JOURNAL OF COMPUTATIONAL PHYSICS 111, 248-255 (1994)

Fast Numerical Solution of KKR-CPA Equations: Testing New Algorithms

E. BRUNO, G. M. FLORIO, B. GINATEMPO, AND E. S. GIULIANO

Dipartimento di Fisica, Universitá di Messina, Messina, Italy

Received September 30, 1991; revised October 15, 1992

Some numerical methods for the solution of KKR-CPA equations are discussed and tested. New, efficient, computational algorithms are proposed, allowing a remarkable reduction of computing time and a good reliability in evaluating spectral quantities. © 1994 Academic Press, Inc.

1. INTRODUCTION

The Korringa–Kohn–Rostoker multiple scattering method, together with the coherent potential approximation (KKR-CPA) [1, 2], is today one of the best theoretical approaches to study the electronic properties in random metal alloys. Moreover, the KKR-CPA calculations, within the density functional theory in local density approximation (LDA) [3, 4], make it feasible to develop an *ab initio* theory for the disordered state of great prediction capabilities. Recently, the extension of the method to the relativistic case [5, 6] allowed the study of heavy metal alloys, although with a big increase in the computational resources needed.

In the literature, several accurate KKR-CPA calculations of the electronic properties (e.g., density of states, Bloch spectral functions, Fermi surfaces, etc.) [7] of random alloys are reported. Thermodynamical properties, such as heat of mixing, phase diagrams, etc. also can be obtained in the framework of the LDA [8].

Notwithstanding such a favourable circumstance, the applicability of this method is actually limited by the amount of computing time needed to solve the double system of non-linear equations imposed by the LDA and the KKR-CPA: the first being relative to the achievement of the charge self-consistency and total energy minimization (Euler–Lagrange equations); the second, the central issue of the KKR-CPA theory, being relative to the achievement of the configurational ensemble averages [3, 4, 7]. Relative to the last, in a recent paper, Akai [9] introduced a "fast" KKR-CPA technique that reduces drastically the amount of computing time *without sacrificing the accuracy too much.* In the present paper we discuss the sources of such an inaccuracy and propose a modified version of Akai's method *without sacrificing the accuracy at all.*

This paper is organised as follows: in Section 2 we discuss some numerical aspects of the KKR-CPA method and outline a fast numerical algorithm for its practical implementation; in Section 3 the results of numerical tests are reported and we study the advantages and the limits of our computational scheme; Section 4 contains some concluding remarks and a brief discussion about the further possible extensions of our numerical scheme.

2. NUMERICAL SOLUTION OF KKR-CPA EQUATIONS

2a. *The KKR-CPA Method*

According to multiple scattering theory, the site diagonal matrix elements $t_c^{ii}(E)$ and $\tau_c^{i}(E)$ of the "on energy shell" scattering path operator and the corresponding single-site effective scattering matrix are related by

$$\tau^{ii}_{c\gamma\gamma'}(E)=\frac{1}{\Omega_{BZ}}\int_{\Omega_{BZ}}[t_c^{-1}(E)-G(\mathbf{q},E)]^{-1}_{\gamma\gamma'}\,d\mathbf{q}, \quad (2.1)$$

where $G_{\gamma\gamma'}(\mathbf{q}, E)$ are the KKR structure constants, $\gamma=\Gamma_n$ labels the nth irreducible representation of the lattice (double) point group and the integral runs over the volume of the first Brillouin zone, Ω_{BZ}. The coherent potential approximation (CPA) states that for a random alloy, whose different components of concentration c_α are labeled by α, the *coherent* scattering matrices t_c may be determined by imposing that the substitution of a single scatterer α in the effective medium does not modify the total scattering, so that

$$\sum_\alpha c_\alpha(D_\alpha(E)\,\tau_c(E))_{\gamma\gamma'}=\tau_{c\gamma\gamma'}(E), \quad (2.2)$$

0021-9991/94 $6.00

where the "CPA projectors" [7, 10] are defined by

$$D_{\alpha\gamma\gamma'}(E)=[1+\tau_\alpha(t_\alpha^{-1}-t_c^{-1})]^{-1}_{\gamma\gamma'}. \tag{2.3}$$

Equations (2.1) and (2.2) can be solved iteratively. This scheme is the best way [1, 2, 7, 10] of formulating a *mean field* theory for the ensemble average of the electronic Green's function.

