Optimal meshes for integrals in real- and reciprocal-space unit cells

Juana Moreno* and José M. Soler†

Departamento de Física de la Materia Condensada, Universidad Autónoma de Madrid, E-28049 Madrid, Spain

(Received 18 February 1992)

We present a detailed method to construct uniform meshes for fast Fourier transforms in electronic-structure calculations. We show that a drastic reduction in the mesh size can be frequently achieved by using an unconventional set of primitive lattice vectors. The same method can be applied also to obtain optimum sets of special points for Brillouin-zone integrals, generalizing previous schemes.

I. INTRODUCTION

The fast Fourier transformation (FFT) is a basic tool in most methods of electronic-structure calculation. They are amply used when plane-wave basis sets are employed and, more generally, in the solution of Poisson's equation. They are especially important in the iterative solution of Schrödinger's equation and for *ab initio* molecular dynamics.[1] In many cases, the time spent doing FFT's represents a very substantial fraction of the total computation, and this fraction generally increases when the codes are highly optimized and vectorized. Therefore, an important practical issue is the minimization of this time, which increases as $N \ln N$ with the number N of mesh points used to represent the function transformed. The first subject of this paper is the construction of optimum meshes, with minimum N,[2] compatible with the lattice periodicity and symmetry and with the requirements of FFT algorithms.

An equally important issue is the choice of "special points" for Brillouin-zone (BZ) integrations of electron density, total energy, and many other quantities.[3-8] These special points are representative of a uniform mesh of points in the BZ, completely similar to that used in the real-space unit cell to perform FFT's. Therefore, we will see that our method can be equally used to generate sets of special **k** points, generalizing and optimizing previous schemes in a systematic way.

II. OPTIMAL MESH FOR FFT

Let $\mathbf{A}_i$, $i=1,2,3$ be a set of primitive lattice vectors. A uniform mesh of points **p** will be defined by a similar set of three (smaller) vectors $\mathbf{a}_i$ such that all mesh points can be written as

$$\mathbf{p}=\sum_{i=1}^{3} n_i \mathbf{a}_i \tag{1}$$

with n_i integer numbers. For the mesh to be acceptable, it must be commensurate with the lattice, i.e., the lattice vectors $\mathbf{A}_i$ must be mesh vectors. This implies that there is some integer matrix N_{ij} such that

$$\mathbf{A}_j=\sum_{i=1}^{3} \mathbf{a}_i N_{ij} \ . \tag{2}$$

Generally, the mesh is also required to possess all the symmetry operations of the lattice space group (including nonsymmorphic translations). Furthermore, in order to be used for FFT's, matrix N_{ij} must be *diagonal* $N_{ij}=N_i\delta_{ij}$. The standard way of constructing the mesh is then to make $\mathbf{A}_i$ the conventional primitive lattice vectors and

$$\mathbf{a}_i=\mathbf{A}_i/N_i \ , \tag{3}$$

with N_i some suitable integers. Roughly speaking, one wants the mesh points to be as evenly distributed as possible. Figure 1 shows a two-dimensional (2D) rectangular unit cell with two possible mesh partitions. In both cases the number of mesh points is the same but the points in mesh (b) are obviously more evenly distributed than those in mesh (a). It can be seen that mesh (a) allows the description of functions with very-short-wavelength oscillations in some directions but with only very long wavelengths in other directions, while mesh (b) permits more or less equal wavelengths in all directions. Since in general there is no *a priori* reason to expect shorter-wavelength oscillations in any particular direction, mesh (b) is generally preferable.

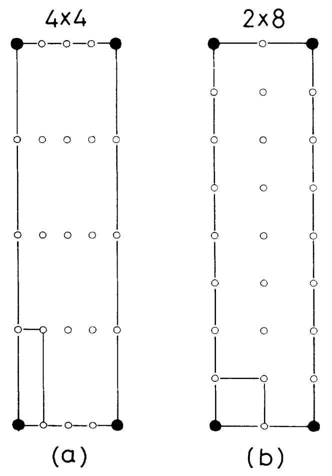

FIG. 1. Two-dimensional rectangular lattice (larger dots) and two possible meshes (a) and (b) (smaller dots). The lines indicate the unit cells of the lattice and of the mesh.

