

https://dspace.org/

Theoretical study of the electronic nature of non-stoichiometric metal hydrides

Anagnostaras, Philip

2015-05-28
http://hdl.handle.net/10539/17894

Downloaded from a DSpace repository. https://dspace.org/

Because of the complexity of the algebraic manipulations required by LCAO methods, it is sensible from the start to derive modified Bloch sums B_ℓ which are *orthogonal* to each other; they do not have this property in general because of the lack of orthogonality in the set $\{\phi_\ell\}$. To obtain ϕ_ℓs which are orthogonal, the *Löwdin* method is used[A1.2]; the resultant Löwdin functions $\{\psi_\ell\}$ have the same symmetry properties as the atomic orbitals from which they are derived, thus retaining the symmetry features of the atomic formalism.

Rewriting equation (A1.1) in terms of Löwdin functions, and introducing a suitable normalization factor, we now obtain an *orthogonal* set $\{B_\ell\}$ such that:-

$$B_\ell(\underline{k}) = \frac{1}{\sqrt{N}} \sum_j e^{i\underline{k}\cdot\underline{R}_j} \psi_\ell(\underline{r}-\underline{R}_j) \qquad \text{(A1.3)}$$

where periodic boundary conditions are assumed, the number of unit cells in the repeating region being N. For the fcc structure (that of Palladium metal) Slater and Koster choose a unit cell containing only one atom. The next step is to evaluate the energy matrix element between any given pair of Bloch sums; if H is the Hamiltonian then these matrix elements are given by:-

$$\langle B_\ell|H|B_{\ell'}\rangle = \frac{1}{N} \sum_{j,j'} e^{i\underline{k}\cdot(\underline{R}_{j'}-\underline{R}_j)} \int_V \psi_\ell^*(\underline{r}-\underline{R}_j) H\, \psi_{\ell'}(\underline{r}-\underline{R}_{j'})\,dV \qquad \text{(A1.4a)}$$

Either of the summations in equation (A1.4a) can be replaced by the number of unit cells N, so that this equation becomes:-

$$\langle B_\ell|H|B_{\ell'}\rangle = \sum_j e^{i\underline{k}\cdot(\underline{R}_{j'}-\underline{R}_j)} \int_V \psi_\ell^*(\underline{r}-\underline{R}_j) H\, \psi_{\ell'}(\underline{r}-\underline{R}_{j'})\,dV \qquad \text{(A1.4b)}$$

where $\underline{R}_{j'}-\underline{R}_j$ is the vector joining the atom on which ψ_l is situated to the atom on which $\psi_{l'}$ is located.

Slater and Koster remark at this stage on the complexity of equation (A1.4b): firstly the Löwdin functions $\{\psi_l\}$ must be determined, which is a considerable task; secondly each ψ_l consists of a linear combination of ϕ_ls, so that the integral of equation (A1.4b) itself breaks up into a linear combination of several integrals of the form $\int_V \phi_l^*(\underline{r}-\underline{R}_j)H\ \phi_{l'}(\underline{r}-\underline{R}_{j'})dV$; and thirdly the Hamiltonian itself consists of a number of terms. In particular, the expression for H includes a sum of spherically-symmetric potential wells situated on all N atoms of the system; thus $\langle B_l|H|B_{l'}\rangle$ consists of a linear combination of complicated energy integrals of the form:-

$$E_{l,l'} = \int_V \psi_l^*(\underline{r}-\underline{R}_j)\ H\ \psi_{l'}(\underline{r}-\underline{R}_{j'})dV \qquad (A1.5)$$

where H contains terms like $v(\underline{r}-\underline{R}_{j''})$, which is the potential well associated with the atom at $\underline{R}_{j''}$. Thus we see that equation (A1.5) describes a so-called *three-centre integral* (that is the integrand has factors relating to in general three different atoms), which Slater and Koster describe as being computationally impractical because of their complexity and because there are so many of them. Thus a *rigorous* BS calculation using the LCAO method is not feasible; hence Slater and Koster's decision to replace these integrals with arbitrary constants with which to fit accurate BS calculations performed at certain special values of $\underline{k}$.

In order to carry out such a fitting procedure, the number of constants (and hence integrals) kept must equal the number of states for which the energy is accurately known. To achieve this we firstly notice that the integral of equation (A1.5) will get smaller as the overlap between ψ_l and $\psi_{l'}$ decreases, that is as the separation between the atoms at $\underline{R}_j$ and $\underline{R}_{j'}$ increases. It can thus be arbitrarily decided to consider only nearest neighbours (n.n.s), or n.n.s and second-n.n.s, or even n.n.s, second-n.n.s and third-n.n.s, depending on the number of constants required. A second means of reducing the total number of integrals is to make use of straightforward group-theory techniques to show the equivalence of many integrals. Thirdly, the number of atomic functions included in the set $\{\phi_l\}$ can be chosen judiciously; for example, some transition metals can be quite well modelled (by BS techniques) using only the five valency d orbitals (the order of the secular equation is equal to the number of atomic orbitals used); and fourthly, the number of parameters can be changed by neglecting three-centre integrals.

We now expand upon this fourth possibility; although three-centre integrals are smaller than *two-centre integrals*, they are not necessarily negligible. However, within the context of this model (in which many terms have already been dropped) Slater and Koster consider this approximation to be quite acceptable, and proceed to apply it; hence we must explain the concept of a two-centre integral. Such an integral is of the form:-

$$(l\ l') = \int_V \psi_l^*(\underline{r}-\underline{R}_j)\ H\ \psi_{l'}(\underline{r}-\underline{R}_{j'})dV \qquad (A1.6)$$

where H contains terms like $v(\underline{r}-\underline{R}_i)$, such that i=j or j' (cf equation (A1.5)).

This is similar to the type of integral which would be used to describe a diatomic molecule, and hence we express the functions $\{\psi_l\}$ as linear combinations of functions space-quantized with respect to the interatomic axis $\underline{R}_j,-\underline{R}_j$. For example, a d function would be expressed as a linear combination of $d\sigma$, $d\pi_\pm$ and $d\delta_\pm$ functions, where σ, π and δ refer to the components of angular momentum about the axis (we must remember that although the ψ_ls are not atomic orbitals, they have the same symmetry properties as their corresponding atomic orbitals). We choose to rewrite the two-centre integrals (equation (A1.6)) in the form (mnλ), where m,n are s,p,d and λ is σ,π,δ; the following are the ten possible cases: (ssσ), (ppσ), (ppπ), (ddσ), (ddπ), (ddδ), (spσ), (sdσ), (pdσ), (pdπ).