An alternative form for the CPA condition (2.2) may be obtained in terms of the excess scattering operator X_α [5]:

$$\sum_\alpha c_\alpha X_{\alpha\gamma\gamma'}(E)=0, \tag{2.4}$$

where

$$X_{\alpha\gamma\gamma'}(E)=[(t_\alpha^{-1}-t_c^{-1})^{-1}-\tau_c(E)]^{-1}_{\gamma\gamma'}. \tag{2.5}$$

Also Eq. (2.4) can be used in an iterative procedure and, of course, it is only satisfied when convergence is reached. However, at the nth iteration there is an excess of scattering to be removed,

$$\sum_\alpha c_\alpha X^{(n)}_{\alpha\gamma\gamma'}(E)=X^{(n)}_{c\gamma\gamma'}(E)\neq 0. \tag{2.6}$$

A good convergence scheme may be obtained [11] if one relates $X_c^{(n)}$ to the difference between two successive values of t_c^{-1}, i.e.,

$$X^{(n)}_{c\gamma\gamma'}(E)=[(t_c^{-1,(n)}-t_c^{-1,(n+1)})^{-1}-\tau_c^{(n)}(E)]^{-1}_{\gamma\gamma'}; \tag{2.7}$$

thus

$$t^{-1,(n+1)}_{c\gamma\gamma'}=t^{-1,(n)}_{c\gamma\gamma'}-[X_c^{-1,(n)}(E)+\tau_c^{(n)}(E)]^{-1}_{\gamma\gamma'}. \tag{2.8}$$

It has been shown [11] that, when iterated using (2.8) the CPA equation always converges, provided that the initial guess is the single site average t-matrix (ATA),

$$t^{\mathrm{ATA}}_{\gamma\gamma'}(E)=\sum_\alpha c_\alpha t_{\alpha\gamma\gamma'}(E). \tag{2.9}$$

The actual calculation starts from a guess for the alloy components site potentials, which determine the atomic scattering matrices $t_\alpha(E)$. Then, for each energy, using Eq. (2.9) as the initial guess, $\tau_c(E)$ is computed (Eq. (2.1)) and a new guess for $t_c(E)$ is produced by means of Eq. (2.8). All this has to be iterated until Eq. (2.4) is satisfied (within an arbitrary tolerance) for each irreducible representation component. At convergence *both* $t_c(E)$ and $\tau_c(E)$ must reach stationary values, but, in a real calculation, it happens with different convergence rates. That is due to the fact that $\tau_c(E)$ might be affected by errors in the volume integration, therefore an accurate computation of Eq. (2.1) seems to be necessary. Integration errors are a consequence of the sharpness of the integrand, and, in order to minimize them, one has to sample finely the integration domain Ω_{BZ}.

Such an iterative cycle has to be repeated for each energy value of a grid, in order to evaluate the charge density and the electron total energy, by means of an energy integration (up to the Fermi level) of the site diagonal Green's function. This is expressed in terms of some single site quantities and $\tau_c(E)$ [7]. Then the site potentials might be reconstructed, by adding up the ionic potentials, the Hartree terms, the exchange-correlation terms, and the Madelung terms [12]. These new potentials are now the new inputs to the Euler–Lagrange procedure, continued until one obtains the charge density stationarity, according to the density functional theory.

In order to speed up the calculation, it is worth to introduce some trick. One of the most noticeable is the following. Due to the analytic properties of the Green's function, involved in the theory [13], the energy integration grid may be efficiently built as a path in the upper halfplane for the complex variable E. The great advantage of such a choice is a smoother behaviour of the integrand of Eq. (2.1) and a faster convergence of the internal KKR-CPA iterative scheme. It can be shown, in effect, that for an imaginary part of the energy arbitrarily high the CPA matrices $t_c(E)$ reduce to Eq. (2.9), the zeroth cycle. However, close to the real axis and, in particular, near the Fermi level, this procedure is still expensive. Top computational technology methods and routines [14] are used to take care of such a delicate integration. Moreover, one can exploit the big advantage of the KKR-CPA method: its intrinsically *parallel* structure (each energy value is a *task*). Very efficient and fast codes have been implemented on parallel supercomputers and workstations [14].

All that notwithstanding, it is still highly desirable to reduce, if possible, the amount of calculations involved in the procedure from Eq. (2.1) to Eq. (2.8). In standard cases, it involves a matrix inversion for a couple of thousands **q**-points per energy value and per CPA iteration. The size of the matrices is, taking into account for terms up to $l=3$ in the angular momentum expansion, $16n$ (or 32 for the relativistic case), with n number of atoms per unit cell. Employing standard schemes of solution, usually this procedure must be iterated 5–20 times; thus each energy value requires about 10^4 matrix inversions. In order to obtain the charge density, one has to perform such calculations for 100–200 energies. Obtaining charge self-consistency requires 10–50 external iterations. Therefore a self-consistent KKR-CPA calculation requires a number of matrix inversions of the order of 10^7, at least. It can also be shown that Eq. (2.1) might take up to the 94% of the full calculations [14]. Of course, improving such a point will greatly reduce the computing time for a full self-consistent field run.

