PATTERN SEARCH METHODS FOR USER-PROVIDED POINTS: APPLICATION TO MOLECULAR GEOMETRY PROBLEMS [*]

PEDRO ALBERTO [†], FERNANDO NOGUEIRA [‡], HUMBERTO ROCHA [§], AND LUíS N. VICENTE [¶]

Abstract. This paper deals with the application of pattern search methods to the numerical solution of a class of molecular geometry problems with important applications in molecular physics and chemistry. The goal is to find a configuration of a cluster or a molecule with minimum total energy.

The minimization problems in this class of geometry molecular problems have no constraints and the objective function is smooth. The difficulties arise from the existence of several local minima, and especially, from the expensive function evaluation (total energy) and the possible non-availability of first-order derivatives.

We introduce a pattern search approach that attempts to exploit the physical nature of the problem by using energy lowering geometrical transformations and to take advantage of parallelism without the use of derivatives. Numerical results with a particular instance of this new class of pattern search methods are presented showing the promise of our approach.

The new pattern search methods can be used in any other context where there is an user-provided scheme to generate points leading to potential objective function decrease.

Key words. pattern search methods, expensive function evaluations, parallel computing, user-provided points, molecular geometry, geometrical transformations

AMS subject classification. 49M37, 65K05, 65Z05, 81V55, 90C30, 90C56, 90C90, 92E99

1. Introduction.

The motivation behind the study of the geometrical arrangement of atoms in a molecule or cluster is its close relation to their chemical and physical properties (e.g. optical response). For example, patterns in the structure of related systems can give a powerful insight to their physical properties. This is the case, for instance, of atomic clusters of different sizes of a single element, or of different elements in the same group of the periodic table. In most cases, clear and unambiguous structural information is difficult to obtain experimentally; theory then plays a particularly important role.

The stable configurations of atoms in any material can be found by minimization of the total energy of the system with respect to the atomic positions. The most stable structure is the one with the lowest total energy. The theoretical procedure can be seen as two separate problems: obtaining the total energy for a given configuration and minimizing it with respect to the atomic coordinates. Only the second problem is to be addressed in this work. There are several geometrically distinct structures (isomers) (i.e. structures with the same number of atoms but different shapes) for which the total energy is locally minimized. As some of these can be simultaneously present in an experiment, it is sometimes desirable not only to find the lowest energy

---

[*] This work was supported by FCT under grant Praxis/P/FIS/14195/1998.

[†] Departamento de Física da Universidade de Coimbra, 3004-516 Coimbra, Portugal; Centro de Física Computacional (`pedro@teor.fis.uc.pt`).

[‡] Departamento de Física da Universidade de Coimbra, 3004-516 Coimbra, Portugal; Centro de Física Computacional (`fnog@teor.fis.uc.pt`).

[§] Universidade Católica, Pólo de Viseu, 3504-505 Viseu, Portugal (`hrocha@mat.uc.pt`).

[¶] Departamento de Matemática da Universidade de Coimbra, 3001-454 Coimbra, Portugal (`lnv@mat.uc.pt`). Support for this author was also provided by Centro de Matemática da Universidade de Coimbra, by FCT under grant POCTI/35059/MAT/2000, by the European Union under grant IST-2000-26063, and by Fundação Calouste Gulbenkian.

structure but also to find other low-lying local minima. The number of these local energy minima grows exponentially with the number of atoms, making it hard to find the lowest energy configuration of a moderately sized cluster or molecule[1], even when using two-body potentials that give rise to smooth energy surfaces. For Lennard-Jones clusters, it has been found that the number of isomers grows from 2 for a 6-atom cluster to 988 for a 13-atom cluster [14], although realistic potentials yield less local minima.

With the exception of noble gases, Lennard-Jones potentials provide very unrealistic descriptions of physical systems. We are interested in more realistic approximations such as the local density plane-wave total energy calculation [18] briefly described in the Appendix B. The expensive numerical minimization of the total energy calculated with this method motivated the work reported in this paper.

Methods commonly used to minimize the total energy include simulated annealing, steepest descent and other gradient based methods, and genetic algorithms. Good results have been obtained coupling some of these. An example is the so-called Langevin dynamics method, proposed some years ago by Biswas and Hamann [3]. This minimization method is a combination of simulated annealing and gradient techniques, and has proved to be very efficient for small molecules. The total energy gradient gives the internal forces on the atoms, that can be used to “guide” the annealing process, i.e., to introduce a bias in the minimization process, turning it into a “smart simulated annealing”, as the system does not evolve at random. Despite the use of the gradient, the Langevin dynamics method retains the possibility to move away from local minima that are not global. But this approach has several drawbacks. On one hand, it is not by itself parallelizable. On the other hand, it is developed to run for a given fixed number of iterations, instead of incorporating an autonomous stopping criterion. Furthermore, the method requires the possibly expensive calculation of the gradient. Another popular method that combines simulated annealing and gradient techniques is the method of Car and Parrinello [6] that also shares these numerical inconvenients.

In many cases, obtaining the gradient of the total energy can be too time consuming. Only in the simpler (least accurate) methods of calculating the total energy of a cluster, the gradient is available at moderate cost. Moreover, there are situations where the gradient is not available [33]. Therefore, many interesting problems in physics are being tackled using methods where no calculation of the total energy gradient is required. Among the methods which do not require the computation of the gradient are pattern search methods. In this paper, we develop a class of pattern search methods suited for molecular geometry problems and apply it to sodium clusters to determine the geometry that minimizes the total energy (sodium clusters are a typical, well known test system in cluster physics). This paper does not address the local refinement that could be achieved by applying local optimization techniques after the conformational searching has been applied, in order to identify more precisely the global optimizer; pattern search methods are used only in the conformational searching.

Pattern search methods are an instance of direct search methods where the step directions are not modified at the end of each iteration. Examples of pattern search methods are the coordinate search with fixed step lengths, evolutionary operation using factorial designs [5], the original pattern search algorithm of Hookes and Jeeves

---

[1] The terms cluster and molecule will be used without any distinction being made between them. In all cases they should be interpreted as referring simply to a collection of atoms.

[16], and the multidirectional search algorithm of Dennis and Torczon [10, 28] also referred to as the parallel direct search (PDS) method. A unified framework for pattern search methods was proposed by Torczon [29] and improved by Audet and Dennis [2] (see also the essay [21]). Surveys of other derivative free methods, including other direct search methods for unconstrained optimization (such as the well-known Nelder-Mead algorithm [23]) can be found in [8, 27, 30, 32].

The application of pattern search methods to molecular geometry is not new. Meza and Martinez [22] have compared PDS, genetic algorithms and simulated annealing using Lennard-Jones potentials, concluding that PDS could also be used in conformational searching, and showing that it performed as well as genetic algorithms and substantially better than simulated annealing for large molecules. Pattern search methods have also been combined with evolutionary techniques (see the work by Hart [12, 13]) and the resulting evolutionary pattern search method compared favourably against evolutionary algorithms.

This paper is divided as follows. We start in Section 2 by introducing pattern search methods in a quite general framework. In Section 3 we introduce our new class of pattern search methods for user-provided points: Section 3.1 presents a family of positive bases with desirable uniform directionality properties; Section 3.2 combines the pattern generated by these positive bases with user-provided points and develops the new class of pattern search methods. The user-provided points computation is illustrated by introducing geometrical transformations with physical meaning in the context of molecular geometry (see Appendix A). In Section 4 we show numerical results with an implementation of our pattern search methods for user-provided points in molecular geometry problems. Finally, in Section 5 we draw conclusions and present prospects of future work. In Appendix B we provide a brief description of the total energy evaluation and comment on its numerical complexity.

This new class of pattern search methods can also be applied to other application contexts where the user can provide a scheme to compute points that can lead to an objective function decrease. We have implemented this class of pattern search methods for general user-provided points as well as for the molecular geometry context described above. The codes and their documentation can be downloaded from the web site `http://www.mat.uc.pt/∼lnv/psm/`. Both versions have been implemented in Fortran 95. The parallel version uses the parallelization protocol MPI. See [1] for more details.