We next introduce the following notation for the states l and l' associated with the integrals $E_{l,l'}$ (equation (A1.5)), defined with respect to Cartesian coordinates: s stands for an s orbital; x,y,z for p_x,p_y,p_z orbitals; xy, yz, zx for the three d_{t_2} orbitals; and x^2-y^2, $3z^2-r^2$ for the two d_e orbitals. Thus for instance the interaction between an s orbital on one atom (l=s) and a d_e orbital on another atom (e.g. $l' = x^2-y^2$) would be written E_{s,x^2-y^2}. Next, let the directional cosines of the interatomic vector $\underline{R}_j,-\underline{R}_j$ be p,q,r (note that these will be determined by the structure of the lattice, for example fcc). We can now

Energy Integral	Two-Centre Approximation
$E_{s,s}$	$(ss\sigma)$
$E_{s,xy}$	$\sqrt{3}pq(sd\sigma)$
E_{s,x^2-y^2}	$\frac{1}{2}\sqrt{3}(p^2-q^2)(sd\sigma)$
$E_{s,3z^2-r^2}$	$\left[r^2-\frac{1}{2}(p^2+q^2)\right](sd\sigma)$
$E_{xy,xy}$	$3p^2q^2(dd\sigma) + (p^2+q^2-4p^2q^2)(dd\pi) + (r^2+p^2q^2)(dd\delta)$
$E_{x^2-y^2,x^2-y^2}$	$\frac{1}{4}(p^2-q^2)^2(dd\sigma) + \left[p^2+q^2-(p^2-q^2)^2\right](dd\pi) + + \left[r^2+\frac{1}{4}(p^2-q^2)^2\right](dd\delta)$
$E_{3z^2-r^2,3z^2-r^2}$	$\left[r^2-\frac{1}{2}(p^2+q^2)\right]^2(dd\sigma) + 3r^2(p^2+q^2)(dd\pi) + + \frac{3}{4}(p^2+q^2)^2(dd\delta)$

Table A1.1 A selection of energy integrals in the two-centre approximation (after Slater and Koster[A1.1], Table I). Note that r in the first column is given by $r = \left[x^2+y^2+z^2\right]^{\frac{1}{2}}$ whereas in the second column it refers to the third of the directional cosines p,q,r.

express $E_{l,l'}$ as a linear combination of two-centre integrals, the weighting factors being functions of p,q,r. Table A1.1 is taken from Table I of the Slater-Koster paper, and contains a sample of energy integrals in the two-centre approximation.

It is further necessary to distinguish between energy integrals $E_{l,l'}$ for nearest-neighbour (n.n.), second-n.n. and third-n.n. atoms. This is done by expressing $E_{l,l'}$ in the more detailed form $E_{l,l'}(n_{1j},n_{2j},n_{3j})$, where n_{1j},n_{2j},n_{3j} are integers which describe the position of an atom in the j^{th}-nearest neighbour shell in terms of the lattice constant a. For example, the nearest neighbours in an fcc lattice would give energy integrals of the form $E_{l,l'}(110)$. We can now rewrite equation (A1.4b) in the form:-

$$\langle B_l|H|B_{l'}\rangle = \sum_j e^{i\underset{\sim}{k}\cdot(\underset{\sim}{R}_j'-\underset{\sim}{R}_j)} E_{l,l'}(n_{1j},n_{2j},n_{3j}) \qquad (A1.7)$$

We note that exponents of the form $i\underset{\sim}{k}\cdot\underset{\sim}{R}_j$ can be rewritten in a more convenient form, viz.:-

$$i\underset{\sim}{k}\cdot\underset{\sim}{R} = i(n_{1j}ak_x + n_{2j}ak_y + n_{3j}ak_z)$$

$$\equiv i(n_{1j}\xi + n_{2j}\eta + n_{3j}\zeta) \qquad (A1.8)$$

In order to expand the $E_{l,l'}$ integrals in equation (A1.7) in terms of two-centre integrals, the definition of the latter must be extended to allow for *near-neighbour* interactions; specifically, we express the two-centre integrals in the more general form $(mn\lambda)_i$, where m,n

are s,p,d, λ is σ,π,δ and i is 1,2,3 (referring to nearest, second-nearest and third-nearest neighbours respectively).

We now clarify these concepts by means of an example. Consider the matrix element $\langle B_s|H|B_s\rangle$ ($l = l' = s$ in equation (A1.7)), for the case of an fcc lattice; if we abbreviate the notation for this matrix element to (s/s), then by equations (A1.7) and (A1.8) we can write:-

$$(s/s) = E_{s,s}(000) + 4E_{s,s}(110)(\cos\xi\cos\eta + \cos\xi\cos\zeta + \cos\eta\cos\zeta) + \\ + 2E_{s,s}(200)(\cos 2\xi + \cos 2\eta + \cos 2\zeta) \qquad \text{(A1.9a)}$$

which with reference to Table 3.1 becomes:-

$$(s/s) = E_{s,s}(000) + 4(ss\sigma)_1(\cos\xi\cos\eta + \cos\xi\cos\zeta + \cos\eta\cos\zeta) + \\ + 2(ss\sigma)_2(\cos 2\xi + \cos 2\eta + \cos 2\zeta) \qquad \text{(A1.9b)}$$

where $E_{s,s}(000)$ is the energy of the s state. The general case of equations (A1.9a) and (A1.9b), viz. that for the matrix element (l/l'), represents the main fruit of Koster and Slater's labour; this is because (l/l') is equivalent to an energy level calculated by a conventional BS technique such as the APW method, and hence we have a link between the LCAO parameters and detailed BS calculations done for simpler values of $\underline{k}$. One last step is required, viz. to ensure that the number of LCAO parameters is made equal to the number of states to be fitted so that the equations can be solved. Comparison

of Tables II and III in the Slater-Koster paper reveal that in general fewer two-centre integrals $(mn\lambda)_i$ are needed to describe a particular interaction than the more accurate $E_{l,l'}(n_{1j},n_{2j},n_{3j})$ integrals; for example, their Table II reveals that six independent $E_{l,l'}$ integrals are needed to describe the interaction between d electrons on second-n.n. atoms, while in Table III this interaction is represented by only three two-centre integrals. Thus by changing to a two-centre representation the number of parameters can usually be reduced; other approximations, for example neglect of certain off-diagonal matrix elements, allow for further reduction of the number of parameters. As discussed above, the number of these integrals can be increased by arbitrarily increasing the number of near-neighbour interactions.