The above concept of "even distribution" of the mesh points can be made more precise with the use of the reciprocal lattice. Let $\mathbf{B}_i$ and $\mathbf{b}_i$ be the reciprocal vectors of $\mathbf{A}_i$ and $\mathbf{a}_i$: $\mathbf{A}_i\mathbf{B}_j=2\pi\delta_{ij}$, $\mathbf{a}_i\mathbf{b}_j=2\pi\delta_{ij}$. Vectors $\mathbf{b}_i$ define a *supercell* of $\mathbf{B}_i$:

$$\mathbf{b}_i=\sum_{j=1}^{3} N_{ij}\mathbf{B}_j \ , \tag{4}$$

Now, any function $f(\mathbf{r})$ with the lattice periodicity can be expanded in plane waves,

$$f(\mathbf{r})=\sum_{\mathbf{G}} f_{\mathbf{G}}e^{i\mathbf{G}\cdot\mathbf{r}} \ , \tag{5}$$

where

$$\mathbf{G}=\sum_{i=1}^{3} n_i\mathbf{B}_i, \quad n_i=-\infty,\ldots,+\infty \ . \tag{6}$$

However, if the function is to be Fourier transformed using only the finite number of points in the mesh $\mathbf{a}_i$, it is possible to include only as many plane waves as there are points in the mesh per unit cell. More specifically, we can only use lattice vectors included in the supercell defined by vectors $\mathbf{b}_i$, since other Fourier components would be superimposed or "aliased" to these.[9] It is customary to use the Brillouin construction for the supercell, taking the $\mathbf{G}$ vectors with smallest modulus among those equivalent by supercell translations. Then, $\max|\mathbf{G}|\le G_{\text{cut}}\equiv\frac{1}{2}\min|\mathbf{b}_i|$, where G^2_{cut} is usually called the plane-wave energy cutoff (in rydbergs when $\mathbf{b}_i$ is in atomic units).[10] If the mesh unit cell defined by $\mathbf{a}_i$ is very "elongated," as in Fig. 1(a), its reciprocal unit cell $\mathbf{b}_i$ will also be very elongated and G_{cut} will be small. Thus, we want to minimize the volume of the supercell (i.e., the number of $\mathbf{G}$ vectors contained in it) but keeping G_{cut} as large as possible. These partly contradictory requirements are accomplished best when the supercell is as "spherical" as possible.

An additional complication is that the most efficient FFT algorithms generally require that the integers N_i in Eq. (3) contain only powers of 2 or, at most, of a few prime factors like 2, 3, and 5. Let us assume for the moment that we use an FFT routine which only accepts powers of 2, and consider the 2D unit cell in Fig. 2. If the 4×4 mesh of Fig. 2(a) is not dense enough to represent our function $f(\mathbf{r})$ (i.e., if the G_{cut} of its reciprocal supercell is too small) then we would usually have to increase it to an 8×8 mesh with *four* times as many points. However, if we define the primitive lattice vectors differently, we can generate the intermediate 4×8 mesh of Fig. 2(b), with only *twice* as many points. If the cutoff of this intermediate mesh is large enough for our purposes, we will have saved by more than a factor of 2 in all our FFT computations. The same procedure can be used in three dimensions for cubic cells: by alternating sc, bcc, and fcc meshes, we can always generate another mesh with only twice the points of the previous one, thus saving by a factor of 4 over the conventional method of doubling each N_i. If our FFT routine permits prime factors other than 2, the savings will be smaller but still substantial.

A more subtle case, with a centered rectangular unit cell, is illustrated in Fig. 3. Although the two sides of the rectangle have very different lengths, the two conventional primitive vectors have equal lengths in this case. In order to preserve in the mesh the lattice symmetry, we are obliged to take the same N_i for both vectors, as in the 4×4 mesh of Fig. 3(a). It is readily apparent that this mesh has the same problem as that in Fig. 1(a), with a poor ratio between G_{cut} and the total number of points N. Again, changing the primitive vectors we can generate the 8×1 mesh of Fig. 3(b), with half the number of points and still the same value of G_{cut}. Alternatively, Fig. 3(c) shows still another mesh with the same number of points (16×1) than that in (a) but with an energy cutoff G^2_{cut} twice as large. Notice that in this case the required primitive vectors have already become quite nontrivial.

We can ask now whether the above procedure of choosing the primitive lattice vectors is possible in general and extensible to three dimensions. To be precise, imagine that we have already found an evenly distributed mesh, with a "rounded" reciprocal supercell and a good ratio between G_{cut} and N, preserving all the symmetry of the lattice and commensurate with it [i.e., accomplishing Eq. (2), but not necessarily Eq. (3)]. The question is whether it is possible in general to find another set of primitive vectors $\mathbf{A}'_i$ and $\mathbf{a}'_i$ of the lattice and mesh respectively, such that

$$\mathbf{a}'_i=\mathbf{A}'_i/N_i \ . \tag{7}$$

The answer is yes. To prove it, let us define two 3×3 ma-

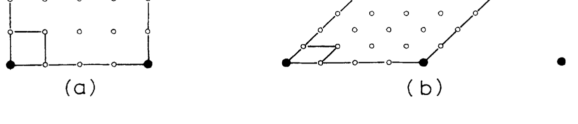

FIG. 2. The same as Fig. 1 for a square lattice.

