

# *Electronic Structure of Multi-Electron Quantum Dots*

**Ranga D. Muhandiramge**  
**Jingbo Wang**

This article describes the use of the *SACI* package—a package for calculating the energy levels and wavefunctions of a multi-electron quantum dot modelled as a 2D harmonic well with electrons interacting through a Coulomb potential and under the influence of a perpendicular magnetic field.

## ■ Introduction

Quantum dots are artificially fabricated atoms, in which charge carriers are confined in all three dimensions just like electrons in real atoms. Consequently, they exhibit properties normally associated with real atoms such as quantised energy levels and shell structures. These properties are described by the electron wavefunctions whose evolution is governed by the Schrödinger equation and the Pauli exclusion principle.

There are many methods available to solve the Schrödinger equation for multiple electrons. They roughly fall into the categories of the diagonalisation method, mean-field density-functional theory, and the self-consistent field approach. One of the first theoretical studies of quantum dots was by Pfannkuche et al. [1], who compared the results of Hartree–Fock self-consistent calculations and exact diagonalisation of the Hamiltonian for two electrons in a circularly symmetric parabolic potential. They found good agreement between the two methods for the triplet state but marked differences for the singlet state, indicating important spin correlations were not included properly in their Hartree–Fock model. This suggests that the proper treatment of electron spins is crucial for correctly obtaining the electronic structures in quantum dots.

Examples of self-consistent field approaches in the literature include Yannouleas and Landman [2, 3], who studied circularly symmetric quantum dots using an unrestricted spin-space Hartree–Fock approach, and McCarthy et al. [4], who developed a Hartree–Fock *Mathematica* package. Macucci et al. [5] studied quantum dots with up to 24 electrons using a mean-field local-density-functional

approach, in which the spin exchange-correlation potential was approximated by an empirical polynomial expression. Lee et al. [6] also studied an  $N$ -electron quantum dot using density-functional theory, where the generalised-gradient approximation was used for exchange-correlation potentials. Exchange interaction comes directly from the antisymmetrisation of wavefunctions as required by Pauli's exclusion principle. In density-functional theory this is a major problem since the mathematical object is the electron density rather than the electron wavefunction, making evaluation of the exchange interaction intrinsically difficult.

Diagonalisation approaches in the literature include Ezaki et al. [7, 8], Eto [9, 10], Reimann et al. [11], and Reimann and Manninen [12], who each applied a brute force approach by numerically diagonalising the  $N$ -electron Hamiltonian using Slater determinants composed of single-electron eigenstates as the basis functions. This approach, namely the configuration interaction (CI) method, takes into account the full interaction and correlation of the electrons in the system as long as the numerical results converge with an increasing number of basis functions. However, such an approach involves the calculation of a very large number of interaction integrals and the inversion of large matrices, which can be prohibitively expensive in terms of computer resources. Reimann et al. [11] employed matrices of dimensions up to 108,375 with 67,521,121 nonzero elements for a six-electron quantum dot. Calculations for any higher number of electrons were not considered numerically viable using the conventional CI formalism, even with state-of-the-art computing facilities.

We have recently developed a spin-adapted configuration interaction (SACI) method to study the electronic structure of multi-electron quantum dots [13]. This method is based on earlier work by quantum chemist R. Pauncz [14], which expands the multi-electron wavefunctions as linear combinations of antisymmetrised products of spatial wavefunctions and spin eigenfunctions. The SACI method has an advantage over using Slater determinants in that a smaller basis is used. This reduces the computational resources required, allowing calculations for dots with more than six electrons on a desktop computer.

After some theoretical background, we present results from a *Mathematica* package which employs the SACI method to calculate the energy levels and wavefunctions of multiple electrons confined by a 2D harmonic well and under the influence of a perpendicular magnetic field. *Mathematica* enables the exact calculation of interaction integrals which greatly improves the speed and accuracy of calculations.

## ■ SACI Package

The *SACI* package encapsulates all the functionality needed to calculate the energy levels and wavefunctions for electrons in a 2D harmonic well potential with/without the influence of a perpendicular magnetic field.

Ensure that `SACI.m` is on your `$Path` and load the package.

```

In[1]:= AppendTo[$Path, DirectoryName[ToFileName[
      "FileName" /. NotebookInformation[EvaluationNotebook[]]]]];
In[2]:= << SACI `;

```

You can also choose Input  $\triangleright$  Get File Path to locate SACI.m on your computer. This package will be used in the Theory section to demonstrate results and produce examples, as well as in the Calculations section to calculate energy levels and wavefunctions.

## ■ Theory

### □ Spin Eigenfunctions

When measuring the component of an electron's spin angular momentum along an axis, the result is either  $+\frac{\hbar}{2}$  or  $-\frac{\hbar}{2}$ . Conventionally, we denote this axis as the  $z$ -axis and work in units in which  $\hbar$  is one. The two spin eigenstates in which we are certain of the electron's spin projection are denoted  $\alpha$  and  $\beta$ , which are the eigenfunctions of the projection  $\hat{S}_z$  operator. To find projected spin eigenfunctions for multiple electrons we can simply multiply single electron spin eigenstates (e.g.,  $\alpha\beta\beta\alpha\alpha$  or  $\beta\alpha\alpha\beta\alpha$  for five electrons). We call these simple multiplications of  $\alpha$ 's and  $\beta$ 's elementary spin eigenfunctions. Quantum mechanics also allows linear combinations of projected spin eigenfunctions (e.g.,  $\varphi = \frac{1}{2}\alpha\beta\beta\alpha\alpha + \frac{1}{2}\beta\alpha\alpha\beta\alpha$ ). As long as each term in the spin function has the same number of  $\alpha$ 's and  $\beta$ 's, we will still have an eigenfunction of  $\hat{S}_z$ , that is  $\hat{S}_z\varphi = M\varphi$ , where  $M$  is the spin projection quantum number given by half the number of spin up electrons minus half the number of spin down electrons ( $M = \frac{1}{2}$  in this case).