Summary

The Slater-Koster interpolation method uses an LCAO approach to generate expressions for the energy matrix elements of a perfect crystalline solid. These expressions contain large numbers of complicated integrals, which for all practical purposes are impossible to evaluate rigorously. Consequently, these integrals are represented by parameters $E_{l,l'}$, where l and l' refer to a pair of interacting atomic-like orbitals. To further simplify matters, the two-centre approximation is introduced, which allows the parameters $E_{l,l'}$ to be expressed in terms of simpler integrals, of the type which describe interactions in diatomic molecules. The complete energy matrix elements are thus expressed as linear combinations of either the

parameters $E_{\ell,\ell'}$ or of the two-centre integrals (which are also taken to be parameters); the coefficients depend on components of the momentum vector $\underset{\sim}{k}$. The values of the energy matrix elements are known at certain high-symmetry values of $\underset{\sim}{k}$ (from band structure calculations), say n in number; it is then necessary to express these energy levels in terms of exactly n LCAO parameters. To achieve this, the number of these parameters can be varied in several ways: the number of near-neighbour interactions can be arbitrarily increased or decreased; the energy matrix elements can be arbitrarily expressed either in terms of the parameters $E_{\ell,\ell'}$ or in terms of two-centre integrals; and various approximations can be made, for example careful choice of orbitals $\{\phi_n\}$ and the neglect of certain off-diagonal matrix elements. Once these n LCAO parameters have been chosen, equations like equation (A1.9a) or (A1.9b) can be solved to obtain expressions for the parameters in terms of the components (k_x, k_y, k_z); substituting these into the original LCAO expressions for the energy matrix elements, we are finally left with interpolation formulae in the variables (k_x, k_y, k_z), which allow continuous plots of the energy bands to be drawn. We emphasize the fact that the parameters for which expressions are obtained have some physical significance, since they represent energy matrix elements; however they are by no means unique, depending as they do on the number n of band structure energy levels used as well as on the variation of these levels according to the BS technique employed.

REFERENCES (APP. 1)

A1.1) Slater J C and Koster G F 1954 *Phys. Rev.* 94 1498

A1.2) Löwdin P O 1950 *J. Chem. Phys.* 18 365

APPENDIX 2

NUMERICAL CONSIDERATIONS

A2.1 INTRODUCTION

The bulk of the calculations performed in the present work have required *integration* of either the Local Densities of States (for example equations (4.13) and (4.15)) or of the LDOS multiplied by energy (for instance equations (4.17)). We have already seen in Appendix 4.1 that the LDOS cannot be integrated analytically, and it can easily be inferred from that discussion that the same applies to the LDOS times energy; hence we have employed *numerical quadrature* for these calculations. It can be appreciated from our LDOS plots (see Figures 5.1 and 5.2) that our integrands are highly peaked, leading us to suspect the inadequacy of simple quadrature techniques for our purposes. In what follows we firstly compare various quadrature procedures, leading up to a choice of one particular procedure for all our calculations (Section A2.2). We proceed in Section A2.3 to explain how we have implemented this routine in our calculation of the Fermi energy. Then in Section A2.4 we test the sensitivity of the routine to the chosen *limits of integration*, which leads us in a natural manner to Section A2.5 in which we discuss the methods we have employed to fit *curves* to our discrete charge and energy results. In Section A2.6 we explain our choice of (α,p) values for our *two-phase energy minimization procedure* (Section 5.2.2) in the light of the computational requirements of this procedure; and finally in Section A2.7 we summarize the important issues discussed in this Appendix.

A2.2 CHOICE OF NUMERICAL QUADRATURE PROCEDURE

A2.2.1 Simple Composite Quadrature Methods

In the early stages of the present work we used a number of *composite quadrature methods* to evaluate our energies and charges. Such composite methods divide the interval of integration into several subintervals of *equal length*, and then apply a standard *quadrature rule* (such as *Simpson's rule* or *5-point Gaussian quadrature*) to each subinterval. For smooth, well-behaved integrands such techniques produce good results; however, they are less successful when dealing with integrands with *sharp peaks and singularities*. This is because the subintervals are of equal length, so that a highly-peaked region of the integrand is dealt with in the same way as a smooth, well-behaved region; thus if the number of subintervals is too small the peaked regions will not be integrated properly, while if the number of subintervals is sufficient to deal with the peaks, there will be much unnecessary work done on the well-behaved parts of the integrand. Furthermore, simple quadrature methods of this type are not equipped to deal with singularities in the integrand.

We recall that our heat of formation expressions are important in that they allow us to compare our electronic models directly with experiment; because of the subtle *cancellation effects* that occur in these exp. ssions (Section 4.5.3) we appreciate the importance of a *highly reliable* quadrature technique for our ΔH calculations. In Figure A2.1 we have plotted discrete values of ΔH (one-phase model, parameter set (a.1)) versus x, for which we have used a simple composite quadrature method. We see that the ΔH values generated in this way are numerically unstable; these instabilities can be

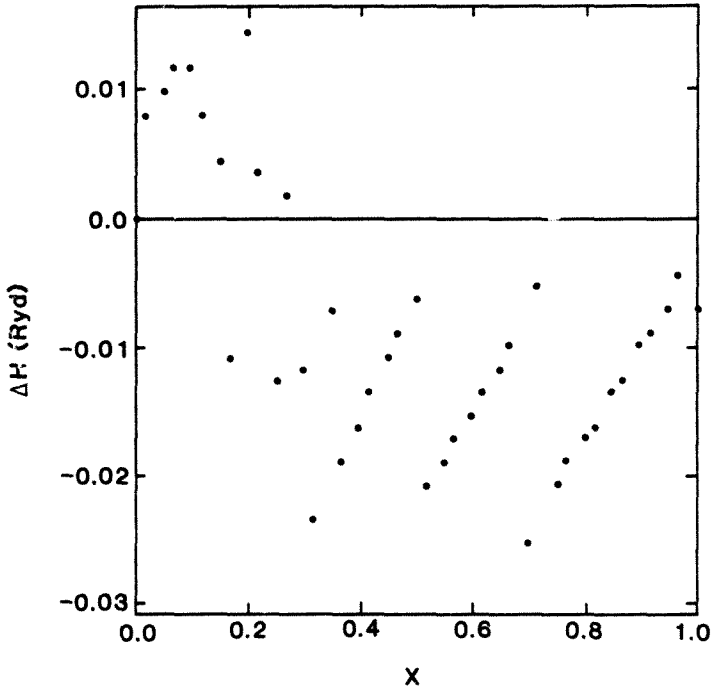


Figure A2.1 Heat of formation (one-phase model, prm.set (a.1)) vs x for simple composite quadrature method.