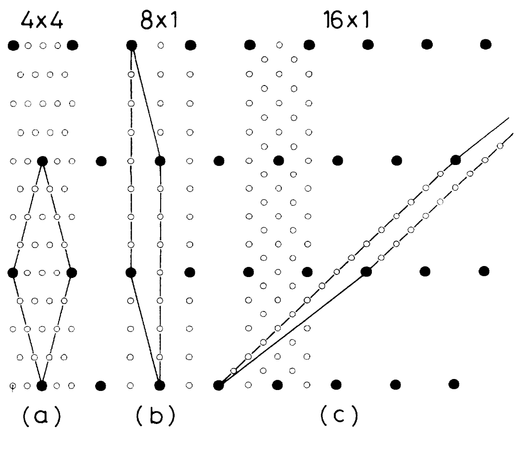

FIG. 3. The same as Fig. 1 for a centered rectangular lattice.

trices $\underline{A}$ and $\underline{a}$ containing the components of vectors $\mathbf{A}_i$ and $\mathbf{a}_i$ with each vector in a column. In this notation, Eq. (2), becomes $\underline{A}=\underline{a}\underline{N}$. Then we want two integer matrices $\underline{U}_A$ and $\underline{U}_a$ with determinant ± 1 and a *diagonal* integer matrix $\underline{N}'$ such that

$$\underline{A}'=\underline{A}\underline{U}_A \ , \tag{8}$$

$$\underline{a}'=\underline{a}\underline{U}_a \ , \tag{9}$$

$$\underline{A}'=\underline{a}'\underline{N}' \ . \tag{10}$$

The fact that $\underline{U}_A$ and $\underline{U}_a$ have determinant ± 1 means that $\underline{A}'$ and $\underline{a}'$ are primitive basis, completely equivalent to $\underline{A}$ and $\underline{a}$. It also implies that the inverse matrices $\underline{U}_A^{-1}$ and $\underline{U}_a^{-1}$ exist and are also integer. After a few straightforward manipulations with Eqs. (2), (8), (9), and (10), we obtain

$$\underline{N}'=\underline{U}_a^{-1}\underline{N}\underline{U}_A \ . \tag{11}$$

We have reduced our problem to that of diagonalizing an arbitrary integer matrix by multiplying it by two unitary integer matrices at left and right. This is a well-known problem in linear algebra, whose solution method is similar to Gaussian elimination and can be found in some textbooks.[11] Furthermore, since $\underline{U}_A$ and $\underline{U}_a$ are unitary and $\underline{N}'$ is diagonal,

$$N \equiv N_1'N_2'N_3'=\det(\underline{N}')=\det(\underline{N}) \ . \tag{12}$$

This implies that, in order to have only permitted prime factors in N_i' we simply must choose mesh vectors $\mathbf{a}_i$ such that the determinant of $\underline{N}$ in Eq. (2) contains only permitted prime factors. Therefore, we conclude that the choice of the unit-cell primitive vectors should *follow from* the choice of the optimum FFT mesh, not vice versa.

It is more difficult to find a simple general algorithm to generate the optimum mesh vectors $\mathbf{a}_i$ and we will instead illustrate the general methods with a particular example. Suppose that our lattice is centered tetragonal, with basis vectors $\mathbf{A}_i$ of the form

$$\underline{A}=\frac{1}{2}\begin{bmatrix} -A_0 & A_0 & A_0 \\ A_0 & -A_0 & A_0 \\ C_0 & C_0 & -C_0 \end{bmatrix} \ . \tag{13}$$

To preserve the lattice symmetry, our mesh must be also tetragonal, either simple or centered.[12] Let us study the simple one, defining our mesh basis as

$$\underline{a}=\begin{bmatrix} a_0 & 0 & 0 \\ 0 & a_0 & 0 \\ 0 & 0 & c_0 \end{bmatrix} \ . \tag{14}$$

Multiplying by $\underline{a}^{-1}$ in Eq. (2) we obtain

$$\underline{N}=\underline{a}^{-1}\underline{A}=\frac{1}{2}\begin{bmatrix} -A_0/a_0 & A_0/a_0 & A_0/a_0 \\ A_0/a_0 & -A_0/a_0 & A_0/a_0 \\ C_0/c_0 & C_0/c_0 & -C_0/c_0 \end{bmatrix} \ . \tag{15}$$

Since we want all matrix elements to be integers we must make $a_0=A_0/2N_1$, $c_0=C_0/2N_3$, with N_1, N_3 integer numbers. Also, $\det(\underline{N})=4N_1^2N_3$ so that we want N_1 and N_3 to have only prime factors permitted by our FFT routine. Furthermore, we want the cutoff associated with our mesh to be large enough, say larger than some prespecified value G_{cut}. This implies that $a_0, c_0 \le \pi/G_{\text{cut}}$, or

$$N_1 \ge \frac{G_{\text{cut}}}{2\pi}A_0, \quad N_3 \ge \frac{G_{\text{cut}}}{2\pi}C_0 \ . \tag{16}$$

