 (L)))...)
```

4.3 Challenging Issues

Here we outline some of the challenges we encountered during our formalisation of Thm. 2. Several of these issues involve the proofs of the remaining lemmas, which all require some user intervention beyond simple algebraic manipulation. Here we discuss two such instances. The others are omitted because we solve them similarly. Finally, we discuss the final form of Thm. 2 and the various considerations regarding it and alternative approaches.

4.3.1 Instantiating Inequalities

The proof of Nest. 2 implies Nest. 3 amounts to adding two copies of Nest. 2 with x, y swapped. This induces issues regarding the proof of the implication. The natural form of the lemma would involve Nest. 2 among the hypotheses as in Prog. 4.8.

However, to instantiate a copy of Nest. 2 with swapped x, y in such a form would be equivalent to

$$\forall x, y, (P(x,y) \implies P(y,x)) \tag{21}$$

[3]The rest of the Skolem functions can be found in `nesterov-4.lisp`

Program 4.8 An "obvious" way to state Nest. 2 implies Nest. 3.

```
(defthm ineq-2-implies-ineq-3
  (implies (and (real-listp x) (real-listp y)
                (= (len y) (len x)) (= (len x) (DIM))
                (ineq-2 (L)))
           (ineq-3 (L)))...)
```

where P is a predicate (in this case equivalent to Nest. 2), which is not necessarily true. The form we wish to have is

$$(\forall x, y, P(x,y)) \implies (\forall x, y, P(y,x)). \tag{22}$$

In order to instantiate another copy of Nest. 2 within the implication requires quantifiers within the theorem statement. The usual approach involves using Skolem functions to introduce quantified variables. We can now instantiate the two copies with swapped variables as in Prog. 4.9.

Program 4.9 Instantiating two copies of Nest. 2 with swapped variables.

```
(defthm ineq-2-expanded-v1
  (implies (ineq-2 (L))
           (and (<= (+ (mvfn x)
                       (dot (nabla-mvfn x) (vec-- y x))
                       (* (/ (* 2 (L)))
                          (metric^2 (nabla-mvfn x) (nabla-mvfn y))))
                    (mvfn y))
                (<= (+ (mvfn y)
                       (dot (nabla-mvfn y) (vec-- x y))
                       (* (/ (* 2 (L)))
                          (metric^2 (nabla-mvfn y) (nabla-mvfn x))))
                    (mvfn x))))...)
```

We also considered simply including two copies of the inequality with swapped variables among the hypotheses. This has two advantages. Firstly, with such a form, the lemma becomes stronger because the hypothesis $P(x,y) \wedge P(y,x)$ is weaker than $\forall x, y, P(x,y)$. Secondly, the lemma is slightly easier to pass in ACL2(r). However, the primary drawback is that this form is inconsistent with the other lemmas and the final form of Nesterov's theorem becomes less elegant (eg. showing Nest. 1 implies Nest. 2 would also require two copies of Nest. 1).

The lemmas Nest. 1 implies Nest. 6 and Nest. 2 implies Nest. 5 also requires instantiating multiple copies of the antecedent inequalities (albeit with different vectors).

4.3.2 Taking Limits

In the language of non-standard analysis, limits amount to taking standard-parts. For example, $\lim_{x \to a} f(x)$ is equivalent to $\mathrm{st}(f(x))$ when $x - a$ is an infinitesimal. However, for products, say, xy, the identity $\mathrm{st}(xy) = \mathrm{st}(x)\,\mathrm{st}(y)$ only holds when x, y are both finite reals. In the proof of Nest. 6 implies Nest. 1, there is a step that requires taking the limit of $(1-\alpha)\|y-x\|^2$ as $\alpha \to 0$. Now, if $\alpha > 0$ is an infinitesimal,

$$\mathrm{st}((1-\alpha)\|y-x\|^2) = \mathrm{st}(1-\alpha)\,\mathrm{st}(\|y-x\|^2) = \|y-x\|^2 \tag{23}$$

is easy to satisfy when x, y are vectors with standard real components. Moreover, requiring variables to be standard is consistent with some instances of single variable theorems (eg. the product of continuous functions is continuous). It then remains to state such a hypothesis using, say, a recognizer `standard-vecp` as in Prog. 4.10. The natural approach to defining `standard-vecp` would be to simply

Program 4.10 Introducing `standard-vecp` into the hypotheses.