The square of the total spin is also quantised:

$$\hat{S}^2 X = S\left(S + \frac{1}{2}\right)X, \quad (1)$$

where  $X$  is a multi-electron spin eigenfunction (spin eigenfunction) and  $S$  is the total spin quantum number. Elementary spin eigenfunctions with the same number of  $\alpha$ 's and  $\beta$ 's will be eigenfunctions of  $\hat{S}_z$ , each with the same projected spin quantum number  $M$ , but they are not in general eigenfunctions of  $\hat{S}^2$ . Thus  $X$  will in general be a linear combination of elementary spin eigenfunctions.

One way to calculate the spin eigenfunction is to use the Dirac identity to write the  $\hat{S}^2$  operator in terms of permutations

$$\hat{S}^2 X = \sum_{i < j}^{N-1} \hat{P}_{ij} X + \frac{N}{4} (4 - N) X, \quad (2)$$

where  $N$  is the number of electrons confined in the quantum dot and  $\hat{P}_{ij}$  is the permutation operator. To form eigenfunctions of  $\hat{S}^2$ , we can calculate the action

of  $\hat{S}^2$  on each of the elementary spin eigenfunctions. We convert  $\hat{S}^2$  into a matrix with  $X$  the vector of coefficients of the elementary spin eigenfunctions in the linear combination. We can then solve the matrix eigenvalue problem to calculate all the spin eigenfunctions. There may be more than one eigenfunction for a given total spin quantum number, and so we carry out an orthonormalisation procedure to calculate an orthonormal spin eigenspace.

In the *SACI* package, we define  $N$  and  $M$  implicitly using `ElectronList`, in which 1 represents a spin up electron and 0 represents a spin down electron.

```
In[3]:= ElectronList = {1, 1, 0};
```

$N$  is the length of the list and  $M$  is defined by the number of ones and zeros in the list. In this case  $N = 3$  and  $M = \frac{1}{2}$ . We then define the total spin quantum number  $S$ .

```
In[4]:= S = 1/2;
```

$S$  must be compatible with the number of electrons and  $M$  to get meaningful results. With  $S$ ,  $M$ , and  $N$  in place we can calculate the spin basis and then display it.

```
In[5]:= CalculateSpinBasis[]
```

```
In[6]:= ShowSpinBasis[]
```

```
Out[6]//TableForm=
```

$$\frac{\langle \alpha, \beta, \alpha \rangle}{\sqrt{2}} - \frac{\langle \beta, \alpha, \alpha \rangle}{\sqrt{2}} - \sqrt{\frac{2}{3}} \langle \alpha, \alpha, \beta \rangle + \frac{\langle \alpha, \beta, \alpha \rangle}{\sqrt{6}} + \frac{\langle \beta, \alpha, \alpha \rangle}{\sqrt{6}}$$

We can see that we have a 2D spin eigenspace as there are two basis vectors produced. Calculating the spin eigenfunctions is a prerequisite to the rest of the calculations. Henceforth, we label spin eigenvectors  $X(N, S, M; i)$ , where  $N$  is the number of electrons,  $M$  and  $S$  are the spin quantum numbers, and  $i$  is an index to identify vectors in a multi-dimensional basis.

## □ Fock–Darwin Solutions

The solution for a single electron moving in a 2D harmonic well under the influence of a perpendicular magnetic field was solved independently by Fock [15] and Darwin [16]. With the Hamiltonian

$$\hat{H}_0 = \frac{1}{2m^*} \left( \hat{p} - \frac{e}{c} \hat{A} \right)^2 + \frac{1}{2} m^* \omega_0^2 \hat{r}^2 \quad (3)$$

we get the energy eigenvalues

$$E_{nm} = (2n + |m| + 1) \hbar \sqrt{\frac{1}{4} \omega_c^2 + \omega_0^2} - \frac{1}{2} m \hbar \omega_c \quad (4)$$

and the wavefunctions

$$\psi_{nm}(r, \theta) = \left( 2 k^{|m|+1} \frac{n!}{(n+|m|)!} \right)^{\frac{1}{2}} r^{|m|} e^{-\frac{k}{2} r^2} L_n^{|m|}(k r^2) \frac{1}{\sqrt{2\pi}} e^{im\theta}, \quad (5)$$

where  $m^*$  is the effective mass,  $\omega_0$  the harmonic well constant,  $\omega_c = \frac{eB}{m^*c}$  the cyclotron frequency,  $k = \frac{b}{2l_0}$  the effective harmonic well constant,  $l_0 = \sqrt{\frac{\hbar c}{eB}}$  the magnetic length,  $b = \sqrt{1 + 4 \frac{\omega_0^2}{\omega_c^2}}$ ,  $B$  the magnetic field, and  $L_n^{|m|}(x)$  the generalised Laguerre polynomials.

To coincide with experiments done on a circularly symmetric quantum, we will choose solid-state parameters appropriate for GaAs (i.e.,  $m^* = 0.067 m_e$  and  $\epsilon = 13.1 \epsilon_0$ , where  $m_e$  is the electron mass and  $\epsilon_0$  is the vacuum permittivity).

```
In[7]:= relativeeffectivemass = 0.067;
```

```
In[8]:= relativepermittivity = 13.1;
```

When we are working in effective atomic units, the unit of length is the effective Bohr radius. This is given by the function `lengthscale` with the result in nanometers.

```
In[9]:= lengthscale[relativeeffectivemass, relativepermittivity]
```

```
Out[9]= 10.3466
```

Here is a plot of the probability density for  $\psi_{2,1}(r, \theta)$  in GaAs in a harmonic well in which the gap between the first and second energy levels is 10 meV and there is no magnetic field. Fixing the first energy gap is a convenient way to define the harmonic well constant as it can be determined experimentally. We also set the magnetic field to 1 Tesla and plot six effective Bohr radii from the centre of the well.