The solution to our optimization problem is then given by the two smallest integers N_1, N_3 verifying the inequalities (16) and containing only allowed prime factors. Similar (but more involved) equations are obtained for a centered mesh. In general, the problem is to minimize the product $N_1N_2N_3$, with N_1, N_2, N_3 integer numbers with only allowed prime factors and subject to inequalities of the general form

$$\sum_{i,j=1}^{3} \alpha_{ij}N_iN_j \ge \beta \ , \tag{17}$$

where α_{ij}, β are real coefficients which depend on the lattice and mesh types, on the lattice constants, and on the required cutoff G_{cut}. In some cases, as in our previous example, some of the integers N_1, N_2, N_3 must be equal. Once the coefficients α_{ij}, β in Eqs. (17) are determined for a given mesh type, it is straightforward to perform an exhaustive computer search of combinations N_1, N_2, N_3 to obtain the optimum mesh of that type. After trying this with all mesh types compatible with the lattice symmetry, that with the lowest value for $\det(\underline{N})$ is finally selected. Then, the previously described procedure for obtaining the diagonal form $\underline{N}'$ and the modified primitive vectors $\mathbf{A}_i'$ is applied.

To be fully specific in our example, let us consider the case of La_2CuO_4, which has a body-centered-tetragonal unit cell with $A_0=7.15$ a.u., $C_0/A_0=3.50$, and suppose that we have concluded that we need $G_{\text{cut}}^2=50$ Ry to represent accurately the electron density (i.e., 12.5 Ry for the wave functions). We will also assume that our FFT routine handles efficiently powers of 2, 3, and 5. Then, from Eq. (16) we obtain $N_1 \ge 8.05$ and $N_3 \ge 28.16$, and the next integers containing only allowed prime factors are $N_1=9$ and $N_3=30$. The mesh vectors are given by Eq. (14), with $a_0=A_0/18=0.397$ a.u. and $c_0=C_0/60=0.417$ a.u. Matrix $\underline{N}$, given by Eq. (15), becomes

$$\underline{N}=\begin{bmatrix} -9 & 9 & 9 \\ 9 & -9 & 9 \\ 30 & 30 & -30 \end{bmatrix} \ . \tag{18}$$

In order to "diagonalize" this matrix, we use two basic operations: (1) adding (or subtracting) n times one column to another, and (2) adding or subtracting n times one row to another. The first operation can be accomplished by multiplying our matrix on the right by a matrix with determinant ± 1, and the second by a similar left multiplication. For example, right multiplication by matrix

$$\begin{bmatrix} 1 & n & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

adds n times the first column to the second. Using only these two basic operations we can transform our matrix $\underline{N}$ in (18) the sequence

$$\underline{N} \to \begin{bmatrix} -9 & 0 & 9 \\ 9 & 0 & 9 \\ 30 & 60 & -30 \end{bmatrix} \to \begin{bmatrix} -9 & 0 & 0 \\ 9 & 0 & 18 \\ 30 & 60 & 0 \end{bmatrix}$$
$$\to \begin{bmatrix} -9 & 0 & 0 \\ 0 & 0 & 18 \\ 30 & 60 & 0 \end{bmatrix} \to \begin{bmatrix} -9 & 0 & 0 \\ 0 & 0 & 18 \\ 3 & 60 & 0 \end{bmatrix}$$
$$\to \begin{bmatrix} 0 & 180 & 0 \\ 0 & 0 & 18 \\ 3 & 60 & 0 \end{bmatrix} \to \begin{bmatrix} 0 & 180 & 0 \\ 0 & 0 & 18 \\ 3 & 0 & 0 \end{bmatrix} . \tag{19}$$

This last matrix can be finally diagonalized by a column permutation, which can be also accomplished by right-multiplication by a matrix of determinant 1. Notice that the crucial step for making the above process converge is to use the smaller nonzero element in a row or column to reduce the other elements in that row or column. After multiplying all the left matrices and all the right matrices to get one single matrix on each side, we obtain

$$\underline{N}' = \begin{bmatrix} 180 & 0 & 0 \\ 0 & 18 & 0 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 3 \\ -1 & 1 & 0 \\ -3 & 0 & -1 \end{bmatrix} \underline{N} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 20 & 1 & 1 \end{bmatrix} . \tag{20}$$