considerably reduced by increasing the number of subintervals by a factor of ten, but we recall from the above discussion that this is an *inefficient* means of evaluating the integrals. We will therefore proceed to consider more sophisticated approaches to numerical quadrature.

A2.2.2 *Adaptive Integration Methods*[A2.1]

Let us consider the integration of our Local Density of States or LDOS times energy over an energy interval (E1,E2); because the integrands are highly peaked over certain parts of the interval but relatively flat in other regions, we can appreciate the attractiveness of a quadrature method which would "work hard" on the subintervals where the function is less well-behaved while doing less work on the better-behaved regions. Such techniques are available and are referred to as *adaptive integration methods*; typically, such a method will divide the interval of integration into subintervals and apply a *sequence* of related quadrature rules to each subinterval. This sequence may consist for example of Romberg or Gaussian quadrature rules of increasing order, in which functional values evaluated for lower-order rules are used again in subsequent, more accurate approximations. The sequence of rules is continued either until the difference between successive approximations becomes less than a specified absolute (or fractional) value, or until all the rules have been implemented.

We have employed and compared three such adaptive integration methods, viz. the DCADRE package of IMSL, Inc[A2.2], and the D01AHF and D01AJF packages of the Numerical Algorithms Group[A2.3]. All three packages have the desired *relative error* in the integral as an input parameter; in addition,

considerably reduced by increasing the number of subintervals by a factor of ten, but we recall from the above discussion that this is an *inefficient* means of evaluating the integrals. We will therefore proceed to consider more sophisticated approaches to numerical quadrature.

A2.2.2 *Adaptive Integration Methods*[A2.1]

Let us consider the integration of our Local Density of States or LDOS times energy over an energy interval (E1,E2); because the integrands are highly peaked over certain parts of the interval but relatively flat in other regions, we can appreciate the attractiveness of a quadrature method which would "work hard" on the subintervals where the function is less well-behaved while doing less work on the better-behaved regions. Such techniques are available and are referred to as *adaptive integration methods*; typically, such a method will divide the interval of integration into subintervals and apply a *sequence* of related quadrature rules to each subinterval. This sequence may consist for example of Romberg or Gaussian quadrature rules of increasing order, in which functional values evaluated for lower-order rules are used again in subsequent, more accurate approximations. The sequence of rules is continued either until the difference between successive approximations becomes less than a specified absolute (or fractional) value, or until all the rules have been implemented.

We have employed and compared three such adaptive integration methods, viz. the DCADRE package of IMSL, Inc[A2.2], and the D01AHF and D01AJF packages of the Numerical Algorithms Group[A2.3]. All three packages have the desired *relative error* in the integral as an input parameter; in addition,

DCADRE and D01AJF allow the user to specify a desired *absolute error* value.

cautious adaptive Romberg extrapolation and is based on work by de Boor[A2.4)]

cautious adaptive Romberg extrapolation and is based on work by de Boor[A2.4)].

For a user-supplied function f(x) and integration limits a and b, DCADRE attempts to find a number INT which satisfies the following equation:-

$$\left| \int_a^b f(x)dx - \text{INT} \right| \leq \max \left(\text{ABSERR}, \text{RELERR} \times \left| \int_a^b f(x)dx \right| \right) \qquad \text{(A2.1)}$$

where:-

ABSERR = desired absolute error

RELERR = desired relative error

DCADRE first uses cautious Romberg extrapolation on the interval (a,b) to attempt to find a value for INT which satisfies equation (A2.1); if this attempt fails then the interval is divided into *two subintervals of equal length*, each of which is examined separately. This procedure of subdivision is continued until an acceptable value of INT is obtained or until the internal working storage allocated to the package is exhausted, in which case a specific error message is generated. DCADRE is often able to deal with integrands containing *jump discontinuities*.

We next consider D01AJF and D01AHF; the former is somewhat more *reliable* than the latter, though D01AHF tends to be more *efficient*[A2.3)]. The package D01AJF employs the *Gauss 10-point* and *Kronrod 21-point* quadrature

rules[A2.3] and is based on an algorithm described by de Doncker[A2.5]; it can be used for integrands with *singularities*, particularly when these are algebraic or logarithmic in nature. D01AJF attempts to satisfy an inequality very similar to that of equation (A2.1), in other words it allows the user to specify both absolute and relative error requirements.

D01AHF employs a sequence of *interlacing* high precision quadrature rules, making use of 1,3,7,15,31,63,127 and 255 nodes[A2.3] (that is 2^n-1 nodes, where n is the number of the rule); it is based on a technique described by Patterson [A2.6]. This package allows the user to specify a *relative error* only; when two successive rules in the sequence have a relative difference less than the specified relative error, then the value given by the last rule is taken as the value of the integral. If all the rules are applied without the desired result being obtained then the interval of integration is divided into two subintervals (*not necessarily of equal length*) and the sequence of rules is applied to each subinterval. This process of subdivision continues until either the desired accuracy is attained or until the work-space available to the package is exhausted.

D01AHF is only equipped to deal with sharp peaks and singularities at the *end points* of a given subinterval; such peaks and singularities are used as criteria for the selection of interval sizes in the process of subdivision.

In Table A2.1 we compare the results of the above three packages for the integral $\int_0^1 4/(1+x^2)dx$ which can be integrated analytically, giving the value $\pi(= 3.14159265358979)$. We see that for this smooth, well-behaved integrand, all three packages are highly reliable, giving values for π with relative errors far smaller than the user-supplied input relative errors.

