Computational Methods in Astrophysics

Linear Algebra - Matrix Inversion

and

The rate equations

Keith Butler: December 2020

1 Introduction

The solution of a set of linear algebraic equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2N}x_N &= b_2 \\ \vdots & \quad \vdots \\ a_{M1}x_1 + a_{M2}x_2 + a_{M3}x_3 + \cdots + a_{MN}x_N &= b_M. \end{aligned} \tag{1}$$

is the subject of this practical. Here there are M equations for the N unknowns x_i. The coefficients a_{ij} and the numbers on the right-hand side are assumed to be known. The set of equations can be written in matrix form

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{2}$$

Note that we can swap rows without affecting the set of equations at all while swapping the columns means that we need to change the ordering of the variables. In addition we can form linear combinations of the equations without changing the information content.

If the number of equations is larger than the number of unknowns ($M > N$) then the system is overdetermined and can only be solved in the sense of a least squares fit. In the opposite case ($M > N$) there is no unique solution. We thus restrict ourselves to the case where $M = N$, the matrix $\mathbf{A}$ is square.

Even with $M = N$ a solution is not guaranteed since the matrix might be singular, i.e. it's determinant might be zero. This happens when two or more rows are linear combinations of each other (row degeneracy) or the equations define one or more variables only in a linear combination of each other (column degeneracy). For square matrices column degeneracy implies row degeneracy and vice versa. Since our computer models are only of limited numerical accuracy the equations we wish to solve can be degenerate numerically even if the "true" equations are not.

Formally, as long as $\mathbf{A}$ is not singular, the solution may be written

$$\mathbf{x} = \mathbf{A^{-1}}\mathbf{b}. \tag{3}$$

and $\mathbf{A}^{-1}$ is the inverse of $\mathbf{A}$. However, in most cases it is not necessary to obtain $\mathbf{A}^{-1}$ explicitly since we can obtain a solution without it. There are several ways of doing this, the first being Gaussian elimination which we discuss next.

2 Gaussian Elimination

The easiest way to learn about Gaussian elimination is to see it in action. We solve the set of equations

$$\begin{array}{rcrcrcr} 2x & + & 3y & - & 4z & = & 12 \\ x & + & 5y & - & z & = & 12 \\ 3x & + & 7y & - & 3z & = & 20 \end{array} \tag{4}$$

First we subtract half of row one from row 2 and obtain

$$\begin{array}{rcrcrcr} 2x & + & 3y & - & 4z & = & 12 \\ 0x & + & 7/2y & + & z & = & 6 \\ 3x & + & 7y & - & 3z & = & 20 \end{array} \tag{5}$$

Note that there is now a zero in the first column. We can do the same with rows 1 and 3, this time with $3/2$ as the factor to find

$$\begin{array}{rcrcrcr} 2x & + & 3y & - & 4z & = & 12 \\ 0x & + & 7/2y & + & z & = & 6 \\ 0x & + & 5/2y & + & 3z & = & 2 \end{array} \tag{6}$$

The important thing here is that x no longer appears in the the last two rows which we must now solve for y and z. Subtracting $5/7$ of row 2 from row 3 then gives

$$\begin{array}{rcrcrcr} 2x & + & 3y & - & 4z & = & 12 \\ 0x & + & 7/2y & + & z & = & 6 \\ 0x & + & 0y & + & 16/7z & = & -16/7 \end{array} \tag{7}$$

The last equation now only contains z and can be solved trivially to give $z = -1$. Knowing z in row two allows us to solve for y, $\frac{7}{2}y = 7$ or $y = 2$ and finally the first row implies that $x = 1$. Using the matrix form the reductions look like this

$$\left(\begin{array}{ccc|c} 2 & 3 & -4 & 12 \\ 1 & 5 & -1 & 12 \\ 3 & 7 & -3 & 20 \end{array}\right) \xrightarrow{\text{row2}-\frac{1}{2}\cdot\text{row1},\text{row3}-\frac{3}{2}\cdot\text{row1}} \left(\begin{array}{ccc|c} 2 & 3 & -4 & 12 \\ 0 & 7/2 & 1 & 6 \\ 0 & 5/2 & 3 & 2 \end{array}\right)$$

$$\xrightarrow{\text{row3}-\frac{5}{7}\cdot\text{row2}} \left(\begin{array}{ccc|c} 2 & 3 & -4 & 12 \\ 0 & 7/2 & 1 & 6 \\ 0 & 0 & 16/7 & -16/7 \end{array}\right)$$

and we have saved a little space by writing the right hand side (RHS) as a fourth column.

Thus we have

- reduced the initial matrix to *triangular* form
- solved the system by *back substitution*

The algorithm in the general case is then

for all row i **do**
  for all row $j > i$ **do**
    for all column $k > i$ **do**
      subtract $l_{ji} = a_{ji}/a_{ii}$ times a_{ik} from a_{jk} and b_j
      and store the result in a_{jk}, b_j
    end for
  end for
end for

for the reduction to U (for upper) form followed by

for all row i in descending order **do**
  for all column $j > i$ **do**
    subtract a_{ij} times b_j from b_i
    and store the result in b_i
  end for
  $b_i = b_i/a_{ii}$
end for

There are a few points to note

- the solution is performed in place so no additional storage is needed
- in consequence both $\mathbf{A}$ and $\mathbf{b}$ are destroyed, occasionally a copy may be required
- the necessary computation time is proportional to N^3.
- since $\mathbf{U}$ is triangular the determinant is simply the product of the diagonal elements.
- as written, the algorithm is only applied to a single RHS but this is only for didactical purposes. Any number of RHSs may be treated. In particular, setting $\mathbf{b}$ equal to the unit matrix will give the inverse of $\mathbf{A}$ since the solution of

$$\mathbf{Ax} = \mathbf{I}$$

is

$$\mathbf{x} = \mathbf{A}^{-1}$$

In fact elements of $\mathbf{A}$ below the diagonal remain unchanged so that all the information needed to reduce any given RHS is available. We can perform the U reduction for $\mathbf{A}$ alone and later reduce and solve for $\mathbf{b}$. This scheme is implemented in the pair of routines LURED/RESLV (to be found in lured.f90 and reslv.f90).