2b. *Fast Numerical Solution of KKR-CPA Equations*

Mostly accepting Akai's suggestions [9], we point out two facts. First, when the (initial) guess for t_c is far from the solution, there is no need to compute the integral (2.1) with great accuracy. It might be more efficient to improve progressively the accuracy in the evaluation of the integral, together with the increase of the confidence in t_c. Second, and perhaps more important, the above scheme, cycle by cycle, erases all the expensive calculations *already* done for $\tau_c(E)$. In effect, $\tau_c^{(n)}(E)$, before convergence is reached, is not exact, nevertheless it is *not completely wrong* and it might be clever to trace those *good* contributions to it and take advantage of them during the next iterations, as we now show.

In the quoted codes [14], the techniques used for the evaluation of the integral (2.1) are the special directions method and the tetrahedron method. The first, the most expensive, is used for energy values close to the positive real axis, and the second, which involves a linearization, is used elsewhere in the grid. In the directions method, several rays, starting from the Γ point in the irreducible Brillouin zone are selected, and the volume integral is computed as a weighted summation of the corresponding line integrals. In the tetrahedron method, the integration domain is decomposed into tetrahedra of different sizes, at whose centers the integrand is computed, and the volume integral turns out to be the sum of the contributions, depending on $\tau_{c\gamma\gamma}$, cofactors, weighted by some function which depends upon the corresponding tetrahedra volumes and the determinant of the KKR matrix. However, in order to have a continuous improvement of the integration accuracy, it would be preferable to have an integration method which allows a much easier handling of the reciprocal space grid. Such a request is satisfied by Monte Carlo and Weyl integration methods, which allow us to use grids without any geometrical restrictions. Both these methods sample the integrand function domain D uniformly, equally weighting each contribution. They produce an estimate of the integral, convergent to the exact value when N goes to infinity, by means of

$$\frac{1}{D}\int_D F(\mathbf{q})\, d^3\mathbf{q} \approx \frac{1}{N}\sum_{i=1}^{N} F(\mathbf{q}_i). \tag{2.10}$$

Opposite to standard Monte Carlo methods, where the $\mathbf{q}_i$ are chosen randomly, the Weyl method chooses

$$q_{i,\lambda} = \mathbf{i}\cdot\xi_\lambda - \operatorname{int}(i\cdot\xi_\lambda), \tag{2.11}$$

where λ runs over the cartesian coordinates of the integration domain and the ξ_λ are irrational incommensurate numbers. This choice ensures uniform sampling and an increased convergence rate, which is proportional to N [9], whilst in the Monte Carlo method it is proportional only to $N^{1/2}$ [15].

The idea is now the following: exploiting the flexibility of Eq. (2.11), we can achieve convergence of the integral (2.1) *and* of the CPA equation *contemporarily*, by means of the following algorithm. We start from the ATA guess $t_c^{(0)}$ and, selecting the first s_1 (~ 10) points out of the N points Weyl grid (2.11), we compute

$$\tau_c^{(1)} = \frac{1}{s_1}\sum_{i=1}^{s_1}\left[t_c^{(0)-1} - G(\mathbf{q}_i)\right]^{-1}. \tag{2.12}$$

This crude estimate of τ_c is used to obtain $t_c^{(1)}$, by means of Eq. (2.8). This new value will be used to evaluate the integrand in a set of s_2 grid points, different from the former ones. Now, we are going to add to the estimate of τ_c new contributions, hopefully *more reliable*, without wasting the old value.

$$\tau_c^{(2)} = \frac{a_2}{s_2}\sum_{i=1}^{s_2}\left[t_c^{(1)-1} - G(\mathbf{q}_i)\right]^{-1} + (1-a_2)\,\tau_c^{(1)}. \tag{2.13}$$

For the generic jth iteration the last equation might be rewritten as

$$\tau_c^{(j)} = \frac{a_j}{s_j}\sum_{i=1}^{s_j}\left[t_c^{(j-1)-1} - G(\mathbf{q}_i)\right]^{-1} + (1-a_j)\,\tau_c^{(j-1)}, \tag{2.14}$$

where the coefficients a_j have to be chosen in order to give a bigger weight to the last, *more precise*, s_j contributions. We observe that to give equal weight to each point contribution would be equivalent to assuming that

$$a_j = s_j \Big/ \sum_{k=1}^{j} s_k. \tag{2.15}$$

We want to find out the properties of the coefficients a_j. In order to study an appropriate choice for these coefficients we define

$$w_j = \left(\sum_{k=1}^{j} s_k \Big/ s_j\right) a_j. \tag{2.16}$$

It is clear that, for $j = 1$, it has to be $a_1 = w_1 = 1$. Moreover, when j is large, in order to obtain convergence in Eq. (2.12), we must weight equally each contribution. That is,

$$\lim_{j\to\infty} w_j = 1 \tag{2.17}$$

When j is finite, the newest contributions must have a bigger weight. That is,

$$w_j > 1 \qquad (\forall j > 1). \tag{2.18}$$

This succession must be decreasing, because the accuracy of the integral must increase with j, i.e.,