```
(defthm ineq-6-implies-ineq-1-expanded
 (implies (and (real-listp x) (real-listp y)
               (= (len y) (len x)) (= (len x) (DIM))
               (realp alpha) (i-small alpha)
               (< 0 alpha) (<= alpha 1)
               (standard-vecp x) (standard-vecp y)
               (<= (+ (* alpha (mvfn y))
                      (* (- 1 alpha) (mvfn x)))
                   (+ (mvfn (vec-+ (scalar-* alpha y)
                                   (scalar-* (- 1 alpha) x)))
                      (* (/ (L) 2) alpha (- 1 alpha) (metric^2 y x)))))
          (<= (mvfn y)
              (+ (mvfn x)
                 (dot (nabla-mvfn x) (vec-- y x))
                 (* (/ (L) 2) (metric^2 y x))))))...)
```

recurse on the length of a vector applying `standardp` to each entry. However, `standardp` is non-classical and this definition encounters a common issue throughout our ACL2(r) formalisation in that it is a non-classical recursive function. We discuss the subtleties of this problem more in [8]. Because `standard-vecp` is dependent on the dimension, our solution is to encapsulate the function and prove the necessary theorems involving it (eg. `metric^2` is `standardp` on `standard-vecp` values). For the case $n = 2$, we simply check the length of the vector is two and that both entries are standard reals. A final note regarding the lemma is the hypothesis $\alpha > 0$ replacing the weaker $\alpha \geq 0$ since part of the proof depends on dividing by α.

The other lemma that requires similar hypotheses is the implication Nest. 5 implies Nest. 2.

4.3.3 Final Form of Nesterov's Theorem

Finally, we discuss our final form of Thm. 2 as well as several alternatives and their considerations. The final form can be seen in Prog. 4.11. The function `hypotheses` ensure that we are dealing with real vectors of the correct dimension. The function `st-hypotheses` ensure that the vectors have standard entries due to the necessity of taking limits. The function `alpha-hypotheses` is the same as `hypotheses` but includes the hypothesis that $\alpha \in [0,1]$. The function `alpha->-0-hypotheses` ensures that $\alpha > 0$ for the case of taking limits.

In Sec. 4.2, we already mentioned, for each lemma, the basic structure involving Skolem functions. In Sec. 4.3.1, we cited elegance and ease of instantiation as reasons for using Skolem functions. Because stating even the shorter inequalities in Polish notation would quickly become awkward and unintelligible (eg. Prog. 4.12), it became desirable for us to define the inequalities in a clean and clear manner. One simple approach would be to define the inequalities as ACL2 functions or macros. Unfortunately, during the course of our formalisation, we found that the rewriter would be tempted to "simplify" or otherwise

Program 4.11 The final statement of Thm. 2.

```
(defthm nesterov
 ;; theorem statement
 (implies (and (hypotheses (ineq-0-witness (L)) (DIM))
                   (hypotheses (ineq-1-witness (L)) (DIM))
                   (hypotheses (ineq-2-witness (L)) (DIM))
                   (hypotheses (ineq-3-witness (L)) (DIM))
                   (hypotheses (ineq-4-witness (L)) (DIM))
                   (st-hypotheses (ineq-1-witness (L)))
                   (st-hypotheses (ineq-2-witness (L)))
                   (alpha-hypotheses (ineq-5-witness (L)) (DIM))
                   (alpha-hypotheses (ineq-6-witness (L)) (DIM))
                   (alpha->-0-hypotheses (ineq-5 (L)))
                   (alpha->-0-hypotheses (ineq-6 (L)))
                   (or (ineq-0 (L)) (ineq-1 (L)) (ineq-2 (L)) (ineq-3 (L))
                       (ineq-4 (L)) (ineq-5 (L)) (ineq-6 (L))))
          (and (ineq-0 (L)) (ineq-1 (L)) (ineq-2 (L)) (ineq-3 (L))
               (ineq-4 (L)) (ineq-5 (L)) (ineq-6 (L))))
 ;; hints, etc. for ACL2(r)
 ...)
```

change the form of the inequality via arithmetic rules. This made applications of certain theorems more involved and arduous than necessary. Therefore, it would be necessary to disable the function definitions anyways. In addition to permitting instantiations of inequalities with different vectors within a single theorem statement, Skolem functions would allow us to suppress or "hide" the explicit inequality thus providing a clear, concise, and compact package.

Program 4.12 Nest. 5 in Polish notation.