There is one additional very important point about the algorithm as written

- **It can fail!**

To see why we change our example slightly

$$\left(\begin{array}{ccc|c} 0 & 3 & -4 & 10 \\ 1 & 5 & -1 & 12 \\ 3 & 7 & -3 & 20 \end{array}\right)$$

If we try to divide by the diagonal element a_{11} we have a problem. The same could happen if any of the a_{ii} are zero or very small compared to the non-diagonal elements. The solution to this problem is *pivoting*.

3 Pivoting

Pivoting is the re-ordering of the equations to avoid divisions by small or zero elements during the reduction process. This is done by searching for a the largest element in the present column and bringing it onto the diagonal. Rows and columns that have already been treated are not considered. Looking at the example again

$$\left(\begin{array}{ccc|c} 0 & 3 & -4 & 10 \\ 1 & 5 & -1 & 12 \\ 3 & 7 & -3 & 20 \end{array}\right)$$

we see that the largest element in the first column is 3 and it appears in the third row. So we swap rows 1 and 3, obtain

$$\left(\begin{array}{ccc|c} 3 & 7 & -3 & 20 \\ 1 & 5 & -1 & 12 \\ 0 & 3 & -4 & 10 \end{array}\right)$$

and then proceed as before.

Here we have only looked for the largest element in the present column, the procedure is called *partial* pivoting. Of course it is possible to look for the largest element in the remaining sub-matrix, *full* pivoting. In this case, the 7 to be found in row 3 and column 2 would be the chosen pivot. The two columns 1 and 2 and rows 1 and 3 would then be swapped. The advantage in doing this, numerical stability in all cases, is far outweighed by the disadvantages that the increase in bookkeeping brings with it. This is particularly so, since, in practice, partial pivoting is equally stable (artificial examples can be constructed for which partial pivoting also fails).

It is instructive to look at a very simple example to see just how badly things can go wrong. We imagine that we are computing to three significant figures and want to solve the following system

$$\left(\begin{array}{cc} 0.1 & 100 \\ 1 & 2 \end{array}\right)\left(\begin{array}{c} x_1 \\ x_2 \end{array}\right) = \left(\begin{array}{c} 100 \\ 3 \end{array}\right)$$

The usual procedure leads to the new system

$$\left(\begin{array}{cc} 0.1 & 100 \\ 0 & -1000 \end{array}\right)\left(\begin{array}{c} x_1 \\ x_2 \end{array}\right) = \left(\begin{array}{c} 100 \\ -1000 \end{array}\right)$$

since the -997 will be rounded to -1000. Solving for $x - 2$ gives $x_2 = 1$ which is correct but for x_1 we have

$$0.1x_1 + 100 = 100 \tag{8}$$

or $x_1 = 0$ which is incorrect.

Now we use pivoting which in this case simply implies that we swap the two rows.

$$\begin{pmatrix} 1 & 2 \\ 0.1 & 100 \end{pmatrix} \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 100 \end{pmatrix}$$

Now the reduction leads to

$$\begin{pmatrix} 1 & 2 \\ 0 & 99.8 \end{pmatrix} \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 99.7 \end{pmatrix}$$

and the solution for x_1 is 1 while x_2 is given by

$$x_2 + 2 = 3. \tag{9}$$

$x_2 = 1$ which is also correct. It is the ratio of a_{11} to a_{12} which is important here. The same situation can, in principle, occur in real applications but is somewhat more unlikely since single precision delivers roughly 6.5 significant figures and double precision something like 14. At some point, however, if the matrix is large enough rounding errors are capable of producing such a situation so partial pivoting at least is a must.

Finally, we note that the numerical value of the pivots depends on the scaling of the equations, we can multiply row 2 throughout by 1000, say, without changing the actual solution vector. But in this case we would choose row 2 as the pivot instead of 3. To deal with this *implicit* pivoting can used. The pivot candidates are chosen as if they were normalized so that the sum of the absolute values of the row elements is 1. We look at

$$\frac{|a_{ik}|}{\sum_k |a_{ik}|} \tag{10}$$

Turning again to our example, the sum of the elements in row 2 is 7 and the first factor is $1/7$. For row 3 the sum is 13, the factor is $3/13$ so we would once again choose row 3 as our pivot.

The row interchanges may be represented succinctly by the use of a *permutation* matrix $\mathbf{P}$. For $N = 2$ the matrix

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{11}$$

swaps two rows as a simple test will show

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} a_{21} & a_{22} \\ a_{11} & a_{12} \end{pmatrix} \tag{12}$$

In the general case we notice that $\mathbf{PI} = \mathbf{P}$ by definition so that P for any permutation can be found simply by performing the same operations in the same order on the unit matrix $\mathbf{I}$. For instance, the $\mathbf{P}$ corresponding to the exchange of row 1 with row 3 and then row 2 with row 3 is given by

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{\text{swap1and3}} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \xrightarrow{\text{swap2and3}} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = P \tag{13}$$

Thus having begun with the system

$$\mathbf{Ax} = \mathbf{b}. \tag{14}$$

the row interchanges correspond to multiplication of both sides by $\mathbf{P}$

$$\mathbf{PAx} = \mathbf{Pb} \tag{15}$$

and it is this system of equations which is solved. As a final comment, it may be noted that swapping two rows multiplies the determinant by -1, so that counting the number of swaps allows the determinant to be determined.