By identification with Eq. (11) we see that the last matrix is $\underline{U}_A$. Applying Eqs. (8) and (10) we obtain the transformed primitive and mesh vectors

$$\underline{A}' = \frac{1}{2} \begin{bmatrix} 20A_0 & 0 & A_0 \\ 20A_0 & 2A_0 & A_0 \\ -18C_0 & 0 & -C_0 \end{bmatrix} , \tag{21}$$

$$\underline{a}' = \begin{bmatrix} A_0/18 & 0 & A_0/6 \\ A_0/18 & A_0/18 & A_0/6 \\ -C_0/20 & 0 & -C_0/6 \end{bmatrix} . \tag{22}$$

This mesh contains $180 \times 18 \times 3 = 9720$ points. Actually, this is not the best possibility and we have worked it out only as an illustration because it is the simplest one. The optimum mesh is of body-centered-tetragonal type and contains only $240 \times 12 \times 2 = 5760$ points. We will give only the final result:

$$\underline{a} = \begin{bmatrix} -a_0 & a_0 & a_0 \\ a_0 & -a_0 & a_0 \\ c_0 & c_0 & -c_0 \end{bmatrix} \tag{23}$$

with $a_0 = A_0/24 = 0.298$ a.u. and $c_0 = C_0/80 = 0.313$ a.u. A set of primitive lattice vectors appropriate for this mesh is

$$\underline{A}' = \frac{1}{2} \begin{bmatrix} -20A_0 & 2A_0 & A_0 \\ -20A_0 & 0 & -3A_0 \\ 6C_0 & 0 & C_0 \end{bmatrix} . \tag{24}$$

On the other hand, it is not difficult to see that the minimum mesh of the standard form (3), which is consistent with the conditions imposed, contains $30 \times 30 \times 30 = 27\,000$ points, i.e., almost five times more than the optimum one.

III. SPECIAL POINTS IN THE BRILLOUIN ZONE

There have been a number of works[3-8] proposing methods to generate efficient sets of points in the BZ, each one generalizing the previous ones. Although the terminology has been frequently different, referring to "special points," all these methods in fact generate *uniform periodic meshes,* which are completely analogous to the real-space meshes that we have considered previously. Furthermore, the criterion for the completeness or "quality" of a mesh is also exactly the same: the maximum planewave cutoff in its reciprocal space. Notice that now the reciprocal of the mesh unit cell is a supercell of the real-space lattice unit cell[6] and that the cutoff radius is a length. This supercell and length have a simple physical interpretation: instead of using a mesh in the BZ, one might equivalently use one single **k** point and a real-space supercell. This method would be very inefficient because the computation time increases much faster with the size of the supercell than with the number of **k** points, but the results should be exactly the same. Then, the "length cutoff" is half the minimum distance between equivalent atoms connected by (super)lattice vectors. This cutoff is frequently referred to as the "radius of the first star not integrated exactly" but we prefer the simpler name of "length cutoff" by analogy to the "energy cutoff." It is a rigorous measure of the "quality" or completeness of the **k** mesh and provides a simple way of comparing meshes for different lattices and systems.[8] By contrast, what is more generally given in electronic structure calculations is the "number of special points in the irreducible wedge of the BZ (IBZ)." This number depends strongly on the lattice unit-cell size and symmetry and on the method used to generate the mesh, making it rather difficult to judge the mesh quality on the basis of this single number. In fact, we think that giving this number is as useless as giving the number of points used in FFT's, while a knowledge of the energy cutoff of the plane-wave basis set is definitely helpful. Therefore we propose that we specify the length cutoff, instead of the number of special points, in reporting results of electronic-structure calculations.

It is generally not necessary to perform FFT's in the BZ and this eliminates the complication of requiring the determinant of the supercell matrix $\underline{N}$ to have only some prime factors. However, there is a different and essential complication in the "special-points" issue. The hardest part of most electronic-structure computations involves a series of separate calculations for each **k** point. When several **k** points are equivalent by symmetry, it is necessary to make the calculation only for one of them, reducing drastically the computational effort. It is therefore essential to take into account this reduction in order to determine the efficiency of a **k** mesh, and to maximize it if possible. It is frequently possible to increase the multiplicity of the stars of mesh points by displacing the mesh origin, as shown in Fig. 4 for a 2D square lattice. Obviously, the length cutoff of the mesh is not affected by the displacement but the number of inequivalent **k** points (and the computation time) has been halved. Notice that this trick is useless for the real-space mesh because the FFT algorithms use all the mesh points independently of whether they are equivalent by symmetry.