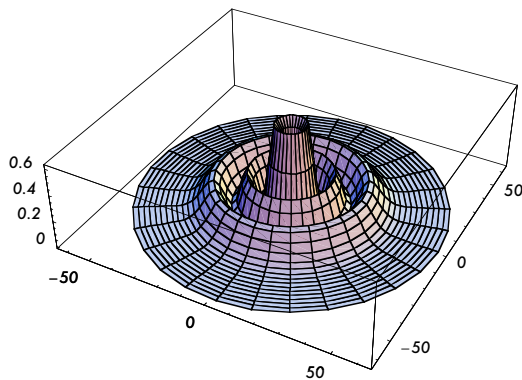
```
In[10]:= firstenergygap = 10.0;
```

```
magneticfield = 1;
```

```
In[12]:= SingleElectronProbabilityDensity[2, 1,
```

```
firstenergygap, magneticfield, relativeeffectivemass,
```

```
relativepermittivity, 6, BoxRatios -> {1, 1, .3}]
```



The base scale is in nanometers and indicates the scale of the quantum dot.

Multiplying one-electron functions together (e.g.,  $\Phi = \psi_{1,0} \psi_{1,0} \psi_{2,0} \psi_{2,0} \psi_{2,1}$ ) allows us to form multi-electron spatial wavefunctions that can be combined with the spin eigenfunctions. However, the combinations of spin and spatial wavefunctions must obey Pauli's exclusion principle as described in the next section.

## □ The Antisymmetry and Representation Matrices

The Pauli exclusion principle states that because electrons are indistinguishable fermions (half-integer spin particles), any wavefunction that describes the motion of electrons must change sign if both the spin and spatial coordinates of any pair of electrons are interchanged. Consequently in the  $N$ -electron case, if we apply any permutation  $P$  to the electrons, the following holds:

$$P\Psi = (-1)^p \Psi, \quad (6)$$

where  $\Psi$  is the multi-electron wavefunction and  $(-1)^p$  is the parity of the permutation. To antisymmetrise a wavefunction we apply the antisymmetrisation operator

$$A = \frac{1}{\sqrt{N!}} \sum_{P \in S_N} (-1)^p P, \quad (7)$$

where  $S_N$  is the symmetric group (the set of all permutations of  $N$  electrons under composition). A wavefunction must be an eigenfunction of the antisymmetrisation operator if it is to fully describe the physical properties of the electrons.

Since the permutation and spin operators commute, if we apply a permutation to a spin eigenfunction, the result is another spin eigenfunction. This new spin eigenfunction is not, in general, one of the functions already calculated but a linear combination of them. In fact, the coefficients of the linear combination form matrices which together form a representation of the symmetric group. Explicitly

$$P X(N, S, M; k) = \sum_{i=1}^d U(P)_{ik} X(N, S, M; i), \quad (8)$$

where  $X(N, S, M; k)$  is the  $k^{\text{th}}$  spin eigenfunction for  $N$  electrons with total spin quantum numbers  $S$  and  $M$  in an eigenspace of dimension  $d$ , and  $U(P)_{ik}$  is the  $(i, k)^{\text{th}}$  element of the representation matrix  $U(P)$ . It can be shown that for any two permutations  $R$  and  $P$

$$U(RP) = U(R)U(P), \quad (9)$$

where  $RP$  is the composite permutation derived by acting with  $P$  first then  $R$ . Equation (9) satisfies the condition for a representation of the symmetric group. These representation matrices and their properties are useful in simplifying the calculation of the Hamiltonian matrix.

We can see that our spin eigenfunctions are eigenfunctions of  $P_{1,2}$  by the representation matrix of  $P_{1,2}$ . The permutation is given by a reordering of the list  $\{1, 2, 3, \dots, N\}$ . For example, in a three-electron system,  $\{2, 1, 3\}$  represents the transposition of electrons 1 and 2 and  $\{1, 3, 2\}$  the transposition of electrons 2 and 3.

```
In[13]:= RepresentationMatrix[{2, 1, 3}] // TableForm
```

```
Out[13]//TableForm=
  -1    0
   0    1
```

This shows spin eigenfunction two is unchanged by the permutation and spin eigenfunction one changes sign under the permutation. In general the action of a permutation can lead to a linear combination of the spin eigenfunctions.

```
In[14]:= RepresentationMatrix[{1, 3, 2}] // TableForm
```

```
Out[14]//TableForm=
  1/2    -sqrt(3)/2
 -sqrt(3)/2  -1/2
```

This shows that under the permutation  $P_{2,3}$  the spin eigenfunctions transform via a linear combination. The following demonstrates the relation given by equation (9).

```
In[15]:= RepresentationMatrix[{2, 1, 3}] . RepresentationMatrix[{1, 3, 2}] ==
  RepresentationMatrix[{3, 1, 2}]
```

```
Out[15]= True
```

## □ Spin-Adapted Basis

To get an  $N$ -electron spin-spatial basis function we can multiply  $N$  single-electron wavefunctions, given by equation (5), combine them with a spin eigenfunction of  $\hat{S}^2$  and  $\hat{S}_z$ , and then antisymmetrise the result. The antisymmetrisation process restricts which spatial and spin combinations give nonzero results as discussed in detail in reference [13].