4 LU decomposition

The elements below the diagonal are zero by definition after the reduction. By leaving them unchanged we were able to deal with any number of RHSs. There is however another alternative, we can store the multiplicative factors l_{ij} in the appropriate elements. Our example was

$$\begin{pmatrix} 2 & 3 & -4 \\ 1 & 5 & -1 \\ 3 & 7 & -3 \end{pmatrix} \tag{16}$$

and we reduced the first column to 0 by subtracting $1/2$ and $3/2$ times the first row from the second and third rows respectively. We store these factors l_{21},l_{31} in place of a_{21},a_{31}

$$\begin{pmatrix} 2 & 3 & -4 \\ 1/2 & 7/2 & 1 \\ 3/2 & 5/2 & 3 \end{pmatrix} \tag{17}$$

The final step (for this 3×3 matrix) was to subtract $5/7$ of the second row from the third. Our final matrix is

$$\begin{pmatrix} 2 & 3 & -4 \\ 1/2 & 7/2 & 1 \\ 3/2 & 5/7 & 16/7 \end{pmatrix} \tag{18}$$

But what is the advantage in doing this? We complete the matrix below the diagonal (L) with 1s on the diagonal and calculate LU (hence the name)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 3/2 & 5/7 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & -4 \\ 0 & 7/2 & 1 \\ 0 & 0 & 16/7 \end{pmatrix} = \begin{pmatrix} 2 & 3 & -4 \\ 1 & 5 & -1 \\ 3 & 7 & -3 \end{pmatrix} \tag{19}$$

i.e. we have $\mathbf{A} = \mathbf{LU}$. Now we can solve

$$\mathbf{Ax} = \mathbf{LUx} = \mathbf{b} \tag{20}$$

in two stages: first we find $\mathbf{y}$ from $\mathbf{Ly} = \mathbf{b}$ followed by x from $\mathbf{Ux} = \mathbf{y}$.

The second equation is solved by the same algorithm as before

for all row i in descending order **do**
  for all column $j > i$ **do**
    subtract a_{ij} times b_j from b_i
    and store the result in b_i
  end for
  $b_i = b_i / a_{ii}$
end for

Since $\mathbf{L}$ is also triangular, the algorithm to solve $\mathbf{Ly} = \mathbf{b}$ is similar

for all row i in ascending order **do**
  for all column $j < i$ **do**
    subtract a_{ij} times b_j from b_i
    and store the result in b_i
  end for
end for

and we have explicitly set the diagonal elements to 1.

In general **L** has only elements on and below the diagonal, we call them l_{ij} with the understanding that $l_{ij} = 0$ for $j > i$. Similarly the elements of **U** are u_{ij} with $u_{ij} = 0$ for $j < i$. This gives us N^2 equations for $N^2 + N$ unknowns. They read

$$\begin{array}{lrcl} i < j: & l_{i1}u_{1j} + l_{i2}u_{2j} + \cdots + l_{ii}u_{ij} & = & a_{ij} \\ i = j: & l_{i1}u_{1j} + l_{i2}u_{2j} + \cdots + l_{ii}u_{jj} & = & a_{jj} \\ i > j: & l_{i1}u_{1j} + l_{i2}u_{2j} + \cdots + l_{ij}u_{jj} & = & a_{ij} \end{array} \tag{21}$$

We are free to choose N of these unknowns and we set $l_{ii} = 1$. The solution is then found using Crout's algorithm

for all column j in ascending order **do**
  for all row i $i < j$ **do**
    calculate

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj} \tag{22}$$

  end for
  for all row i $i > j$ **do**
    calculate

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj} \right) \tag{23}$$

  end for
end for

Then u_{ij} is the reduced upper matrix and the l_{ij} are the corresponding multiplicative factors as described above. For $i = j$ the two equations are identical apart from the division by the pivot element u_{jj} so pivoting can be introduced by performing the U reduction completely and then for $i = j$ make the choice for the pivot and continue with the L reduction. Subroutines implementing this scheme with implicit pivoting are to be found in ludcmp.f90 and lubksb.f90 (from Numerical Recipes).

5 Special cases

If you have information about the structure of the matrix you should use it as the speed and accuracy can be increased dramatically. In the solution of differential equations using differencing band matrices occur frequently. These are such that the matrix has non-zero elements only in bands on and close to the diagonal. A diagonal matrix is the simplest, the next simplest being a tridiagonal matrix and we look at this example in more detail. The equations are

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} \tag{24}$$

with $a_1 = c_n = 0$. In matrix form this reads

$$\begin{bmatrix} b_1 & c_1 & & & & & 0 \\ a_2 & b_2 & c_2 & \cdots & & & \\ & a_3 & b_3 & c_3 & \cdots & & \\ \vdots & & & & & & \\ & & & \cdots & b_{n-1} & c_{n-1} & \\ 0 & & & \cdots & a_n & b_n & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{25}$$

The solution is a simple application of the Gaussian elimination we saw earlier.