2. Pattern search methods and positive bases.

We use $\| \cdot \|$ and $\langle \cdot, \cdot \rangle$ to represent the Euclidean norm and inner product, respectively. By abuse of notation, if A is a matrix, $a \in A$ means that the vector a is a column of A. It will be also convenient to assume that $[a_1 \cdots a_r]$ represents, not only the matrix with r columns, but also, depending on the context, the set of r vectors $\{a_1, \ldots, a_r\}$. The identity matrix is denoted by I and its i-th column by e_i. Finally, we write e to represent a vector of ones with appropriate size.

2.1. Positive bases.

We present a few basic properties of positive bases from the theory of positive linear dependence developed by Davis [9] (see also Lewis and Torczon [20]). The *positive span*[2] of a set of vectors $[v_1 \cdots v_r]$ is the convex cone

$$\{v \in \mathbb{R}^n : \ v = \alpha_1 v_1 + \cdots + \alpha_r v_r, \quad \alpha_i \geq 0, \ i = 1, \ldots, r\}.$$

---

[2] Strictly speaking we should have written *nonnegative* instead of positive, but we decided to follow the notation in [9], [20]. We also note that by *span* we mean *linear span*.

The set $[v_1 \cdots v_r]$ is said to be *positively dependent* if one of the vectors is in the convex cone positively spanned by the remaining vectors, i.e., if one of the vectors is a positive combination of the others; otherwise the set is *positively independent.* A *positive basis* is a positively independent set whose positive span is $\mathbb{R}^n$. Alternatively, a positive basis for $\mathbb{R}^n$ can be defined as a set of nonzero vectors of $\mathbb{R}^n$ whose positive combinations span $\mathbb{R}^n$, but no proper set does. The following theorem [9] indicates that a positive spanning set contains at least $n+1$ vectors in $\mathbb{R}^n$.

THEOREM 2.1. *If $[v_1 \cdots v_r]$ positively spans $\mathbb{R}^n$, then it contains a subset with $r-1$ elements that spans $\mathbb{R}^n$.*

It can also be shown that a positive basis cannot contain more than $2n$ elements [9]. Positive basis with $n+1$ and $2n$ elements are referred to as *minimal* and *maximal* positive basis, respectively.

We present now three necessary and sufficient characterizations for a set that spans $\mathbb{R}^n$ to also span $\mathbb{R}^n$ positively [9].

THEOREM 2.2. *Let $[v_1 \cdots v_r]$, with $v_i \neq 0$ for all $i \in \{1, \ldots, r\}$, span $\mathbb{R}^n$. Then the following are equivalent:*

(i) *$[v_1 \cdots v_r]$ positively spans for $\mathbb{R}^n$.*

(ii) *For every $i = 1, \ldots, r$, $-v_i$ is in the convex cone positively spanned by the remaining $r-1$ vectors.*

(iii) *There exist real scalars $\alpha_1, \ldots, \alpha_r$ with $\alpha_i > 0, i \in \{1, \ldots, r\}$, such that $\sum_{i=1}^r \alpha_i v_i = 0$.*

(iv) *For every nonzero vector $b \in \mathbb{R}^n$, there exists an index i in $\{1, \ldots, r\}$ for which $b^\top v_i > 0$.*

The following result provides a simple mechanism for generating different positive bases. The proof can be found in [20].

THEOREM 2.3. *Suppose $[v_1 \cdots v_r]$ is a positive basis for $\mathbb{R}^n$ and $B \in \mathbb{R}^{n \times n}$ is a nonsingular matrix. Then $[Bv_1 \cdots Bv_r]$ is also a positive basis for $\mathbb{R}^n$.*

From Theorems 2.2 and 2.3, we can easily deduce the following corollary:

COROLLARY 2.1. *Let $B = [b_1 \cdots b_n] \in \mathbb{R}^{n \times n}$ be a nonsingular matrix. Then $[B \ \ -\sum_{i=1}^n b_i]$ is a positive basis for $\mathbb{R}^n$.*

A trivial consequence of this corollary is that $[I \ \ -e]$ is a positive basis.

2.2. Pattern search methods.

We present pattern search methods for unconstrained optimization problems of the form

$$\min \ f(x), \quad x \in \mathbb{R}^n,$$

and briefly describe their main convergence properties. Pattern search methods are iterative methods generating a sequence of iterates $\{x_k\}$. Given the current iterate x_k, at each iteration k, the next point x_{k+1} is chosen from a finite number of candidates on a given *mesh* M_k. The next iterate, if iteration k is *successful*, must provide a decrease on the objective function: $f(x_{k+1}) < f(x_k)$.

In order to define the mesh M_k, let us consider a set $\mathcal{V}$ of m positive bases. For convenience, let us abuse notation and also denote by $\mathcal{V}$ the matrix whose columns correspond to the vectors in the m positive bases. The number of columns of $\mathcal{V}$, denoted by $|\mathcal{V}|$, is the sum of the number of vectors in all positive bases. The mesh at iteration k is then defined as

$$M_k \ = \ \{x_k + \Delta_k \mathcal{V} z : \ z \in W \subseteq \mathbb{Z}^{|\mathcal{V}|}\}, \tag{2.1}$$

where $\Delta_k > 0$ is the mesh size parameter. Possible choices for W are

$$W = \mathbb{Z}^{|\mathcal{V}|}, \qquad W = \mathbb{N}^{|\mathcal{V}|}.$$

The choice we actually use in our implementation is

$$W = \{ne_i : n \in \mathbb{N},\ i = 1, \ldots, |\mathcal{V}|\}. \tag{2.2}$$

The mechanism of pattern search methods is best explained by considering two phases at every iteration. The first phase, or step, consists of a finite search on the mesh, with the goal of finding a new iterate that decreases the value of the objective function at the current iterate. This step, called the *search step*, is free of any other rules, as long as it searches only a finite number of points in the mesh. If the search step is unsuccessful, a second phase or step, called the *poll step*, is performed around the current iterate with the goal of decreasing the objective function.

The poll step follows stricter rules and appeals to the concept of a positive basis described in the previous section. In this step the candidate for a new iterate x_{k+1} is chosen in the *mesh neighborhood* around x_k

$$\mathcal{N}(x_k) \ = \ \{x_k + \Delta_k v : \text{ for all } v \in V_k(x_k)\},$$

where $V_k(x_k)$ is a positive basis chosen from the finite set $\mathcal{V}$ of positive bases. This set $\mathcal{V}$ of positive bases is specified *a priori*, but the choice of each $V_k(x_k) \in \mathcal{V}$ may depend on k and x_k. Note that the poll step also searches points in the mesh since every column v of any of the positive bases in $\mathcal{V}$ is of the form $\mathcal{V}z$ with $z = e_i$ for a given $i \in \{1, \ldots, |\mathcal{V}|\}$.

We have now all the ingredients to describe pattern search methods.

Algorithm 2.1 (Pattern search methods).

0. Initialization Choose a rational number $\tau > 1$ and an integer number $m_{max} \geq 1$. Choose $x_0 \in \mathbb{R}^n$ and $\Delta_0 \in \mathbb{R}_+$. Set $k = 0$.

1. Search step (in current mesh) With the goal of decreasing $f(x_k)$, try to obtain x_{k+1}^{trial} by evaluating f at a finite number of points in M_k. If $x_{k+1}^{trial} \in M_k$ is found satisfying $f(x_{k+1}^{trial}) < f(x_k)$, then set $x_{k+1} = x_{k+1}^{trial}$, and go to step 3, expanding M_k (search step and iteration are declared successful).

2. Poll step (in mesh neighborhood given by the positive basis) This step is reached only if the search step is unsuccessful. If $f(x_k) \leq f(x)$ for every x in the mesh neighborhood $\mathcal{N}(x_k)$, go to step 4, shrinking M_k (poll step and iteration are declared unsuccessful). Otherwise, choose a point $x_{k+1} \in \mathcal{N}(x_k)$ such that $f(x_{k+1}) < f(x_k)$ and go to step 3, expanding M_k (poll step and iteration are declared successful).

3. Mesh expansion (at successful iterations) Let $\Delta_{k+1} = \tau^{m_k^+} \Delta_k$ (with $0 \leq m_k^+ \leq m_{max}$). Increase k by one, and move to step 1 for a new iteration.

4. Mesh reduction (at unsuccessful iterations) Let $\Delta_{k+1} = \tau^{m_k^-} \Delta_k$ (with $-m_{max} \leq m_k^- \leq -1$). Increase k by one, and move to step 1 for a new iteration.

