Journal of Computational Physics 228 (2009) 4858–4880

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Efficient algorithms for discrete lattice calculations

M. Arndt [a], V. Sorkin [b], E.B. Tadmor [b,*]

[a] School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA
[b] Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 29 September 2008
Received in revised form 25 March 2009
Accepted 27 March 2009
Available online 7 April 2009

PACS:
02.60.Cb
61.50.Ah

Keywords:
Lattice algorithms
Lattice reduction
Nearest neighbor
Cluster computation
Quasicontinuum

ABSTRACT

We discuss algorithms for lattice-based computations, in particular lattice reduction, the detection of nearest neighbors, and the computation of clusters of nearest neighbors. We focus on algorithms that are most efficient for low spatial dimensions (typically $d = 2, 3$) and input data within a reasonably limited range. This makes them most useful for physically oriented numerical simulations, for example of crystalline solids. Different solution strategies are discussed, formulated as algorithms, and numerically evaluated.

1. Introduction

Scientific computing often involves lattices and algorithms operating on them. For example, the atoms in a crystalline solid are arranged in the form of a lattice, and numerical codes that simulate the behavior of such crystals need to perform many operations on this lattice. This includes the identification of all atoms within a given radius of a lattice site for energy and force calculations and the location of the nearest lattice site to an arbitrary point. In particular, the current work was motivated by lattice algorithms needed for the quasicontinuum (QC) method [13–15]. Lattices are also used in other physical applications, such as the Ising model [9], lattice Monte Carlo [7], lattice protein folding algorithms [16], as well as in more abstract settings, such as integer linear programming problems in operations research, lattice-based cryptography and communication theory [1].

Lattice algorithms and their computational complexity have been intensively studied from the algebraic point of view. This resulted in the development of highly-sophisticated algorithms that have good scaling properties as the spatial dimension d gets large or the lattice gets highly distorted. For an overview of such lattice algorithm, see, *e.g.*, [1,6]. A more general classical reference for lattices and their properties is [2]. However, these sophisticated algorithms are not always optimal for many applications that are oriented more towards physics or engineering, where the spatial dimension d is typically low (mostly $d = 2, 3$), and lattice distortions are typically limited to a physically-relevant range so that the scaling properties

* Corresponding author. Tel.: +1 612 625 6642.
E-mail address: tadmor@aem.umn.edu (E.B. Tadmor).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.03.039

of the algorithms do not come into play. In these cases, simpler approaches that are significantly easier to implement can be superior to the more sophisticated techniques.

This paper describes lattice algorithms that are tailored to practical physical applications in numerical simulation and evaluates their performance on a set of test problems. The authors feel that there is a gap between the mathematically complex algorithms mentioned above and the naive "brute-force" approaches that are often used in practical applications. The purpose of this paper is to fill this gap for certain common lattice problems and to make algorithms readily-available for application.

We deal with three problems: lattice reduction, detection of nearest lattice sites, and computation of clusters of nearest neighbors. As noted above, this study was motivated by the QC method, nevertheless, these problems have been formulated in a general way to make them applicable and useful for a large variety of applications. We consider both simple lattices and multilattices, also called "lattices with a basis", which correspond to a set of inter-penetrating simple lattices.

A key starting point for many algorithms is a suitable lattice reduction that determines an optimal set of lattice vectors that can considerably accelerate subsequent operations. In Section 2, we discuss two variants of lattice reduction: the classical LLL reduction [8] that provides approximate, globally-optimized lattice vectors and a pairwise reduction approach that results in lattice vectors that are pairwise optimal but not necessarily globally optimal. The advantages and disadvantages of these approaches are discussed. Numerical studies of their performance appear in subsequent sections where they are incorporated into other algorithms.

Section 3 deals with the detection of nearest lattice sites, also known as the closest vector problem, for both simple lattices and multilattices. We discuss two strategies: a naive brute-force algorithm and a new approach which we refer to as the *short-list algorithm*. The latter is based on investing some computation time in advance to determine a small set of candidate lattice sites that is subsequently used to speed up the actual process of neighbor detection. This is advantageous when multiple lattice sites need to be detected for the same lattice structure. Both algorithms are evaluated numerically.

