LINEAR AND NON-LINEAR PROBLEMS

Geert Brocks

December 15, 2020

General part.

1. **Theory.** In the following “LN” stands for “lecture notes”. Start by reading LN sec. 1.1 (but skip the “starred” sections).

   (a) The LU decomposition given in LN sec. 1.1.1 is called *Doolittle's algorithm.* It is also possible to write $\mathbf{T} = \mathbf{L}'\mathbf{U}'$, with

$$\mathbf{L}' = \begin{pmatrix} l'_{11} & 0 & 0 & 0 & 0 \\ l'_{21} & l'_{22} & 0 & 0 & 0 \\ 0 & l'_{32} & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & l'_{N,N-1} & l'_{N,N} \end{pmatrix}; \; \mathbf{U}' = \begin{pmatrix} 1 & u'_{12} & 0 & 0 & 0 \\ 0 & 1 & u'_{23} & 0 & 0 \\ 0 & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \ddots & u'_{N-1,N} \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}. \quad (1)$$

   Write out efficient algorithms, analogous to Eqs. (1.7), (1.8), and (1.9), to solve the linear problem $\mathbf{Tx} = \mathbf{b}$, which make use of the matrices given by Eq. 1.

   (b) Consider the 2×2 linear problem

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 + 10^{-10} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \quad (2)$$

   Find the analytical solution to this problem for the case (i) $b_1 = 1, b_2 = 1$, and for the case (ii) $b_1 = 1, b_2 = 1 + 10^{-10}$.

   (c) The condition number κ of a Hermitian or symmetric matrix $\mathbf{A}$ is given by $\kappa(\mathbf{A}) = |\lambda_{\max}/\lambda_{\min}|$, where $\lambda_{\max}, \lambda_{\min}$ are the eigenvalues with maximum and minimum absolute value, respectively.
   Find the condition number of the matrix of Eq. 2. *Is this* an ill-conditioned matrix?

   (d) *Explain* the large difference between the solutions you found in (b) for (i) and (ii) in terms of the condition number of the matrix. *Explain* the phrase “ill-conditioned” if you think of a practical application, where b_1, b_2 are given by physical measurements, and x_1, x_2 are parameter values you wish to extract from a physical model.

2. **A boundary value problem as a linear problem.** Consider a differential equation of the type

$$\frac{d^2 f}{dy^2}(y) = g(y); \quad 0 \leq y \leq 1, \quad (3)$$

where the solution $f(y)$ has to obey the boundary conditions $f(0) = a$ and $f(1) = b$. We can convert this into a linear problem, using a finite difference approximation

$$\frac{d^2 f}{dy^2}(y) \approx \frac{f(y+h) - 2f(y) + f(y-h)}{h^2}.$$

Defining a grid $y_j = jh;\ j = 0, 1, ..., N;\ h = 1/N$, Eq. 3 becomes

$$\begin{cases} f(y_{j+1}) - 2f(y_j) + f(y_{j-1}) = h^2 g(y_j); & j = 1, ..., N-1 \\ f(y_0) = a;\ f(y_N) = b. \end{cases} \tag{4}$$

This can be written as a linear problem

$$\mathbf{T}\mathbf{f} = \mathbf{g}; \quad \text{with} \quad \mathbf{T} = \begin{pmatrix} -2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix}, \tag{5}$$

a tridiagonal $(N-1) \times (N-1)$ matrix, and

$$\mathbf{f} = \begin{pmatrix} f(y_1) \\ f(y_2) \\ \vdots \\ f(y_{N-1}) \end{pmatrix}; \quad \mathbf{g} = \begin{pmatrix} h^2 g(y_1) - a \\ h^2 g(y_2) \\ \vdots \\ h^2 g(y_{N-1}) - b \end{pmatrix}, \tag{6}$$

vectors of length $N-1$.

(a) *Description:* Write a matlab routine for LU decomposition of a tridiagonal matrix using the algorithms of LN sec. 1.1.1. Don't waste any computer memory by storing the zeros!
Deliverables: In terms of N, how much memory is needed to store the results of the LU decomposition? Show that for the particular matrix $\mathbf{T}$ of eq. 5 one has the recurrence relations

$$l_{i,i-1} = -\left(2 + l_{i-1,i-2}\right)^{-1} \quad \text{and} \quad u_{i,i} = -\left(2 + u^{-1}_{i-1,i-1}\right),$$

and verify these relations (for small N) for the data that come out of your matlab routine. [Hint: start by showing that $u_{i-1,i} = 1$.]