$$w_j > w_{j+1} \qquad (\forall j > 1). \tag{2.19}$$

A proper choice for the set w_j must satisfy the above conditions from (2.16) to (2.19). In this work we fixed $s_j = s$ for all j's and used the succession

$$w_j = j\,\frac{h_j - h_{j-1}}{h_j} \tag{2.20}$$

$$h_j = j \cdot s - \frac{\mu}{1-\mu}\,(1-\mu^j), \tag{2.21}$$

where $0 < \mu < 1$. μ is a parameter that should be adjusted to optimize the rate of convergence. For $s = 9$ we found a good choice to be $0.9 < \mu < 0.99$. In our calculations we found that such an algorithm could cause numerical instabilities in the CPA equation during the very first iterations, when the integral estimate is still bad. In order to avoid these problems, instead of Eq. (2.8), we used a smoother mixing for t_c, i.e.,

$$t_{c\gamma\gamma'}^{-1,(j+1)} = t_{c\gamma\gamma'}^{-1,(j)} - \eta[X_c^{-1,(j)}(E) + \tau_c^{(j)}(E)]_{\gamma\gamma'}^{-1}, \tag{2.22}$$

where $0 < \eta < 1$ should also be adjusted to achieve the fastest convergence. This form always converges starting from ATA, as the Mill's algorithm represented by Eq. (2.8), although the effect of the η parameter is to slow the convergence rate a little. We notice, in effect, that this algorithm reduces to the Mills iterative scheme (Eq. 2.8) when s goes to infinity and $a_j = 1$. With our choice of a small value for s, of course, the number of iterations necessary to obtain convergence is expected to be fairly large; nevertheless the total amount of operations should drop.

This algorithm, as we already mentioned, is essentially the one suggested by Akai [9], the main difference being the choice of the CPA equation in the form (2.4) rather than (2.2). The advantages of this choice have been discussed in connection with non-zero off-diagonal elements of t_c in Ref. [5]. As Akai remarked, this scheme of solution of the KKR-CPA equations is quite effective for evaluating energy-integrated quantities, such as total energy or magnetization. However, we point out that the convergence rate for the energy-dependent quantities, such as density of states or spectral functions could be slower than the former. The analysis of these last quantities, indeed, plays a crucial role in understanding the electronic properties of an alloy, and one should evaluate them to a high degree of reliability. We are going to test, in the following section, such a reliability, but, first, we want to comment on another point.

We note that, usually, the convergence of the CPA equation is expected to be slow in those energy ranges where "A" and "B" states strongly hybridize, because the ATA is far from the CPA solution. However, in such case, the Weyl integration formula should work well, because the integrands are broad functions. On the contrary, when ATA works well, usually the integrands of Eq. (2.10) are sharp functions and the Weyl formula shows its weakness, needing a big N for an accurate integral. We hope that such weaknesses and advantages will compensate each other in such a mixed scheme.

3. NUMERICAL TESTS

Section 2b algorithm has been coded out and plugged into the KKR-CPA code. We chose $Ni_{0.5}$–$Pt_{0.5}$ random alloy as a good candidate [14, 16] to check the method reliability for energy-dependent quantities as the density of states (DOS) and the Bloch spectral functions (BSF). The DOS is the k-space integral of the BSF. The last is the imaginary part of the lattice Fourier transform of the Green's function. For a pure system it reduces to a sum of delta functions centered at the band structure points $\mathbf{k}(E)$. The effects of the disorder in a random alloy are to broaden the BSFs unequally at each $\mathbf{k}$ and E. In other words, the disorder can differently affect different states. As a consequence, the BSFs may be broad or sharp or both in the same system. $Ni_{0.5}$–$Pt_{0.5}$, a "common band" alloy, shows BSFs with such a variety of behaviour, with broad and sharp states at different energies and $\mathbf{k}$-points. In other words (see the last comment of Section 2), this is a system which presents good and bad features for this method.

A reasonable observable for testing purposes is the DOS. As we mentioned, it is the BSF integral in the reciprocal space; therefore, if for some energy value either the Weyl integration failed or the CPA convergence was bad, we expect spikes or non-analytical behaviour to show up in the DOS. One can define the DOS in terms of the α-projected DOS as

$$n(E) = \sum_\alpha c_\alpha n_\alpha(E) \tag{3.1}$$

$$n_\alpha(E) = -\frac{1}{\pi}\,\mathrm{Im}\,\mathrm{Tr}\int_{\mathrm{u.c.}} d\mathbf{r} \times [F_\alpha(E,\mathbf{r},\mathbf{r})\,D_\alpha(E)\,\tau_c(E) - J_\alpha(E,\mathbf{r},\mathbf{r})]. \tag{3.2}$$

We refer the reader to [7] for the meaning of the quantities in the last equation. All the results we present have been produced using as input the non-relativistic self-consistent site potentials obtained in Ref. [16].

