



Efficient calculation of regular simplex gradients

Ian Coope¹ · Rachael Tappenden¹

Received: 4 October 2017 / Published online: 30 January 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Simplex gradients are an essential feature of many derivative free optimization algorithms, and can be employed, for example, as part of the process of defining a direction of search, or as part of a termination criterion. The calculation of a general simplex gradient in \mathbf{R}^n can be computationally expensive, and often requires an overhead operation count of $\mathcal{O}(n^3)$ and in some algorithms a storage overhead of $\mathcal{O}(n^2)$. In this work we demonstrate that the linear algebra overhead and storage costs can be reduced, both to $\mathcal{O}(n)$, when the simplex employed is regular and appropriately aligned. We also demonstrate that a gradient approximation that is second order accurate can be obtained cheaply from a combination of two, first order accurate (appropriately aligned) regular simplex gradients. Moreover, we show that, for an arbitrarily aligned regular simplex, the gradient can be computed in $\mathcal{O}(n^2)$ operations.

Keywords Positive bases · Numerical optimization · Derivative free optimization · Regular simplex · Simplex gradient · Least squares · Well poised

Mathematics Subject Classification 52B12 · 65F20 · 65F35 · 90C56

1 Introduction

Estimating derivatives is important in a wide variety of applications and many successful numerical optimization algorithms rely on gradient information and/or directional derivatives. When analytical derivatives are not directly available, it is useful to be able to obtain gradient estimates, for example, by using difference methods. Furthermore, simplex gradients are often used in derivative-free optimization as search directions,

✉ Rachael Tappenden
rachael.tappenden@canterbury.ac.nz
Ian Coope
ian.coope@canterbury.ac.nz

¹ School of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch 8041, New Zealand

like is the case of the implicit filtering algorithm [4], as descent indicators for reordering the poll directions in directional direct search [8], or in the definition of stopping criteria for algorithms [7]. A first comprehensive study on the computation of general simplex gradients was provided in [22]. In this work we investigate computationally efficient approaches to estimating the gradient at either the centroid or vertex of an appropriately aligned regular simplex.

To obtain a first order approximation to the gradient (of the function f at some point x_0) one considers the first order Taylor approximation about x_0 :

$$f(x) = f(x_0) + (x - x_0)^T \nabla f(x_0) + \mathcal{O}(\|x - x_0\|_2^2).$$

Consider a set of $m + 1$ points ($m \geq n$), $x_0, x_1, \dots, x_m \in \mathbf{R}^n$. Using the notation $g \approx \nabla f(x_0)$ to denote an approximation to the gradient, $f_j := f(x_j)$, and ignoring the order terms, the previous expression leads to the following system of equations

$$f_j - f_0 = (x_j - x_0)^T g \quad \text{for } j = 1, \dots, m. \quad (1)$$

Expression (1) is a linear regression model, and determining a least squares solution to the system (1) results in an approximation to the gradient of the underlying function. If $m = n$ and the $n + 1$ points are affinely independent then (1) is a determined system with unique solution independent of the ordering of the points. When $m > n$ the order of the points used is important because the least squares solution to the system (1) depends on which point is labeled x_0 [22].

In this work, attention is restricted to the case where $m = n + 1$ and the points x_1, x_2, \dots, x_{n+1} defining the regression model in (1) are the vertices of an appropriately aligned regular simplex and x_0 is its centroid (This will be defined in Sect. 2.). The main theme here is to determine a least squares solution to the system (1) efficiently, both in terms of the linear algebra costs and in terms of storage requirements, to determine an appropriately aligned regular simplex gradient.

For the regular simplices discussed in this work, the centroid of the simplex is denoted by x_0 , and each vertex x_1, x_2, \dots, x_{n+1} is equidistant from the centroid with

$$h := \|x_j - x_0\|_2, \quad j = 1, 2, \dots, n + 1, \quad (2)$$

where the distance h is sometimes referred to as the ‘radius’ or ‘arm length’ of the regular simplex.

The system in (1) is central to many derivative free optimization algorithms, but solving it can be a computational challenge. Firstly, usually the vectors x_0, \dots, x_{n+1} (or the differences $x_1 - x_0, \dots, x_{n+1} - x_0$) must be explicitly stored, which can be costly in terms of memory requirements, and also poses a limitation in terms of the size of problems that can be solved using such algorithms. Secondly, the computational cost (number of floating point operations) of solving such problems can also be high (e.g., if the problem is unstructured or if a general simplex is used).

Here, the use of regular simplices is investigated. The computation of regular simplex gradients was proposed in the context of derivative-free optimization of molecular

geometries [1]. One advantage of using a regular simplex is that it provides a uniform, economic ‘tiling’ of n -dimensional space, each n -dimensional tile having only $n + 1$ vertices compared to 2^n vertices for a hypercube tile. A disadvantage is that storing the vertices of the simplex is usually less efficient than that for a hypercube because it is possible to align the edges of the hypercube with the coordinate axes. However, if the orientation of the regular simplex is free to be chosen also, then we will show that it is possible to generate each vertex from a single vector by simple adjustment of one component. This enables considerable savings in storage requirements for several lattice search algorithms for optimization. For example, the multidirectional search (MDS) method of Torczon [26,27] can be implemented using either a rectangular or a regular simplex based lattice but the usual construction for the regular simplex lattice requires $\mathcal{O}(n^2)$ storage (see for example [26]). Similarly, the Hooke and Jeeves [13] lattice search method, although originally implemented in a rectangular lattice framework, can also be implemented using a regular simplex lattice (It is anticipated that each of these methods will benefit, in terms of memory requirements and computational effort, if the particular simplex construction used in this work is employed.). The added advantage of being able to compute an aligned regular simplex gradient in $\mathcal{O}(n)$ housekeeping operations using only $n + 1$ function evaluations makes it attractive for many numerical gradient based algorithms for optimization, including the recent minimal positive basis based direct search conjugate gradient methods described in [18].

The vertices of a simplex are often explicitly required during the initialization of simplex based algorithms for optimization, including the algorithms in [19–21,25,26]. Using the technique described later, the vertices of an aligned regular simplex can be constructed explicitly, whenever required, efficiently. However, it will also be shown that the vertices of the aligned regular simplex do not have to be stored in order to calculate the simplex gradient.

In derivative free optimization, one must always be mindful of the cost of function evaluations. There exist real-world applications for which computing a single function evaluation can be very costly, and in such cases it is clear that the linear algebra and memory requirements may be very small in comparison. In this work, we focus on algebraically efficient methods to compute the simplex gradient *after function evaluations are complete*. In most situations, function evaluations will dominate the overall time to compute a simplex gradient. However, this trend should not be used to justify performing other portions of the computation inefficiently. It is prudent to be as economical as possible at every stage of the optimization process.

1.1 Contributions

We state the main contributions of this work (listed in order of appearance).

1. *Aligned regular simplex gradient in $\mathcal{O}(n)$ operations* A simplex gradient is the (least squares) solution of a system of linear equations, which, in general, comes with an associated $\mathcal{O}(n^3)$ computational cost. In this work we show that, if one employs a regular simplex that is appropriately aligned, then the linear system simplifies, and the aligned regular simplex gradient can be computed in $\mathcal{O}(n)$

operations. Indeed, the gradient of the aligned regular simplex is simply a weighted sum of a vector containing function values (measured at the vertices of the simplex) and a constant vector. This is an important saving over the $\mathcal{O}(n^3)$ computational cost of solving a general unstructured linear system (see Sect. 3).

2. *Aligned regular simplex need not be explicitly stored* Here, we demonstrate that the storage needed for the computation of the aligned regular simplex gradient is $\mathcal{O}(n)$, whereas the usual storage requirements for the computation of a general simplex gradient are at least $\mathcal{O}(n)$ vectors [i.e., $\mathcal{O}(n^2)$]. In particular, it is simple and inexpensive to construct the vertices of the aligned regular simplex on-the-fly, and the simplex need not be stored explicitly at all. This is because all that is required to uniquely specify (and construct) each simplex vertex is the centroid x_0 , the simplex radius h (2) and the problem dimension n . To compute the aligned regular simplex gradient, the function values at the vertices of the simplex are required, but once a vertex has been constructed and the function value found, the vertex can be discarded. Therefore, the storage requirements of this approach are low (see Sect. 3).
3. *Function value f_0 is not required* The function value f_0 at the centroid of the regular simplex is not required in the calculation of the regular simplex gradient (at the point x_0). Moreover, we extend this result to show that it also applies to any general simplex, and not just a regular one (see Sect. 3.2).
4. *Regular simplex gradient in $\mathcal{O}(n^2)$ operations* In some applications, it may not be possible to ensure the particular alignment of the regular simplex. In this case, we show that it is still possible to calculate the regular simplex gradient in $\mathcal{O}(n^2)$ floating point operations (see Sect. 4.1).
5. *Inexpensive $\mathcal{O}(h^2)$ gradient approximation* One can efficiently compute an accurate (order h^2) gradient approximation using a Richardson extrapolation type approach. Specifically, if two first order accurate aligned regular simplex gradients are combined in a particular way, then a second order accurate approximation to the true gradient $\nabla f(x_0)$ is obtained. That is, an $\mathcal{O}(h^2)$ gradient approximation is simply the weighted sum of two $\mathcal{O}(h)$ aligned regular simplex gradients. Moreover, no additional storage is required to generate the $\mathcal{O}(h^2)$ gradient approximation (see Sect. 4.3).