We use a five-electron quantum dot as an example, where the system wavefunction  $\Phi = \psi_{1,0} \psi_{1,0} \psi_{2,0} \psi_{2,0} \psi_{2,1}$  and there are two doubly occupied spatial orbitals. Two possible spin eigenfunctions for  $S = \frac{1}{2}$ ,  $M = \frac{1}{2}$  are

$$\begin{aligned} X(1) &= -\frac{1}{2} \alpha\beta\alpha\beta\alpha + \frac{1}{2} \alpha\beta\beta\alpha\alpha + \frac{1}{2} \beta\alpha\alpha\beta\alpha - \frac{1}{2} \beta\alpha\beta\alpha\alpha, \\ X(2) &= -\frac{\alpha\beta\alpha\alpha\beta}{\sqrt{3}} + \frac{\alpha\beta\alpha\beta\alpha}{2\sqrt{3}} + \frac{\alpha\beta\beta\alpha\alpha}{2\sqrt{3}} + \frac{\beta\alpha\alpha\alpha\beta}{\sqrt{3}} - \frac{\beta\alpha\alpha\beta\alpha}{2\sqrt{3}} - \frac{\beta\alpha\beta\alpha\alpha}{2\sqrt{3}}. \end{aligned} \quad (10)$$

Both  $X(1)$  and  $X(2)$  are antisymmetric under permutation  $P_{1,2}$ , but  $X(1)$  is antisymmetric under permutation  $P_{3,4}$ , while  $X(2)$  is symmetric under permutation  $P_{3,4}$ . This means  $A\Phi X(2)$  vanishes but  $A\Phi X(1)$  is nonzero.

The `CalculateSpinBasis` function applies a procedure to ensure the spin basis functions are also eigenfunctions of the permutations  $P_{1,2}, P_{3,4}, \dots$ . This property is required for the Hamiltonian matrix element formulae, described in the Matrix Elements Rules section, to be valid. A basis whose spin eigenfunctions are simultaneously eigenfunctions of  $P_{1,2}, P_{3,4}, \dots$  are guaranteed to exist as the permutation operators together with  $\hat{S}^2$  and  $\hat{S}_z$  pairwise commute.

## □ Diagonalisation

The multi-electron Hamiltonian is given as the following:

$$\hat{H} = \sum_{i=1}^N \hat{H}_{0i} + \sum_{j=2}^N \sum_{i=1}^j \frac{e^2}{4\pi\epsilon} \frac{1}{|\vec{r}_i - \vec{r}_j|}. \quad (11)$$

This is simply the sum of the single-electron energies defined by equation (3) plus the pairwise Coulomb interactions. The fact that we are working in a solid state media is modelled by using the relative effective mass  $m^*$  and the effective permittivity  $\epsilon$ .

We wish to solve the following equation:

$$\hat{H} \Psi = E \Psi \quad (12)$$

for the energy levels  $E$  and wavefunctions  $\Psi$ . The standard diagonalisation method is to approximate  $\Psi$  as a sum of a finite sum of basis functions with the convergence improving as we increase the size of the basis. This converts the problem into a matrix eigenvalue problem. However, to get a good approximation to the solution we need to choose the basis wisely as there is a limit to the size of the matrix that is computationally feasible. We also need to be able to efficiently calculate the Hamiltonian matrix elements.

## □ Basis Choice

The SACI method uses normalised, antisymmetrised products of spin eigenfunctions and spatial wavefunctions given by products of Fock–Darwin solutions as basis elements. The elements of our Hamiltonian matrix are then  $\langle N_\Phi A\Phi X(i) | \hat{H} | N_\Psi A\Psi X(j) \rangle$  for various spatial wavefunctions  $\Phi$  and  $\Psi$ , spin eigenfunctions  $X(i)$  and  $X(j)$ , and normalisation constants  $N_\Phi$  and  $N_\Psi$ .

The SACI method reduces the number of basis elements required by ensuring the wavefunction satisfies the Pauli exclusion principle from the outset and restricting the calculation to specific spin quantum numbers  $S$  and  $M$ .

Although each of the basis elements is a sum of  $N!$  terms, the properties of the antisymmetriser and the spin eigenfunctions allow enormous simplification of the calculation similar to the Slater–Condon rules for matrix elements between Slater determinates. In effect, only the product spatial wavefunction and spin eigenfunction need to be specified for each basis element with the normalisation



constant and antisymmetrisation built into the rules for calculating the Hamiltonian elements.

We need to be careful in the selection of basis elements, however, as we need an orthonormal basis for the diagonalisation expansion to be valid. Remember, some spin and spatial combinations vanish and thus cannot be included in the basis. We must also make sure the set of basis elements is mutually orthogonal and normalised (i.e.,  $\langle N_\Phi A\Phi X(i) | N_\Psi A\Psi X(j) \rangle = 1$  if  $\Phi = \Psi$  and  $i = j$ , and 0 otherwise). To do this, we select only spatial wavefunctions that have doubly occupied orbitals in the first and second position, the third and fourth position, ... to match the spin eigenfunctions symmetry or antisymmetry under the operators  $P_{1,2}$ ,  $P_{3,4}$ , ... . The properties of the antisymmetriser then guarantee the production of a complete orthonormal basis.

The *SACI* package automates the process of the orthonormal basis generation. As we can select only a finite number of basis elements out of an infinite set, we need to have some method of selecting appropriate basis elements. As we are usually interested in the ground and first few excited states of a quantum dot, we have chosen to use basis elements in increasing order of the sum of their component one-electron energies.