Quadrature Package	Relative Error (Input)	Computed value of π	Actual Relative Error
DCADRE	10^{-6}	3.14159266	2.0×10^{-9}
	10^{-9}	3.1415926537	3.5×10^{-11}
D01AJF	10^{-6}	3.14159265	1.1×10^{-9}
	10^{-8}	3.1415926536	3.2×10^{-12}
D01AHF	10^{-6}	3.14159265	1.1×10^{-9}
	10^{-8}	3.1415926536	3.2×10^{-12}

Table A2.1 Values of π computed by numerical evaluation of $\int_0^1 4/(1+x^2)\,dx$ using three different quadrature packages.

A2.2.3 Choice of Adaptive Integration Package

We shall choose between the above three packages on the basis of both *efficiency* and *accuracy*. To compare the *efficiencies* we use each package in the same quadrature-intensive computer programme and record the CPU time required for each case. The results are displayed in Table A2.2, from which we see that DCADRE is distinctly more efficient than the other two packages.

Quadrature Package		DCADRE	D01AJF	D01AHF
CPU time (arbitrary units)	Rel. error = 10^{-3}	1.0	2.7	3.3
	Rel. error = 10^{-4}	1.4	4.6	5.5

Table A2.2 Relative efficiencies of different quadrature packages, measured in terms of CPU time required for a given set of calculations. "Rel. error" is the relative error requirement and is an input parameter for each package.

We next examine the relative *accuracies* of the three methods by employing each in the same *total electronic energy* calculation (two-phase model, prm.set (a.1)); we do this for *two* cases, viz. for relative error requirements of both 10^{-3} and 10^{-4}. The results are plotted in Figures A2.2 and A2.3 respectively. We see that for a relative error requirement of 10^{-3} (Figure A2.2) the package D01AJF gives a distinctly smoother curve than the other two packages, implying that it is decidedly more accurate for this fairly large choice of relative error. It is also clear that D01AHF gives the *least stable* results in this case; we recall that this particular package is only designed to deal with certain limited cases of sharp peaks and singularities, and hence we infer that it is the presence of such sharp peaks (and possibly singularities) in our LDOS that causes D01AHF to produce unstable results.

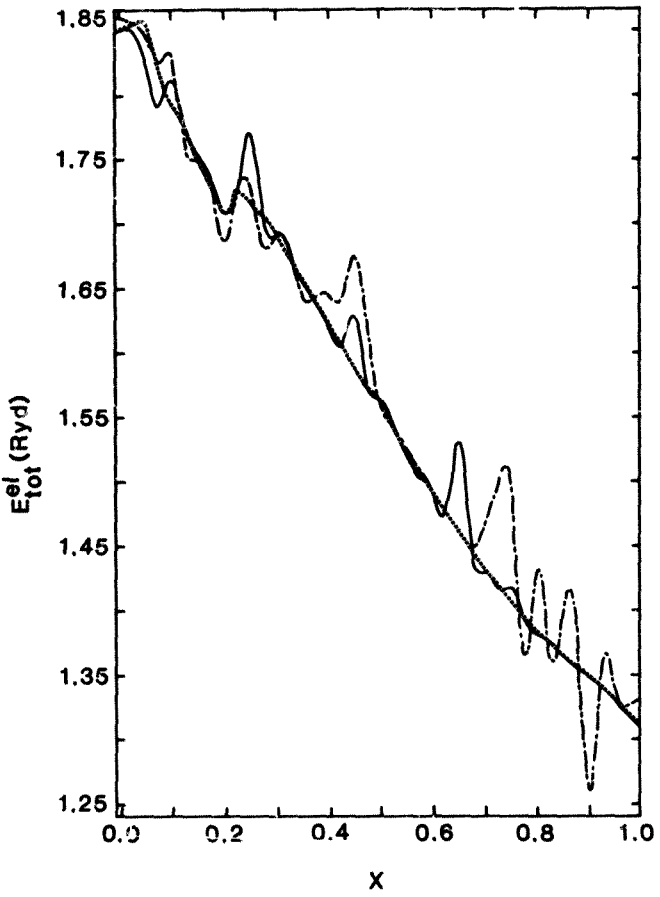


Figure A2.2 Total electronic energy (two-phase model, prm.set (a.1)) vs x for relative error requirement of 10^{-3}. Full curve: DCADRE; broken curve: D01AJF ; chained curve: D01AHF.

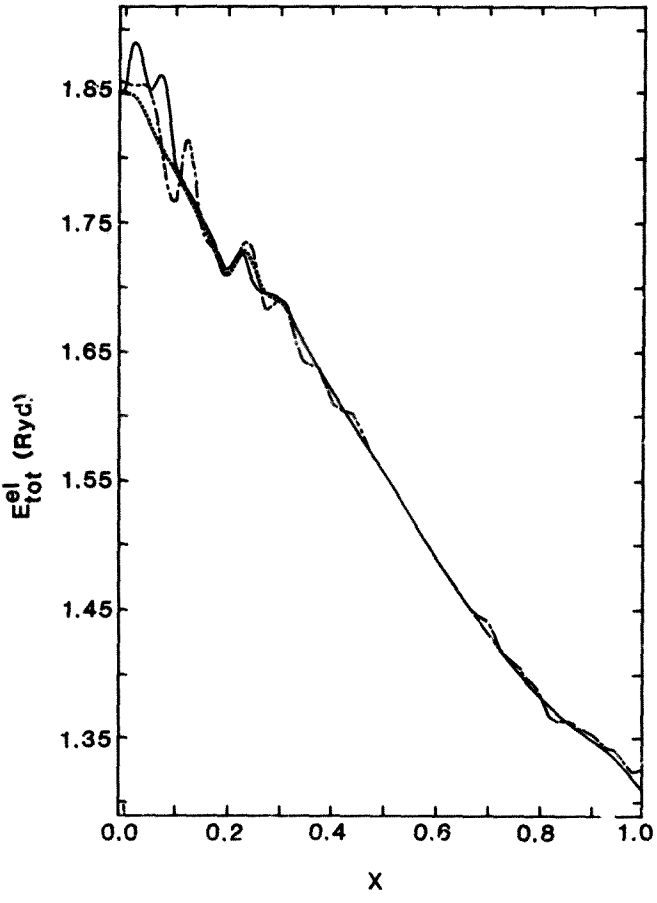


Figure A2.3 Total electronic energy (two-phase model, prm.set (a.1)) vs x for relative error requirement of 10^{-4}. Full curve: DCADRE; broken curve: *D01AJF*; chained curve: D01AHF.