The first equation has solution

$$\boldsymbol{x_1 = -b_1^{-1} c_1 x_2 + b_1^{-1} d_1 \equiv e_1 x_2 + v_1.} \tag{26}$$

The equation is written in this way because the system could be *block* tridigonal with $\boldsymbol{a_i}$,$\boldsymbol{b_i}$,$\boldsymbol{c_i}$ matrices and $\boldsymbol{x_i}$,$\boldsymbol{b_i}$ vectors. Substituting this result into the second equation we obtain

$$\begin{aligned} \boldsymbol{x_2} &= \boldsymbol{e_2 x_3 + v_2} && (27) \\ \boldsymbol{e_2} &= \boldsymbol{-(b_2 + a_2 e_1)^{-1} c_2} && (28) \\ \boldsymbol{v_2} &= \boldsymbol{(b_2 + a_2 e_1)^{-1}(d_2 - a_2 v_1)} && (29) \end{aligned}$$

and in the general case

$$\begin{aligned} \boldsymbol{x_i} &= \boldsymbol{e_i x_{i+1} + v_i} && (30) \\ \boldsymbol{e_i} &= \boldsymbol{-(b_i + a_i e_{i-1})^{-1} c_i} && (31) \\ \boldsymbol{v_i} &= \boldsymbol{(b_i + a_i e_{i-1})^{-1}(d_i - a_i v_{i-1})} && (32) \end{aligned}$$

For $i = n$ we have $c_n = 0$ and thus $e_n = 0$ and $x_n = v_n$. Having found v_n the equations 30 can then be used to derive all the x_i, the back-substitutions. There are several points to note here

- a,b,c can be stored as vectors
- The algorithm is now O(n)
- The algorithm can be extended to more bands in an obvious way

6 Astrophysical application

A star with mass not too far from that of the sun will, at the end of its life, throw off its outer shell leaving the central star to cool as a white dwarf. The shell will be illuminated by the remnant and is observed as a planetary nebula. They come in all shapes and sizes as the pictures show. They are of considerable interest since the elemental abundances are representative of the outer layers of the star as the nebula was thrown off giving clues about stellar evolution. Also since they have large diameters they are very bright and with their distinctive forbidden line spectra (see below) they are easy to spot in galaxies well outside the local group. Their optical properties make them good standard candles with well known physics allowing the Hubble constant to be determined independently of Cepheids.

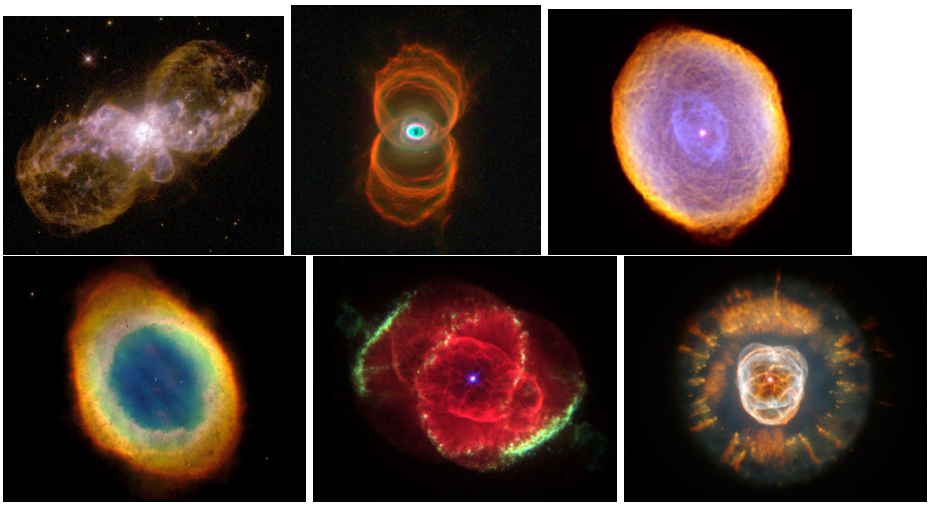

Figure 1: A selection of planetary nebulae images from the HST. They are a) hb5; b) mycn18; c) spirograph; d) ring; e) cat's eye; f) eskimo.

The physics of such a nebular is particularly simple. Photoionization by stellar photons is negligible to a first approximation due to geometrical effects (the nebula is large, the star small) and densities are small so that photons escape immediately. Thus radiative transfer need not be considered. Under such circumstances the strengths of well-chosen line pairs can be used to give a direct measure of the temperature and the electron density.

The strength of a spectral emission line is given by the particle number density of the upper level n_j multiplied by the transition probability

$$j = n_j A_{ji}. \tag{33}$$

The latter is an atomic physical quantity and can be taken as given. Normally ratios of two lines belonging to the same ion are taken as in this case abundances

and other complicating factors do not play a rôle. Thus we have

$$\frac{j_1}{j_2} = \frac{n_1 A_1}{n_2 A_2} \tag{34}$$

and since the As are known, the line strength ratio depends on the population ratio n_1/n_2.

In thermodynamical equilibrium this ratio is given by the Boltzmann formula

$$\frac{n_1}{n_2} = \frac{g_1}{g_2} \exp -[(E_1 - E_2)/kT] \tag{35}$$

g_1,g_2 being the statistical weights and E_1, E_2 the excitation energies of levels 1,2 respectively. The Boltzmann constant is k and the temperature is T. Of course, thermodynamical equilibrium is not possible in a nebula but if the densities are high enough so that collisional processes are more efficient than radiative ones, then *Local Thermodynamic Equilibrium* can apply and the Boltzmann formula can be used with local values of the temperature, the line ratio does not depend on the density.