In principle, it is not necessary that the displaced mesh have all the symmetry of the lattice point group. This is because one can always "symmetrize" the mesh by completing the stars with additional points. Since these additional points are equivalent by construction to the old ones, no calculations are needed at these points and the symmetrization does not cost anything. However, if the displacement initially reduces the symmetry of the mesh,

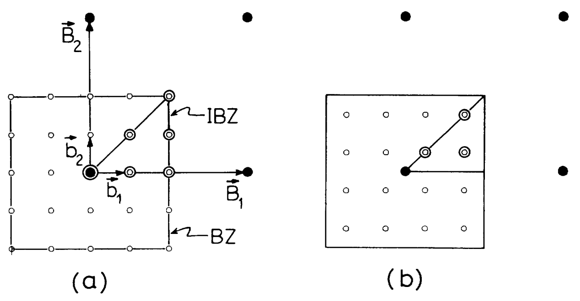

FIG. 4. Reduction of the number of inequivalent mesh points (circles) by displacing the mesh origin. The lattice is square and the long arrows are the primitive vectors of the reciprocal lattice. The square is the first BZ; and the triangle, the IBZ.

TABLE I. All possible **k** meshes for a simple cubic lattice, up to some length cutoff. The **k** points are given by

$$\mathbf{k}=(2\pi/NA_0)\left[\mathbf{b}_0+\sum_{i=1}^{3} n_i\mathbf{b}_i\right],$$

where N is given in the second column, A_0 is the lattice constant in real space, n_i are integer numbers, and $\mathbf{b}_i$ are three primitive vectors of the specified mesh type: $\mathbf{b}_i=(2,0,0)$, $(0,2,0)$, $(0,0,2)$, for sc; $\mathbf{b}_i=(0,1,1)$, $(1,0,1)$, $(1,1,0)$ for fcc; and $\mathbf{b}_i=(-1,1,1)$, $(1,-1,1)$, $(1,1,-1)$, for bcc. $\mathbf{b}_0$ is one of the following points: p_0, (0,0,0); p_i, (1,0,0); and p_2, (1,1,1). When several of these points are indicated for a given mesh, they all produce the same number $N_{\rm IBZ}$ of points in the IBZ. $l_{\rm cut}$ is the "length cutoff" and $N_{\rm star}$ is the number of real-space stars integrated exactly. Asterisks in the last column indicate the optimum sequence of meshes with increasing $N_{\rm IBZ}$.

Mesh type	N	$\mathbf{b}_0$	$N_{\rm BZ}$	$N_{\rm IBZ}$	$(2l_{\rm cut}/A_0)^2$	$N_{\rm star}$	
sc	2	p_0, p_2	1	1	1	1	
bcc	2	p_0	2	2	2	2	
fcc	2	p_0, p_1, p_2	4	2	3	3	
sc	4	p_2	8	1	4	4	*
bcc	4	p_0	16	5	8	7	
sc	6	p_0, p_2	27	4	9	8	
fcc	4	p_1, p_2	32	4	12	11	
sc	8	p_2	64	4	16	14	*
bcc	6	p_0	54	8	18	16	*
sc	10	p_0, p_2	125	10	25	22	
fcc	6	p_0, p_1, p_2	108	10	27	24	
bcc	8	p_0	128	14	32	27	
sc	12	p_2	216	10	36	31	*
fcc	8	p_1, p_2	256	16	48	41	*
sc	14	p_0, p_2	343	20	49	42	
bcc	10	p_0	250	20	50	43	
sc	16	p_2	512	20	64	54	*
bcc	12	p_0	432	30	72	61	
fcc	10	p_0, p_1, p_2	500	28	75	64	*
sc	18	p_0, p_2	729	35	81	69	
bcc	14	p_0	686	40	98	83	
sc	20	p_2	1000	35	100	85	*
fcc	12	p_1, p_2	864	40	108	92	*
sc	22	p_0, p_2	1331	56	121	102	
bcc	16	p_0	1024	55	128	107	*
sc	24	p_2	1728	56	144	121	*
fcc	14	p_0, p_1, p_2	1372	60	147	124	*

it is very unlikely to be an improvement, because the equivalences between at least some of the points will be reduced and the number of inequivalent points will more likely increase rather than decrease. In some cases, it may occur that, after displacement and symmetrization, a denser and *periodic* mesh results with a larger length cutoff (see Fig. 5). But this mesh can always be generated also by a symmetry-conserving displacement of an initially denser mesh. Therefore, following Froyen,[8] we will only consider symmetry-conserving displacements. The problem is then to find the optimum displacement vector $\mathbf{b}_0$ which minimizes the number of points in the IBZ. To achieve this, we find *all* possible symmetry-conserving displacement vectors $\mathbf{b}_0$ and select the best one. By construction (by the same method described in the previous section), the undisplaced mesh has the whole lattice symmetry, i.e., for all mesh points

$$\mathbf{k}=\sum_{i=1}^{3} n_i \mathbf{b}_i \ , \tag{25}$$

and for every symmetry operation S, $S\mathbf{k}$ is also a mesh point. Based on this, it is easy to show[8] that a sufficient condition for the displaced mesh to conserve symmetry S is that the mesh origin $\mathbf{b}_0$ transform under S into a point of the displaced mesh, i.e., $\underline{S}\mathbf{b}_0=\mathbf{b}_0+\sum_{i=1}^{3} n_i\mathbf{b}_i$. Or,