(b) *Description:* Choose $g(y) = \sin(\pi y)$, $a = 0$, $b = 0$, $N = 101$. Solve Eq. 5 numerically using forward and back substitution. Also solve the differential equation, Eq. 3, analytically.
Deliverables: Plot the numerical solution $f(y)$, and, in a separate figure, the difference $f_{\text{ana}}(y) - f(y)$, where $f_{\text{ana}}(y)$ is the analytical solution. What is the largest deviation of $f(y)$ from $f_{\text{ana}}(y)$?

(c) *Description:* We apply a small random perturbation to the function $g(y) \to g(y) + \left[\text{rand}_y - \frac{1}{2}\right] \times 10^{-4}$, where "rand" is the matlab random number generating routine (for each y a different random number), and $g(y)$ is the function of (b).
Deliverables: As in (b). Do we have a well-conditioned problem?

3. **A "non-linear" problem.** This problem serves as an introduction to iterative methods. Start by reading LN sec. 1.3 (again, without the "starred" sections). Consider the problem

$$\frac{d^2 f}{dy^2}(y) = g(y) = -\pi^2 f(y); \quad 0 \leq y \leq 1, \tag{7}$$

with boundary conditions $f(0) = f(1) = 0$. Don't normalize the functions. We use the finite difference technique as outlined in the previous problem to convert the present problem into a matrix problem; use $N = 201$. We are going to pretend that it is a non-linear problem of the type explained in LN sec. 1.3.1, which we are going to solve by **fixed-point iteration**.[1] We use of a particularly idiotic starting vector: $f_0(y_j) = 2\texttt{rand}_j - 1$; $j = 1, .., N-1$ (where `rand` is the matlab random number generator, for each j a different random number). Use as a measure of the remaining error

$$c_n = \frac{\int_0^1 \left[f_n(y) - f_{n-1}(y)\right]^2 dy}{\int_0^1 \left[f_n(y)\right]^2 dy}. \tag{8}$$

Obviously Eq. 7 can also be solved analytically.

(a) *Deliverables:* Give the analytical solution $f(y)$ to eq. 7. Plot the starting function $f_0(y)$. How idiotic is it?

(b) *Description:* Write a matlab routine to solve eq. 7 by fixed-point iteration, starting from $f_0(y)$. Use as convergence criterion $c_n < 10^{-8}$.
Deliverables: Plot (in the same plot) all functions $f_n(y)$, obtained during iterations $n = 1, 2, ..., \text{last}$. Plot the difference $A_n f_n(y) - f(y)$, where $f(y)$ is the analytical solution, and $A_n = \text{sign}\left[f_n(\frac{1}{2}) f(\frac{1}{2})\right] \sqrt{\int_0^1 \left[f(y)\right]^2 dy / \int_0^1 \left[f_n(y)\right]^2 dy}$ is a scaling factor. Discuss the results.

(c) *Deliverables:* Focus on c_n, $n = 1, 2, ..., \text{last}$. What sort of convergence de we have here? [Hint: LN sec. 1.3.1.1; the matlab "semilog" functionality may be useful here.]

[1] This problem is a bit artificial. It is an excellent problem to illustrate numerical techniques. However, as you will have noticed, Eq. 7 is in fact an eigenvalue equation, with eigenvalue $-\pi^2$. One can solve it using eigenvalue/eigenvector routines.

Screening project 1: Poisson-Boltzmann.