1.2 Paper outline

This paper is organised as follows. Section 2 introduces the notation and technical preliminaries that are necessary to describe and set up the problem of interest. In particular, the concepts of a minimal positive basis and how a minimal positive basis is related to a simplex are discussed, and the definition of a simplex gradient is given. In Sect. 3 the main results of this work are presented, including how to construct the simplex and how to compute the aligned regular simplex gradient in $\mathcal{O}(n)$ operations. Section 4 describes several extensions to the work presented in Sects. 1–3, including a special case of a regular simplex with integer entries, as well as a technique to obtain an $\mathcal{O}(h^2)$ gradient approximation from two $\mathcal{O}(h)$ aligned regular simplex gradients. Numerical experiments are presented in Sect. 5 to demonstrate how the aligned regular

simplex and its gradient can be computed in practice, as well as how to generate an $\mathcal{O}(h^2)$ gradient approximation. Finally we make our concluding remarks in Sect. 6, and we also discuss several ideas for possible future work.

2 Notation and preliminaries

Here the variables that are used in this work are defined, and the notation is fixed. Several preliminary results that will be used later in this work are also presented.

2.1 Notation

Consider a set of $n + 2$ points $x_0, x_1, \dots, x_{n+1} \in \mathbf{R}^n$, where x_0 is the centroid of the $n + 1$ points x_1, \dots, x_{n+1} , and suppose that the function values f_1, \dots, f_{n+1} are known. The function value f_0 also appears in this work, although we will present results to confirm that it is not used in the computation of the aligned regular simplex gradient, so it is unnecessary to assume that f_0 is known. Let e be the (appropriately sized) vector of all ones and define the following vectors,

$$\mathbf{f} := \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix}, \quad \text{and} \quad \delta\mathbf{f} := \mathbf{f} - f_0e = \begin{bmatrix} f_1 - f_0 \\ \vdots \\ f_n - f_0 \end{bmatrix}, \tag{3}$$

along with their ‘extended’ versions,

$$\mathbf{f}_+ = \begin{bmatrix} \mathbf{f} \\ f_{n+1} \end{bmatrix}, \quad \text{and} \quad \delta\mathbf{f}_+ := \mathbf{f}_+ - f_0e = \begin{bmatrix} \delta\mathbf{f} \\ f_{n+1} - f_0 \end{bmatrix}. \tag{4}$$

For a general simplex (to be defined precisely in the next section), the internal ‘arms’ of the simplex are $v_j = x_j - x_0$ for $j = 1, \dots, n + 1$. However, this paper only considers *regular* simplices. In this case it is convenient to denote the ‘arms’ of the regular simplex using the vectors v_1, \dots, v_{n+1} , which satisfy the relationship

$$x_j = x_0 + hv_j, \quad \text{for } j = 1, \dots, n + 1, \tag{5}$$

for some (fixed) $h \in \mathbf{R}$, with $\|v_j\|_2 = 1$ and $\|x_j - x_0\|_2 = h$ for $j = 1, \dots, n + 1$. Thus $h > 0$ is the radius of the circumscribing hypersphere of the regular simplex and each v_j denotes a *unit* vector defining the direction of each vertex from the centroid of the simplex.

Now we can define the matrix

$$V = [v_1 \dots v_n] \in \mathbf{R}^{n \times n} \tag{6}$$

and the vector

$$v_{n+1} = - \sum_{j=1}^n v_j \equiv -Ve, \quad (7)$$

along with the ‘extended’ matrix

$$V_+ := [V \ -Ve] \in \mathbf{R}^{n \times (n+1)}. \quad (8)$$

2.2 Technical preliminaries

Here we outline several technical preliminaries that will be used in this work. These properties are known but they are stated here for completeness. For further details on the results discussed here, see, for example, [2], [5, pp. 32–34], [6, Chapter 2].

Definition 1 (*Affine independence, pg. 29 in [6]*) A set of $m + 1$ points $y_1, y_2, \dots, y_{m+1} \in \mathbf{R}^n$ is called affinely independent if the vectors $y_2 - y_1, \dots, y_{m+1} - y_1$ are linearly independent.

Definition 2 (*Definition 2.15 in [6]*) Given an affinely independent set of points $\{y_1, \dots, y_{m+1}\}$, its convex hull is called a simplex of dimension m .

Definition 3 A regular simplex is a simplex that is also a regular polytope.

A regular simplex has many interesting properties, see for example [11].

Proposition 1 *A regular simplex satisfies the following properties.*

1. *The distance between any two vertices of the simplex is constant.*
2. *The centroid of a regular simplex is equidistant from each vertex.*
3. *The angle between the vectors formed by joining the centroid to any two vertices of the simplex is constant.*

Proof The first property is a direct consequence of the definition. The second property is established in Theorem 10 in [11]. The third property follows from the first and second properties. \square

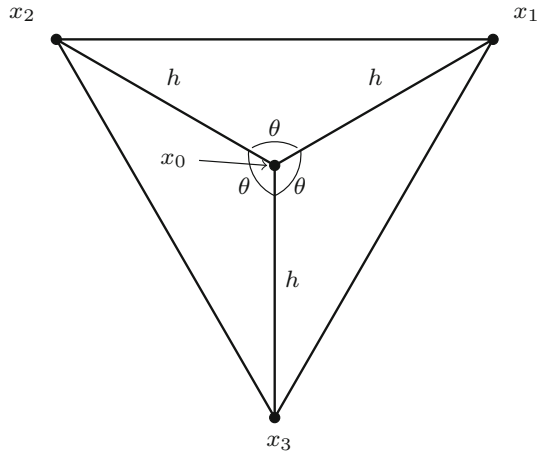
Thus, for a regular simplex, using Proposition 1 it can be established (see for example, [11]) that the centroid of the simplex x_0 is equidistant from each vertex of the simplex, and we will say that each (internal) simplex ‘arm’ (vectors v_j for $j = 1, \dots, n + 1$) is of equal length, and the angles between any two arms of the simplex are equal.

The positive span of a set of vectors $\{y_1, \dots, y_m\}$ in \mathbf{R}^n is the convex cone

$$\{y \in \mathbf{R}^n : y = \alpha_1 y_1 + \dots + \alpha_m y_m, \alpha_i \geq 0, \ i = 1, \dots, m\}.$$

Definition 4 (*Definition 2.1 in [6]*) A positive spanning set in \mathbf{R}^n is a set of vectors whose positive span is \mathbf{R}^n . The set $\{y_1, \dots, y_m\}$ is said to be positively dependent if one of the vectors is in the convex cone positively spanned by the remaining vectors,

Fig. 1 A regular simplex in \mathbf{R}^2 generated by a minimal positive basis with uniform angles. Points (vertices) x_1, x_2, x_3 are affinely independent and their convex hull is the regular simplex, while x_0 is the centroid. Each ‘arm’ of the simplex $x_j - x_0$, for $j = 1, 2, 3$ has the same length h , and the angle between any two arms is equal



i.e., if one of the vectors is a nonnegative combination of the others; otherwise, the set is positively independent. A positive basis in \mathbf{R}^n is a positively independent set whose positive span is \mathbf{R}^n .

Remark 1 Definition 4 is taken directly from [6, Definition 2.1]. As is stated in Footnote 2 of that work, “strictly speaking we should have written *nonnegative* instead of positive, but we decided to follow the notation in [9,17]”.

Lemma 1 (Minimal positive basis, Corollary 2.5 in [6])

- (i) $[I - e]$ is a minimal positive basis.
- (ii) Let $W = [w_1 \dots w_n] \in \mathbf{R}^{n \times n}$ be a nonsingular matrix. Then $[W - We]$ is a minimal positive basis for \mathbf{R}^n .

Proving the existence of a regular simplex in \mathbf{R}^n is equivalent to proving the existence of a minimal positive basis with uniform angles in \mathbf{R}^n , which is established in [1]. Moreover, the work [16] establishes the existence of a regular simplex by an induction argument.

This work considers the set-up where x_0 is the centroid of the regular simplex in \mathbf{R}^n with vertices x_1, \dots, x_{n+1} . The arms of the simplex v_1, \dots, v_{n+1} (defined in (5)) form a minimal positive basis (This will be discussed in more detail in the sections that follow.). To make this more concrete, Fig. 1 shows a regular simplex in \mathbf{R}^2 .

2.3 Simplex gradients

The following defines a simplex gradient.

Definition 5 (Simplex gradient, Sect. 2.6 in [6] and its generalization [7]) When there are $n + 2$ (or more) points, $y_1, \dots, y_m \in \mathbf{R}^n$ with $m \geq n + 2$, containing a proper subset of affinely independent points, the simplex gradient is defined as the least-squares solution of the linear system

$$f(y_j) - f(y_1) = (y_j - y_1)^T g, \quad \text{for } j = 2, \dots, m.$$

This definition depends upon whichever point is labeled y_1 and as a consequence, it is sometimes referred to as the simplex gradient *at the point* y_1 .

This work considers only $n + 1$ or $n + 2$ points (either with or without the centroid). When $m = n + 1$ the aligned regular simplex gradient is independent of the ordering of the points because the system generated by these points is a determined system. When $m = n + 2$, the centroid is explicitly used, but it can be shown that the system generated by the $n + 2$ points is equivalent to a determined system (see Sect. 3.4).

Using the results in Sects. 2.1 and 2.2, (1) can be rewritten in matrix notation as

$$V_+^T g = \frac{1}{h} \delta \mathbf{f}_+. \quad (9)$$

Definition 5 shows that, for the setup used in this work with the $n + 2$ points $x_0, \dots, x_{n+1} \in \mathbf{R}^n$ and centroid x_0 , g satisfies the normal equations form of (9):

$$V_+ V_+^T g = \frac{1}{h} V_+ \delta \mathbf{f}_+. \quad (10)$$

For further discussion on simplex gradients in a more general setting, see for example [6,7,22].