From Figure A2.3 (relative error 10^{-4}) we see that the oscillations in DCADRE and D01AHF are considerably reduced, with the chief fluctuations occurring for *low* x. Furthermore, for $x \geq 0.3$ the values generated by DCADRE and D01AJF coincide on the scale of Figure A2.3. For the bulk of this work a relative input error of 10^{-5} was employed, and for this error requirement we would expect the agreement between DCADRE and D01AJF to be considerably improved for $x \leq 0.3$.

In summary, we firstly have that D01AHF is unsuitable for the integrals under consideration, from the points of view of both *efficiency* (Table A2.2) and *accuracy/stability*. Secondly, DCADRE is seen to be significantly more efficient than D01AJF (Table A2.2) as well as being of comparable accuracy for a relative error of 10^{-4}; we furthermore expect that the agreement between these two methods would be noticeably improved for a relative error requirement of 10^{-5} (which we have used throughout the present work). Recalling the considerable computational requirements of our two-phase model (a consideration which will be more fully appreciated in the following sections), we see that in our case the above arguments are strongly in favour of DCADRE; hence our choice of this package for all our Fermi energy, charge and total electronic energy calculations.

We now proceed to examine the implementation of DCADRE in our various energy calculations.

A2.3 EVALUATION OF THE FERMI ENERGY

We recall from Sections 4.4.2 and 5.2.2 that the Fermi energy was found by solving the following equation numerically for E_F:-

$$\int_{-\infty}^{E_F} n_{tot}(x,\varepsilon)d\varepsilon = \frac{10+x}{10+2x}\int_{-\infty}^{+\infty} n_{tot}(x,\varepsilon)d\varepsilon \qquad (A2.2)$$

We recall from our LDOS plots (for example, Figures 5.1 and 5.2) that $n_{tot}(x,\varepsilon)$ is only *non-zero* over a fairly constant energy interval, and hence we may replace the limits $(-\infty, +\infty)$ in equation (A2.2) by a convenient though *arbitrary* pair of *finite* numbers (E_1, E_2) so that equation (A2.2) becomes:-

$$\int_{E_1}^{E_F} n_{tot}(x,\varepsilon)d\varepsilon = \frac{10+x}{10+2x}\int_{E_1}^{E_2} n_{tot}(x,\varepsilon)d\varepsilon \qquad (A2.3)$$

We use the package DCADRE[A2.2] to evaluate the integrals in equation (A2.3); for clarity we therefore write:-

$$\int_{E_1}^{E_F} n_{tot}(x,\varepsilon)d\varepsilon \simeq \text{DCADRE}\ (n_{tot},\ E_1,\ E_F) \qquad (A2.4a)$$

and

$$\int_{E_1}^{E_2} n_{tot}(x,\varepsilon)d\varepsilon \simeq \text{DCADRE}\ (n_{tot},\ E_1,\ E_2) \qquad (A2.4b)$$

so that equation (A2.3) becomes:-

$$\text{DCADRE}\ (n_{tot},\ E_1,\ E_F) \simeq \left(\frac{10+x}{10+2x}\right)\ \text{DCADRE}\ (n_{tot},\ E_1,\ E_2) \qquad (A2.5)$$

We solve equation (A2.5) using a *bisection method*, the algorithm for which is given in Figure A2.4. Since our charge and energy calculations require the Fermi energy as a parameter it is clear that E_F must be determined accurately, that is the total number of iterations (NMAX in Figure A2.4) must be sufficiently large. At the same time the number of iterations should not be excessive because of the large number of E_F calculations required by the two-phase model (see Section A2.6).

To estimate a sensible value for the number of iterations NMAX, we have carried out a typical *total electronic energy* calculation using NMAX and the relative error as the only parameters; the results are displayed in Table A2.3.

Rel. error (Input)	No. Iterations (NMAX)	$^{+}E^{el}_{tot}$ (x=0.1) (Ryd)	Difference (Ryd)
10^{-4}	10	1.79102	+0.01889
	14	1.80991	-0.00001
	18	1.80990	
10^{-5}	10	1.79098	-0.00040
	14	1.79138	-0.00002
	18	1.79136	
10^{-6}	10	1.79094	+0.00045
	14	1.79139	-0.00001
	18	1.79138	

Table A2.3 Convergence of total electronic energy data as a function of the number of iterations employed to calculate the Fermi energy; the quadrature package DCADRE was used.

+1-Phase, prm.set (a.1)

$E_1 = -0.4$ TOP = 10.0

$E_2 = +0.6$ NMAX = 18

$E_{old} = 0.3$ N = 0

STEP = 0.1 $I_{tot} = (10+x)/(10+2x)DCADRE(n_{tot},E_1,E_2)$

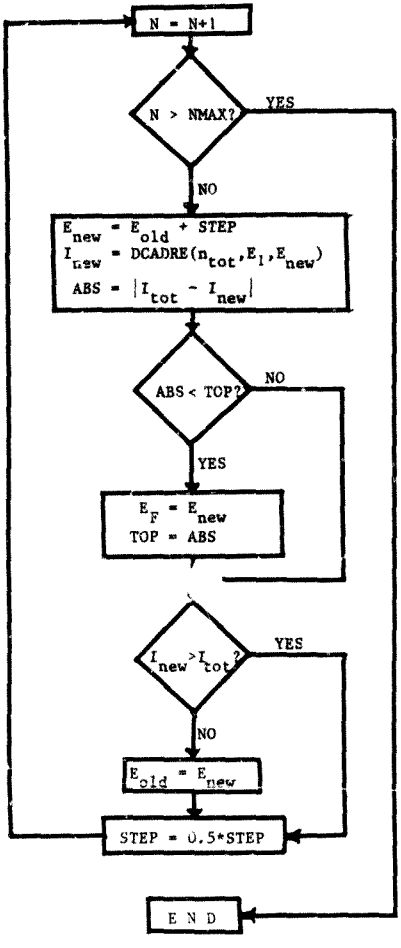


Figure A2.4 Algorithm and typical input data for the evaluation of the Fermi energy using DCADRE; energies are in Rydberg.