3a. *The Plain Weyl Method*

We calculated the $Ni_{0.5}$–$Pt_{0.5}$ DOS for real energy and, therefore, we attached the special directions method, rather than the tetrahedron method. In Fig. 1a we show our calculation of the DOS using 36 directions (about 2000 **q**-points per integral). We name this the *exact* method. In Fig. 1b we show the result of the calculation performed using the scheme illustrated in Section 2. We name this the *plain Weyl* method. An upperbound, for the number of **q**-points, of 90,000 per energy value was imposed in the last run. These calculations agree well, apart from a few energy points in the region between 0.47 and 0.55 ry above the muffin-tin level. We are going to investigate the sources of such errors.

In Fig. 2 we plot the CRAY-YMP time (single tasking) needed for those runs versus the energy (essentially the time spent to solve the CPA equation). We observe that the *plain Weyl* method does achieve a remarkable overall gain in the total CPU time (the difference of the areas). We are aware that the units used as ordinates could be misleading. However, the *exact* method routines are highly efficient on the YMP, as opposed to the *plain Weyl* method whose loop sizes are small (cf. s_j in Section 2). Hence, the gain in terms of CPU time on a scalar machine would increase.

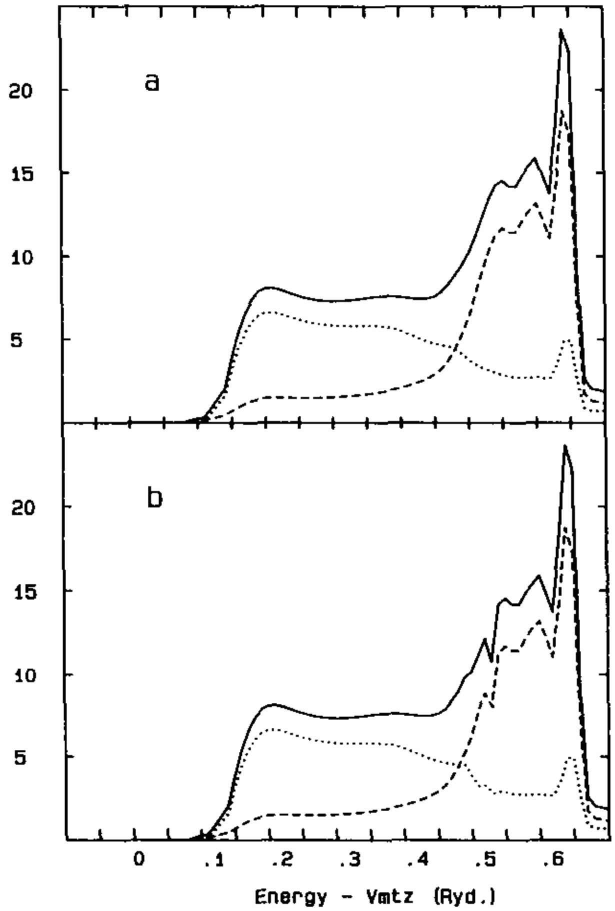

FIG. 1. Total (full line) and partial (dashed = Ni, dotted = Pt) DOS's for $Ni_{0.5}-Pt_{0.5}$ alloy: (a) *exact* DOS's; (b) *plain Weyl* DOS's (see text).

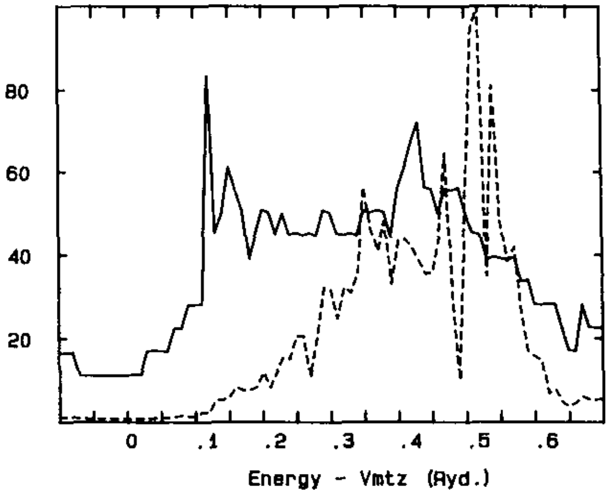

FIG. 2. CRAY-YMP CPU time (seconds) vs energy for *exact* (full line) and *plain Weyl* (dashed line) methods.

Noticeably, the *plain Weyl* execution times are much larger in the energy region between 0.35 and 0.55 ry, even higher than the *exact* method times for a bunch of energy values in the same interval. At those points the achievement of the double convergence (CPA equation and BZ integral) is very slow. Eventually, for some points, the upperbound of 90,000 **q**-points was reached without perfect convergence. This is the source of the mismatches in Fig. 1, to eliminate which a much bigger upperbound was necessary, but at the price of an inconvenient increase of computing time.