In the opposite extreme, at very low densities, every excitation of the atom by a collision will lead directly to a line photon. Only collisions involving the ground state (population n_g) need be considered since none of the other levels will be populated. Thus we have

$$n_1 A_1 = n_g n_e q_{g1}(T) \tag{36}$$

$$n_2 A_2 = n_g n_e q_{g2}(T) \tag{37}$$

and the line ratio is given by

$$\frac{j_1}{j_2} = \frac{q_{g1}(T)}{q_{g2}(T)}. \tag{38}$$

There is again no dependence on the density n_e. The collision rates $q_{g1}(T)$,$q_{g1}(T)$ are calculated or measured by atomic physicists and are given. Normally, the downward rate

$$q_{1g}(T) = \frac{8.631 \times 10^{-6}}{g_1 T^{1/2}} \Upsilon(T) \tag{39}$$

is to be preferred since $\Upsilon(T)$ is a slowly varying function of temperature. The upward rate is then given by detailed balance to be

$$q_{g1}(T) = \frac{g_1}{g_g} \exp -[(E_1 - E_g)/kT) q_{1g}(T) \tag{40}$$

At intermediate densities the picture is more complicated and the populations must be derived from the equations of *statistical equilibrium*. They simply balance

the number of transitions into a level and those leaving it. So for a level i the equation reads

$$n_i \sum_j P_{ij} = \sum_j n_j P_{ji}. \tag{41}$$

The sum extends, in principle, over all possible levels j but in practice only a few terms are necessary. The transition rate P_{ij} includes contributions from collisional $C_{ij} = n_e q_{ij}$ and radiative processes but the latter are only to be included in the downward rates. For three levels the three equations will read (n_1 is the ground level)

$$\begin{aligned} n_1(C_{12} + C_{13}) &= n_2(C_{21} + A_{21}) + n_3(C_{31} + A_{31}) && (42)\\ n_2(C_{21} + A_{21} + C_{23}) &= n_1 C_{12} + n_3(C_{32} + A_{32}) && (43)\\ n_3(C_{31} + A_{31} + C_{32} + A_{32}) &= n_1 C_{13} + n_2 C_{23} && (44) \end{aligned}$$

Note that these are linearly dependent as, for example, the third is equal to the sum of the other two. So we need a further equation to close the system e.g. for the total number density N

$$n_1 + n_2 + n_3 = N. \tag{45}$$

If we only deal with line ratios the actual value of N is not important. Having derived T and n_e from suitable line ratios N, the element abundance, is determined separately from the individual line strengths. This is a much more complicated question however, for instance, several ionization stages will need to be considered, and we shall not investigate it here.

Instead we will look at line ratios in doubly (O III)and singly O II ionized oxygen the line ratios of which are temperature and density sensitive respectively.

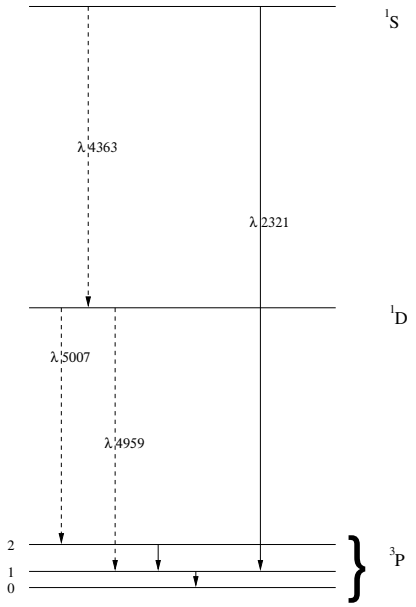

Figure 2: Energy level (Grotrian) diagram for O III

6.1 Temperature dependent O III

Two lines will be sensitive to temperature if the respective upper levels have a large energy difference. In O III the lines at $\lambda 4959/5007$ and $\lambda 4363$ (see fig. 2) have upper levels ^{1}D and ^{1}S that are more than 20000 cm^{-1} apart. The ratio of the populations of the two levels is proportional to the Boltzmann factor $\exp -20000/kT$ which will be sensitive up to temperatures where $kT \approx 20000$ (with kT in cm^{-1}). Since the electron density only appears linearly in the equations the exponential factor dominates.

The relevant atomic data are shown in Table 1. They have been taken from the book on Gaseous Nebulae by Osterbrock. Of particular interest is the fact that all the levels belong to the 2p^2 configuration, the two valence electrons can both be labelled with 2p. Transitions among the 5 levels are *forbidden* since no change in the electron configuration takes place. As a consequence, the radiative transition probilities are very small, the largest is only 1.82 sec^{-1}. For comparison, the transition probabilities for allowed transitions are of order 10^8 sec^{-1}. The small radiative probabilities allow collisions to produce relatively large populations in the excited states even though the electron densities are low (perhaps 10^3 cm^{-3}). The collisional rates are so small that the atoms are far from thermodynamic equilibrium so forbidden lines in emission are among the strongest to be seen in Planetary nebulae as illustrated in fig. 3.

```
 5                          !  Number of levels
 3P0      1 0.0             ! For each level a label, the statistical wei
 3P1      3 113.178
 3P2      5 306.174
 1D2      5 20273.27
 1S0      1 43185.74
1 2   0.54   2.6e-5         ! For each pair of levels two indices, an eff
1 3   0.27   3.0e-11        ! and a transition probability
1 4   0.24   2.7e-6
1 5   0.03   0.0
2 3   1.29   9.8e-5
2 4   0.72   6.7e-3
2 5   0.09   0.22
3 4   1.21   2.0e-2
3 5   0.16   7.8e-4
4 5   0.62   1.8
```

Table 1: Input data for O III

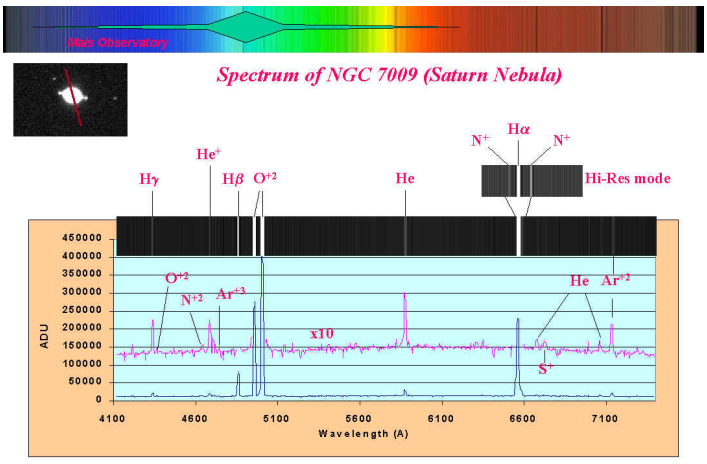

Figure 3: An amateur spectrum of the Saturn nebula. The Mais Observatory. Note the strength of the O III(=O^{2+}) lines.