$$(\underline{S}-\underline{I})\mathbf{b}_0=\sum_{i=1}^{3} n_i \mathbf{b}_i \ , \tag{26}$$

where $\underline{I}$ is the identity matrix. Since S conserves the norm and we can always take $\mathbf{b}_0$ within the first BZ, it is easy to see that $-1 \le n_i \le +1$. The method for finding suitable vectors $\mathbf{b}_0$ is then as follows: a particular symmetry operation S of the lattice point group is selected such that the 3×3 matrix $\underline{S}-\underline{I}$ is nonsingular. Then, for each possible combination of values of n_1,n_2,n_3 (27 in total), the system of Eqs. (26) is solved, obtaining a candidate vector $\mathbf{b}_0$. Then, all other symmetry operations of

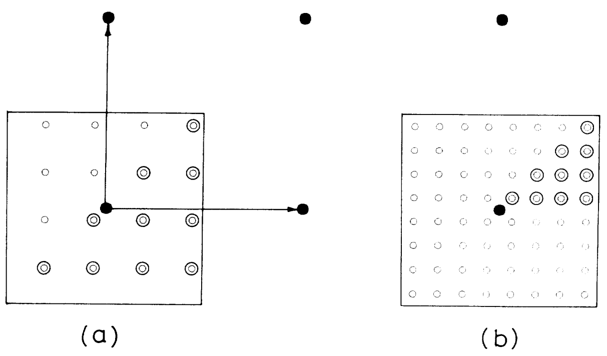

FIG. 5. (a) The same original mesh of Fig. 4 with a different displacement. Notice that the symmetry has been reduced and the number of inequivalent mesh points has increased from 6 to 10. (b) The mesh, after completing the star of each mesh point in (a) (symmetrization). A denser mesh has been generated without any further increase of the number of inequivalent mesh points. But notice that mesh (b) could have been equally well generated by displacing the denser mesh from the beginning and without any symmetrization.

TABLE II. The same as Table I for a fcc **k** lattice (i.e., reciprocal of a bcc lattice in real space).

Mesh type	N	$\mathbf{b}_0$	$N_{\rm BZ}$	$N_{\rm IBZ}$	$(2l_{\rm cut}/A_0)^2$	$N_{\rm star}$	
fcc	1	p_0, p_1, p_2	1	1	0.75	1	
sc	2	p_2	2	1	1.00	2	*
bcc	2	p_0	4	3	2.00	3	
fcc	2	p_1, p_2	8	2	3.00	5	
sc	4	p_2	16	2	4.00	6	*
fcc	3	p_0, p_1, p_2	27	4	6.75	10	*
bcc	4	p_0	32	7	8.00	11	
sc	6	p_2	54	5	9.00	13	*
fcc	4	p_1, p_2	64	6	12.00	17	*
sc	8	p_2	128	8	16.00	22	*
bcc	6	p_0	108	13	18.00	25	
fcc	5	p_0, p_1, p_2	125	10	18.75	26	*
sc	10	p_2	250	14	25.00	35	
fcc	6	p_1, p_2	216	14	27.00	38	*
bcc	8	p_0	256	22	32.00	43	
sc	12	p_2	432	20	36.00	49	
fcc	7	p_0, p_1, p_2	343	20	36.75	50	*
fcc	8	p_1, p_2	512	26	48.00	65	*
sc	14	p_2	686	30	49.00	67	*
bcc	10	p_0	500	34	50.00	68	*
fcc	9	p_0, p_1, p_2	729	35	60.75	82	*
sc	16	p_2	1024	40	64.00	86	*
bcc	12	p_0	864	50	72.00	97	
fcc	10	p_1, p_2	1000	44	75.00	102	*

TABLE III. The same as Table I for a bcc **k** lattice (reciprocal of a real-space fcc lattice).