4. **DeBye-Hückel**. Start by reading LN sec. 4.1. We consider a solution of NaCl (cations Na^+ and anions Cl^-) in water, between two parallel metal plates at a distance $2d = 50$ nm. [Beware of the factor 2! This is different from the lecture notes.] The two plates have a fixed potential $\Phi(0) = \Phi_1 = 0$ V and $\Phi(2d) = \Phi_2 = 0.2$ V, respectively. The concentration of the solution is $c_0 = 10^{-2}$ mol/l. The other physical parameters in this problem are $q = e$ (the elementary unit charge), $\epsilon = 80\epsilon_0$ (the static permittivity of water), $T = 25$°C, and k_B the Boltzmann constant.
   Consider the *Poisson-Boltzmann equation*

$$\frac{d^2\Phi}{dx^2}(x) = g\left(\Phi(x)\right) \quad \text{with} \quad g_{\text{PB}}\left(\Phi(x)\right) = \frac{2qc_0}{\epsilon}\sinh\left[\frac{q\left(\Phi(x) - \Phi_0\right)}{k_BT}\right]; \; 0 \le x \le 2d, \tag{9}$$

   and $\Phi_0 = \frac{1}{2}\left(\Phi_1 + \Phi_2\right)$. Its linearization, where

$$g_{\text{DH}}\left(\Phi(x)\right) = \frac{1}{\lambda_D^2}\left(\Phi(x) - \Phi_0\right) \quad \text{with} \quad \lambda_D = \sqrt{\frac{\epsilon k_BT}{2q^2c_0}}, \tag{10}$$

   is the *DeBye-Hückel equation*, with λ_D the DeBye screening length. The DeBye-Hückel equation is a linear equation, which can be solved by a direct method, such as LU decomposition.

   (a) *Deliverables:* Calculate λ_D (in nm). Using λ_D as the natural unit of length, and Φ_0 as the natural unit of potential, convert the DeBye-Hückel equation into a dimensionless equation.

   (b) *Description:* Rewrite the DeBye-Hückel equation as the linear equation

$$\frac{d^2\Phi}{dx^2}(x) - \frac{\Phi(x)}{\lambda_D^2} = -\frac{\Phi_0}{\lambda_D^2},$$

   and solve this equation with the techniques you used in problem 2 (using $N = 201$ points will do). Obviously this equation can also be solved analytically; call that solution $\Phi_{\text{ana}}(x)$.
   Deliverables: Plot the converged numerical solution $\Phi(x)$ (in V) and, in a separate figure, $\Phi_{\text{ana}}(x) - \Phi(x)$ (in V) as functions of x (in nm). A short description of the results is welcome.

   (c) *Description:* Calculate $\rho_{\text{DH}}(x)$; $0 \le x \le d$ from the solution of the DeBye-Hückel equation you found in (b).
   Deliverables: Plot the numerically obtained $\rho_{\text{DH}}(x)$ in units of e/nm^3 (where e is the elementary charge), and give a short description of the results. [Hint: first show that

$$\rho_{\text{DH}}(x) = -\frac{\epsilon}{\lambda_D^2}\left(\Phi(x) - \Phi_0\right)$$

   (d) *Description:* Calculate $\sigma_{\text{DH}} = \int_0^d \rho_{\text{DH}}(x)dx$ (in units of e/nm^2).
   Deliverables: What is the induced charge density in the metal plate, and where exactly do I find these charges in the metal? What is the average charge on the charged metal atoms, if the distance between nearest neighbor atoms is 0.25 nm and the metal atoms are close packed?

5. **DeBye-Hückel (iterative solution).** The Poisson-Boltzmann equation, eq. 9, is non-linear, and it has to be solved by an iterative method. As a preparation, we solve the DeBye-Hückel equation by iterative techniques, as from the previous problem we know exactly what the outcome should be. We use the same parameters, discretization techniques and grid size as in the previous problem. We try a linear function as our initial starting guess

$$\Phi_{\text{ini}}(x) = \Phi_1 + \frac{x}{d}(\Phi_0 - \Phi_1),$$

and we monitor our iterative processes by looking at

$$e_n = \int_0^{2d} \left[\Phi_n(x) - \Phi_{n-1}(x)\right]^2 dx, \tag{11}$$

as function of the iteration number n. Analogous to eq. 8, we use as convergence criterion

$$c_n = \frac{e_n}{\int_0^{2d} \left[\Phi_n(x)\right]^2 dx} < 10^{-8}. \tag{12}$$

   (a) *Description:* Write and execute a matlab routine to solve the problem using Gauss-Seidel iteration.
   Deliverables: Plot $\Phi_{\text{ana}}(x) - \Phi(x)$ (in V) as functions of x (in nm), where $\Phi(x)$ is the converged numerical solution. Comment on the differences with what you found in problem 4(b). Plot e_n as function of n on a logarithmic plot (using the matlab "semilog" functionality). What sort of convergence do we have?