3 Constructing the simplex

The central goal of this work is to determine a least squares solution to the system (10) in $\mathcal{O}(n)$ operations/computations, while maintaining $\mathcal{O}(n)$ storage for the aligned regular simplex. One cannot hope to achieve this for a *generic* simplex. However, if one is free to choose a *regular simplex* that is *oriented* in a particular way, then this goal can be achieved. This section is devoted to the construction of an aligned regular simplex that can be stored in $\mathcal{O}(n)$ and whose gradient can be evaluated in $\mathcal{O}(n)$ operations.

3.1 Positive basis with uniform angles

Several properties of a positive basis with uniform angles are stated now. The description uses several of the concepts already presented in [6, Chapter 2].

Consider $n + 1$ *normalized* vectors $v_1, \dots, v_{n+1} \in \mathbf{R}^n$, where the angle θ between any pair of vectors v_i, v_j , for $i \neq j$ is equal. It can be shown that (see [6, Exercise 2.7(4)])

$$\cos \theta = v_i^T v_j = -\frac{1}{n}, \quad i, j \in \{1, \dots, n + 1\}, \quad i \neq j. \quad (11)$$

If (5), (7) and (11) hold, then $x_1, \dots, x_{n+1} \in \mathbf{R}^n$ are the vertices of a regular simplex with centroid x_0 . Thus, we seek to construct a positive basis of $n + 1$ normalized vectors $v_1, \dots, v_{n+1} \in \mathbf{R}^n$ such that properties (7) and (11) hold. With (11) in mind, first the aim is to find a matrix V satisfying (see (2.2) in [6])

$$A = V^T V = \begin{bmatrix} 1 & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & 1 & & \vdots \\ \vdots & & \ddots & -\frac{1}{n} \\ -\frac{1}{n} & \dots & -\frac{1}{n} & 1 \end{bmatrix}. \tag{12}$$

From (12), one may write

$$A = V^T V = \left(1 + \frac{1}{n}\right) I - \frac{1}{n} ee^T = \alpha^2 (I - \beta ee^T), \tag{13}$$

where

$$\alpha := \sqrt{\frac{n+1}{n}} \quad \text{and} \quad \beta := \frac{1}{n+1}. \tag{14}$$

Using (7) and (12), a positive basis with uniform angles exists. In particular, A in (12) is symmetric and positive definite (see, for example, [6, pg. 20], [12]) so it has a Cholesky decomposition $A = R^T R$. Taking $V = R$, which is nonsingular, combined with (7) and applying Lemma 1, establishes that $R_+ = [R - Re]$ is a normalized minimal positive basis with uniform angles, as pointed out in [6, p. 20]. The particular structure of A allows the Cholesky factor R to be calculated efficiently.

There is, however, another factorization of A that comes from the fact that any symmetric positive definite matrix has a *unique* symmetric positive definite square root [12, p. 149]. We search for a square-root matrix with similar structure to A . In particular, let

$$V = \alpha (I - \gamma ee^T), \tag{15}$$

where we must now specify $\gamma \in \mathbf{R}$. Since $A = V^T V = V^2$ it is clear that $\gamma \in \mathbf{R}$ must satisfy

$$I - \beta ee^T = (I - \gamma ee^T)^2 = I - 2\gamma ee^T + n\gamma^2 ee^T.$$

Equating the coefficients of ee^T one sees that γ is a root of the quadratic equation

$$n\gamma^2 - 2\gamma + \beta = 0, \tag{16}$$

giving two possible solutions:

$$\gamma = \frac{1}{n} \left(1 \pm \frac{1}{\sqrt{n+1}} \right). \tag{17}$$

Letting γ_1, γ_2 denote these two solutions and V_1, V_2 the corresponding matrices defined in (15) it is easy to show that $V_1 = H V_2$ where $H = I - \frac{2}{n} ee^T$ is an

elementary Householder reflection matrix (V_2 is the reflection of V_1 in the hyperplane through the origin with normal vector e and vice-versa).

Choosing the negative sign for γ in (17) yields the unique positive definite square-root matrix as the following lemma shows.

Lemma 2 *Let α, β and γ be defined in (14) and (17). The matrix $V = \alpha(I - \gamma ee^T)$ is nonsingular. Moreover, V has $n - 1$ eigenvalues equal to α and one eigenvalue satisfying*

$$\lambda_n(V) = \begin{cases} \frac{1}{\sqrt{n}} & \text{if } \gamma = \frac{1}{n}(1 - \sqrt{\beta}), \\ -\frac{1}{\sqrt{n}} & \text{if } \gamma = \frac{1}{n}(1 + \sqrt{\beta}). \end{cases}$$

Proof The matrix $-\alpha\gamma ee^T$ has $n - 1$ zero eigenvalues, and one eigenvalue equal to $-\alpha\gamma n$. Further, adding αI to $-\alpha\gamma ee^T$ simply shifts the spectrum by α . Therefore, V has $n - 1$ eigenvalues equal to α , and the remaining eigenvalue is $\alpha(1 - \gamma n) \stackrel{(17)}{=} \alpha(1 - (1 \pm \sqrt{\beta})) = \mp \alpha\sqrt{\beta} = \mp 1/\sqrt{n}$. Finally, all the eigenvalues are nonzero, so V is nonsingular. □

Corollary 1 *If $\gamma = \frac{1}{n}(1 - \sqrt{\beta})$ then V is positive definite.*

Lemma 3 *Let α, β and γ be defined in (14) and (17) and let V be defined in (15). Then*

$$Ve = \begin{cases} \frac{1}{\sqrt{n}}e & \text{if } \gamma = \frac{1}{n}(1 - \sqrt{\beta}) \\ -\frac{1}{\sqrt{n}}e & \text{if } \gamma = \frac{1}{n}(1 + \sqrt{\beta}). \end{cases} \tag{18}$$

Moreover,

$$Vee^T V^T = \frac{1}{n}ee^T. \tag{19}$$

Proof With some abuse of notation, for $\gamma = \frac{1}{n}(1 \pm \sqrt{\beta})$ we have

$$Ve \stackrel{(15)}{=} \alpha(I - \gamma ee^T)e = \alpha(1 - n\gamma)e = \mp \frac{1}{\sqrt{n}}e,$$

which proves (18). The result (19) follows immediately. □

Now we present the main result of this subsection, which shows that the choice V in (15) leads to a minimal positive basis with uniform angles.

Theorem 1 *Let α, β, γ and V be defined in (14), (17) and (15) respectively. Then $V_+ = [V - Ve]$ is a minimal positive basis with uniform angles.*

Proof By Lemma 2, V is nonsingular, so applying Lemma 1 shows that V_+ is a minimal positive basis.

It remains to show the uniform angles property. By construction, V defined in (15) satisfies (12). Then

$$V_+^T V_+ = \begin{bmatrix} V^T & \\ -(Ve)^T & \end{bmatrix} [V \ -Ve] = \begin{bmatrix} V^2 & -V^2e \\ -(V^2e)^T & e^T V^2e \end{bmatrix} \in \mathbf{R}^{(n+1) \times (n+1)}.$$

Furthermore, by (18),

$$V^2e = V(Ve) = V\left(\frac{1}{\sqrt{n}}e\right) = \frac{1}{n}e,$$

and $e^T V^2e = e^T e/n = 1$ so that

$$V_+^T V_+ = \begin{bmatrix} 1 & -\frac{1}{n} & \dots & -\frac{1}{n} \\ -\frac{1}{n} & 1 & & \vdots \\ \vdots & & \ddots & -\frac{1}{n} \\ -\frac{1}{n} & \dots & -\frac{1}{n} & 1 \end{bmatrix} \in \mathbf{R}^{(n+1) \times (n+1)}.$$

Hence, v_1, \dots, v_{n+1} also satisfy (11), so the positive basis has uniform angles. □

Although not explicitly stated, the positive basis derived from (15) has essentially been used (with a scaling factor and origin shift), for setting up initial regular simplices by several authors ([3, p. 267], [10], [14, p. 80]).

Remark 2 Lemma 3 and Theorem 1 explain why the terminology ‘aligned regular simplex’ is used in this work. Theorem 1 shows that V_+ is a minimal positive basis with uniform angles, so the resulting simplex is regular. Moreover, Lemma 3 demonstrates that Ve , which is an ‘arm’ of the simplex (recall Fig. 1), is always proportional to e ; one arm of the regular simplex is always aligned with the vector of all ones. Finally, the choice of γ simply dictates whether the simplex arm is oriented in the ‘+ e ’ or ‘− e ’ direction.

3.2 Weight attached to centroid

Here we present a general result regarding the weight attached to the centroid when solving the normal equations defining a least squares solution in linear regression. It is well known to linear regression analysts in statistics that a linear (affine) function, fitted by least squares, passes through the centroid of the data points. Adding an extra ‘observation’ at the centroid does not affect the solution for the normal of the fitted affine function—it does, of course, affect the offset. This is irrespective of the number of data points but has important consequences for calculating the simplex gradient at the centroid when fitting an affine function to $n + 2$ data points in \mathbf{R}^n . The following result generalises to any least squares system with $p > n$ data points (V need not be a normalized invertible matrix), however, we avoid introducing extra notation by focusing on the result relating to simplex gradients.

In order to define a general simplex the following equations are used:

$$v_{n+1} = - \sum_{j=1}^n v_j, \quad \text{where } v_j = x_j - x_0 \text{ for } j = 1, \dots, n + 1. \quad (20)$$

The vertices of the simplex are $\{x_i, \ i = 1, \dots, n + 1\}$ and its centroid is x_0 . Here, it is *not* assumed that $\|v_j\|_2 = 1$ for all j , so the simplex is *not necessarily* a regular simplex [i.e., (5) need not hold].