The search step provides the flexibility for a global search, and influences the quality of the local minimizer or stationary point found by the method. The poll step is applied when the search step fails to produce a better point. The poll step attempts to perform a local search in a mesh neighborhood that, for a sufficient small mesh parameter Δ_k, is guaranteed to provide a function reduction, unless the current iterate is at a stationary point (a fact that can be inferred by Theorem 2.2.iv with $b = -\nabla f(x_k)$). So, if the poll step also fails, the mesh parameter Δ_k must be decreased.

An interesting feature of pattern search methods is the simple way in which they can be parallelized. The poll step and the search step can be implemented by requiring different processors to evaluate the objective function at different points; their strategies can actually depend on the number of processors available.

Pattern search methods, as described above, share the following convergence result, provided the following assumption is made on the mesh: each column i of $\mathcal{V}$ is given by $G\bar{z}_i$, where $G \in \mathbb{R}^{n \times n}$ is a nonsingular generating matrix and $\bar{z}_i$ is an integer vector in $\mathbb{Z}^n$.

THEOREM 2.4. *Suppose that the level set* $L(x_0) = \{x \in \mathbb{R}^n : \ f(x) \leq f(x_0)\}$ *is compact and that* f *is continuously differentiable in an open set containing* $L(x_0)$*. Then*

$$\liminf_{k \longrightarrow +\infty} \|\nabla f(x_k)\| \ = \ 0,$$

and there exists at least one limit point x_* *such that* $\nabla f(x_*) = 0$*.*

Furthermore, if $\lim_{k \longrightarrow +\infty} \Delta_k = 0$*,* $\|x_{k+1} - x_k\| \leq C\Delta_k$ *for some constant* $C > 0$ *independent of the iteration counter* k*, and* $x_{k+1} = \mathit{argmin}_{x \in \mathcal{N}(x_k)} f(x)$ *in the poll step, then*

$$\lim_{k \longrightarrow +\infty} \|\nabla f(x_k)\| \ = \ 0,$$

and every limit point x_* *satisfies* $\nabla f(x_*) = 0$*.*

The proof can be found, for instance, in [2, 20, 29].

We note finally that the condition $x_{k+1} = \mathrm{argmin}_{x \in \mathcal{N}(x_k)} f(x)$ can be implemented in the poll step and that the condition $\|x_{k+1} - x_k\| \leq C\Delta_k$ is verified for some positive constant C if the choice of z in (2.1) is limited to a bounded set.

The results of Theorem 2.4 concern the ability of pattern search methods to converge globally (i.e. from arbitrary points) to local minimizers candidates. We recall, despite the inexistence of any supporting theory, that there is numerical evidence about the capability of pattern search methods to compute global minimizers (see the papers [12, 13, 22] and the numerical experiments reported in this paper).

3. Pattern search methods for user-provided points: Application to molecular geometry problems.

Having described pattern search methods in a general framework, we turn now to its application to the situation where one would like to take advantage of an user-provided points calculation, like the one we will describe in the context of molecular geometry problems. Our goal is to develop a class of pattern search methods especially tailored to these problems, where each optimization step is physically meaningful.

We accomplish our intention by identifying a set of geometrical transformations — the user-provided points — viewed as deformations of the molecular shape with physical meaning that may provide an energy lowering path. However, as we will see in Appendix A, these geometrical transformations are dependent on the data of the current configuration. In other words, they depend on each optimization point x_k — that stores the coordinates of the current configuration — and therefore they cannot themselves define a pattern and a mesh. (Asymptotically, the dependence would be on the sequence $\{x_k\}$, ruining the finiteness property of the pattern matrices.)

As we will see in Section 3.1, we then define a pattern with interesting uniform directionality properties to fit the geometrical transformation procedure, or any other user-provided points calculation.

A new trial point for the search step is computed by geometrical transformation followed by a computation that determines approximately the closest point in the patterned mesh to the point calculated by geometrical transformation.

The positive basis needed to define the mesh neighborhood in the poll step is identified after a point is computed again by a geometrical transformation: among all the vectors in the set of positive bases, the one that makes the smallest angle with the vector defined by the current point and the point computed by the geometrical transformation is identified. This vector in turn identifies the positive basis to be used in the poll step.

The search and poll steps of this new class of pattern search methods for user-provided points (e.g. geometrical transformations) are described in Section 3.2.

3.1. Positive bases with uniform angles. We start by introducing the pattern onto which geometrical transformations will be projected. Consider $n+1$ vectors $v_1, \ldots, v_{n+1}$ in $\mathbb{R}^n$ for which all the angles between pairs v_i, v_j $(i \neq j)$ have the same amplitude α. Assuming that the $n+1$ vectors are normalized, this requirement is expressed as

$$a \;=\; \cos(\alpha) \;= \langle v_i, v_j \rangle, \qquad i, j \in \{1, \ldots, n+1\},\; i \neq j, \tag{3.1}$$

where $a \neq 1$. One can show that $a = -1/n$: Let us assume, without loss of generality, that

$$v_{n+1} \;=\; \sum_{i=1}^{n} \alpha_i v_i, \tag{3.2}$$

for some scalars $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. From (3.1) and (3.2), we obtain:

$$1 = a \sum_{i=1}^{n} \alpha_i, \tag{3.3}$$

$$a = \sum_{i=1, i\neq j}^{n} a\alpha_i + \alpha_j, \qquad j = 1, \ldots, n. \tag{3.4}$$

Adding all the rows in (3.4) yields

$$na \;=\; (1 + (n-1)a) \sum_{i=1}^{n} \alpha_i. \tag{3.5}$$

From (3.3) and (3.5) we have that $na^2 + (1-n)a - 1 = 0$, and thus, since $a \neq 1$, we conclude that $a = -1/n$.

Now we seek a set of $n+1$ normalized vectors $[v_1 \cdots v_{n+1}]$ satisfying property (3.1) with $a = -1/n$. Let us first compute $v_1, \ldots, v_n$, i.e., let us compute a matrix $V = [v_1 \cdots v_n]$ such that

$$V^\top V \;=\; A,$$

where A is the matrix given by

$$A = \begin{bmatrix} 1 & -1/n & -1/n & \cdots & -1/n \\ -1/n & 1 & -1/n & \cdots & -1/n \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ -1/n & -1/n & -1/n & \cdots & 1 \end{bmatrix}.$$

The matrix A is symmetric and diagonally dominant with positive diagonal entries, and, therefore, it is positive definite [11]. Thus, we can make use of its eigenvalue decomposition

$$A \;=\; Q\Lambda Q^\top,$$

where $Q \in \mathbb{R}^{n\times n}$ is orthogonal and Λ is a diagonal matrix of order n with positive diagonal entries. Given this decomposition, one can easily see that a choice for V is determined by

$$V \;=\; [v_1 \cdots v_n] \;=\; Q\Lambda^{\frac{1}{2}}Q^\top. \tag{3.6}$$

The vector v_{n+1} is then computed by

$$v_{n+1} \;=\; -\sum_{i=1}^{n} v_i. \tag{3.7}$$

It is obvious that $\langle v_i, v_{n+1}\rangle = -1/n$, $i = 1,\ldots,n$, and $\langle v_{n+1}, v_{n+1}\rangle = 1$.

Since V is nonsingular and v_{n+1} is determined by (3.7), we can apply Corollary 2.1 to establish that $[v_1 \cdots v_{n+1}]$ is a (minimal) positive basis.