   (b) *Description:* We now pretend that the DeBye-Hückel equation is a non-linear problem, and write it again as Eqs. 9 and 10. All the rest (parameters, etc.) we keep the same as in (a). Write a matlab routine to solve this non-linear problem by fixed-point iteration with damping (a constant damping parameter is sufficient). You will experience that without damping the algorithm diverges, so the trick is to find a decent damping parameter.
   Deliverables: Determine a decent damping parameter, i.e., one that gives convergence in around 30 cycles or so. Plot $\Phi_{\text{ana}}(x) - \Phi(x)$ (in V) as functions of x (in nm), where $\Phi(x)$ is the converged numerical solution. Plot e_n as function of n on a logarithmic plot. Discuss the differences with (a). Which of the methods do you prefer for this particular problem, the Gauss-Seidel method, or fixed-point iteration with damping?

6. **Poisson-Boltzmann.** We are now ready to attack the Poisson-Boltzmann equation, Eq. 9. By now (hopefully) we have our method of choice for solving this problem numerically. We use the same convergence criterion, eq. 12, as in the previous problem. We start from an intelligent (?) guess of our initial function $\Phi_{n=0}(x) = \Phi_{\text{ana}}(x)$, the analytical solution of the DeBye-Hückel equation.

   (a) *Description:* Convert the Poisson-Boltzmann equation to dimensionless units, using the same natural units as before. Solve this problem by a method of your choice (using your own routines).
   Deliverables: Plot the converged numerical solution $\Phi(x)$ and $\Phi_{\text{ana}}(x)$ in one plot. Comment on the differences, using a plot of the difference $\Phi_{\text{ana}}(x) - \Phi(x)$ in a separate plot.

(b) *Description:* Calculate $\rho_{\mathrm{PB}}(x);\ 0 \leq x \leq d$ from the solution of the Poisson-Boltzmann equation you found in (a). Calculate $\sigma_{\mathrm{PB}} = \int_0^d \rho_{\mathrm{PB}}(x)dx$. [Hint: first show that

$$\rho_{\mathrm{PB}}(x) = -2qc_0 \sinh\left[\frac{q\left(\Phi(x) - \Phi_0\right)}{k_B T}\right]$$

Deliverables: Plot $\rho_{\mathrm{PB}}(x)$ in units of e/nm^3 and compare it to $\rho_{\mathrm{ana}}(x)$ you can find from the analytical solution of the DeBye-Hückel equation (the two functions in the same plot would be handy; zoom in if you like to make things more visible). Compare σ_{PB} and σ_{ana}.

(c) *Description:* Repeat the calculations of (b) for potentials Φ_2 between > 0 V and < 0.44 V (keeping $\Phi_1 = 0$ V; a reasonable step size for Φ_2 would be 0.02 V, for instance).
Deliverables: Show a figure comparing σ_{PB} and σ_{ana} as a function of Φ_2. How good an approximation is σ_{ana} to σ_{PB} ?
One can replace one of the metal plates by a (doped) semiconductor. One would like to induce a charge density $\sigma > 1\ e/\mathrm{nm}^2$ in the semiconductor.[2] Do you think this is possible with the present set-up? (Give your arguments.)

[2] For instance, theories predict that two-dimensional semiconductors such as MoS_2 or black phosphorous, become superconducting if the induced charge density is sufficiently high. In case of MoS_2 this has been demonstrated experimentally.

Screening project 2: self-consistent polarization field (SCPF).