```
(<= (+ (mvfn (vec-+ (scalar-* alpha y) (scalar-* (- 1 alpha) x)))
       (* (/ (* 2 (L))) (* alpha (- 1 alpha) (metric^2 (nabla-mvfn y)
                                                       (nabla-mvfn x)))))
    (+ (* alpha (mvfn y)) (* (- 1 alpha) (mvfn x))))
```

On the other hand, this form has the unfortunate drawback of making the proof of the theorem slightly more involved. By introducing Skolem functions we also introduce the necessity of witness functions; proving the lemmas in terms of the witness functions may occasionally become onerous. For example, to state the hypotheses that the entries of the witness functions are real vectors of the same dimension, we would like to define a `hypotheses` function. However, explicitly exhibiting the witness functions within the definition of `hypotheses` leads to a signature mismatch. We instead pass the witness function to `hypotheses` as an argument.

5 Conclusion

In this paper, we presented a set of theorems for reasoning about convex functions in ACL2(r). We also discussed some of the challenges of formalising a proof that relies heavily on informally well-established

and intuitive notions. Examples include translating statements in classical multivariate analysis into non-standard analysis and instantiating quantified statements using Skolem functions. Our particular interest in this work are the potential applications to verifying, among other areas, optimization algorithms used in machine learning. To this end, we chose a theorem of Nesterov's to serve as an example of the analytical reasoning possible in our formalisation. The natural next step would be to develop a proper theory of multivariate calculus to further automate the reasoning about optimisation algorithms.

References

[1] Leif O. Arkeryd, Nigel J. Cutland & C. Ward Henson (Eds.) (1997): *Nonstandard Analysis: Theory and Applications*, 1st edition. *Nato Science Series C: 493* , Springer Netherlands, doi:10.1007/978-94-011-5544-1.

[2] Stephen Boyd & Lieven Vandenberghe (2004): *Convex Optimization*. Cambridge University Press, doi:10.1017/CBO9780511804441.

[3] Ruben Gamboa (2000): *Continuity and Differentiability*. In Matt Kaufmann, Panagiotis Manolios & J. Strother Moore, editors: *Computer-Aided Reasoning: ACL2 Case Studies*, Springer US, Boston, MA, pp. 301–315, doi:10.1007/978-1-4757-3188-0_18.

[4] Ruben A. Gamboa & Matt Kaufmann (2001): *Nonstandard Analysis in ACL2*. *Journal of Automated Reasoning* 27(4), pp. 323–351, doi:10.1023/A:1011908113514.

[5] John Harrison (2007): *Formalizing Basic Complex Analysis*. In R. Matuszewski & A. Zalewska, editors: *From Insight to Proof: Festschrift in Honour of Andrzej Trybulec*, *Studies in Logic, Grammar and Rhetoric* 10(23), University of Białystok, pp. 151–165. Available at `http://mizar.org/trybulec65/`.

[6] Nathan Jacobson (1985): *Basic Algebra I*, 2nd edition. Dover Publications.

[7] Matt Kaufmann (2000): *Modular Proof: The Fundamental Theorem of Calculus*. In Matt Kaufmann, Panagiotis Manolios & J. Strother Moore, editors: *Computer-Aided Reasoning: ACL2 Case Studies*, Springer US, Boston, MA, pp. 75–91, doi:10.1007/978-1-4757-3188-0_6.

[8] Carl Kwan & Mark R. Greenstreet (2018): *Real Vector Spaces and the Cauchy-Schwarz Inequality in ACL2(r)*. In: Proceedings 15th International Workshop on the *ACL2 Theorem Prover and its Applications*, Austin, Texas, USA, November 5-6, 2018, *Electronic Proceedings in Theoretical Computer Science* 280, Open Publishing Association, pp. 111–127, doi:10.4204/EPTCS.280.9.

[9] Serge Lang (2002): *Algebra*, 3rd edition. *Graduate Texts in Mathematics 211* , Springer-Verlag New York, doi:10.1007/978-1-4613-0041-0.

[10] Peter A. Loeb & Manfred P. H. Wolff (2015): *Nonstandard Analysis for the Working Mathematician*, 2nd edition. Springer Netherlands, doi:10.1007/978-94-017-7327-0.

[11] Yurii Nesterov (2004): *Introductory Lectures on Convex Optimization*, 1st edition. *Applied Optimization 87* , Springer US, doi:10.1007/978-1-4419-8853-9.

[12] Abraham Robinson (1966): *Non-Standard Analysis*. North-Holland Publishing Company.

[13] Steven Roman (2008): *Advanced Linear Algebra*, 3rd edition. *Graduate Texts in Mathematics 135* , Springer-Verlag New York, doi:10.1007/978-0-387-72831-5.

[14] Walter Rudin (1976): *Principles of Mathematical Analysis*, 3rd edition. *International Series in Pure and Applied Mathematics* , McGraw-Hill.

[15] Georgi E. Shilov (1977): *Linear Algebra*. Dover Publications.