Our goal is now to generate, from the positive basis $[v_1 \cdots v_{n+1}]$ given by (3.6)-(3.7), a set of positive bases such that: (i) the overall set of vectors captures the directionality of $\mathbb{R}^n$ as well as possible; (ii) each element of the set is itself a positive basis satisfying the uniform angle property (3.1) with $a = -1/n$. First, let us consider a "rotation" $U[e_1 \cdots e_n] = [u_1 \cdots u_n]$ of the coordinate axes $[e_1 \cdots e_n]$ given by the *a priori* fixed orthogonal matrix $U = [u_1 \cdots u_n]$. The first positive basis is computed by $U_1[v_1 \cdots v_{n+1}]$, where U_1 is an orthogonal matrix that "rotates" v_1 into u_1:

$$U_1 v_1 \;=\; u_1.$$

A choice for U_1 is the Householder transformation

$$U_1 \;=\; I - \pi^{-1}uu^\top, \quad u \;=\; v_1 - u_1, \quad \pi \;=\; \frac{1}{2}\|u\|^2.$$

The i-th positive basis is then obtained by "rotating" v_1 into u_i. However, since $u_i = Ue_i$ and $e_i = U^\top u_i$, there is no need to compute another Householder transformation. In fact, we easily see that

$$U\mathcal{P}_{1i}U^\top U_1 v_1 \;=\; u_i,$$

where $\mathcal{P}_{1i}$ is the permutation matrix obtained from the identity by interchanging rows 1 and i. Thus the i-th positive basis is given by $U_i[v_1 \cdots v_{n+1}]$, where U_i is the orthogonal matrix

$$U_i = U\mathcal{P}_{1i}U^\top U_1.$$

The desired set of positive bases is given by these n positive bases and their corresponding symmetrical counterparts:

$$\mathcal{V} \;=\; [\,U_1[v_1 \cdots v_{n+1}] \cdots U_n[v_1 \cdots v_{n+1}] \;\; -U_1[v_1 \cdots v_{n+1}] \cdots -U_n[v_1 \cdots v_{n+1}]\,]. \tag{3.8}$$

The number of positive bases is therefore $m = 2n$.

The vectors in $\mathcal{V}$ are reasonably well distributed by amplitude in $\mathbb{R}^n$. In Fig. 3.1 we depict the mesh (2.1) with the choices of $\mathcal{V}$ and W respectively given by (3.8) and (2.2); the matrix U given above was set to the identity.

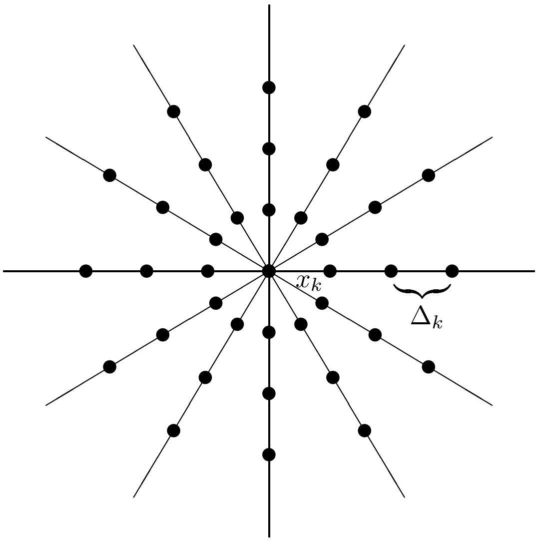

FIGURE 3.1. *Mesh for $n = 2$. There are 4 uniform positive bases.*

3.2. The new pattern search framework. Finally, we combine the procedure introduced in Section 3.1 with the technique described in Appendix A, and define our class of pattern search methods for molecular geometry problems. We describe how the computation of new points (by geometrical transformations) can determine a pattern search method using, for instance, the pattern described in Section 3.1. The same ideas can be used in any application where the user has a scheme to provide the calculation of new points (see also [1]).

The new search and poll steps are described in a parallel environment with N_p processors. We start by showing how the computation of a trial point x_{k+1}^{trial} can be carried out in the search step.

Search step: computation of x_{k+1}^{trial}

For each processor p in $\{1, \ldots, N_p\}$:
1. Compute a trial point $u_{p,k+1}^{gt}$ by a geometrical transformation.
2. Solve the integer programming problem

$$\min_{z \in W} \|u_{p,k+1}^{gt} - (x_k + \Delta_k \mathcal{V} z)\| \tag{3.9}$$

to determine a point $x_{p,k+1}^{gt}$, in M_k, closest to $u_{p,k+1}^{gt}$.
3. Set

$$x_{k+1}^{trial} \;=\; \text{argmin}_{x_{p,k+1}^{gt}} \, f(x_{p,k+1}^{gt}) \, .$$

Using the mesh (2.1) with the choices of $\mathcal{V}$ and W respectively given by (3.8) and (2.2), as we do in our implementation, the computation of $x_{p,k+1}^{gt}$ as the solution of the integer programming problem (3.9) can be carried out with relatively little computational effort (see also Fig. 3.2). In fact, it can be easily checked that the linear algebra cost is of the order of n^3, which for small n is relatively low compared

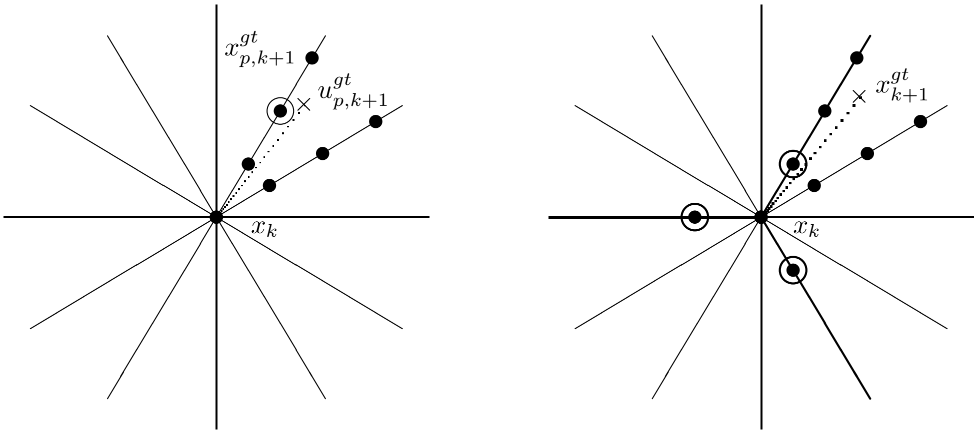

FIGURE 3.2. *Search step (left) and poll step (right).*

to the cost of expensive function evaluations such as the total energy by local density plane-waves (Appendix B).

In the poll step, the geometrical transformation technique defines the positive basis $V_k(x_k)$, that in turn defines the mesh neighborhood $\mathcal{N}(x_k)$. The procedure is described below and depicted in Fig. 3.2.

Poll step: choice of mesh neighborhood $\mathcal{N}(x_k)$

1. Compute one trial point x^{gt}_{k+1} by a geometrical transformation.

2. Determine v^{gt}_k in $\mathcal{V} = [\,U_1[v_1\cdots v_{n+1}]\cdots U_n[v_1\cdots v_{n+1}] \quad -U_1[v_1\cdots v_{n+1}]\cdots -U_n[v_1\cdots v_{n+1}]\,]$ such that

$$\frac{\langle x^{gt}_{k+1} - x_k,\, v^{gt}_k\rangle}{\|x^{gt}_{k+1} - x_k\|} \;=\; \max_{v\in\mathcal{V}} \frac{\langle x^{gt}_{k+1} - x_k,\, v\rangle}{\|x^{gt}_{k+1} - x_k\|}$$

3. Set $V_k(x_k)$ to the positive basis in $\mathcal{V}$ that contains v^{gt}_k, and then set $\mathcal{N}(x_k) = \{x_k + \Delta_k v : \text{ for all } v \in V_k(x_k)\}$.

Poll step: evaluation of f in the mesh neighborhood $\mathcal{N}(x_k)$

4. List the points in $\mathcal{N}(x_k)$ by increasing order of the values of the angles between $x^{gt}_{k+1} - x_k$ and the corresponding vectors in $V_k(x_k)$.

5. Following the list given above, divided in groups of N_p points, start evaluating in parallel the function f in $\mathcal{N}(x_k)$.
Stop if a point $x_{k+1} \in \mathcal{N}(x_k)$ is found such that $f(x_{k+1}) < f(x_k)$. In this case go to step **3**, expanding M_k (poll step and iteration are declared successful).
If $f(x_k) \leq f(x)$ for every x in the mesh neighborhood $\mathcal{N}(x_k)$, go to step **4**, shrinking M_k (poll step and iteration are declared unsuccessful).

Mesh expansions and reductions could also be designed to take advantage of problem information obtained from geometrical transformations.