Theorem 2 *Let $\delta\mathbf{f}_+$ and $\delta\mathbf{f}$ be defined in (3) and (4), respectively, where f_0, \dots, f_{n+1} are the function values at the points x_0, \dots, x_{n+1} . Let V and V_+ be structured as in (6) and (8), respectively, but using the vectors v_1, \dots, v_{n+1} defined in (20). Then the simplex gradient g in (10) is independent of f_0 .*

Proof Clearly, the term $(V_+ V_+^T)^{-1}$ in (10) does not involve f_0 . Now,

$$\begin{aligned} V_+ \delta\mathbf{f}_+ &= V \delta\mathbf{f} - (f_{n+1} - f_0) V e \\ &= V \mathbf{f} - f_0 V e - f_{n+1} V e + f_0 V e \\ &= V (\mathbf{f} - f_{n+1} e). \end{aligned} \quad (21)$$

□

Theorem 2 shows that, if the relationship (7) holds [equivalently, the summation property in (20)], and V_+ is a minimal positive basis, then the function value at the centroid x_0 is not used when computing the simplex gradient. That is, the weight attached to f_0 is zero when calculating a simplex gradient.

3.3 Aligned regular simplex gradient

Here we state and prove the main result of this work, that the aligned regular simplex gradient can be computed in $\mathcal{O}(n)$ operations. We begin with the following result.

Lemma 4 *Let α, β and γ be defined in (14) and (17) and let V be defined in (15). Then, for V_+ defined in (8),*

$$V_+ V_+^T = \alpha^2 I.$$

Proof Note that

$$\begin{aligned} V_+ V_+^T &= [V - V e] \begin{bmatrix} V^T \\ -(V e)^T \end{bmatrix} \\ &= V V^T + V e e^T V^T \\ &\stackrel{(19)}{=} V^2 + \frac{1}{n} e e^T \\ &\stackrel{(13)}{=} \alpha^2 \left(I - \beta e e^T \right) + \frac{1}{n} e e^T \end{aligned}$$

$$\begin{aligned}
 &= \alpha^2 I - \left(\alpha^2 \beta - \frac{1}{n}\right) ee^T \\
 &\stackrel{(16)}{=} \alpha^2 I.
 \end{aligned}$$

□

Our main result follows, which shows that the aligned regular simplex gradient can be computed in $\mathcal{O}(n)$ operations.

Theorem 3 *Let α , β and γ be defined in (14) and (17), respectively, let V and V_+ be defined in (15) and (8) respectively, and let*

$$c_1 = \frac{1}{h\alpha} \quad \text{and} \quad c_2 = c_1 \left((\gamma n - 1) f_{n+1} - \gamma e^T \mathbf{f} \right).$$

Then, the aligned regular simplex gradient g is computed by

$$g = c_1 \mathbf{f} + c_2 e, \tag{22}$$

which is an $\mathcal{O}(n)$ computation.

Proof We have

$$\begin{aligned}
 g &\stackrel{(10)}{=} \frac{1}{h} \left(V_+ V_+^T \right)^{-1} V_+ \delta \mathbf{f}_+ \\
 &\stackrel{\text{Lemma 4}}{=} \frac{1}{h\alpha^2} V_+ \delta \mathbf{f}_+ \\
 &\stackrel{(21)}{=} \frac{1}{h\alpha^2} V (\mathbf{f} - f_{n+1} e) \\
 &\stackrel{(15)}{=} \frac{1}{h\alpha} (I - \gamma ee^T) (\mathbf{f} - f_{n+1} e) \\
 &= \frac{1}{h\alpha} \left(\mathbf{f} - f_{n+1} e - \gamma (e^T \mathbf{f}) e + \gamma f_{n+1} ne \right) \\
 &= \frac{1}{h\alpha} \left(\mathbf{f} + ((\gamma n - 1) f_{n+1} - \gamma (e^T \mathbf{f})) e \right).
 \end{aligned}$$

Note that the gradient is simply the sum of two (scaled) vectors, which is an $\mathcal{O}(n)$ computation (see, for example [28, p. 3]). □

Theorem 3 shows that the gradient of the aligned regular simplex can be expressed very simply as a weighted sum of the function values (measured at the vertices of the simplex) and a constant vector. Thus, it is very cheap to obtain the simplex gradient once function values have been calculated.

These results also demonstrate that using this particular simplex leads to efficiencies in terms of memory requirements. Neither the vertices of the simplex x_1, \dots, x_{n+1} , nor the arms of the simplex v_1, \dots, v_{n+1} , appear in the calculation of the aligned regular simplex gradient. All that is needed is the function values computed at the vertices of the simplex. Note that the vertices of the simplex need not be stored because they

can be computed easily on-the-fly as follows. Recall that $V = \alpha(I - \gamma ee^T)$ (15). Therefore, each arm of the simplex is

$$v_j = \alpha(e_j - \gamma e), \tag{23}$$

where e_j is the j th column of I . The j th vertex of the simplex is recovered via

$$x_j \stackrel{(5)}{=} x_0 + hv_j \stackrel{(23)}{=} x_0 + h\alpha(e_j - \gamma e) = (x_0 - h\alpha\gamma e) + h\alpha e_j. \tag{24}$$

Expression (24) shows that x_j is simply the sum of a constant vector $(x_0 - h\alpha\gamma e)$ whose j th component has been modified by $h\alpha$. The only quantities necessary to uniquely determine each vertex are x_0, h and n . To compute the aligned regular simplex gradient, the j th vertex can be generated [via (24)], the function value f_j evaluated and stored in \mathbf{f} , and subsequently, the vertex can be discarded. This confirms that the storage requirements for the aligned regular simplex gradient are $\mathcal{O}(n)$.

3.4 An alternative formulation

In Sect. 3.2 it was shown that the weight attached to the centroid is zero so that only the function values at the vertices of the simplex feature in the regular simplex gradient calculation. But $n + 1$ affinely independent points in \mathbf{R}^n define a unique interpolating affine function with constant gradient and this must, therefore, coincide with the definition of the simplex gradient defined by the $n + 2$ points used in the least-squares solution (10). This means that the regular simplex gradient could also be calculated as the solution to the square system of equations

$$(x_j - x_{n+1})^T g = (f_j - f_{n+1}), \quad j = 1, \dots, n. \tag{25}$$

It is not immediately obvious that this is an equivalent formulation. To show this equivalence algebraically we use the identity $x_j - x_{n+1} = x_j - x_0 - (x_{n+1} - x_0) = h(v_j - v_{n+1})$, and the definition of V (15) and v_{n+1} (7). The linear system of Eq. (25) can then be rewritten

$$h(v_j - v_{n+1})^T g = (f_j - f_{n+1}), \quad j = 1, \dots, n.$$

or in matrix form (after dividing by h),

$$\left(V + Vee^T \right)^T g = \left(V + ee^T V \right) g = \frac{1}{h} (\mathbf{f} - f_{n+1}e).$$

Premultiplying by the invertible matrix V then gives

$$\left(V^2 + Vee^T V \right) g = \frac{1}{h} V (\mathbf{f} - f_{n+1}e). \tag{26}$$

Lemma 4 showed that $(V^2 + Vee^T V) = \alpha^2 I$, and it is then clear that solving Eq. (26) is equivalent to finding the solution of the normal Eq. (10) by the method described in the previous section.

Remark 3 We remark that a *linear model* is being used throughout this work, so an affine function is fitted through the $n + 1$ simplex vertices, and the simplex gradient is the gradient of the affine function. Furthermore, note that if the centroid x_0 is included in the calculation of the simplex gradient at x_0 , then the offset of the affine function is affected, but this *does not affect the gradient*, i.e., the simplex gradient at the centroid is the same as the simplex gradient at any vertex when the centroid is not included (If the simplex gradient is calculated at $x_j, j \neq 0$, using the $n + 2$ points then the simplex gradient will be affected.). However, inclusion of the centroid does simplify the derivation of error bounds as is now shown.

3.5 Error bounds

Here we state explicit bounds on the error in the regular simplex gradient, compared with the analytic gradient. First we give the following result providing an error bound for the aligned regular simplex gradient at the centroid x_0 and follow with an extension giving an error bound at any vertex.