In Fig. 3 we show the scattering phase shifts for the site potentials used. As one can see, the energy region, where the *plain Weyl* method shows its drawbacks, lies in between the *d* phase shift resonances of the Pt and Ni components. In such a region the disorder scattering is strong and plenty of CPA iterations are required. This implies that the subtle

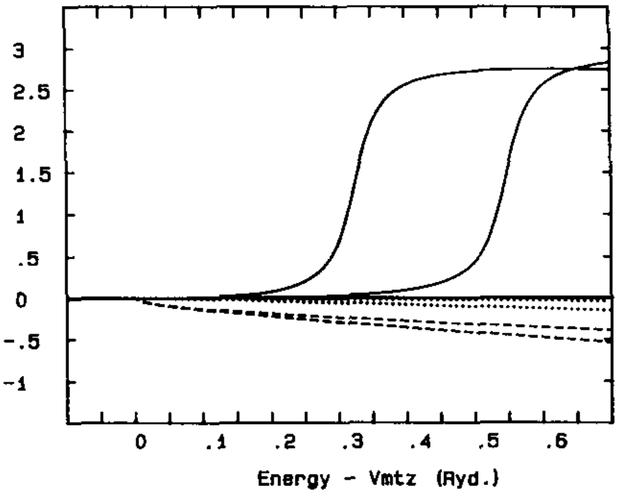

FIG. 3. Single site scattering phase shifts up to $l = 2$: *s*, dotted lines; *p*, dashed lines; *d*, full lines; Pt and Ni *d* phase shifts have their resonance at 0.33 and 0.55 Rydbergs, respectively.

error correction mechanism of Eq. (2.14), in practice, dramatically fails in such a situation.

Nevertheless, out of this relatively small interval the method does work with a remarkable saving of computing time. To recover these advantages, we are going to explore, in the next subsection, how fast this method could be without affecting the results too much, in those ranges where it already works. In the second next subsection, we will design new tricks to achieve the same accuracy of standard calculations, still maintaining the computer time savings.

3b. *Rough Weyl Method*

In order to push this method to its limits, we fix the Weyl integral upperbound to 2000 (i.e., the number of points needed *per iteration* by the *exact* method), in place of 90,000 *even if convergence is not reached.* We name this the *rough Weyl* run. The resulting DOS, shown in Fig. 4, is surprisingly, close to the *exact* one in large energy ranges and, as expected, very bad in between the Pt and Ni resonances.

We want to remark that (i) this *rough Weyl* method involves a much smaller number of matrix inversions than the *exact* one, and (ii) in the regions where it works, the convergence to the *exact* values of t_c and τ_c is very good and reliable. That confirms Akai's results [9] and strongly supports this method. But the algorithm fails badly exactly at those energies where the CPA is far from ATA. Our analysis shows that this algorithm cannot be used as a *black box*, because it would lead either to an unpredictable increase of computing time or to unphysical results, as in Fig. 4. That might prevent its application to a full self-consistent field calculation.

On the other hand, as we will discuss next, up in the complex energy plane, when the CPA solution is closer to the ATA and the integrands are smooth, this scheme works fairly well at any energy, producing also big time savings. This is the reason why one can still have good results for energy-integrated quantities and the fast convergence Akai found for those.

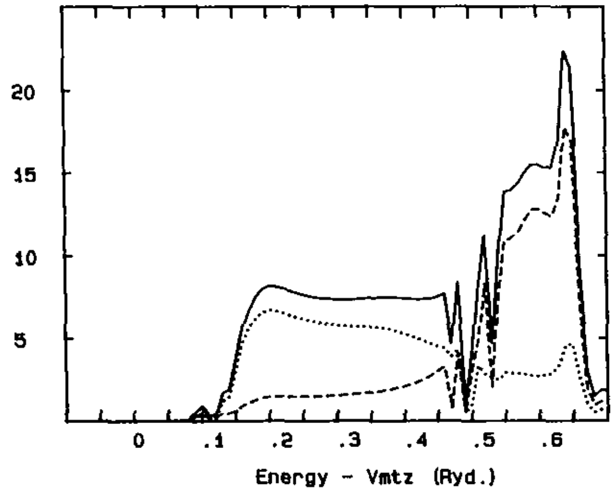

FIG. 4. The same as Fig. 1, but using the *rough Weyl* method.