4. Numerical experiments. In order to define a pattern search method for molecular geometry we need to be more specific about the geometrical transformations. The simplest geometrical transformations used in our calculations were the uniform expansions and compressions of the cluster in the plane perpendicular to the l-axis (Fig. 5.1(b)). These deformations correspond to putting $c_1 = c_2 = 0$ in (5.1) and setting $c_3 = 1.1$ for expansions and $c_3 = 0.9$ for compressions. For the linear stretches (Fig. 5.1(c)), c_3 was set to 1 and $c_2 = 0.1$ or $c_2 = -0.1$ (Fig. 5.1(c), top and bottom, respectively). The quadratic stretches were done using $c_3 = 1$, $c_2 = 0$, and $c_1 = 0.1$ (Fig. 5.1(d), top) or $c_1 = -0.1$ (Fig. 5.1(d), bottom). The last deformation considered was the torsion of the cluster around the l-axis. This torsion was accomplished rotating atom α around the l-axis by an angle $\theta = c_2\, r_l^{\alpha,k} + c_3$, with c_2 and c_3 chosen so that the topmost atom would be rotated by $\pi/8$ clockwise (Fig. 5.1(e), bottom) or counter-clockwise (Fig. 5.1(e), top).

All the values mentioned above for c_1, c_2, and c_3 were the values used in the poll step of our pattern search methods. The search step should be much more aggressive than the poll step as an attempt for global search. To try to accomplish this goal, the parameters used in the search step were the poll step parameters, scaled by a factor of 5.

A random rearrangement of the atoms was also considered at every iteration in an attempt to capture geometries very different from the current one. During the poll step, these rearrangements consisted in multiplying each coordinate of the atoms by a random value between 0.9 and 1.1, i.e., whenever a random deformation was performed, the $3N - 6$ coordinates of the cluster were scaled by a set of $3N - 6$ random values between 0.9 and 1.1. The random scaling factors used in the search step were between 0.5 and 1.5.

The mesh used in our implementation is defined by (2.1) with the choices of $\mathcal{V}$ and W respectively given by (3.8) and (2.2). The set of positive bases has $m = 2n$ uniform positive bases each with $n + 1$ vectors. To ensure that all deformations are tried in the search steps, the set of N_p deformations used is changed in a consistent way in consecutive search steps.

The stopping criterion used in our pattern search method followed the one implemented in PDS:

$$\sqrt{\frac{2(n-1)}{n}}\,\frac{\Delta_k}{\max\{1, \|x_k\|\}} \leq 10^{-2},$$

where $\sqrt{2(n-1)/n}\Delta_k$ is the length of the longest edge in the simplex defined in the current poll step by the corresponding uniform positive basis.

We applied our pattern search methods (PSM:MGP) to the minimization of the total energy of sodium clusters of dimension 4, 8, 16, and 32. The calculation of the total energy followed the process summarized in Appendix B. Results are given in Tables 4.1, 4.2, 4.3, and 4.4. We provide results for eight initial points, except for Na32, for which we present only two initial points. We list in these tables the number of iterations (iters), the number of total energy function evaluations (fevals), and the best value of the total energy found (f). The calculations were done in a cluster of 12 2.266 GHz Intel Pentium IV personal computers connected through a switched full-duplex 100 Mb/s ethernet network, running LINUX, and using the Message Passing Interface (MPI) as the parallelization protocol. We point out once again that we are dealing with expensive function evaluations: one evaluation of the total energy for the Na4 (resp. Na8, Na16, and Na32) cluster took on average 16 (resp. 59, 114, and

186) seconds of CPU time.

x_0	iters	fevals	f
Na4a	24	397	-1.689
Na4b	38	589	-1.697
Na4c	42	711	-1.685
Na4d	12	193	-1.698
Na4e	16	262	-1.696
Na4f	20	312	-1.697
Na4g	28	444	-1.682
Na4h	52	875	-1.694

Table 4.1

Numerical results obtained by PSM:MGP for Na4. The numbers of processors used was $N_p = 12$.

x_0	iters	fevals	f
Na8a	125	3385	-3.524
Na8b	105	2913	-3.522
Na8c	101	2748	-3.528
Na8d	73	1783	-3.467
Na8e	11	293	-3.504
Na8f	169	4558	-3.521
Na8g	108	2502	-3.489
Na8h	133	3305	-3.515

Table 4.2

Numerical results obtained by PSM:MGP for Na8. The numbers of processors used was $N_p = 12$.

x_0	iters	fevals	f
Na16a	260	11195	-7.119
Na16b	285	11740	-7.147
Na16c	283	12091	-7.135
Na16d	371	15783	-7.159
Na16e	239	10373	-7.136
Na16f	245	10268	-7.143
Na16g	255	10846	-7.122
Na16h	304	13134	-7.138

Table 4.3

Numerical results obtained by PSM:MGP for Na16. The numbers of processors used was $N_p = 12$.

These preliminary results show that the method PSM:MGP is able to find a configuration nearly optimal for a significant number of initial points. The optimal value is approximately -1.698 for the Na4 cluster and -3.534 for the Na8 cluster, but these values are only attained after applying a local optimization code. The Na16 and Na32 cluster geometries are not well-established.

x_0	iters	fevals	f
Na32a	462	33417	-14.558
Na32b	477	34509	-14.580

TABLE 4.4
Numerical results obtained by PSM:MGP for Na32. The numbers of processors used was $N_p = 12$.

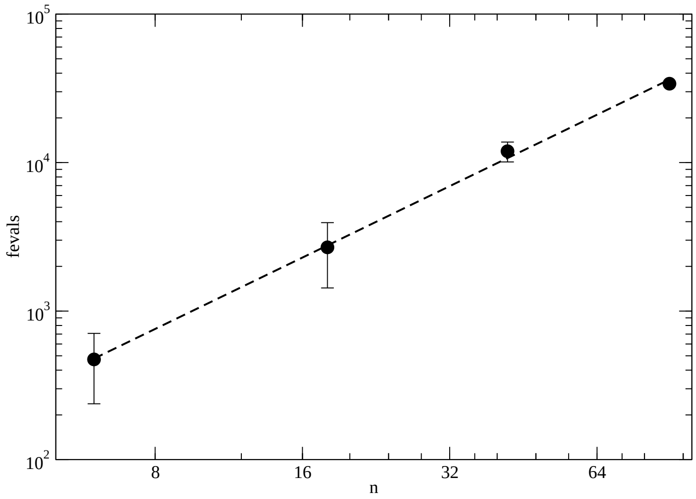

FIGURE 4.1. *Scaling of the average number of function evaluations (fevals) with the number of variables (n). Dashed line is a fit of a power law, fevals =* $A\mathrm{n}^B$*, to the data (*$A = 27.336$ *and* $B = 1.5977$*).*

Due to limited access to our cluster we are unable to provide the full results for Na32. We were only able to finish two of the runs, for two given initial configurations, which terminated with 33417 and 33509 function evaluations, respectively. We used this information, however, in a derivation of an estimate for the rate of growth in the number of (average) function evaluations in terms of the number of variables $n = 3N - 6$. The number of function evaluations (*fevals*) seems to grow with n under a rate slower than quadratic ($fevals \approx n^{1.6}$). A fit of $\log(fevals)$ to $A + B \log n$ has given $A = 27.336$ and $B = 1.5977$ and the least squares regression error was 0.008322 (see Fig. 4.1).

For the sake of comparison with other methods, we ran PSM:MGP and PDS for another set of initial configurations for Na4 and Na8. These calculations were done in a different computer system (a cluster of 24 DIGITAL/Compaq Alpha 500au Personal Workstations connected through a switched full-duplex 100 Mb/s ethernet network, running DIGITAL UNIX, and using the Message Passing Interface (MPI) as the parallelization protocol) and with different parameters for the plane-wave code (corresponding to a different version of the application code, slower due to the use of a more accurate model for the electron-ion interactions). As a result, the total energy values presented now are not comparable to the ones reported above. A simple comparison of the total energy values obtained with both methods shows that the comparison between PDS and PSM:MGP is mildly favorable to the latter, as we indicated in the last column of Tables 4.5 and 4.6 (in one instance the values of the objective function coincided and we used as a second criterion the number of function evaluations). Both PDS and PSM:MGP were able to find, for the Na8 cluster, the two best known local minimizers for different starting configurations.