Theorem 4 *Let x_0 be the centroid of the aligned regular simplex with radius $h > 0$ and vertices $x_j = x_0 + hv_j, j = 1, 2, \dots, n + 1$. Assume that f is continuously differentiable in an open domain Ω containing $B(x_0; h)$ and ∇f is Lipschitz continuous in Ω with constant $L > 0$. Then, g , obtained by solving the system of linear Eq. (10), satisfies the error bound*

$$\|\nabla f(x_0) - g\|_2 \leq \frac{1}{2} Lh\sqrt{n}.$$

Proof Using the normal Eq. (10) defining g we can write

$$V_+ V_+^T (g - \nabla f(x_0)) = \frac{1}{h} V_+ \left(\delta \mathbf{f}_+ - h V_+^T \nabla f(x_0) \right). \tag{27}$$

The integral form of the mean value theorem provides the identity

$$f_j - f_0 = \int_0^1 (x_j - x_0)^T \nabla f(x_0 + t(x_j - x_0)) dt, \quad j = 1, \dots, n + 1.$$

Therefore, the j th component of the vector in brackets on the right-hand-side of Eq. (27) is

$$\begin{aligned} \left(\delta \mathbf{f}_+ - h V_+^T \nabla f(x_0) \right)_j &= f_j - f_0 - (x_j - x_0)^T \nabla f(x_0), \\ &= (x_j - x_0)^T \int_0^1 (\nabla f(x_0 + t(x_j - x_0)) - \nabla f(x_0)) dt, \end{aligned}$$

$$\begin{aligned} &\leq \|x_j - x_0\|_2 \int_0^1 L \|t(x_j - x_0)\| dt, \\ &= L \|x_j - x_0\|_2^2 \int_0^1 t dt, \\ &= \frac{1}{2} L h^2, \quad j = 1, \dots, n + 1, \end{aligned}$$

which provides the bound

$$\|\delta \mathbf{f}_+ - h V_+^T \nabla f(x_0)\|_2 \leq \frac{1}{2} L h^2 \sqrt{n + 1}. \tag{28}$$

Because $V_+ V_+^T = \alpha^2 I$ Eq. (27) and the bound (28) lead to the inequality

$$\alpha^2 \|g - \nabla f(x_0)\|_2 \leq \frac{1}{2} L h \sqrt{n + 1} \|V_+\|_2.$$

By Lemma 4, $\|V_+\|_2 = \alpha$, so

$$\|\nabla f(x_0) - g\|_2 \leq \frac{1}{2\alpha} h L \sqrt{n + 1}.$$

The definition of α in (14) gives the required result. □

An error bound at any vertex x_j , $j = 1, \dots, n + 1$, of the regular simplex is then easily derived from the Lipschitz continuity of the gradient of f and the triangle inequality.

$$\|\nabla f(x_j) - g\|_2 \leq \|\nabla f(x_j) - \nabla f(x_0)\|_2 + \|\nabla f(x_0) - g\|_2 \leq \left(1 + \frac{1}{2}\sqrt{n}\right) Lh.$$

4 Extensions

In this section we describe several extensions of the work presented so far. In particular, we show that a regular simplex gradient, where the simplex is arbitrarily oriented, can be computed in $\mathcal{O}(n^2)$ operations, we show that one can easily construct a regular simplex with integer entries when $n + 1$ is a perfect square, and we also show that it is computationally inexpensive to calculate an $\mathcal{O}(h^2)$ approximation to the gradient using a Richardson extrapolation type approach.

4.1 A regular simplex gradient in $\mathcal{O}(n^2)$

In practice, it may not be desirable to use the oriented regular simplices discussed so far. However, any regular simplex is related to that particular simplex formed from the aligned positive basis V_+ by a scale factor, an orientation (orthogonal matrix), a permutation of the columns, and a shift of origin. In fact the permutation can be dispensed with because if P is a permutation matrix then

$$\left(I - \gamma ee^T\right) P = P - \gamma ee^T P = P \left(I - \gamma P^T ee^T P\right) = P \left(I - \gamma ee^T\right).$$

Thus, if $W_+ = [W - We]$ is any normalized minimal positive basis with uniform angles then,

$$W = QVP = (QP)V$$

so that W is linked to V by an orthogonal transformation QP (and hence W_+ to any other normalized minimal positive basis with uniform angles). These observations enable any regular simplex gradient to be calculated in $\mathcal{O}(n^2)$ operations.

Theorem 5 Let $Z_+ = [z_1 \dots z_n z_{n+1}] = [Z z_{n+1}] \in \mathbf{R}^{n \times (n+1)}$ be any regular simplex with radius h and centroid z_0 and let $f_j = f(z_j)$, $j = 1, \dots, n+1$ be known function values. Further, let

$$u = \frac{1}{\alpha^2 h^2} (\mathbf{f} - f_{n+1}e). \tag{29}$$

Then the simplex gradient is

$$g = Zu - (e^T u) z_0, \tag{30}$$

which can be calculated in $\mathcal{O}(n^2)$ floating point operations.

Proof The interpolation conditions for the simplex gradient can be written as

$$((z_j - z_0) - (z_{n+1} - z_0))^T g = f_j - f_{n+1}, \quad j = 1, \dots, n. \tag{31}$$

Let $Y_+ = [Y - Ye]$ be the regular simplex with unit radius and with centroid at the origin defined by

$$Y = \frac{1}{h} (Z - z_0 e^T), \tag{32}$$

and let $Q \in \mathbf{R}^{n \times n}$ be the orthogonal transformation linking Y_+ to the oriented simplex $V_+ = [V - Ve]$ where $V = \alpha(I - \gamma ee^T)$ so that

$$Y = QV.$$

The square system of Eq. (31) can be written in matrix form as

$$h (Y + Yee^T)^T g = \mathbf{f} - f_{n+1}e.$$

Pre-multiplying by the invertible matrix Y and dividing by h we get

$$(YY^T + Yee^T Y^T) g = \frac{1}{h} Y (\mathbf{f} - f_{n+1}e). \tag{33}$$

Now

$$\begin{aligned} YY^T &= QV^2Q^T \\ &\stackrel{(13)}{=} \alpha^2 Q(I - \beta ee^T)Q^T \\ &= \alpha^2 (I - \beta Qee^TQ^T). \end{aligned}$$

But $Q = YV^{-1}$ so $Qe = YV^{-1}e$. By (18), $Ve = \pm \frac{1}{\sqrt{n}}e$, so that $V^{-1}e = \pm\sqrt{n}e$ and we have

$$Qee^TQ^T = nYee^TY^T.$$

Therefore,

$$YY^T = \alpha^2 I - \alpha^2 \beta n Yee^TY^T.$$

Using the definitions (14), $\alpha^2 \beta n = 1$, so that

$$YY^T = \alpha^2 I - Yee^TY^T.$$

Inserting this result in (33) we get

$$g = \frac{1}{\alpha^2 h} Y(\mathbf{f} - f_{n+1}e),$$

which is a simple matrix-vector product costing $\mathcal{O}(n^2)$ flops. In fact we do not need to calculate Y . Substituting for Y from Eq. (32) gives

$$g = \frac{1}{\alpha^2 h^2} (Z - z_0 e^T)(\mathbf{f} - f_{n+1}e).$$

Letting u be as defined in (29) gives the result (30). Finally, note that the dominant computation in (30) is the matrix-vector product Zu , which has a computational complexity of $\mathcal{O}(n^2)$ (see for example, [28, p. 2]). \square

In practice, the centroid z_0 will often be known but even if it is not given initially, its calculation is at most $\mathcal{O}(n^2)$ flops because $z_0 = \frac{1}{n+1} \sum_{j=1}^{n+1} z_j$. If a new simplex is formed by resizing a given simplex but keeping one vertex in common then the new centroid can be easily calculated from the old centroid and the resizing parameter in $\mathcal{O}(n)$ flops. Finally, we note that if h is unknown it can be calculated as $h = \|z_j - z_0\|_2$ for any j , which is an additional cost of $\mathcal{O}(n)$ flops.

4.2 Regular simplices with integer entries

The results of Sect. 3 show that one can construct a regular simplex with *integer* coordinate vertices in n -space when $n + 1$ is a perfect square. Simply let $x_0 = 0$ be

the centroid of the simplex so that $x_j = hv_j, j = 1, \dots, n + 1$ are the $n + 1$ vertices. Writing $X_+ = [x_1, \dots, x_{n+1}]$, we choose X_+ to be proportional to the rational matrix $\frac{1}{\alpha} V_+$. For example, when $n = 3$, then $n + 1 = 4$ is a perfect square, so two examples of regular simplices in \mathbf{R}^3 with integer coordinates, corresponding to the two choices for γ in (17), are

$$X_+ = \begin{bmatrix} 5 & -1 & -1 & -3 \\ -1 & 5 & -1 & -3 \\ -1 & -1 & 5 & -3 \end{bmatrix} \in \mathbf{Z}^{3 \times 4}$$

and

$$X_+ = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \in \mathbf{Z}^{3 \times 4}.$$

Schöenberg [24] proved that a regular n -simplex exists in \mathbf{R}^n with integer coordinates in the following cases, and no others:

- (i) n is even and $n + 1$ is a square;
- (ii) $n \equiv 3 \pmod{4}$;
- (iii) $n \equiv 1 \pmod{4}$ and $n + 1$ is a sum of two squares.

In particular, the first few values of n for which integer coordinate vertices exist are $n = 1, 3, 7, 8, 9, 11, 15, 17, 19, \dots$, and do not exist for $n = 2, 4, 5, 6, 10, 12, 13, 14, 16, 18, 20 \dots$

4.3 Order $\mathcal{O}(h^2)$ gradient approximation

At certain stages of an optimization algorithm an accurate gradient may be required. This is the case, for example, when deciding whether to reduce the mesh/grid size in mesh/grid based optimization algorithms, or for deciding whether a gradient based stopping condition has been satisfied. In such cases, an $\mathcal{O}(h)$ gradient approximation may not be sufficient, and a more accurate gradient, say an $\mathcal{O}(h^2)$ gradient approximation, may be desired.

The construction proposed in this paper allows one to obtain an inexpensive aligned regular simplex gradient, which is an $\mathcal{O}(h)$ approximation to the true gradient. However, it is well known in the statistics community that a Richardson’s extrapolation approach can be used to increase the accuracy of an approximation or iterative method by (at least) an order of magnitude, see for example [23,29]. Indeed, using the set-up in this paper, we now demonstrate how to obtain an $\mathcal{O}(h^2)$ approximation to the true gradient in $\mathcal{O}(n)$ operations and storage, although extra function evaluations will be required.