We see from this table that for all three relative errors chosen, the E_{tot}^{el} values for 14 and 18 iterations agree to five significant figures (cf agreement to only two or four significant figures between E_{tot}^{el} values for 10 and 14 iterations). We thus choose to use NMAX = 18 for all our E_F calculations, both one- and two-phase.

Having selected a suitable quadrature package and chosen an appropriate number of iterations for the evaluation of E_F by a bisection method, we now consider the effects of varying our arbitrary limits of integration.

A2.4 LIMITS OF INTEGRATION

From a purely *mathematical* point of view the various charge and energy calculations should be *identical* for all choices of integration limits (E_1, E_2) which satisfy the following conditions:-

$$\left.\begin{aligned} n_{tot}(x,\varepsilon) &= 0.0 \text{ for } \varepsilon < E_1 \\ n_{tot}(x,\varepsilon) &= 0.0 \text{ for } \varepsilon > E_2 \end{aligned}\right\} \qquad (A2.6)$$

However we note that *numerical* quadrature procedures evaluate the integrand at *discrete* intervals determined by some step-length h. Thus if we were for example to *increase* the interval of integration but keep the number of allowed functional evaluations constant, the procedure would re-evaluate the integrand at a *different* set of points. This would have a negligible effect if we were dealing with a well-behaved integrand; it might however make a very considerable difference if the integrand has singularities or very sharp peaks.

In order to investigate the effect of this numerical phenomenon on our calculations we have performed two complete two-phase total electronic energy calculations, both for parameter set (a.1); the only difference between these calculations is that the first uses a value for the *arbitrary lower limit of integration* of $E_1 = -0.4$ Ryd while the second uses $E_1 = -0.5$ Ryd. The results of these calculations are plotted in Figure A2.5 in which we see that for most values of x the two curves coincide (as we would expect from purely mathematical considerations); nevertheless at certain x values the curve corresponding to $E_1 = -0.5$ Ryd has very distinct *kinks* in it. The presence of these kinks suggests that we

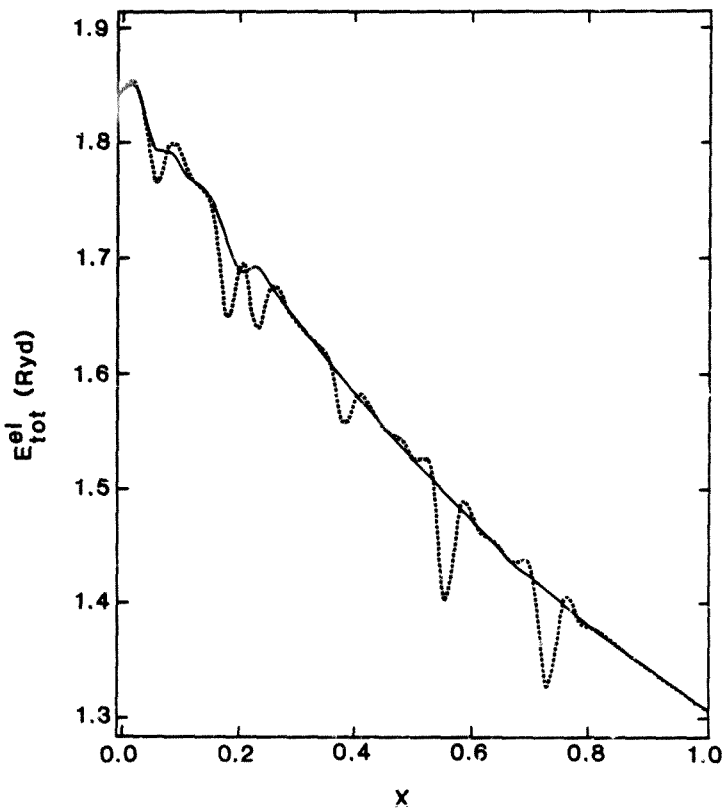


Figure A2.5 Total electronic energy (two-phase model, prm.set (a.1)) vs x for two different lower limits of integration (E1). Full curve: E1 = - 0.4 Ryd; broken curve: E1 = -0.5 Ryd.

should expect some numerical "noise" in all our energy calculations; however we observe that the shape of the physically more correct smooth curve (corresponding to E_l = -0.4 Ryd) is easily discernible from the *unperturbed points* of the E_l = -0.5 Ryd curve because these unperturbed points are in the majority.

The above observations now lead us to consider appropriate ways of *smoothing* our calculated curves so as to highlight the essential physics of these calculations.

A2.5 CURVE FITTING AND SMOOTHING

Our approach in plotting the various integrated quantities we have calculated has been to *exclude individual* points which are far removed from the trend of the data; we do this on the assumption that such isolated deviations are purely numerical in origin. Where there is clear *oscillatory* behaviour of *successive* points we have chosen *to dampen* the oscillations rather than smooth them out completely, thereby giving some idea of numerical effects without obscuring the underlying physics. This "damping" has in some cases been achieved by simply drawing a curve through points roughly midway between consecutive pairs of discrete data points; in other cases the *spline* or combined *spline-least-square* interpolation facilities of the SAS/GRAPH plotting routine GPLOT[A2.7] have been used (all the LDOS plots were done in this manner).

Isolated peaks and troughs in the data have been damped in the same way as the oscillatory behaviour.

In summary we emphasize that we have smoothed our various integrated quantities just sufficiently to reduce the more serious numerical effects, so that important physical trends are not obscured.

A2.5 CURVE FITTING AND SMOOTHING

Our approach in plotting the various integrated quantities we have calculated has been to *exclude individual* points which are far removed from the trend of the data; we do this on the assumption that such isolated deviations are purely numerical in origin. Where there is clear *oscillatory* behaviour of *successive* points we have chosen to *dampen* the oscillations rather than smooth them out completely, thereby giving some idea of numerical effects without obscuring the underlying physics. This "damping" has in some cases been achieved by simply drawing a curve through points roughly midway between consecutive pairs of discrete data points; in other cases the *spline* or combined *spline-least-square* interpolation facilities of the SAS/GRAPH plotting routine GPLOT[A2.7] have been used (all the LDOS plots were done in this manner).

Isolated peaks and troughs in the data have been damped in the same way as the oscillatory behaviour.

In summary we emphasize that we have smoothed our various integrated quantities just sufficiently to reduce the more serious numerical effects, so that important physical trends are not obscured.