We point out that the implementation of PSM:MGP used in these computations is far from being exhaustively tuned. We did not play with the code to try to come up with the best strategies (geometrical transformations, etc.) and with the best values for the different parameters. We expect that a method like PSM:MGP has plenty of room for improvement.

	PDS			PSM:MGP			
x_0	iters	fevals	f	iters	fevals	f	best
Na4i	26	324	-2.58191	28	533	-2.58251	PSM:MGP
Na4j	9	120	-2.58751	18	343	-2.58772	PSM:MGP
Na4k	6	84	-2.58920	4	77	-2.58868	PDS
Na4l	7	96	-2.58860	4	77	-2.58862	PSM:MGP
Na4m	25	312	-2.58098	32	609	-2.57487	PDS

TABLE 4.5

Numerical results obtained by PDS (with 12 pattern points) and PSM:MGP for Na4. The computation of the total energy was carried out differently than the one reported in Tables 1-4. The numbers of processors used was $N_p = 12$.

	PDS			PSM:MGP			
x_0	iters	fevals	f	iters	fevals	f	best
Na8i	31	1135	-5.25543	35	868	-5.23153	PDS
Na8j	5	199	-5.31805	3	88	-5.31805	PSM:MGP
Na8k	24	883	-5.30026	58	1546	-5.31268	PSM:MGP
Na8l	3	118	-1.27962	3	88	-5.31022	PSM:MGP

TABLE 4.6

Numerical results obtained by PDS (with 36 pattern points) and PSM:MGP for Na8. The computation of the total energy was carried out differently than the one reported in Tables 1-4. The numbers of processors used was $N_p = 10$.

5. Conclusions and future work.

We designed a class of pattern search methods for molecular geometry by taking advantage of physically meaningful energy lowering geometrical transformations, and by combining them with appropriate patterns for minimization purposes. The preliminary numerical results obtained with a particular pattern search method in the class have indicated that this approach could lead to very promising algorithms for molecular geometry. We hope to obtain better numerical results by considering more elaborate search steps. In fact, our approach has the flexibility to incorporate several types of global optimization algorithms in the search step to enhance the selection of the geometrical transformations and their defining values. We have in mind, for instance, the use of evolutionary algorithms like evolutionary programming or evolutionary strategies.

The new pattern search methods can be used in any other context where there is a user-provided scheme to generate points leading to potential objective function decrease.

We plan to apply our pattern search methods to the total energy minimization of other clusters and to develop analogues of this approach in other molecular geometry contexts. We also plan to investigate patterns with similar interesting uniform directionality properties.

Appendix A. The current point x_k in the optimization process stores the atomic positions $r_i^{\alpha,k}$ of a set of N atoms, where k denotes the iteration counter and r_i^α is the i-th coordinate of atom α ($i = 1, 2, 3$). The set of atomic positions specifies not only the shape of the system of atoms but also its location and orientation in space. Since shapes that result from translations or rotations about a fixed point have the same energy, there are six redundant coordinates in a molecular geometry optimization process. Three of these refer to the location of the set of atoms with a certain shape in space and the other three are the angles that define the orientation of this set with respect to some fixed three-dimensional reference frame. The easiest way to get rid of these additional degrees of freedom is to fix one of the atoms at the origin of a three-dimensional reference frame, to keep another atom on one of the axis of this frame (the x-axis, for example), and to force a third atom to move only on a plane containing the above mentioned axis (the xy-plane, for example). These restrictions do not introduce constraints in shape space, they merely exclude atomic configurations representing the same system translated and/or rotated in space. Without loss of generality, the three constrained atoms are chosen to be atoms N, $N-1$, and $N-2$. The vector x_k can then be related to the atomic positions, $r_i^{\alpha,k}$, in the following way:

$$x_k = \begin{bmatrix} r_1^{1,k} \\ r_2^{1,k} \\ r_3^{1,k} \\ r_1^{2,k} \\ \vdots \\ r_3^{N-3,k} \\ r_1^{N-2,k} \\ r_2^{N-2,k} \\ r_1^{N-1,k} \end{bmatrix}.$$

The corresponding "constraints" are:

$$r_3^{N-2,k} = r_2^{N-1,k} = r_3^{N-1,k} = r_i^{N,k} = 0, \ i = 1, 2, 3.$$

An optimization step, $x_k \to x_{k+1}$, can be viewed as a deformation of the molecular shape described by x_k. This deformation may not have any physical meaning, corresponding simply to a random rearrangement of the atoms. The space spanned by an algorithm where only this type of moves is present is unrelated to shape space, i.e., a given path in this space is not related in a simple way to a shape space path, a path where the molecule undergoes a recognizable shape transformation. Physically meaningful deformations (as, for example, a simple uniform compression or expansion of the molecule), i.e., paths in shape space, are expected to be closer to (total energy) downhill directions than simple paths in x_k-space. In fact, a path in shape space will in general correspond to a non-trivial path in x_k-space that can even connect very distant x_k-space points.

A simple way to introduce physically meaningful and energy lowering deformations of a given molecule or cluster is to consider just stretches and twists along some direction. An obvious choice for the directions along which the cluster is to be stretched or twisted is its principal axes system[3]. In order to deform the molecule in this way it is necessary to refer the atomic positions to the principal axes system:

$$\bar{r}_i^{\alpha,k} = \sum_{j=1}^{3} \mathcal{R}_{ij}^{(k)} r_j^{\alpha,k},$$

where $\mathcal{R}^{(k)}$ is the rotation to the principal axes system. The deformations of the molecule can then be written as

$$\bar{r}_i^{\alpha,k+1} = \sum_{j=1}^{3} \epsilon_{ij}^{\alpha,k} \bar{r}_j^{\alpha,k}$$

or, returning to the non-principal axes system, as

$$r_i^{\alpha,k+1} = \sum_{j,l,m=1}^{3} \mathcal{R}_{ij}^{(k)-1} \epsilon_{jl}^{\alpha,k} \mathcal{R}_{lm}^{(k)} r_m^{\alpha,k}.$$

Alternatively, using matrix notation, we can write

$$r^{\alpha,k+1} = \mathcal{R}^{(k)-1} \epsilon^{\alpha,k} \mathcal{R}^{(k)}\, r^{\alpha,k}.$$

The form for the deformations assumed above is very broad. Some simple and physically meaningful particular forms can be written simply as:

$$\epsilon_{ij}^{\alpha,k} = \left[\left(c_1 \left(\bar{r}_l^{\alpha,k} \right)^2 + c_2\, \bar{r}_l^{\alpha,k} + c_3 - 1 \right) (1 - \delta_{jl}) + 1 \right] \delta_{ij} \tag{5.1}$$

where $l \in \{1, 2, 3\}$ is the label of the principal axes about which the transformations are made. The effect of these transformations is simply to put the atoms closer or farther from the principal axis l (see Figs. 5.1(a)–5.1(d)).

Another physically meaningful deformation that can be considered is a torsion of the molecule around some axis (Fig. 5.1(e)), in a way that forces different parts of the molecule to be rotated around that axis with different angles:

$$\epsilon^{\alpha,k} = \mathcal{R}_{\hat{e}_l}(\theta(\bar{r}_l^{\alpha,k})). \tag{5.2}$$

The axis $l \in \{1, 2, 3\}$ is the torsion axis and $\mathcal{R}_{\hat{e}_l}(\theta)$ is a rotation by an angle θ around that axis. For example:

$$\mathcal{R}_{\hat{e}_3}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

---

[3] The principal or inertial axes system of a given molecule is the set of eigenvectors of the matrix

$$\mathcal{I}_{ij} = \sum_{\alpha=1}^{N} m_\alpha \left(\|r^\alpha\|^2 \delta_{ij} - r_i^\alpha r_j^\alpha \right),$$

where δ_{ij} is the Kronecker tensor and m_α is the mass of atom α. For convenience, we choose a reference frame whose origin is the center of mass of the molecule, i.e., where the atomic coordinates satisfy the relation $\sum_{\alpha=1}^{N} m_\alpha r^\alpha = 0$.

The angle θ must be a function of the l-coordinate of each atom (a quadratic function of $r_l^{\alpha,k}$, for example). Contrary to the previous forms, this type of deformation can break any axial symmetry that the molecule at iteration k might possess.