Elaborating, with no magnetic field the energy levels of the Fock–Darwin wavefunctions are given by  $E_{nm} = (2n + |m| + 1)E_0$ . If there were no interaction between the electrons, the energy of a multi-electron state would simply be the sum of the one-electron energies, for example, the energy of  $\psi_{1,0} \psi_{0,-1} \psi_{2,1}$  would be  $(2 \times 1 + |0| + 1)E_0 + (2 \times 0 + |-1| + 1)E_0 + (2 \times 2 + |1| + 1)E_0 = 11E_0$ . We select a basis consisting of all those elements with a noninteracting energy less than a certain cutoff.

```
In[16]:= cutoffenergy = 5;
```

The variable `cutoffenergy` is set to a cutoff energy of  $5E_0$ . We can then create a basis.

```
In[17]:= BasisCreate[cutoffenergy]
```

```
Out[17]= {{0, 0}, {0, 0}, {0, -1}}, 1},
  {{0, 0}, {0, 0}, {0, 1}}, 1}, {{0, 0}, {0, -1}, {0, 1}}, 1},
  {{0, 0}, {0, -1}, {0, 1}}, 2}, {{0, 0}, {0, 0}, {0, -2}}, 1},
  {{0, 0}, {0, 0}, {0, 2}}, 1}, {{0, 0}, {0, 0}, {1, 0}}, 1},
  {{0, -1}, {0, -1}, {0, 0}}, 1}, {{0, 1}, {0, 1}, {0, 0}}, 1}
```

`BasisCreate` produces a list of basis elements with noninteracting energy less than the specified cutoff. The spatial component of each basis element is given by a list of pairs of integers denoting the quantum numbers  $n$  and  $m$  for each orbital. The number at the end of the list denotes the index of the spin eigenfunction (its position when `ShowSpinStatus` runs) with which it is combined.

We notice that spin eigenfunction “1” is antisymmetric with respect to the permutation  $P_{1,2}$ , so it can be combined with the doubly occupied orbitals, but spin eigenfunction “2” is symmetric with respect to  $P_{1,2}$ , so it can only be combined with nondoubly occupied orbitals.

## □ Matrix Elements Rules

As mentioned previously, there are general rules for calculating Hamiltonian matrix elements. These rules are coded in the function `InteractionHamiltonianMatrixElement`, and their derivation can be found in [13]. In brief, when the spatial wavefunctions of the two basis elements differ by more than two orbitals (i.e., after being sorted so the orbitals are in maximum correspondence), then the matrix element is zero; when the spatial orbitals differ by two or less orbitals, a different rule applies for each of the three cases (i.e., zero, one, and two orbitals differ). These formulas are quite general and can be used for any interaction Hamiltonian that acts pairwise.

In the Hamiltonian element formulas we must calculate the two-electron interaction integral. For the case of electrons in a 2D harmonic well interaction through a Coulomb potential under the influence of a perpendicular magnetic field, the integral is in the following form:

$$\int_0^\infty \int_0^\infty \int_0^{2\pi} \int_0^{2\pi} r s \frac{\psi_{nm}^*(r, \theta) \psi_{pq}^*(s, \phi) \psi_{ab}(r, \theta) \psi_{cd}(s, \phi)}{\sqrt{r^2 + s^2 - 2 r s \cos(\theta - \phi)}} d\theta d\phi dr ds, \quad (13)$$

where

$$\psi_{nm}(r, \theta) = \left( 2 k^{|m|+1} \frac{n!}{(n + |m|)!} \right)^{\frac{1}{2}} r^{|m|} e^{-\frac{k}{2} r^2} L_n^{|m|}(k r^2) \frac{1}{\sqrt{2\pi}} e^{i m \theta}. \quad (14)$$

We need to be able to do this integral for many combinations of the quantum numbers  $n$ ,  $m$ ,  $p$ ,  $q$ ,  $a$ ,  $b$ ,  $c$ , and  $d$ . We notice that when  $r = s$  and  $\theta = \phi$ , we have a singularity as the  $\frac{1}{r^2}$  Coulomb interaction goes to infinity as the electrons move closer together.

However by using *Mathematica* and a little ingenuity, we can do this integral exactly in all cases needed in computation. The detailed derivation (breaking down and transforming the integral into a sum of components) can be found in [13]. For example, we set the quantum numbers  $n$ ,  $m$ ,  $p$ ,  $q$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  in order.

```
In[18]:= InteractionIntegral[2, 1, 2, 1, 0, 0, 0, 2]
```

```
Out[18]=  $\frac{735 \sqrt{\pi}}{32768}$ 
```

We also note that if  $b + d \neq m + q$  (i.e., the sum or the orbital angular momentum quantum numbers are not equal), the interaction integral is zero.

```
In[19]:= InteractionIntegral[2, 1, 2, 1, 0, 1, 0, 2]
```

```
Out[19]= 0
```

These integrals are done with the effective harmonic well constant  $k$  set to one. Values of the interaction integral for the effective harmonic well constant  $k = 1$  can easily be transformed to any value of  $k$ . Thus when we do an interaction integral, it is done for  $k = 1$  and transformed to the correct value of  $k$  afterwards. This allows us to store interaction integral values for various combinations of

$n$ ,  $m$ ,  $p$ ,  $q$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  as function definitions and save them to use later. This process is done dynamically so that values are saved as definitions only if they are used.

Here are some examples of interaction Hamiltonian matrix elements indicating that they can be done exactly.

```
In[20]:= InteractionHamiltonianMatrixElement[
  <{0, 0}, {0, 0}, {0, -1}>, 1, <{0, 0}, {0, 0}, {0, -1}>, 1]
```

$$\text{Out[20]} = \frac{9\sqrt{\frac{\pi}{2}}}{4}$$

```
In[21]:= InteractionHamiltonianMatrixElement[
  <{0, 0}, {0, 0}, {1, 1}>, 1, <{0, 0}, {0, 0}, {0, 1}>, 1]
```

$$\text{Out[21]} = \frac{7\sqrt{\pi}}{32}$$

```
In[22]:= InteractionHamiltonianMatrixElement[
  <{0, 0}, {0, 0}, {0, 1}>, 1, <{0, 0}, {0, 0}, {0, 2}>, 1]
```

$$\text{Out[22]} = 0$$

It can also be proved that the interaction element is zero if the sum of the orbital angular momentum quantum numbers are not the same, hence, the last calculation leads to zero.

```
In[23]:= InteractionHamiltonianMatrixElement[
  <{0, 0}, {0, -1}, {0, 1}>, 2, <{0, 0}, {0, 0}, {1, 0}>, 1]
```

$$\text{Out[23]} = -\frac{\sqrt{3}\pi}{32}$$

```
In[24]:= InteractionHamiltonianMatrixElement[
  <{0, 0}, {0, -1}, {0, 1}>, 1, <{0, 0}, {0, 0}, {1, 0}>, 1]
```

$$\text{Out[24]} = -\frac{\sqrt{\pi}}{32}$$

The preceding calculations show that the spin eigenfunction with which a spatial wavefunction is combined makes a nontrivial contribution to the Hamiltonian matrix element calculation.