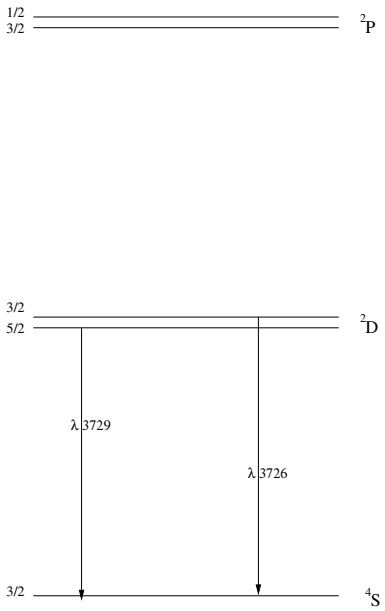

Figure 4: Energy level (Grotrian) diagram for O II

6.2 Density dependent O II

The levels of O II all belong to the $2p^3$ configuration so that the lines are forbidden as was the case for the O III. On the other hand, the energy level structure is completely different (see fig. 4. The energy levels appear in pairs, doublets. The ratio of the strengths of the two lines from the ^{2}D levels to the ground state is primarily sensitive to the electron density. This is simply because the energy difference is only $20\,\mathrm{cm}^{-1}$, as can be seen from Table 2, so that the Boltzmann factor, $\exp -20/kT$, is approximately 1 even at very low temperatures. Thus there is almost no sensitivity to temperature.

As we shall see later, the ratio of the two lines and their wavelength separation is small so that the observations are difficult making O II a far from ideal case.

As a final comment, the collision rates are in fact dependent on temperature and for accurate work a table should be read in and interpolated upon. However, the principle is the same, only the numerical values appearing in the rate equations are slightly different and we omit this detail here.

```
 5                      !  Number of levels
 4S3_2   4 0.0          ! For each level a label, the statistical wei
 2D5_2   6 26810.55
 2D3_2   4 26830.57
 2P3_2   4 40468.01
 2P1_2   2 40470.00
1 2  0.80 3.6e-5        ! For each pair of levels two indices, an eff
1 3  0.54 1.8e-4        ! and a transition probability
1 4  0.27 5.8e-2
1 5  0.13 2.4e-2
2 3  1.17 1.3e-7
2 4  0.73 0.11
2 5  0.30 5.6e-2
3 4  0.41 5.8e-2
3 5  0.28 9.4e-2
4 5  0.29 1.4e-10
```

Table 2: Input data for O II

7 Exercises

Exercise 1

Solve the equations

$$\begin{array}{rcrcrcrcr} 2w & + & 2x & + & 3y & + & 1z & = & 0 \\ 3w & + & 4x & - & 2y & + & 5z & = & 4 \\ -5w & + & 5x & - & 1y & - & 2z & = & 3 \\ -w & - & x & - & 3y & + & 3z & = & -2 \end{array} \tag{46}$$

using Gaussian elimination.

Exercise 2

Solve the same system of equations as before

$$\begin{array}{rcrcrcrcr} 2w & + & 2x & + & 3y & + & 1z & = & 0 \\ 3w & + & 4x & - & 2y & + & 5z & = & 4 \\ -5w & + & 5x & - & 1y & - & 2z & = & 3 \\ -w & - & x & - & 3y & + & 3z & = & -2 \end{array} \tag{47}$$

this time using LU decomposition.

Download the programs (la_progs.tar) from the web page. When you have unpacked the archive (tar xvf la_progs.tar) you will have two directories, TEST and PN we will begin with TEST. Enter the following

```
cd TEST
ifort -c precision.f90
ifort -o test *.f90
```

or

```
cd TEST
gfortran -c precision.f90
gfortran -o test *.f90
```

to compile the package. You should inspect the various subroutines to find out that they do and how they do it. With the help of the script this should be straightfoward. Note that ludcmp and lubksb have been taken from Numerical recipes. You can change from single to double precision by changing the value of the variable 1.e0 to 1.d0 in precision.f90. Read the comments carefully for more information. Pay particular attention to the warning about compiling the module first before the rest of the routines.

You can start the program with

```
./test
```

and you will be offered 4 options

```
1) fill the matrix with random values (default)
2) the hilbert matrix
3) a user-defined matrix: edit user.f90 first
4) read a matrix and a RHS from mat.dat
```

for each of the first 3 you will then need to give the size of the required matrix and which pair ludcmp/lubksb,lured/reslv you wish to use.

An intermezzo: the Hilbert matrix

The Hilbert matrix is a very strange beast. It's elements are easy to define

$$H_{ij} = 1/(i+j-1) \tag{48}$$

and its inverse has elements

$$(H^{-1})_{ij} = (-1)^{i+j}(i+j+1)\binom{n+i-1}{n-j}\binom{n+j-1}{n-i}\binom{n-i}{i-1}^2 \tag{49}$$

where the notation $\binom{n}{j}$ is the binomial coefficient defined by

$$\binom{n}{j} = \frac{n!}{j!(n-j)!}. \tag{50}$$

The interesting property of the matrix is that its determinant is one divided by 1,12,2160,6048000,266716800000 and so on making it an extremely stringent

Table 3: Determinant of the first few Hilbert matrices

n	det(H)
1	1
2	8.33333e-2
3	4.62963e-4
4	1.65344e-7
5	3.74930e-12
6	5.36730e-18

test for any numerical linear algebra solution routine.