As we said before, the geometrical transformations (5.1) are performed with the center of mass of the cluster at the origin of the $\bar{r}^{\alpha,k}$ coordinates. Thus, an atom sitting on the plane containing the center of mass and perpendicular to the l-axis of this system of coordinates — in turn of which these geometrical transformations are performed — would remain unaffected by most deformations (see Figs. 5.1(c)–5.1(e)). The exceptions are the uniform expansions and compressions (see Fig. 5.1(b)).

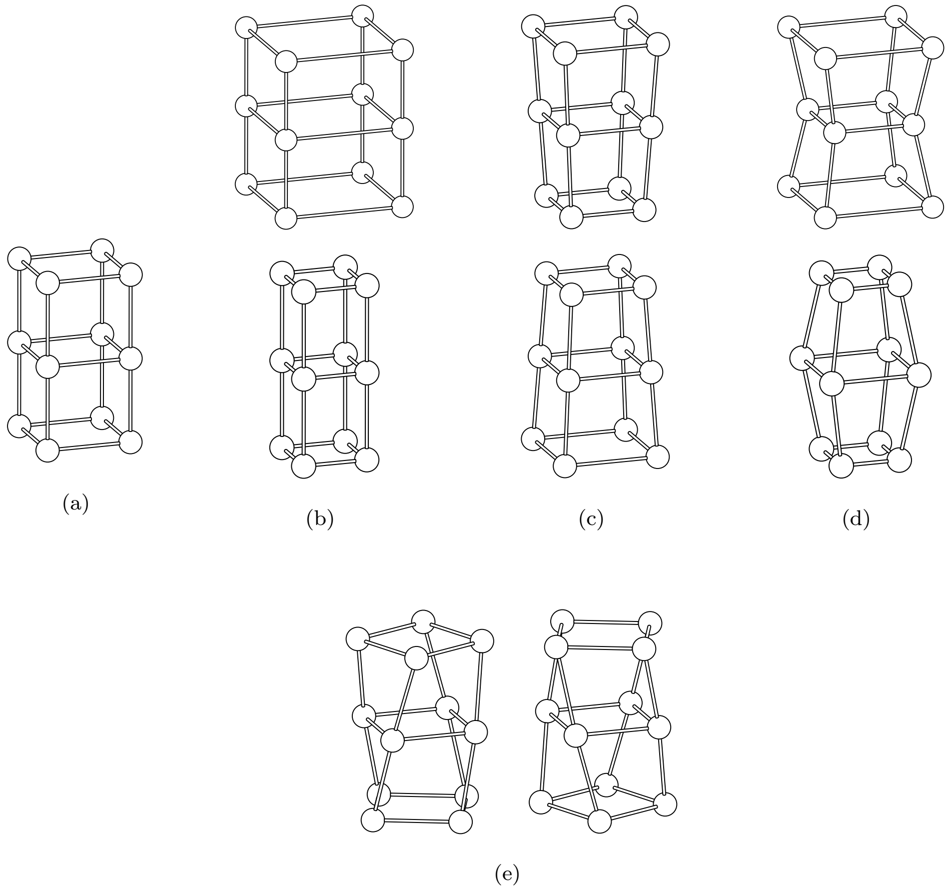

FIGURE 5.1. *Simple example of the deformations (5.1) and (5.2). In Fig. 5.1(b), the reference molecule (Fig. 5.1(a)) is expanded sideways, which corresponds to setting $c_1 = c_2 = 0$, $c_3 \neq 0$ in (5.1). The parameter c_3 can be greater (top) or lower (bottom) than 1. Putting $c_2 \neq 0$ results in deformations similar to Fig. 5.1(c) (with c_2 positive — top — or negative — bottom), while the use of a full quadratic form gives rise to deformations like Fig. 5.1(d) (with c_1 positive — top — or negative — bottom). (In this example, $\theta = c_2\, r_l^{\alpha,k} + c_3$, with c_2 positive — top — or negative — bottom.) Fig. 5.1(e) is an example of the deformations that can be achieved with (5.2). In all these examples, the l-axis is the vertical axis.*

A set of new coordinates $r_i^{\alpha,k+1}$ computed by geometrical transformation from the previous coordinates $r_i^{\alpha,k}$ (stored in x_k) can then be used as a trial point x_{k+1}^{gt} for the search and poll steps of the $k+1$ pattern search iteration.

Appendix B. We will provide a brief description of the main issues in local density plane-wave total energy calculation [18]. The Hamiltonian $\mathbf{H}$ of an N-electron system with M nuclei of charge Z_I and mass m_I can be written as

$$\mathbf{H}(r_1,\ldots,r_N,R_1,\ldots,R_M)=\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{2}+\sum_{I=1}^{M}\frac{\mathbf{P}_I^2}{2m_I}-\sum_{i,I=1}^{N,M}\frac{Z_I}{|r_i-R_I|}+ \\ +\sum_{\substack{i,j=1\\ i<j}}^{N}\frac{1}{|r_i-r_j|}+\sum_{\substack{I,J=1\\ I<J}}^{M}\frac{Z_IZ_J}{|R_I-R_J|},$$

where r_i and R_I are the coordinates of the electrons and of the atomic nuclei, and $\mathbf{p}_i$ and $\mathbf{P}_I$ are their linear momenta. (Spin was not considered for simplicity; atomic units are used throughout the calculations.)

By solving the time-independent Schrödinger equation

$$\mathbf{H}\Psi=E\Psi,$$

one obtains the set of eigenvalues (energies, E) and eigenvectors (wavefunctions, Ψ) of the system. This equation gives a good description of non-relativistic many-electron systems subject to electric fields produced by atomic nuclei, like atoms, molecules, and solids. However, this equation is in general unsolvable. If the mass difference between electrons and nuclei is taken into account [4], the time-independent Schrödinger equation can be separated in two equations: one for the electrons

$$\left(\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{2}-\sum_{i,I=1}^{N,M}\frac{Z_I}{|r_i-R_I|}+\sum_{\substack{i,j=1\\ i<j}}^{N}\frac{1}{|r_i-r_j|}+E_{\mathrm{nn}}\right)\Psi(r_1,\ldots,r_N;R_1,\ldots,R_M)= \\ =E(R_1,\ldots,R_M)\ \Psi(r_1,\ldots,r_N;R_1,\ldots,R_M), \qquad (5.3)$$

where

$$E_{\mathrm{nn}}=\sum_{\substack{I,J=1\\ I<J}}^{M}\frac{Z_IZ_J}{|R_I-R_J|},$$

and another for the nuclei, of no interest in this context.

In (5.3), the nuclear coordinates R_i are just parameters, and the electronic wavefunctions and eigenvalues are different for each arrangement of nuclei. In order to find the lowest energy state of the system (the ground state), one can solve (5.3) for a given set of nuclear coordinates, and assume that $E(R_1,\ldots,R_M)$ is a function of the nuclear coordinates to be subsequently minimized.

Hohenberg and Kohn [15] proved a theorem that legitimizes the use of the electronic density

$$\rho(r)=N\int|\Psi(r,r_2,\ldots,r_N)|^2\,dr_2\ldots dr_N,$$

as fundamental variable, instead of the wavefunction $\Psi(r_1, \ldots, r_N)$. The theorem states that any observable (e.g., the energy) is a functional of the ground state density. In particular, the ground state energy functional of an N-electron system in an external potential $v_{\text{ext}}(r)$ (representing the interaction of the nuclei with the electrons, for example) can be written as

$$E_{v_{\text{ext}}}[\rho] = F_{\text{HK}}[\rho] + \int \rho(r) v_{\text{ext}}(r) dr + E_{\text{nn}},$$

where $F_{\text{HK}}[\rho]$ is an universal functional, i.e., a functional that does not depend on the external potential. Therefore, $F_{\text{HK}}[\rho]$ is the same for atoms, molecules, and solids. The ground state is obtained through the variational principle:

$$E_* = \min_{\{\rho\}} \; E_{v_{\text{ext}}}[\rho], \tag{5.4}$$

and the variational search is performed over all the admissible electronic densities.