The key idea behind Richardson’s extrapolation is to take two approximations that are $\mathcal{O}(h)$, and use these to construct an $\mathcal{O}(h^2)$ approximation. To this end, fix x_0 , let $G = \nabla^2 f(x_0)$ and choose $h_1 = \mathcal{O}(h)$. Then one can form a regular simplex with

centroid x_0 and diameter h_1 with the vertices and ‘arms’ satisfying $x_j - x_0 = h_1 v_j$ for $j = 1, \dots, n + 1$. Now, consider the Taylor series of f about x_0 :

$$\begin{aligned} f_j &= f_0 + (x_j - x_0)^T \nabla f(x_0) + \frac{1}{2}(x_j - x_0)^T G(x_j - x_0) + \mathcal{O}(h^3) \\ &= f_0 + h_1 v_j^T \nabla f(x_0) + \frac{h_1^2}{2} v_j^T G v_j + \mathcal{O}(h^3). \end{aligned}$$

Rearranging the above and dividing by h_1 gives

$$v_j^T \nabla f(x_0) = \frac{1}{h_1}(f_j - f_0) - \frac{h_1}{2} v_j^T G v_j + \mathcal{O}(h^2). \tag{34}$$

An expression of the form (34) can be written for each $j = 1, \dots, n + 1$. Combining the $n + 1$ equations, using the notation established previously, gives

$$V_+^T \nabla f(x_0) = \frac{1}{h_1} \delta \mathbf{f}_+ - \frac{h_1}{2} \text{diag} \left(V_+^T G V_+ \right) e + \mathcal{O}(h^2), \tag{35}$$

where $\text{diag}(V_+^T G V_+)$ is a diagonal matrix with $(\text{diag}(V_+^T G V_+))_{jj} = v_j^T G v_j$. Let

$$C = -\frac{1}{2} \left(V_+ V_+^T \right)^{-1} V_+ \left(\text{diag} V^T G V \right) e, \tag{36}$$

so that (35) becomes

$$\nabla f(x_0) = g_1 + h_1 C + \mathcal{O}(h^2), \tag{37}$$

where $g_1 = \frac{1}{h_1} (V_+ V_+^T)^{-1} V_+ \delta \mathbf{f}$. By (10), g_1 is an $\mathcal{O}(h)$ approximation to the gradient at the point x_0 .

Now, fix the same x_0 and direction vectors v_1, \dots, v_{n+1} , and choose some $h_2 = \mathcal{O}(h)$. Then, constructing a simplex of diameter h_2 and following the same arguments as above, we arrive at the expression

$$\nabla f(x_0) = g_2 + C h_2 + \mathcal{O}(h^2), \tag{38}$$

where C is defined in (36), and $g_2 = \frac{1}{h_2} (V_+ V_+^T)^{-1} V_+ \delta \mathbf{f}$ is an $\mathcal{O}(h)$ approximation to the gradient at the point x_0 .

Finally, multiplying (37) by h_2 , multiplying (38) by h_1 and subtracting the second expression from the first, results in

$$\nabla f(x_0) = g_{12} + \mathcal{O}(h^2), \quad \text{where} \quad g_{12} = \frac{h_2 g_1 - h_1 g_2}{h_2 - h_1}, \tag{39}$$

i.e., g_{12} is an order h^2 accurate approximation to the true gradient at x_0 .

Moreover, if h_2 is chosen to be a multiple of h_1 (i.e., $h_2 = \eta h_1$) then

$$g_{12} = \frac{\eta h_1 g_1 - h_1 g_2}{\eta h_1 - h_1} = \frac{\eta}{\eta - 1} g_1 - \frac{1}{\eta - 1} g_2. \tag{40}$$

To make the previous arguments concrete, an algorithmic description of the procedure to find an $\mathcal{O}(h^2)$ approximation to the gradient is given in Algorithm 1. Briefly, the algorithm proceeds as follows. In Steps 2–3, an $\mathcal{O}(h)$ aligned regular simplex gradient is formed via Eq. (22) (i.e., using the procedure developed previously in this work). To obtain an $\mathcal{O}(h^2)$ gradient approximation, a second (related) $\mathcal{O}(h)$ aligned regular simplex gradient approximation is also needed, and this is computed in Steps 4–5 of Algorithm 1. Finally, in Step 6, a weighted sum of the two $\mathcal{O}(h)$ gradients is formed, resulting in an $\mathcal{O}(h^2)$ regular simplex gradient approximation.

Algorithm 1 Forming an $\mathcal{O}(h^2)$ gradient approximation from two $\mathcal{O}(h)$ gradient approximations via Richardson’s extrapolation.

- 1: **Input:** Centroid x_0 , problem dimension n , scalars $h_1 \sim \mathcal{O}(h)$ and $h_2 \sim \mathcal{O}(h)$.
 - 2: Input or compute: $f(x_j) \stackrel{(24)}{=} f(x_0 + h_1\alpha(e_j - \gamma e))$ for $j = 1, \dots, n + 1$.
 - 3: Compute g_1 via (22) using h_1 .
 - 4: Input or compute: $f(x_j) \stackrel{(24)}{=} f(x_0 + h_2\alpha(e_j - \gamma e))$ for $j = 1, \dots, n + 1$.
 - 5: Compute g_2 via (22) using h_2 .
 - 6: Compute g_{12} via (39).
-

Remark 4 We make the following comments.

1. The $\mathcal{O}(h^2)$ gradient approximation (40) is simply a weighted sum of two $\mathcal{O}(h)$ gradient approximations. The coefficients of g_1 and g_2 sum to 1.
2. For Richardson’s extrapolation, the user defined parameter η in (40) can be either positive or negative, but to avoid division by zero it cannot be set to 1. However, in this work h_1 and h_2 denote the radii of simplices, so they ‘should’ be positive (recall that $h_2 = \eta h_1$). This apparent anomaly (i.e., $\eta < 0$) is no issue in practice, but η must be interpreted carefully. Geometrically, if $\eta > 0$ then the simplex generated using h_2 (see Steps 4–5 in Algorithm 1) is simply a scaled version of the original simplex defined using h_1 (both simplices sharing the common centroid x_0). For the case $\eta < 0$, the new simplex radius is $|h_2|$, with the common centroid x_0 , but the simplex orientation changes. In this work the simplex is aligned with the direction $\pm e$. Thus, if the simplex generated using h_1 is aligned with e (resp. $-e$), then the simplex generated using h_2 will be aligned with $-e$ (resp. e) (In \mathbf{R}^2 , this corresponds to a rotation of 180° about x_0 ; see the numerical example in Sect. 5.2 and Fig. 3).
3. In this section the derivation proceeds by assuming that the simplex gradients g_1 and g_2 are both computed at the same point x_0 , and thus g_{12} is an $\mathcal{O}(h^2)$ accuracy approximation to $\nabla f(x_0)$ [and by results previously presented in this work, g_1 , g_2 and g_{12} all have a computational cost of $\mathcal{O}(n)$]. However, the arguments in Sect. 4.3 can be generalized to an $\mathcal{O}(h^2)$ approximation to $\nabla f(x)$, for some other point x say, so long as both g_1 and g_2 are $\mathcal{O}(h)$ simplex gradients at the common point x . Of course, the computational cost of obtaining g_1 and g_2 may be higher than $\mathcal{O}(n)$ for general x .

5 Numerical example

Here, two numerical examples are presented to make the ideas of the paper concrete, to highlight the simplicity and economy of the proposed approach, and to demonstrate how an $\mathcal{O}(h^2)$ approximation to the gradient can be constructed from two $\mathcal{O}(h)$ aligned regular simplex gradients. All experiments are performed on Rosenbrock's function, and MATLAB (version 2016a) is used for the calculations.

We temporarily depart from our usual notation and let $y \in \mathbf{R}^2$ with components $y = [y_1 \ y_2]^T$ so that Rosenbrock's function can be written as

$$f(y_1, y_2) = (1 - y_1)^2 + 100(y_2 - y_1^2)^2. \quad (41)$$

The gradient of (41) can be expressed analytically as

$$\nabla f(y_1, y_2) = \begin{bmatrix} -2(1 - y_1) - 400y_1(y_2 - y_1^2) \\ 200(y_2 - y_1^2) \end{bmatrix}. \quad (42)$$

Henceforth, we return to our usual notation.

5.1 Inconsistent simplex gradients

The purpose of this example is to highlight a situation that is not uncommon in derivative free optimization algorithms—that of encountering an iterate where the true (analytic) gradient and the simplex gradient point in opposite directions—and how the construction in Sect. 4.3 can be used to determine an accurate gradient direction from which to make further progress. This situation can arise, for example, when the gradient of a function at the iterate $x^{(k)}$ is close to flat. Indeed, this is one of the motivations for considering Rosenbrock's function, which has a valley floor with a shallow incline.

To highlight the situation previously described, we have selected a test point that is close to the 'floor' of the valley of Rosenbrock's function, where a good approximation to the gradient is required to make progress (Ultimately, descent methods do track this valley floor, so it is not unexpected that a point of this nature may be encountered.), We stress that the loss of accuracy is due to the regular simplex gradient being a first order approximation ($\mathcal{O}(h)$) to the analytic gradient, and is not because of the particular construction proposed in this work.