7. **SCPF; ad hoc iterative scheme.** Start by reading LN sec. 4.2. Derivatives of the molecule C_{60} are being considered for use in organic solar cells. C_{60} is a near-spherical molecule that forms crystals where the molecules are packed in a face-centered cubic (fcc) structure. We use a cubic unit cell with $\|\mathbf{a}_1\| = \|\mathbf{a}_2\| = \|\mathbf{a}_3\| = a = d\sqrt{2}$, where d is the distance between nearest neighbor C_{60} molecules in the lattice. This unit cell contains four molecules at positions

$$(0,0,0)\,;\ \frac{a}{2}(1,1,0)\,;\ \frac{a}{2}(1,0,1)\,;\ \frac{a}{2}(0,1,1)\,.$$

The C_{60} molecule has an isotropic polarizability α, i.e.,

$$\alpha_{i,ab} = \alpha\delta_{ab}.$$

We use $d = 1.0$ nm and $\alpha = 0.08$ nm^3. One of the C_{60} molecules is charged, with a charge $q = e$. All the other molecules in the lattice are neutral.
Lattice construction: As a model of the molecular crystal, we consider a range of clusters of molecules, with the charged molecule in the center. Define a variable N', and construct a lattice of unit cells by letting an index run from $-N'$ to $N'-1$ in each of the three directions. That way we construct a $N \times N \times N$ cluster of unit cells, with $N = 2N'$, where we put the charge in the origin.[3] One should generate a decent range of N'; I propose something like $N' = 1, 2, ..., 10$.
Units: use nm as unit of distance, and e (elementary charge) as unit of charge. $(4\pi\epsilon_0)^{-1}$ has the dimension of Vm/C in SI units; convert this to Vnm/e. The rest you should be able to figure out yourself.

   (a) *Description:* Neglect the dipole-dipole interaction, $\mathbf{T} = \mathbf{0}$. Calculate the polarization energy $\mathcal{E}_P$ as a function of the number of molecules N_D in the cluster (using the range of N' values, constructed as discussed above). Plot $\mathcal{E}_P(N_D)$ versus $N_D^{-1/3}$.
   Deliverables: If you use units as described above, in what units do you get $\mathcal{E}_P$? Plot of $\mathcal{E}_P(N_D)$ versus $N_D^{-1/3}$. Explain why $\mathcal{E}_P(N_D)$ behaves this way. Extrapolate $\mathcal{E}_P(N_D \to \infty)$.

   (b) *Description:* Switch on the dipole-dipole interaction, $\mathbf{T}$. We start with testruns, choosing a small, fixed N_D, derived from $N' = 1$. Calculate the polarization energy of a cluster $\mathcal{E}_P$ using the **ad hoc iterative scheme** (LN sec. 4.2.2). Do not construct the matrix $\mathbf{T}$ explicitly; you run out of computer memory if we increase N_D later on!! Instead, calculate the matrix elements where you need them (but efficiently, of course). Start your iterations with zero dipoles $\boldsymbol{\mu}_i = \mathbf{0}$; $i = 1, ..., N_D$. Each iteration $n = 1, 2, ...$ gives an estimate of the polarization energy $\mathcal{E}_P(N_D, n)$. Use as convergence criterion

$$|\mathcal{E}_P(N_D, n) - \mathcal{E}_P(N_D, n-1)| < \epsilon = 10^{-4}\text{eV}.$$

   The ad hoc iterative scheme is not very stable without damping. One of the issues therefore is to determine a decent damping parameter γ. For this particular problem

[3] The cluster will not be exactly symmetric regarding the surfaces, but don't worry about that. Surface effects will become unimportant if the cluster is sufficiently large.

it should give convergence within 5-10 cycles.
Deliverables: Plot $\mathcal{E}_P(N_D, n)$ versus the iteration n, choosing the damping parameter $\gamma = 1$, i.e., no damping. Discuss what happens. Now set $\alpha = 0.11$ nm^3; again plot $\mathcal{E}_P(N_D, n)$ and discuss what happens. Back to the tue physical value $\alpha = 0.08$ nm^3; plot $\mathcal{E}_P(N_D, n)$ with a value for the damping parameter γ that gives convergence within 6-8 cycles.

(c) *Description:* Fix the damping parameter γ to the value you found. We do production runs, letting N' run again, to calculate $\mathcal{E}_P$ as a function of its size N_D. The program needs to be efficient, if you don't wish to wait forever. Make as much use as you can of the build-in vector capabilities of matlab, instead of nested for-loops cyling on vector elements; that makes things a lot faster. Time (parts of) your program, using matlab's tic and toc, for instance, to explore which parts of the program need to be adapted to shorten runtimes.
Deliverables: Plot $\mathcal{E}_P\left(N_D, n_c\right)$ versus $N_D^{-1/3}$, where n_c is the number of iterations at which $\mathcal{E}_P$ is converged. Extraplolate $\mathcal{E}_P\left(N_D \to \infty\right)$. Compare your results to what you found in (a) and explain the difference in physical terms. Plot the runtimes $t(N_D)$ as a function of N_D. Why is this better than what you would get by solving the linear problem by a standard technique, such as LU factorization?