A2.6 TWO-PHASE PARAMETER VALUES AND CPU TIMES

We recall from Section 5.2.1 that our two-phase model is based on the following equation and inequalities:-

$$x = \alpha p + (1-\alpha)q \qquad \text{(A2.7a)}$$

where $0 < \alpha < 1$ (A2.7b)

and $0 \leq \text{conc} \leq 1$ (A2.7c)

where in turn conc = x, p or q

For a given x we evaluate sets of parameters (α,p,q) subject to equations (A2.7b) and (A2.7c), and for each set we evaluate the total electronic energy, choosing the *lowest* value at the end of the calculation. Now equation (A2.7a) can be rewritten as $q = (x-\alpha p)/(1-\alpha)$, so that for a fixed value of x we only have *two* independent parameters, viz. α and p. Thus the number of energy calculations for each x value is equal to the number of (α,p) pairs we employ in the minimization procedure. For this reason we must choose a sufficiently large number of such pairs to make the minimization statistically meaningful yet not so large as to produce a computationally-intractable problem. We have employed the following values for all our two-phase calculations:-

α = 0.1, 0.2, 0.3,.....,0.9 (9 values)

p = 0.0, 0.1, 0.2,.....,1.0 (11 values)

giving a total of 99 (α,p) parameter sets. Although this α-p "grid" might appear to be a rather coarse "sieve" for finding the minimum energy, we must bear in mind the considerable computational requirements of this

procedure. Specifically, for each (α,p) set, DCADRE must be invoked 25 times (19 times to find E_F, 3 times for the charges and 3 times for the electronic energies). Bearing in mind that we have 99 (α,p) sets and 41 values of x (in steps of 0.025) we have a possible total of 101475 integrations to perform. It will be seen below that roughly a third of the (α,p) values are excluded by equations (A2.7), leaving us with about 68000 non-trivial integrals to evaluate per parameter set. In Table A2.4 we express this in terms of the CPU times required for these calculations for various parameter sets; the computations have all been performed on the IBM 3083 Model E8 mainframe, and the CPU times are measured in hours.

It will be appreciated from Table A2.4 that a finer "grid" of (α,p) values would be computationally prohibitive, at least for the numerical techniques we have employed.

Because the *one-phase* model requires only one set of calculations per x value, we would expect the two-phase model to be a maximum of 99 times more costly in terms of computer time. We see from Table A2.4 that it is in fact only about 66 times more costly, leading us to infer that only about two thirds of the possible (α,p) pairs satisfy equations (A2.7). Two clear trends emerge from the *two-phase* column of Table A2.4: firstly for a given V_{hh} value (refer to Table 5.1), the CPU times vary almost *inversely* with V_{hd}. This implies that the LDOS associated with smaller magnitudes of V_{hd} are more difficult to integrate, an inference which is confirmed by Figures 5.11; and secondly, we see that for a fixed V_{hd} value (Table 5.1) the *negative* choice of V_{hh} requires the least computational effort whereas the *positive* V_{hh} value

Parameter Set	CPU time (hours)		CPU(2-Ph.) / CPU(1-Ph.)
	1-Phase	2-Phase	
a.1	0.172	-	-
a.2	-	-	-
a.3	-	-	-
b.1	-	9.55	-
b.2	0.139	9.27	66.7
b.3	0.169	11.73	69.4
c.1	-	12.77	-
c.2	-	12.38	-
c.3	-	13.11	-
d.1	0.110	7.35	66.8
d.2	0.111	7.28	65.6
d.3	0.121	-	-

le A2.4 CPU times for both models and various parameter sets. Prm.sets (d.1)-(d.3) follow the same pattern as the others, with V_{hd} = 2 V (Tables 3.5 and 5.1)

requires the *most* work. This in turn implies that the negative V_{hh} value causes fewer numerical instabilities in DCADRE than does the positive one.

In summary, we have discussed some of the specific computational details of the two-phase calculations and have explained our choice of (α,p) parameters in terms of computational cost. We have also compared the CPU times for the complete one- and two-phase calculations for various parameter sets, and hence made inferences concerning the relative numerical stabilities of the associated calculations.

A2.7 SUMMARY OF APPENDIX 2

We have shown that simple *composite quadrature methods* are too inaccurate and inefficient for the numerical integration requirements of the present work, and have hence examined and compared three *adaptive integration methods*. The DCADRE[A2.2] package was found to have the best combination of accuracy and efficiency characteristics for our requirements, and has therefore been employed to perform all numerical quadrature in the present work.

DCADRE was implemented in a *bisection method* algorithm for evaluating the Fermi energy, leading to the choice of a sensible number of iterations for this algorithm. The effect of changing the (arbitrary) lower limit of integration was then examined; it was found that the *trend* of the calculations was unaffected, although certain individual points were highly perturbed. Consideration of such perturbations has allowed us to estimate the amount of numerical "noise" in our calculations, and hence to choose sensible strategies for the *smoothing* of our calculated curves.

Finally we have discussed the choice of parameters for our two-phase *energy minimization procedure* in terms of *computational costs* incurred, and have discerned certain numerical stability trends by comparing the CPU times required for calculations with various parameter sets.

REFERENCES (APP.2)

A2.1) Ralston A and Rabinowitz P 1978 *A First Course in Numerical Analysis* (2nd Ed.; McGraw-Hill Kogakusha)

A2.2) *The IMSL Library User's Manual* 1984 Edition 9.2 Vol.1 Chap. D (IMSL, Inc.)

A2.3) *NAG FORTRAN Library Manual* 1983 Mark 10 Vol.1 Chap. D01 (Numerical Algorithms Group)

A2.4) de Boor C 1971 *CADRE : An algorithm for numerical Quadrature* in *Mathematical Software* Chap. 7 (Ed.: Rice J R; publ.: Academic Press, New York)

A2.5) de Doncker E 1978 *An Adaptive Extrapolation Algorithm for Automatic Integration* in *Signum Newsletter* 13 No.2 12

A2.6) Patterson T N L 1968 *The Optimum Addition of Points to Quadrature Formulae* in *Math. Comp.* 22 847

A2.7) *SAS/GRAPH User's Guide* 1981 pp.15-16 (SAS Institute Inc.)





Author Anagnostaras Philip
Name of thesis Theoretical Study Of The Electronic Nature Of Non-stoichiometric Metal Hydrides. 1986

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.