## ■ Calculations

In this section we will demonstrate how the package can be used to do various calculations such as energy levels, wavefunctions, convergence plots, and energy level evolutions with magnetic fields.

### □ Energy Levels

To calculate energy levels, we need to choose a basis. For larger basis sizes the calculation will take longer and require more memory (the size of the matrix is the size of the basis squared) but the convergence will be better.

```
In[25]:= cutoffenergy = 12;
         L = 0;
```

```
In[27]:= Length[basisLzero = BasisCreateOneL[cutoffenergy, L]]
```

```
Out[27]= 123
```

We choose a basis with a single total orbital angular momentum quantum number  $L$ . The properties of the circularly symmetric quantum dot allow us to separate calculations for different values of  $L$ , thus reducing the size of the basis needed for each calculation allowing for greater accuracy. We then calculate the interaction Hamiltonian using the function `InteractionHamiltonian`.

```
In[28]:= LoadIntegralandHamiltonianData[]
```

```
File 'Ideffile' not found in directory:
/Users/TMJ - File being created.
```

```
The size of the Hamiltonian matrix
element definition file 'Ideffile' is 52 Bytes.
```

```
In[29]:= Timing[Hint = InteractionHamiltonian[basisLzero] ]
```

```
Out[29]= {41.328 Second, SparseArray[<7557>, {123, 123}]}
```

```
In[30]:= SaveIntegralandHamiltonianData[];
```

The calculation of this matrix can be time-consuming, so it is worthwhile to save the result, as the interaction Hamiltonian calculated for a particular number of electrons and spin quantum numbers  $S$  and  $L$  can then be used for any value of the harmonic well constant, magnetic field, relative effective mass, and relative permittivity. We have also loaded previously calculated integral definitions using the function `LoadIntegralandHamiltonianData` to speed up our calculation and saved the data afterward using `SaveIntegralandHamiltonianData` to speed up future calculations run in other sessions.

Next we set the magnetic field strength to 0 Tesla remembering that `firstener`·`gygap` was set to 10 meV previously and `relativeeffectivemass` and `relative`·`permittivity` are set to the parameters for GaAs.

```
In[31]:= magneticfield = 0;
```

We then can calculate the Hamiltonian for GaAs with these particular values of the harmonic well constant and magnetic field and plot the energy levels.

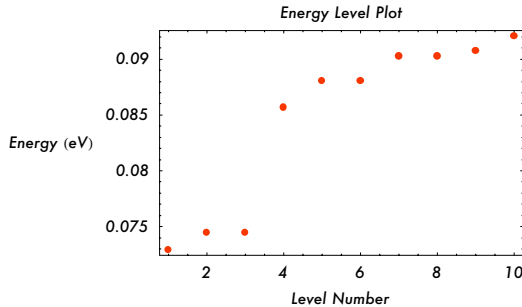
```
In[32]:= H = TotalHamiltonian[Hint, basisLzero, firstenergygap,
         magneticfield, relativeeffectivemass, relativepermittivity];
```

```
In[33]:= numberoflevels = 10;
```

```

In[34]:= EnergyLevelPlot[H, numberoflevels,
  relativeeffectivemass, relativepermittivity,
  PlotStyle -> {PointSize[.02], Hue[.02]}, Frame -> True,
  FrameLabel -> {"Level Number", "Energy (eV)", "Energy Level Plot", ""},
  RotateLabel -> False, Axes -> False]

```



The plot shows the first 10 energy levels for a GaAs quantum dot system of three electrons with spin quantum numbers  $S = \frac{1}{2}$ ,  $M = \frac{1}{2}$ , and total angular momentum quantum number  $L = 0$  in a zero magnetic field. The harmonic well constant is implicitly defined by a first energy gap of 10 meV.

## □ Convergence Plots

As mentioned, the accuracy of our result as well as the computational time required increases with the size of the basis. We can test the convergence with the following series of functions.

As the interaction Hamiltonian for lower cutoff energies is a submatrix of interaction Hamiltonians with higher cutoff energies, we can do convergence tests using a single interaction Hamiltonian. We will use the interaction Hamiltonian from the previous section.

The Convergence function takes submatrices of the Hamiltonian using different basis sizes as specified by the input `sizelist` and calculates the energy of the levels specified by `wantedlevels`. The ground state is specified by 1, the first excited state by 2, and so on. This allows the convergence properties of the solution to be ascertained.

The list basis size can be specified manually, but a convenient method is to choose all the basis sizes that correspond to the different cutoff energies.

```

In[35]:= wantedlevels = {1, 2};
cutoffmin =
  Catch[Do[If[Length[BasisCreateOneL[cut, L]] > Max[wantedlevels],
    Throw[cut]], {cut, cutoffenergy}]];
cutoffmax = cutoffenergy;

```

In the preceding code, we have specified `wantedlevels` to examine the ground state and first excited state. We set `cutoffmin` to be the first cutoff energy that

will produce the highest energy level requested, as we can only approximate up to  $N$  levels when we have  $N$  basis elements.

```
In[38]:= cutoffbasis = Table[Length[BasisCreateOneL[Energy, L]],
      {Energy, cutoffmin, cutoffmax, 2}]
```

```
Out[38]:= {3, 15, 48, 123}
```

As there may be large gaps in the basis sizes, we can also add in points at intermediate intervals. Here, we add a basis size of 85.

```
In[39]:= sizelist = Sort[cutoffbasis~Join~{85}]
```

```
Out[39]:= {3, 15, 48, 85, 123}
```

We can then calculate the energies of the ground state and first excited state as we increase the basis size. The energies are in meV.