Mesh type	N	$\mathbf{b}_0$	N_{BZ}	N_{IBZ}	$(2l_{\mathrm{cut}}/A_0)^2$	N_{star}	
bcc	1	p_0	1	1	0.5	1	
sc	2	p_2	4	1	1.0	2	*
bcc	2	p_0	8	3	2.0	4	
fcc	2	p_0, p_1, p_2	16	3	3.0	6	
sc	4	p_2	32	2	4.0	8	*
bcc	3	p_0	27	4	4.5	9	*
bcc	4	p_0	64	8	8.0	15	
sc	6	p_2	108	6	9.0	17	*
fcc	4	p_1, p_2	128	8	12.0	23	*
bcc	5	p_0	125	10	12.5	24	
sc	8	p_2	256	10	16.0	30	*
bcc	6	p_0	216	16	18.0	34	*
bcc	7	p_0	343	20	24.5	46	
sc	10	p_2	500	19	25.0	47	*
fcc	6	p_0, p_1, p_2	432	22	27.0	51	*
bcc	8	p_0	512	29	32.0	59	
sc	12	p_2	864	28	36.0	67	*
bcc	9	p_0	729	35	40.5	75	*
fcc	8	p_1, p_2	1024	40	48.0	89	*
sc	14	p_2	1372	44	49.0	91	*
bcc	10	p_0	1000	47	50.0	93	*
bcc	11	p_0	1331	56	60.5	112	*
sc	16	p_2	2048	60	64.0	118	*
bcc	12	p_0	1728	72	72.0	133	*
fcc	10	p_0, p_1, p_2	2000	73	75.0	138	*
sc	18	p_2	2916	85	81.0	148	*

the lattice point group are applied to $\mathbf{b}_0$, checking whether Eqs. (26) hold for every $\underline{S}$. If they do, $\mathbf{b}_0$ is a symmetry-conserving displacement and we count the number N_{IBZ} of displaced-mesh points in the IBZ. After all the combinations of n_i are tried and all possible displacements $\mathbf{b}_0$ have been found, the one with smallest N_{IBZ} is finally selected. Notice that the method is general and exhaustive, i.e., guaranteed to find the optimum displacement in every case.

In Tables I, II, and III we give all the possible meshes for the BZ of cubic lattices, up to some length cutoff.[13] An asterisk in the last column indicates the sequence of optimum meshes for increasing N_{IBZ}, and therefore for increasing computational effort. Most of these meshes (and especially the most efficient ones) have been already proposed and can be generated with existing methods. But the main purpose of these tables is to show that, for every possible lattice, an unambiguous *optimum* sequence of **k** meshes exists and can be generated systematically, giving the maximum accuracy for a given computational effort or, alternatively, the minimum effort for a given length cutoff.

IV. CONCLUSIONS

We have presented exact and general methods to generate optimum meshes for two different but closely related purposes: to carry out fast Fourier transforms in a crystal unit cell and to calculate integrals in the crystal Brillouin zone. In the first case, we show that *any* mesh (compatible with the lattice symmetry and periodicity) can be used for FFT's with an appropriate set of primitive lattice vectors. Thus, we argue that the primitive lattice vectors should follow from the choice of an optimum mesh, and not vice versa. Also, we propose to use the same measure of mesh quality in both direct and reciprocal space: the maximum cutoff of plane waves representable with that mesh. For this purpose, we propose the use of a "length cutoff," analogous to the "energy cutoff," as a measure of **k**-mesh completeness in electronic-structure calculations, instead of the "number of special points," now frequently specified. Finally, we have written a FORTRAN code implementing the methods presented in this work and we will provide it freely upon request.

ACKNOWLEDGMENTS

This work was supported by Grants No. PB86-0109 and No. PB89-0157 from the Comisión Interministerial de Ciencia y Tecnología (CICyT).

*Present address: Serin Laboratory, Physics Department, Rutgers University, Piscataway, NJ 08854.

†Electronic address: JSOLER at EMDUAM11.

[1]R. Car and M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985).

[2]The $N \ln N$ dependence is only asymptotic for large N. In practice, the optimum mesh may be not that with the lowest N, and it will depend on the actual FFT routine employed. Although it would not be difficult to implement the optimization with a more complex dependence, we will assume here that this is achieved by minimizing N under the constraint of using only powers of certain allowed prime factors.

[3]A. Baldereschi, Phys. Rev. B **7**, 5212 (1973).

[4]D. J. Chadi and M. L. Cohen, Phys. Rev. B **8**, 5747 (1973).

[5]H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5188 (1976).

[6]R. A. Evarestov and V. P. Smirnov, Phys. Status Solidi B **119**, 9 (1983).

[7]P. J. H. Denteneer and W. Van Haeringen, Solid State Commun. **59**, 829 (1986).

[8]S. Froyen, Phys. Rev. B **39**, 3168 (1989).

[9]W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes. The Art of Scientific Computing* (Cambridge University Press, Cambridge, England, 1986).

[10]This cutoff is for the plane-wave expansion of the density. The more frequently referred cutoff, for the expansion of the wave functions, is generally smaller by as much as a factor of 4. This is because the density is given by the square of the wave functions and contains components with wave vectors up to twice as large.

[11]A. G. Kurosch, *Higher Algebra* (MIR, Moscow, 1975).

[12]The mesh unit cell may have either the same orientation of the lattice unit cell or be rotated 45° around the c axis. Here we only consider the first case.

[13]The full cubic symmetry has been assumed. Otherwise, N_{IBZ} would change and the relative efficiency of the different meshes might vary.