A good approximation to the functional $F_{\text{HK}}[\rho]$ was suggested by Kohn and Sham [19]. Their main hypothesis is that, for each interacting ground state density $\rho(r)$, there exists a non-interacting electron system with the same ground state density. The Kohn-Sham $F_{\text{HK}}[\rho(r)]$ functional is

$$F_{\text{HK}}[\rho(r)] = -\frac{1}{2} \sum_{i=1}^{N} \int \phi_i^*(r) \nabla^2 \phi_i(r) dr + \frac{1}{2} \int \frac{\rho(r_1)\rho(r_2)}{|r_1 - r_2|} dr_1 dr_2 + E_{xc}[\rho(r)],$$

with

$$\sum_{i=1}^{N} |\phi_i(r)|^2 = \rho(r). \tag{5.5}$$

$E_{xc}[\rho(r)]$ is the so-called exchange and correlation functional, for which many approximations exist [7, 24, 25].

The ground state is obtained solving the Euler-Lagrange equation that results from the minimization (5.4):

$$\left[-\frac{1}{2} \nabla^2 + v_{\text{ext}}(r) + \int \frac{\rho(r)}{|r - r'|} dr' + \frac{\delta E_{xc}[\rho(r)]}{\delta \rho(r)} \right] \phi_i(r) = \epsilon_i \phi_i(r), \tag{5.6}$$

and the total energy of the system is therefore:

$$E_{\text{KS}}[\rho(r)] = -\frac{1}{2} \sum_{i=1}^{N} \int \phi_i^*(r) \nabla^2 \phi_i(r) dr + \frac{1}{2} \int \frac{\rho(r_1)\rho(r_2)}{|r_1 - r_2|} dr_1 dr_2 + \\ + E_{xc}[\rho(r)] + \int \rho(r) v_{\text{ext}}(r) dr + E_{\text{nn}}.$$

The coupled nonlinear equations (5.5)-(5.6) are the so-called Kohn-Sham equations.

To calculate the total energy of solids, a plane-wave expansion of the Kohn-Sham wavefunctions is very useful, as it takes advantage of the translation symmetry of the crystal [17, 18, 26]. For finite systems, such as atoms, molecules, and clusters, plane waves can also be used in a supercell approach. In the supercell method, the finite system is placed in a unit cell of a fictitious crystal, and this cell is made large enough to avoid interactions between neighboring cells. However, for finite systems a very large number of plane waves is needed as the electronic density spans only a small fraction of the total volume of the supercell. The plane-wave expansion of the wavefunctions amounts simply to Fourier transforming them and all the other quantities involved in the Kohn-Sham equations, thereby converting the differential equation (5.6) into a matrix diagonalization problem. For finite systems, as many plane waves are needed, this matrix is very large, of the order of hundreds for small clusters.

But even for extended systems, many plane waves may be needed. The valence wavefunctions of the large Z_I atoms oscillate strongly in the vicinity of the atomic core, due to the orthogonalization to the inner electronic wavefunctions. To describe these oscillations a large number of plane waves is required, making even more difficult the calculation of the total energy. But the inner electrons are almost inert and are not significantly involved in bonding. This makes possible the description of an atom based solely on its valence electrons, which feel an effective potential that includes both the nuclear attraction and the repulsion of the inner electrons. This technique is the so-called pseudopotential approximation. In this work, we used the Troullier-Martins pseudopotential [31].

Although the pseudopotential approximation reduces its computational burden, the calculation of the total energy of a given system in the manner outlined above is still a very demanding task. One can deal with systems with at most a few hundred atoms. There are other methods that are significantly faster, allowing the calculation of the total energy of systems consisting of thousands of atoms. But these methods are much less accurate than the density functional method presented above. There are also some methods more accurate than this one, but they are significantly harder, prohibiting the simulation of systems with more than a few atoms.

Acknowledgments. We would like to thank Charles Audet and John Dennis for their insightful comments. We would also like to thank one referee and the associate editor for their suggestions that led to several improvements.

REFERENCES

[1] P. Alberto, F. Nogueira, H. Rocha, and L. N. Vicente. Pattern search methods for user-provided points. In V. N. Alexandrov et al., editor, *Computational Science – ICCS 2001*, volume 2074 of *Lecture Notes in Computer Science*, pages 95–98. Springer–Verlag, Berlin, 2001.

[2] C. Audet and J. E. Dennis. Pattern search algorithms for mixed variable programming. *SIAM J. Optim.*, 11:573–594, 2001.

[3] R. Biswas and D. R. Hamann. Simulated annealing of silicon atom clusters in Langevin molecular dynamics. *Phys. Rev. B*, 34:895–901, 1986.

[4] M. Born and K. Huang. *Dynamical Theory of Crystal Lattices.* Clarendon Press, Oxford, 1954.

[5] G. E. P. Box. Evolutionary operation: A method for increasing industrial productivity. *Appl. Statist.*, 6:81–101, 1957.

[6] R. Car and M. Parrinello. Unified approach for molecular dynamics and density-functional techniques. *Physical Review Letters*, 55:2471–2474, 1985.

[7] D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45:566–569, 1980.

[8] A. R. Conn, K. Scheinberg, and Ph. L. Toint. Recent progress in unconstrained nonlinear optimization without derivatives. *Math. Programming*, 79:397–414, 1997.

[9] C. Davis. Theory of positive linear dependence. *Amer. J. Math.*, 76:733–746, 1954.

[10] J. E. Dennis and V. Torczon. Direct search methods on parallel machines. *SIAM J. Optim.*, 1:448–474, 1991.

[11] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore, Maryland, third edition, 1996.

[12] W. E. Hart. A generalized stationary point convergence theory for evolutionary algorithms. In *Proc. 7th Intl. Conf. on Genetic Algorithms, San Francisco*, pages 127–134, 1997.

[13] W. E. Hart. Comparing evolutionary programs and evolutionary pattern search algorithms: A drug docking application. In *Proc. Genetic and Evolutionary Computation Conf.*, pages 855–862, 1999.

[14] M. R. Hoare and J. A. McInnes. Morphology and statistical statics of simple microclusters. *Adv. Phys.*, 32:791–821, 1983.

[15] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev. B*, 136:864–871, 1964.

[16] R. Hooke and T. A. Jeeves. “Direct search” solution of numerical and statistical problems. *J. ACM*, 8:212–229, 1961.

[17] J. Ihm. Total energy calculations in solid state physics. *Rep. Prog. Phys.*, 51:105–142, 1988.

[18] J. Ihm, A. Zunger, and M. L. Cohen. Momentum-space formalism for the total energy of solids. *J. Phys. C: Solid State Phys.*, 12:4409–4422, 1979. (E) *ibid.* 13 (1980) 3095.

[19] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev. A*, 140:1133–1138, 1965.

[20] R. M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. Technical Report TR96-71, ICASE, NASA Langley Research Center, 1999.

[21] R. M. Lewis, V. Torczon, and M. W. Trosset. Why pattern search works. *OPTIMA, MPS Newsletter*, (59):1–7, 1998.

[22] J. C. Meza and M. L. Martinez. On the use of direct search methods for the molecular conformation problem. *Journal of Computational Chemistry*, 15:627–632, 1994.

[23] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, 7:308–313, 1965.

[24] J. P. Perdew and Y. Wang. Accurate and simple analytic representation of the electron-gas correlation energy. *Phys. Rev. B*, 45:13244–13249, 1992.

[25] J. P. Perdew and A. Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23:5048–5079, 1981.

[26] W. E. Pickett. Pseudopotential methods in condensed matter applications. *Comp. Phys. Rep.*, 9:115–198, 1989.

[27] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numer.*, 7:287–336, 1998.

[28] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM J. Optim.*, 1:123–145, 1991.

[29] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optim.*, 7:1–25, 1997.

[30] M. W. Trosset. I know it when I see it: Toward a definition of direct search methods. *SIAG/OPT Views-and-News*, (9):7–10, 1997.

[31] N. Troullier and J. L. Martins. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B*, 43:1993–2006, 1991.

[32] M. H. Wright. Direct search methods: Once scorned, now respectable. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1995, Proceedings of the 1995 Dundee Biennal Conference in Numerical Analysis*, pages 191–208. Pitman Res. Notes Math. Ser. 344, CRC Press, Boca Raton, Florida, 1996.

[33] W. Zhong, G. Overney, and D. Tománek. Structural properties of Fe crystals. *Phys. Rev. B*, 47:95–99, 1993.