The example proceeds as follows. Suppose one wishes to compute a regular simplex gradient at the point

$$x_0 = \begin{bmatrix} 1.1 \\ 1.1^2 + 10^{-5} \end{bmatrix}. \quad (43)$$

Note that, from (42), the true gradient at the point x_0 is (to the accuracy displayed)

$$\nabla f(1.1, 1.1^2 + 10^{-5}) \approx \begin{bmatrix} 0.195599999999971 \\ 0.002000000000013 \end{bmatrix}. \tag{44}$$

The aligned regular simplex is constructed using the approach presented in Sect. 3. In particular, $n = 2$ for Rosenbrock’s function so that

$$\alpha \stackrel{(14)}{=} \sqrt{\frac{3}{2}} \quad \beta \stackrel{(14)}{=} \frac{1}{3} \quad \gamma \stackrel{(17)}{=} \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right). \tag{45}$$

Then, recalling that $V = \alpha(I - \gamma ee^T)$ [see (15)] we have

$$V_+ = [V - Ve] \approx \begin{bmatrix} 0.2588 & -0.9659 & 0.7071 \\ -0.9659 & 0.2588 & 0.7071 \end{bmatrix}. \tag{46}$$

Recall that the connection between the arms of the simplex and vertices of the simplex is given in (5) as $x_j = x_0 + h_1 v_j$ for $j = 1, 2, 3$, where we choose $h_1 = 10^{-3}$. The three vertices of the simplex are the columns of

$$X_+ = [x_1 \ x_2 \ x_3] \approx \begin{bmatrix} 1.1003 & 1.0990 & 1.1007 \\ 1.2090 & 1.2103 & 1.2107 \end{bmatrix}. \tag{47}$$

The aligned regular simplex gradient (at the point x_0) can be computed in $\mathcal{O}(n)$ operations using Theorem 3 [which requires the function values f_1, f_2, f_3 computed at the points x_1, x_2, x_3 via (41)], and is as follows:

$$g_1 \approx \begin{bmatrix} -0.095750884326868 \\ -0.017496117072893 \end{bmatrix}. \tag{48}$$

Notice that the regular simplex gradient is different from the true gradient (44). Not only are the magnitudes of the numbers different but the regular simplex gradient (48) even has the opposite sign from the true gradient. This loss of accuracy is inevitable for any first order numerical method used to approximate a gradient close to a stationary point and the usual remedy is to switch to a second order method.

However, using the techniques previously presented, another possible approach is as follows. Construct a second approximation to the gradient, again at the centroid x_0 (43), but using a different simplex diameter, $h_2 = \frac{1}{2}h_1$ say, (h_1 and h_2 are of the same order). Thus, V_+ remains unchanged but the simplex vertices become:

$$X'_+ = [x'_1 \ x'_2 \ x'_3] \approx \begin{bmatrix} 1.1001 & 1.0995 & 1.1004 \\ 1.2095 & 1.2101 & 1.2104 \end{bmatrix}. \tag{49}$$

The function values f'_1, f'_2, f'_3 are computed at the points x'_1, x'_2, x'_3 and then the aligned regular simplex gradient (at the point x_0) can be computed in $\mathcal{O}(n)$ operations via Theorem 3:

$$g_2 \approx \begin{bmatrix} 0.049842074409398 \\ -0.007735568480143 \end{bmatrix}.$$

Notice that g_2 is different from that given in (44); again, the signs and numbers do not match. In practice one does not have access to the true gradient so g_1 and g_2 are compared instead. Notice the sign of the first component g_1 is opposite from that of g_2 (they point in different directions) and the numerical values of the components are also different.

In this situation it is beneficial to use the ideas from Sect. 4.3 to improve the accuracy of the simplex gradient at x_0 . To this end, from (40) one can compute the $\mathcal{O}(h^2)$ approximation:

$$g_{12} = 2g_2 - g_1 \approx \begin{bmatrix} 0.195435033145664 \\ 0.002024980112607 \end{bmatrix}.$$

Clearly, g_{12} is a good approximation to the true gradient (44); the sign of g_{12} matches that of $\nabla f(x_0)$, and the magnitude of the components aligns well too, agreeing to 3 significant figures.

The experiment described above was repeated for fixed x_0 and V_+ , but for varying values of h_1 (with $h_2 = \frac{1}{2}h_1$ holding for each choice of h_1). The results are shown in Fig. 2. The error is measured as the difference between the true gradient $\nabla f(x_0)$ stated in (44) and ‘ g ’, where g is a notational placeholder for g_1 (22), g_2 (22) or g_{12} (39) as appropriate. The purpose of this experiment is to show that, as h_1 shrinks, the error decreases linearly, as proven in Theorem 4. The upper bound on the error (Theorem 4) is $\frac{1}{2}Lh\sqrt{n}$, and the value 2000 was selected to approximate the Lipschitz constant, because $L \approx \|\nabla^2 f(x_0) - \nabla^2 f(x_1)\|_2 / \|x_0 - x_1\|_2 \approx 1.0769 \times 10^3 \leq 2000$, where x_1 was the simplex vertex computed for $h_1 = 10^{-3}$. Theoretically, the error grows at the rate $\mathcal{O}(h)$ for g_1 and g_2 , and at the rate $\mathcal{O}(h^2)$ for g_{12} . However, in practice the error grows as $\mathcal{O}(h^m)$, and the values ‘ m ’ for g_1 , g_2 and g_{12} are reported for this experiment in the legend of Fig. 2. Figure 2 shows that the error in g_1 , g_2 and g_{12} closely matches what is predicted in theory.

5.2 High accuracy near the solution

In this example we show how the techniques in Sect. 4.3 can be used to hone in on a stationary point. Suppose one wishes to compute the regular simplex gradient at the point

$$x_0 = \begin{bmatrix} 0.9 \\ 0.81 \end{bmatrix}, \tag{50}$$

which is close to the solution $x^* = [1 \ 1]^T$. Using (42), the analytic gradient at x_0 (50) is

$$\nabla f(0.9, 0.81) \approx \begin{bmatrix} -0.2000000000000000 \\ 0 \end{bmatrix}. \tag{51}$$

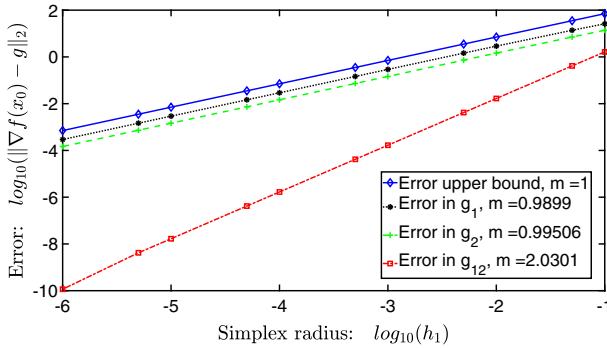


Fig. 2 Plot showing the error in the gradient approximation as the simplex radius varies from $h_1 = 10^{-6}$ (left) to $h_1 = 10^{-1}$ (right). From left to right along the x -axis (corresponding to a growing simplex radius) the error in the aligned regular simplex gradient increases, as expected by Theorem 4. The slopes m correspond to the order of the accuracy in the gradient approximations, $\mathcal{O}(h_1^m)$. The gradients g_1 and g_2 have m values of 0.9899 and 0.99506, which closely match their predicted linear accuracy, while the theoretical quadratic accuracy of g_{12} is mirrored in practice by the value $m = 2.0301$

Now, construct the aligned regular simplex using the approach in Sect. 3. Here, $n = 2$, α, β and γ are the same as in (45), V_+ is the same as in (46), and $h_1 = 10^{-6}$ was chosen. The vertices of the simplex are computed as $x_j = x_0 + h_1 v_j$ for $j = 1, 2, 3$ [see (5)], and are the columns of

$$X_+ = [x_1 \ x_2 \ x_3] \approx \begin{bmatrix} 0.90000026 & 0.89999903 & 0.90000071 \\ 0.80999903 & 0.81000026 & 0.81000071 \end{bmatrix}. \tag{52}$$

The simplex gradient is computed in $\mathcal{O}(n)$ operations using Theorem 3 and is

$$g_1 \approx \begin{bmatrix} -0.200206828472801 \\ -0.000047729764447 \end{bmatrix}. \tag{53}$$

The regular simplex gradient g_1 is a good approximation to the true gradient (51). The first component of (53) has the same sign as the first component of (51), and they match to 3 significant figures. Also, the second component of (53) is $\sim -5 \times 10^{-5}$, which while not exactly zero, is still small.

Now consider computing a second aligned regular simplex gradient, again at the point x_0 , but now with $h_2 = -\frac{1}{2}h_1$, recall Remark 4(2) (A negative multiple was chosen for demonstration purposes only.). The vertices of the simplex are computed as $x_j = x_0 + h_2 v_j$ for $j = 1, 2, 3$ [see (5)], and are the columns of

$$X_+ = [x_1 \ x_2 \ x_3] \approx \begin{bmatrix} 0.89999987 & 0.90000048 & 0.89999965 \\ 0.81000048 & 0.80999987 & 0.80999965 \end{bmatrix}. \tag{54}$$

Using Theorem 3, the regular simplex gradient is

$$g_2 \approx \begin{bmatrix} -0.199896585549141 \\ 0.000023864840841 \end{bmatrix}. \tag{55}$$

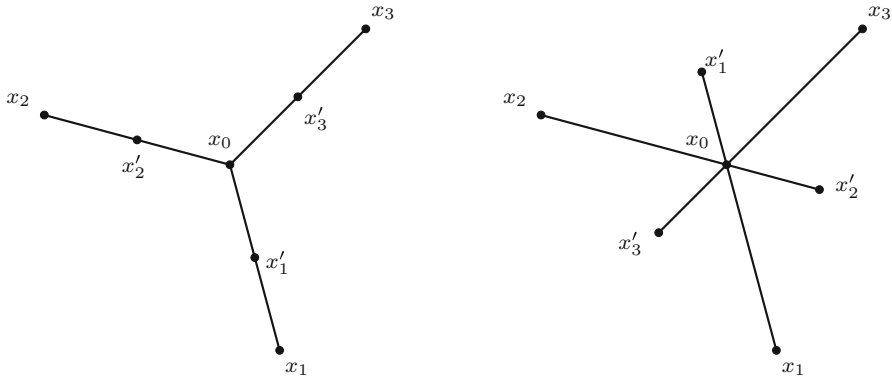


Fig. 3 A schematic of the simplices generated in the numerical experiments. The left plot relates to the experiment in Sect. 5.1, while the right plot relates to the experiment in Sect. 5.2

Again, g_2 is a good approximation to the true gradient. The first components of (55) and (51) are similar, and the second component of (55) is also small. Notice that g_1 and g_2 are also similar, although the sign of the second component of g_1 is opposite that of g_2 . Now (40) can be used to combine g_1 and g_2 and obtain an $\mathcal{O}(h^2)$ approximation to the true gradient:

$$g_{12} \approx \begin{bmatrix} -0.199999999857027 \\ -0.00000000027588 \end{bmatrix}.$$

Clearly, g_{12} is a good approximation to $\nabla f(x_0)$; g_{12} is accurate to 10 d.p.