Exercise 3

Perform the following in single precision

a Choose option 1 and compare the error for a number of matrix sizes (100s to 1000s) and for both algorithm pairs. How fast does it grow? Does the algorithm become unusable. For these runs start the program with `time ./test`, take the user entry as an indication of the computer time used. Plot this logarithmically and compare the slope with the predicted value of 3. (Note that large matrices may not fit into the available memory in which case the system will write data to disk, as needed making the execution times much longer. This is called *paging*. You can use the `top` command in another window to see if this is happening. CPU usage below 90% for any period us a good sign that this is the case).

b Choose option 2 and compare the error for a number of matrix sizes and for both algorithm pairs. Here the matrix size should be small (< 20).

c Choose option 3 and for 3 different choices of matrix perform similar tests. You will need to edit and change user.f90 appropriately and then recompile the program (see above).

d Option 4 is included chiefly for pedagogical purposes but with an appropriate input data set you can check your answers from exercises 1 and 2 (see `mat1.dat`).

Exercise 4

Perform the same tests as in exercise 3 but in double precision and compare your results. You will need to edit precision.f90 and recompile the program.

Exercise 5

Solve the following tridiagonal matrix.

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \tag{51}$$

Exercise 6 Write a subroutine to solve a tridiagonal matrix. Use this routine to check your solution to exercise 5.

Exercise 7 For this exercise

```
cd ../PN
ifort -c precision.f90
ifort -o line *.f90
./line | tee  output
```

The input is simple. You will be asked to calculate data for `oii` or `oiii`. Then for `oiii` which is temperature dependent, you should enter a density, while a temperature is needed for `oii`. For each ion you should perform 2 or 3 runs with a different output file for each run. Then plot your results for each ion, line intensity ratio versus temperature for `oiii` and versus density for `oii`. Comment on your results.

Advanced task 1

The subject of this practical has been to solve the set of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{52}$$

In fact, we have only found an approximate solution which satifisies the perturbed equation

$$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}. \tag{53}$$

The difference between the two gives an equation for δx in terms of δb

$$\mathbf{A}\delta\mathbf{x} = \delta\mathbf{b} \tag{54}$$

while δb is known from Eqn. 53

$$\mathbf{A}\delta\mathbf{x} = \mathbf{A}(\mathbf{x} + \delta\mathbf{x}) - \mathbf{b}. \tag{55}$$

This last equation can be used to improve the current solution and may be applied iteratively. The $\mathbf{A}$ and $\mathbf{b}$ on the RHS are the original matrix and vector so a copy is needed. Since the RHS is the current error vector it must be calculated as accurately as possible, double precision should be used.

Your mission, should you choose to accept it is to write a subroutine to implement this algorithm (no peeking in Numerical Recipes). You should use ludcmp/lubksb and write a test program.

Advanced task 2

S II and S III are very similar to O II and O III in their atomic physical properties and so can be used in a similar way to derive temperatures and densities. Construct input data sets suitable for use with line. The necessary data are to be found in the appendix.

Advanced task 3

lured/reslv as written do not use pivoting. Modify them to use partial or implicit pivoting. You may use ludcmp/lubksb as an example. Once more you should test your routines to ensure that they are correct.

8 Appendix

The energy level notation is ${}^{(2S+1)}L_J$, S being the total spin, L the total angular momentum (this is a letter whereby $L = \mathrm{S} = 0, L = \mathrm{P} = 1, L = \mathrm{D} = 2$. The total angular momentum is $J = L + S$. The statistical weight of any level is simply $2J + 1$ while the statistical weight of a term is $(2S + 1)(2L + 1)$. These weights are such that $\sum(2J + 1) = (2S + 1)(2L + 1)$. The configurations are denoted by $2s^2 2p^3$ and $3s^2 3p^3$. They are not important here but the fact that they are very similar means that the energy levels are very similar and that the two elements are chemically related.

Osterbrock has saved space in his tabulations by making use of some elementary properties of the collision strengths. For instance the ${}^3\mathrm{P} - {}^1\mathrm{D}$ collision strength between the two terms, ^{3}P, ^{1}D splits into three collision strengths between the levels ${}^3\mathrm{P}_{0,1,2} - {}^1\mathrm{D}_2$ according to the statistical weights

$$\begin{aligned}\Omega({}^3\mathrm{P}_0 - {}^1\mathrm{D}_2) &= \frac{(2J+1)}{(2S+1)(2L+1)}\Omega({}^{(2S+1)}\mathrm{L} - {}^{(2S'+1}\mathrm{L}') \\ &= \frac{(2*0+1)}{(3*(2*1+1)}\Omega({}^3\mathrm{P} - {}^1\mathrm{D}) \\ &= \frac{1}{9}\Omega({}^3\mathrm{P} - {}^1\mathrm{D})\end{aligned}$$

and so on. We have used this to write out the tables in full. You can check your values by comparing with the O II, O III numbers given in the text and the input data files.

8.1 Atomic data for O II and S II

Table 4: O II and S II energy levels in cm^{-1} from the NIST website. Note the ordering.