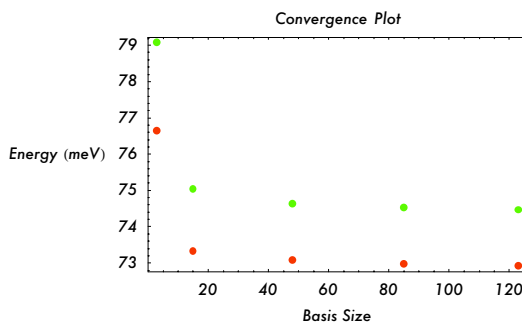
```
In[40]:= (conlist =
      1000 Transpose[Convergence[Hint, basisLzero, firstenergygap,
      magneticfield, relativeeffectivemass, relativepermittivity,
      L, sizelist, wantedlevels]] ) // TableForm
```

```
Out[40]//TableForm=
```

76.6437	73.3202	73.0706	72.9689	72.912
79.0659	75.0315	74.6208	74.529	74.4597

The following plot shows convergence of energies of the two lowest states.

```
In[41]:= g1 = ListPlot[Transpose[{sizelist, conlist[[1]]}], PlotStyle →
      {PointSize[.02], Hue[.02]}, DisplayFunction → Identity];
g2 = ListPlot[Transpose[{sizelist, conlist[[2]]}], PlotStyle →
      {PointSize[.02], Hue[.3]}, DisplayFunction → Identity];
Show[g1, g2, DisplayFunction → $DisplayFunction, Frame → True,
      FrameLabel → {"Basis Size", "Energy (meV)", "Convergence Plot", ""},
      RotateLabel → False, Axes → False]
```



We can see that the convergence is quite good for relatively small basis sizes in this case.

## □ Electron Density Plots

We can extract electron density plots for various energy levels by calculating the normalised energy eigenvectors of the full Hamiltonian and using the function

ElectronDensity. We can then plot the electron density and use LengthScale to give the length dimensions in nanometers.

First we calculate the energy eigenvectors for the first six energy levels.

```
In[44]:= highestlevel = 6;
```

```
In[45]:= energyeigenvectors = Eigenvectors[H, -highestlevel];
```

Next we use ElectronDensityFunction to convert the normalised energy eigenvector into a probability density function by combining it with the basis basisLzero. This is then simplified to cancel out the complex components. As the density function is a measurable quantity, we are guaranteed that the complex components will cancel. Chop is used to remove any complex components that remain due to rounding.

```
In[46]:= ElectronDensityFunctionTable = Table[ElectronDensityFunction[
  Normalize[energyeigenvectors[highestlevel - n]], basisLzero,
  r,  $\alpha$ , firstenergygap, magneticfield, relativeeffectivemass,
  relativepermittivity], {n, 0, highestlevel - 1}];
```

```
In[47]:= ElectronDensityFunctionTableSimplified = Table[
  Expand[ComplexExpand[ElectronDensityFunctionTable[[n]] // Chop]],
  {n, 1, highestlevel}];
```

We can see that we end up with a numerical approximation to the wavefunction. Here is an example.

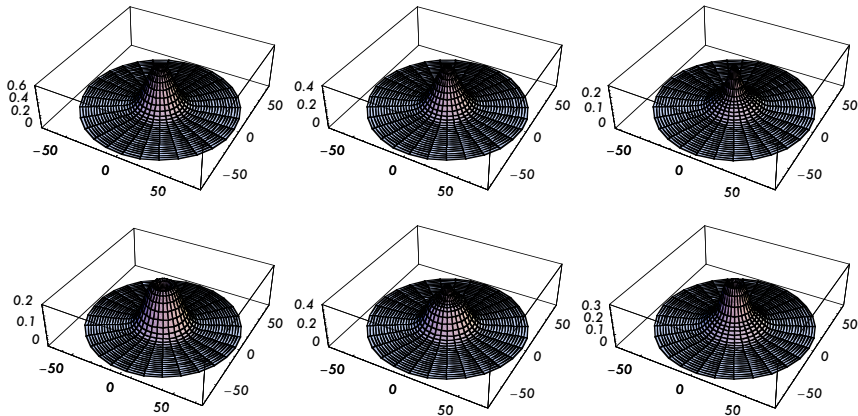
```
In[48]:= ElectronDensityFunctionTableSimplified[[1]]
```

```
Out[48]= 0.647537 e-0.941276 r2 + 0.452761 e-0.941276 r2 r2 + 0.0693917 e-0.941276 r2 r4 +
  0.0160081 e-0.941276 r2 r6 + 0.000113611 e-0.941276 r2 r8 +
  9.19624 × 10-6 e-0.941276 r2 r10 + 7.84521 × 10-7 e-0.941276 r2 r12 -
  6.72418 × 10-11 e-0.941276 r2 r14 + 7.01335 × 10-11 e-0.941276 r2 r16
```

The following plot shows the electron probability density functions for the first six energy levels with  $S = \frac{1}{2}$ ,  $M = \frac{1}{2}$ , and  $L = 0$  for the three-electron quantum dot system that has been the subject of our examples.

```
In[49]:= WavefunctionTable =
  Table [ParametricPlot3D[{r LengthScale[relativeeffectivemass,
    relativepermittivity] Cos[ $\alpha$ ],
    r LengthScale[relativeeffectivemass, relativepermittivity]
    Sin[ $\alpha$ ], ElectronDensityFunctionTableSimplified[[3 m + n]]},
  {r, 0, 7}, { $\alpha$ , 0, 2  $\pi$ }, BoxRatios → {1, 1, .3},
  DisplayFunction → Identity], {m, 0, 1}, {n, 1, 3}];
```

```
In[50]:= Show[GraphicsArray[WavefunctionTable]]
```



## □ Energy Levels versus Magnetic Fields

The *SACI* package also allows the calculation of the evolution of the ground state with a magnetic field using a single interaction Hamiltonian.