8. **SCPF; improved iterative scheme.** Having to determine a convenient damping parameter γ is somewhat of a nuisance, as its value tends to be material-dependent. If you would have a different molecule, or a different crystal structure, you would have to start from scratch determining a decent γ, whereas the success or failure of the algorithm critically depends on it. The **improved iterative scheme** with Pulay iteration avoids this nuisance. Read LN 1.4.1 and 1.4.2 (only methods 1 and 2).

   (a) *Description:* Implement method 2, using matlab's function `linsolve` to solve the linear problem emerging in the Pulay method. We are testing again, so we choose a small, fixed N_D, derived from $N' = 1$.
   Deliverables: Plot $\mathcal{E}_P(N_D, n)$ versus the iteration n, choosing the damping parameter $\gamma = 1$, i.e. no damping, but incorporating Pulay iteration. Now choose $\gamma = -1$, which gives a correction in the totally different direction. What happens? Discuss the difference with (b).

   (b) *Description:* We now do production runs with the Pulay-adapted program, and let N' run again.
   Deliverables: Plot $\mathcal{E}_P\left(N_D, n_c\right)$ versus $N_D^{-1/3}$, where n_c is the number of iterations at which $\mathcal{E}_P$ is converged. Extraplolate $\mathcal{E}_P\left(N_D \to \infty\right)$. Compare your results to what you found in (c). Plot the runtimes $t(N_D)$ as a function of N_D. Does Pulay iteration create any significant overhead?

9. **SCPF; some physics.** The measured polarizability (α), ionization potential (IP), and electron affinity (EA) of an isolated C_{60} molecule are $\alpha = 0.08 \pm 0.01$ nm^3,[4] IP $= 7.58 \pm 0.03$ eV,[5] and EA $= 2.6835 \pm 0.0006$ eV.[6] The double IP (the energy required to remove two electrons from C_{60}) is IP$_2 = 18.98 \pm 0.25$ eV.[7] For more data, see the NIST Chemistry

[4] R. Antoine *et al.*, J. Chem. Phys. 110, 9771 (1999).

[5] J. DeVries *et al.*, Chem. Phys. Lett. 188, 159 (1992)

[6] D. L. Huang, J. Chem. Phys. 140, 224315 (2014).

[7] A. V. Pogulay *et al.*, Int. J. Mass Spectrom. 233, 165–172 (2004)

Webbook. The measured band gap E_g of the molecular C_{60} crystal is 2.3 ± 0.1 eV.[8] The standard models for charge carrier transport are based upon so-called independent electrons (or holes), where each charge carrier experiences a potential resulting from averaging over all other carriers,[9] see your solid state physics course. Whereas such models are accurate in many cases, in some materials the individual charge carriers avoid one another more actively. Such materials are called *correlated*, referring to the motion of these charge carriers. One parameter determining whether this could the case is the Hubbard U, which is the energy difference between putting two holes (or electrons) on one crystal site, and having the two on two sites that are far apart. If U is large, then correlation effects might be expected. The measured U for a molecular C_{60} crystal is 1.6 ± 0.2 eV,[10] which makes it a candidate strongly correlated material.

(a) *Description:* Calculate the band gap E_g of a molecular cystal made from C_{60} molecules, using the SCPF model.
Deliverables: The calculated E_g, and an estimate of the error in E_g. Compare this to the measured value. Discuss whether you think that the SCPF model is appropriate here.

(b) *Description:* Calculate the Hubbard U in a molecular cystal made from C_{60} molecules, using the SCPF model.
Deliverables: The calculated U, and an estimate of the error in U. Compare this to the measured value. Is the SCPF model appropriate here?

[8] R. W. Lof *et al.*, Phys. Rev. Lett. 68, 3924 (1992).

[9] This is called the *mean-field approximation.*

[10] R. W. Lof *et al.*, Phys. Rev. Lett. 68, 3924 (1992).