In Section 4, we discuss the computation of clusters of nearest neighbors, *i.e.*, determination of the set of all lattice sites within a given radius of a specified lattice site for both simple lattices and multilattices. This can be seen as a variation of the closest vector problem in which only the single nearest lattice site is detected. We discuss and numerically assess the performance of two strategies that we developed: the *shell algorithm* and the *on-the-fly algorithm*.

Concluding remarks are given in Section 5. The appendix contains proofs and technical details omitted from the main text in order not to interrupt the flow of reading.

2. Lattice reduction

A *simple lattice* $\mathcal{L}$ in d-dimensional space is an infinite discrete set of points that are integer linear combinations of lattice vectors, $\boldsymbol{a}_i \in \mathbb{R}^d$, for $i = 1, \ldots, d$,[1]

$$\mathcal{L} = \left\{ \sum_{i=1}^{d} \boldsymbol{a}_i n_i : n_i \in \mathbb{Z} \right\} = \{\boldsymbol{A}\boldsymbol{n} : \boldsymbol{n} \in \mathbb{Z}^d\}. \tag{1}$$

In the second term, we subsume the lattice vectors as the column vectors of the matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, *i.e.*, $\boldsymbol{A} = (\boldsymbol{a}_1 \cdots \boldsymbol{a}_d)$. To avoid degenerate lattices, we require the lattice vectors $\boldsymbol{a}_i$ to be linearly independent, or equivalently the matrix $\boldsymbol{A}$ to be invertible.

The lattice definition in (1) is not unique because the lattice vectors are not uniquely determined. Two lattices spanned by $\boldsymbol{A}$ and $\bar{\boldsymbol{A}}$ coincide,

$$\{\boldsymbol{A}\boldsymbol{n} : \boldsymbol{n} \in \mathbb{Z}^d\} = \{\bar{\boldsymbol{A}}\boldsymbol{n} : \boldsymbol{n} \in \mathbb{Z}^d\}, \tag{2}$$

if and only if the matrix $\boldsymbol{M} = \boldsymbol{A}^{-1}\bar{\boldsymbol{A}}$ is *unimodular*, *i.e.*, $\boldsymbol{M}$ and its inverse $\boldsymbol{M}^{-1} = \bar{\boldsymbol{A}}^{-1}\boldsymbol{A}$ are integer matrices, or equivalently if $\boldsymbol{M}$ is an integer matrix with determinant ± 1, *i.e.*, $|\det \boldsymbol{M}| = 1$.

As an example, consider the sheared square lattice spanned by

$$\boldsymbol{a}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{a}_2 = \begin{bmatrix} c \\ 1 \end{bmatrix} \tag{3}$$

for some $c \in \mathbb{R}$. The same lattice is also spanned by

$$\bar{\boldsymbol{a}}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \bar{\boldsymbol{a}}_2 = \begin{bmatrix} 2+c \\ 1 \end{bmatrix}, \tag{4}$$

as shown in Fig. 1. We have

$$\boldsymbol{A} = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix}, \quad \bar{\boldsymbol{A}} = \begin{bmatrix} 1 & 2+c \\ 0 & 1 \end{bmatrix}. \tag{5}$$

[1] $\{\boldsymbol{A}\boldsymbol{n} : \boldsymbol{n} \in \mathbb{Z}^d\}$ denotes the set of all vectors of the form $\boldsymbol{A}\boldsymbol{n}$ that is restricted by the colon notation to all vectors $\boldsymbol{A}\boldsymbol{n}$ for which $\boldsymbol{n}$ is an integer vector, $\boldsymbol{n} \in \mathbb{Z}^d$. A similar notation is used throughout the paper. Note that bold uppercase letters denote matrices, and that bold lowercase letters denote vectors.