3c. *Improved Weyl Methods*

One question is still unanswered. Can this algorithm be improved to produce accurate energy-dependent observables? The answer lies in the accuracy of the integral in the region between the resonances. As we claimed above, *in such a range the CPA solution for t_c and the ATA are* critically different. On top of that, the integral is not precise. That creates an unfavourable situation for the mixed convergence scheme of Section 2b, because in this case there is no advantage in saving the history of τ_c, whose initial values are, of course, far from the solution. On the other hand, a closer look to t_c at each iteration, shows that very soon they obtain values that are not that far from the CPA solution, although they are still critically far. This fact brings down the convergence rate, when used in connection with Eq. (2.14). One sensible trick is then the following: let us use

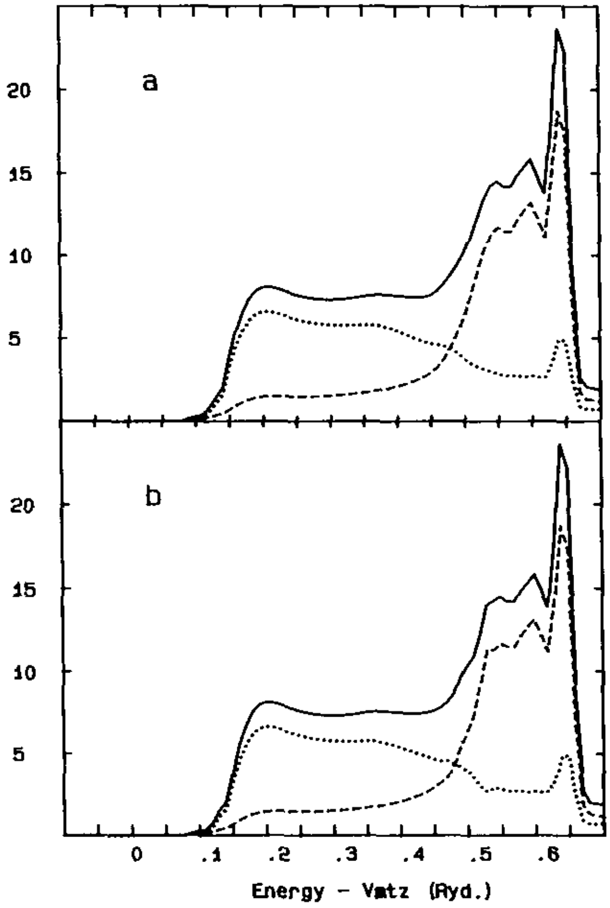

FIG. 5. The same as Fig. 1 but using the (a) *Weyl++* method and (b) *Weyl+* method.

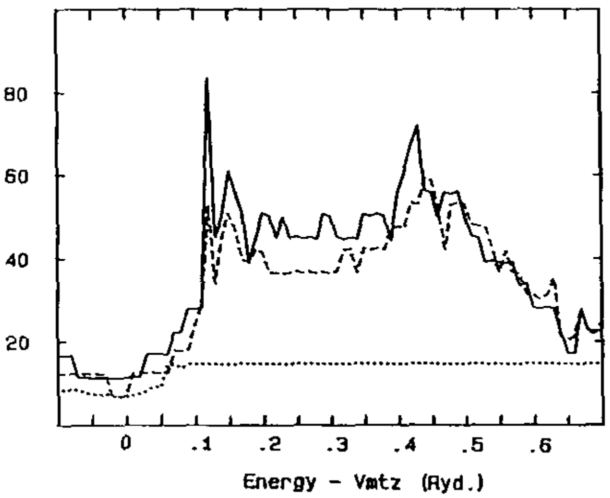

FIG. 6. The same as Fig. 2 but for *exact* method (full line), *Weyl+ +* method (dashed line), and *Weyl+* method (dotted line).

these *rough Weyl* t_c values as input for the *exact* calculation. We name this the *Weyl+ +* method. Another clever trick, as we will show, is the following: by the same input let us perform *just* one complete iteration by the special directions method. We name this the *Weyl+* method.

In Fig. 5 we show the DOS we obtained with the last methods. As expected, the *Weyl+ +* method DOS, Fig. 5a, is undistinguishable from the exact calculation of Fig. 1a. It is somewhat surprising *also* that the *Weyl+* method gives very close results to both the exact DOS. Moreover, going up the complex plane, all these methods give totally identical results.

In Fig. 6 we again plot the CRAY-YMP times needed for the real energy runs of all these methods. As expected, the *Weyl+ +* method allows some saving (10%) with respect to the *exact* one, but the gain of the *Weyl+* method is much greater (more than 80%).

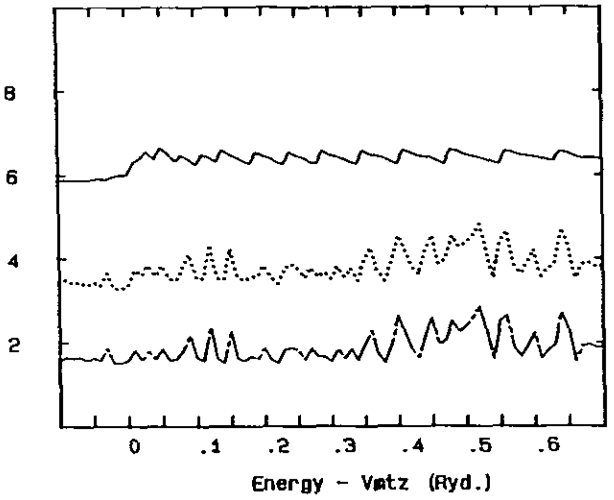

FIG. 7. CRAY-YMP CPU times (seconds) vs the real part of the energy at constant imaginary energy $\gamma = 0.5$ Rydbergs: *exact* method, full line; *Weyl+* (*Weyl+ +*) method, dotted line; *rough Weyl* method, unequally dashed line.