We can use `LevelVsField` to calculate a list of energy levels for a range of magnetic fields defined by `magmin`, `magmax`, and `step` given in Tesla, and for the levels defined by `wantedlevels`. Here we choose the first six energy levels and plot for the magnetic field range of 0 to 4 Tesla at intervals of every 0.5 Tesla.

```
In[51]:= magmin = 0;
magmax = 4;
step = .5;
wantedlevels = {1, 2, 3, 4, 5, 6};

In[55]:= maglevlist =
Transpose[LevelVsField[Hint, basisLzero, firstenergygap,
relativeeffectivemass, relativepermittivity,
L, magmin, magmax, step, wantedlevels]]

Out[55]= {{0, 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4.},
{0.072912, 0.0729707, 0.0731466, 0.0734385,
0.0738447, 0.0743628, 0.07499, 0.0757227, 0.0765571},
{0.0744597, 0.0745195, 0.0746985, 0.0749957, 0.0754092,
0.0759367, 0.076575, 0.0773208, 0.07817},
{0.0744597, 0.0745195, 0.0746985, 0.0749957, 0.0754092,
0.0759367, 0.076575, 0.0773208, 0.07817},
{0.0856832, 0.0857561, 0.0859744, 0.0863367, 0.086841,
0.0874844, 0.0882636, 0.0891743, 0.090212},
{0.0880509, 0.088125, 0.088347, 0.0887156, 0.0892286,
0.0898831, 0.0906756, 0.0916019, 0.0926572},
{0.0880509, 0.088125, 0.088347, 0.0887156, 0.0892286,
0.0898831, 0.0906756, 0.0916019, 0.0926572}}
```



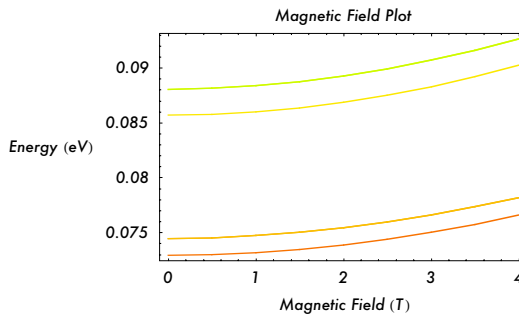
Calculating the energy levels for one  $L$  value at a time has the benefit of producing more accurate results for a given matrix size and allows for easier separation of different energy states from the table of eigenvalues. The following plot shows the results.

```

In[56]:= magplot =
  Table[ListPlot[Transpose[{maglevlist[[1]], maglevlist[[n]]}],
    PlotStyle -> {PointSize[.02], Hue[.03 n]},
    PlotJoined -> True, DisplayFunction -> Identity], {n, 2, 7}];

In[57]:= Show[magplot, DisplayFunction -> $DisplayFunction,
  Frame -> True, FrameLabel ->
  {"Magnetic Field (T)", "Energy (eV)", "Magnetic Field Plot", ""},
  RotateLabel -> False, Axes -> False]

```



Note that although we are plotting six levels, two pairs of these remain degenerate as the magnetic field changes, so we only see four lines in the plot. Running the program several times, it is possible to plot curves with different spin quantum numbers and total orbital angular momentum quantum numbers on the same graph. From this, it is possible to see the graphs for different quantum numbers intersecting as the magnetic field increases. This corresponds to magnetic transitions in the energy levels. Magnetic transitions in the ground state of quantum dots have been measured experimentally.

## ■ Conclusion

This article presents a *Mathematica* implementation of the SACI method for calculating the energies and wavefunctions of multi-electron quantum dot systems. The results obtained from this package are highly accurate with established confidence for up to eight electrons when using a PC with a Pentium IV 2.4GHz processor, which can be readily extended by using more powerful computers. Such numerically exact calculations provide important benchmarks against which one can test other approximate schemes developed to study more complex systems.



## About the Authors

Ranga D. Muhandiramge is a mathematics Ph.D. student at the University of Western Australia. In 2003, Muhandiramge did his honours project jointly in mathematics and physics modelling quantum dots, for which he received first-class honours and a prize for the most outstanding honours graduand. He currently works in the area of operations research on mine field path planning with the assistance of scholarships from the Hackett Foundation and the Defense Science and Technology Organisation.

Jingbo Wang is an associate professor of physics at The University of Western Australia. Her research areas and interests range widely from atomic physics, molecular and chemical physics, spectroscopy, acoustics, chaos, nanostructured electronic devices, and mesoscopic physics to quantum information and computation. Wang has published over 100 research papers in peer-reviewed journals and international conference proceedings. Wang uses *Mathematica* extensively in both her teaching and research, and she is also an editorial board member of the *Journal of Computational and Theoretical Nanoscience*.

### **Ranga D. Muhandiramge**

*School of Mathematics and Statistics, M019  
University of Western Australia  
35 Stirling Highway  
Crawley, WA 6009, Australia  
ranga@student.uwa.edu.au*

### **Jingbo Wang**

*School of Physics  
University of Western Australia  
35 Stirling Highway  
Crawley, WA 6009, Australia  
wang@physics.uwa.edu.au*