These examples make it clear that obtaining a high accuracy aligned regular simplex gradient is cheap (once function evaluations have been computed). Each regular simplex gradient (i.e., g_1 and g_2) is obtained in $\mathcal{O}(n)$ operations, and the $\mathcal{O}(h^2)$ approximation g_{12} is simply a weighted sum of g_1 and g_2 , so it also costs $\mathcal{O}(n)$.

In the above examples the simplices had the same centroid for each first order gradient calculation but this need not always be the case. Sometimes the new simplex is obtained by shrinking (or expanding) the current simplex keeping one of the vertices fixed and/or by rotating the current simplex about a vertex. In such cases the formula (40) can still be applied and gives a second order estimate at the vertex common to the two simplices used in the two first order estimates. The so-called ‘centered difference simplex gradient’ [15, p. 115] is one such example. If the centroid of the simplex is not used it may also be convenient to replace the ‘arm-length’ h by the edge length s . These are simply related through the cosine rule ($s = \sqrt{2}\alpha h = h\sqrt{2 + 2/n}$).

We conclude this section with a schematic of the simplices generated in each of these numerical experiments. The left plot in Fig. 3 relates to the experiment in Sect. 5.1, while the right plot relates to the experiment in Sect. 5.2. In the left plot in Fig. 3, points x_1, x_2, x_3 [see (54)] represent vertices of the simplex with $h_1 = 10^{-3}$. Points x'_1, x'_2, x'_3 [see (49)] represent vertices of the simplex with $h_2 = \frac{1}{2}h_1 = 5 \times 10^{-4}$. This choice of h_2 simply shrinks the regular simplex while maintaining the orientation of the original simplex. On the other hand, the right plot in Fig. 3 corresponds to the

experiment in Sect. 5.2. In particular, points x_1, x_2, x_3 represent vertices of the simplex with $h_1 = 10^{-6}$. However points x'_1, x'_2, x'_3 represent vertices of the simplex with $h_2 = -\frac{1}{2}h_1$. This choice of h_2 shrinks and also rotates the regular simplex. (Because this is an aligned regular simplex, in \mathbf{R}^2 this is equivalent to rotating the simplex about x_0 by 180°).

6 Conclusion

In this work it was shown that a simplex gradient can be obtained efficiently, in terms of the linear algebra and memory costs, when the simplex is regular and appropriately aligned. A simplex gradient is the least-squares solution of a system of linear equations, which can have a computational cost of $\mathcal{O}(n^3)$ for a general and unstructured system. However, due to the properties of the aligned regular simplex, the linear algebra of the least squares system simplifies, and the aligned regular simplex gradient can be expressed as a weighted sum of the function values (measured at the vertices of the simplex) and a constant vector. Therefore, the computational cost of obtaining an aligned regular simplex gradient is only $\mathcal{O}(n)$. Furthermore, the storage costs are low. Indeed, V_+ need not be stored at all; the vertices of the aligned regular simplex can be constructed on-the-fly using only the centroid x_0 and radius h . Moreover, it was shown that if the regular simplex is arbitrarily oriented, then the regular simplex gradient can be computed in at most $\mathcal{O}(n^2)$.

Several extensions were presented, including how to generate a simplex with integer coordinates when $n + 1$ is a perfect square. We also showed that Richardson's extrapolation can be employed to obtain an $\mathcal{O}(h^2)$ accuracy approximation to the true gradient from two regular simplex gradients.

6.1 Future work

The main contribution of this work was to show that a regular simplex gradient can be determined efficiently in terms of the numerical linear algebra and storage costs. Simplex gradients are useful in a wide range of contexts and applications, including using the simplex gradient to determine a search direction, employing the simplex gradient in an algorithm termination condition, and determining when to shrink the mesh size in a grid based method. Future work includes embedding this aligned regular simplex gradient computation into an optimization routine to investigate how this gradient approximation affects overall algorithm performance.

Acknowledgements The authors thank Luis Vicente, and the anonymous referees for their helpful comments and suggestions, leading to improvements in an earlier version of this work.

References

1. Alberto, P., Nogueira, F., Rocha, H.: Pattern search methods for user-provided points: application to molecular geometry problems. *SIAM J. Optim.* **14**(4), 1216–1236 (2004)

2. Audet, C., Hare, W.: *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Cham (2017)
3. Belegundu, A., Chandrupatla, T. (eds.): *Optimization Concepts and Applications in Engineering* (second printing), 1st edn. Prentice Hall, Upper Saddle River (1999)
4. Bortz, D.M., Kelley, C.T.: The simplex gradient and noisy optimization problems. In: Borggaard, J.T., Burns, J., Cliff, E., Schreck, S. (eds.) *Computational Methods in Optimal Design and Control*, Progress in Systems and Control Theory, vol. 24, pp. 77–90. Birkhäuser, Boston (1998)
5. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
6. Conn, A., Scheinberg, K., Vicente, L.: *Introduction to Derivative-Free Optimization*, MPS–SIAM Series on Optimization. SIAM, Philadelphia (2009)
7. Custódio, A.L., Dennis Jr., J.E., Vicente, L.N.: Using simplex gradients of nonsmooth functions in direct search methods. *IMA J. Numer. Anal.* **28**(4), 770–784 (2008)
8. Custódio, A.L., Vicente, L.N.: Using sampling and simplex derivatives in pattern search methods. *SIAM J. Optim.* **18**(2), 537–555 (2007)
9. Davis, C.: Theory of positive linear dependence. *Am. J. Math.* **76**(4), 733–746 (1954). <https://doi.org/10.2307/2372648>
10. Dennis, J., Torczon, V.: Direct search methods on parallel machines. *SIAM J. Optim.* **1**, 448–474 (1991)
11. El-Gebeily, M.A., Fiagbedzi, Y.A.: On certain properties of the regular n -simplex. *Int. J. Math. Educ. Sci. Technol.* **35**(4), 617–629 (2004)
12. Golub, G., Loan, C.V.: *Matrix Computations*, 4th edn. JHU Press, Baltimore (2016)
13. Hooke, R., Jeeves, T.: Direct search solution of numerical and statistical problems. *J. Assoc. Comput. Mach. (ACM)* **8**, 212–219 (1961)
14. Jacoby, S.L.S., Kowalik, J.S., Pizzo, J.T.: *Iterative Methods for Nonlinear Optimization Problems*. Prentice Hall, Englewood Cliffs (1972)
15. Kelley, C.: *Iterative Methods for Optimization*. Frontiers in Applied Mathematics. SIAM, Philadelphia (1999)
16. Lazebnik, F.: On a Regular Simplex in \mathbf{R}^n . Technical report, Department of Mathematical Sciences, University of Delaware (2004). <http://www.math.udel.edu/~lazebnik/papers/simplex.pdf>
17. Lewis, R.M., Torczon, V.: Rank Ordering and Positive Bases in Pattern Search Algorithms. Technical Report TR96-71, ICASE, NASA Langley Research Center, Hampton (1999)
18. Liu, Q.: Two minimal positive basis based direct search conjugate gradient methods for computationally expensive functions. *Numer. Algorithms* **58**, 461–474 (2011)
19. Nelder, J., Mead, R.: A simplex method for function minimization. *Comput. J.* **7**(4), 308–313 (1965)
20. Parkinson, J., Hutchinson, D.: An investigation into the efficiency of variants on the simplex method. In: Lootsma, F.A. (ed.) *Numerical Methods for Non-linear Optimization*, University of Dundee in Scotland, UK, 28 June–1 July 1971. London-New York Academic Press (1972)
21. Price, C., Coope, I., Byatt, D.: A convergent variant of the Nelder–Mead algorithm. *J. Optim. Theory Appl.* **113**(1), 5–19 (2002)
22. Regis, R.G.: The calculus of simplex gradients. *Optim. Lett.* **9**, 845–865 (2015)
23. Richardson, L., Gaunt, J.: The deferred approach to the limit. *Philos. Trans. R. Soc. Lond.* **226**, 299–361 (1927)
24. Schoenberg, I.: Regular simplices and quadratic forms. *J. Lond. Math. Soc.* **12**, 48–55 (1937)
25. Spendley, W., Hext, G., Himsforth, F.: Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics* **4**, 441–461 (1962)
26. Torczon, V.: *Multi-directional Search: A Direct Search Algorithm for Parallel Machines*. Ph.D. thesis, Rice University, Houston, TX (1989)
27. Torczon, V.: On the convergence of the multidirectional search algorithm. *SIAM J. Optim.* **1**, 123–145 (1991)
28. Watkins, D.S.: *Fundamentals of Matrix Computations*, Pure and Applied Mathematics, 3rd edn. Wiley, Hoboken (2010)
29. Zlatev, Z., Dimov, I., Faragó, I., Havasi, A.: *Richardson Extrapolation: Practical Aspects and Applications*, 1st edn. Series in Applied and Numerical Mathematics 2. De Gruyter, Berlin (2018)