At high imaginary energy (0.5 ry), interestingly, the *Weyl+ +* method converges just after one iteration by the special directions method, then coinciding with the *Weyl+* method. Also the *rough Weyl* method gives very close results. Figure 7 shows the CPU times needed for the *exact*, *Weyl+* (*Weyl+ +*), and *rough* methods in the complex plane. This picture shows the great gain in computer time one can obtain by using this scheme, *without losing accuracy*.

4. CONCLUSIONS

After our applications of the Akai scheme, we can draw the following conclusions:

— Akai's idea-based methods can be used to save computing times, but great care must be used for energy-dependent observables;

— Our *Weyl+* method is a fairly good compromise between accuracy and savings in evaluating energy-dependent quantities;

— Energy-integrated quantities produced by all these methods, by using clever complex integration paths, are reliable, and the *rough Weyl* method might be the most convenient;

— the *Weyl+* method, owing to the fact that it takes about the same amount of computing time for each energy, seems the best candidate to exploit the parallel architecture of modern supercomputers (each energy is a task).

The relevance of the present study for a full self-consistent field calculation is that such methods offer the chance to reduce the computational effort for each charge iteration. Perhaps these gains could be much larger if one might combine the external iterative procedure to find the equilibrium charge density, together with the Brillouin zone integration. This would be a natural extension of the scheme presented in Section 2b.

ACKNOWLEDGMENTS

The authors acknowledge the contribution of the Consiglio Nazionale delle Ricerche (CRN-Italy), providing CRAY-YMP/432 computer resources at CINECA, Bologna, Italy.

REFERENCES

1. P. Soven, *Phys. Rev.* **156**, 809 (1967).
2. G. M. Stocks, W. M. Temmermann, and B. L. Gyorffy, *Phys. Rev. Lett.* **41**, 339 (1978).
3. G. M. Stocks and H. Winter, in *Electronic Structure of Complex*

Systems, edited by P. Phariseau and W. M. Temmermann, NATO-ASI Series B, Physics, Vol. 24 (Plenum, New York, 1985).

4. D. D. Johnson, F. J. Pinski, and G. M. Stocks, *Phys. Rev. B* **30**, 5508 (1984).
5. B. Ginatempo and J. B. Staunton, *J. Phys. F: Metals Phys.* **18**, 1827 (1988).
6. G. M. Florio, B. Ginatempo, E.S. Giuliano, and J. B. Staunton, *J. Phys.: Condens. Matter* **1**, 8385 (1989).
7. J. S. Faulkner and G. M. Stocks, *Phys. Rev. B* **21**, 3222 (1980).
8. B. L. Gyorffy, D. D. Johnson, F. J. Pinski, D. M. Nicholson, and G. M. Stocks, in *Alloy Phase Stability*, edited by G. M. Stocks and A. Gonis, NATO-ASI Series E, Applied Sciences, Vol. 163 (Kluwer Academic, Dordrecht, 1989).
9. H. Akai, *J. Phys: Condens. Matter* **1**, 8405 (1989).
10. P. J. Durham, B. L. Gyorffy, and A. J. Pindor, *J. Phys. F: Metals Phys.* **9**, 661 (1980).
11. R. Mills, L. J. Gray, and T. Kaplan, *Phys. Rev. B* **27**, 3252 (1983).
12. D. D. Johnson, D. M. Nicholson, F. J. Pinski, B. L. Gyorffy, and G. M. Stocks, *Phys. Rev. Lett.* **56**, 2088 (1986).
13. F. J. Pinski, D. M. Nicholson, G. M. Stocks, D. D. Johnson, and B. L. Gyorffy, in *Proceedings, Third International Conference on Supercomputing, Boston 1988*, edited by L. Kartashev and S. Kartashev (International Supercomputing Institute, New York, 1988).
14. G. A. Geist, B. Ginatempo, W. A. Shelton, and G. M. Stocks, *IEEE Trans. Software Eng.*, May (1991).
15. C. B. Haselgrove, *Math. Comput.* **15**, 323 (1961).
16. F. J. Pinski, B. Ginatempo, D. D. Johnson, J. B. Staunton, G. M. Stocks, and B. L. Gyorffy, *Phys. Rev. Lett.* **66**, 766 (1991).