O II			S II		
Configuration	Level	Energy	Configuration	Level	Energy
$2s^2 2p^3$	${}^4\mathrm{S}_{3/2}$	0.0	$3s^2 3p^3$	${}^4\mathrm{S}_{3/2}$	0.0
$2s^2 2p^3$	${}^2\mathrm{D}_{5/2}$	26810.55	$3s^2 3p^3$	${}^2\mathrm{D}_{3/2}$	14852.94
$2s^2 2p^3$	${}^2\mathrm{D}_{3/2}$	26830.57	$3s^2 3p^3$	${}^2\mathrm{D}_{5/2}$	14884.73
$2s^2 2p^3$	${}^2\mathrm{P}_{3/2}$	40468.01	$3s^2 3p^3$	${}^2\mathrm{P}_{1/2}$	24524.83
$2s^2 2p^3$	${}^2\mathrm{P}_{1/2}$	40470.00	$3s^2 3p^3$	${}^2\mathrm{P}_{3/2}$	24571.54

Table 5: O II and S II radiative data taken from the book "Astrophysics of Gaseous Nebulae" by Osterbrock. The transition probabilities are given in units of $\sec^{-1}$.

Upper		Lower	O II	S II
$^2P_{1/2}$	–	$^2P_{3/2}$	1.4×10^{-10}	1.0×10^{-6}
$^2D_{5/2}$	–	$^2P_{3/2}$	1.1×10^{-1}	1.8×10^{-1}
$^2D_{3/2}$	–	$^2P_{3/2}$	5.8×10^{-2}	1.3×10^{-1}
$^2D_{5/2}$	–	$^2P_{1/2}$	5.6×10^{-2}	7.8×10^{-2}
$^2D_{3/2}$	–	$^2P_{1/2}$	9.4×10^{-2}	1.6×10^{-1}
$^4S_{3/2}$	–	$^2P_{3/2}$	5.8×10^{-2}	2.2×10^{-1}
$^4S_{3/2}$	–	$^2P_{1/2}$	2.4×10^{-2}	9.1×10^{-2}
$^2D_{5/2}$	–	$^2D_{3/2}$	1.3×10^{-7}	3.3×10^{-7}
$^4S_{3/2}$	–	$^2D_{5/2}$	3.6×10^{-5}	2.6×10^{-4}
$^4S_{3/2}$	–	$^2D_{3/2}$	1.8×10^{-4}	8.8×10^{-4}

Table 6: O II and S II collision strengths taken from the book "Astrophysics of Gaseous Nebulae" by Osterbrock. The collision strength is dimensionless.

Transition	O II	S II	Transition	O II	S II
$\Omega(^4S_{3/2},^2D_{5/2})$	0.80	4.19	$\Omega(^4S_{3/2},^2D_{3/2})$	0.54	2.79
$\Omega(^4S_{3/2},^2P_{3/2})$	0.27	1.52	$\Omega(^4S_{3/2},^2P_{1/2})$	0.13	0.76
$\Omega(^2D_{3/2},^2D_{5/2})$	1.17	7.59	$\Omega(^2D_{3/2},^2P_{1/2})$	0.28	1.52
$\Omega(^2D_{3/2},^2P_{3/2})$	0.41	3.38	$\Omega(^2D_{5/2},^2P_{1/2})$	0.30	2.56
$\Omega(^2D_{5/2},^2P_{3/2})$	0.73	4.79	$\Omega(^2P_{1/2},^2P_{3/2})$	0.29	2.38

8.2 Atomic data for O III and S III

Table 7: O III and S III energy levels in cm^{-1} from the NIST website. Here the ordering is the same

O III			S III		
Configuration	Level	Energy	Configuration	Level	Energy
$2s^2 2p^2$	$^3P_0$	0.0	$3s^2 3p^2$	$^3P_0$	0.0
$2s^2 2p^2$	$^3P_1$	113.178	$3s^2 3p^2$	$^3P_1$	298.69
$2s^2 2p^2$	$^3P_2$	306.174	$3s^2 3p^2$	$^3P_2$	833.08
$2s^2 2p^2$	$^1D_2$	20273.27	$3s^2 3p^2$	$^1D_2$	11322.7
$2s^2 2p^2$	$^1S_0$	43185.74	$3s^2 3p^2$	$^1S_0$	27161.0

Table 8: O III and S III radiative data taken from the book "Astrophysics of Gaseous Nebulae" by Osterbrock. The transition probabilities are given in units of sec^{-1}.

Lower		Upper	O III	S III
$^1D_2$	–	$^1S_0$	1.8×10^{0}	2.2×10^{0}
$^3P_2$	–	$^1S_0$	7.8×10^{-4}	1.0×10^{-2}
$^3P_1$	–	$^1S_0$	2.2×10^{-1}	8.0×10^{-1}
$^3P_2$	–	$^1D_2$	2.0×10^{-2}	5.8×10^{-2}
$^3P_1$	–	$^1D_2$	6.7×10^{-3}	2.2×10^{-2}
$^3P_0$	–	$^1D_2$	2.7×10^{-6}	5.8×10^{-6}
$^3P_1$	–	$^3P_2$	9.8×10^{-5}	2.1×10^{-3}
$^3P_0$	–	$^3P_2$	3.0×10^{-11}	4.6×10^{-8}
$^3P_0$	–	$^3P_1$	2.6×10^{-5}	4.7×10^{-4}

Table 9: O III and S III collision strengths taken from the book "Astrophysics of Gaseous Nebulae" by Osterbrock. The collision strength is dimensionless.

Transition	O III	S III	Transition	O III	S III
$\Omega(^3P_0,^1D_2)$	0.24	0.93	$\Omega(^3P_1,^1D_2)$	0.72	2.80
$\Omega(^3P_2,^1D_2)$	1.21	4.66	$\Omega(^3P_0,^1S_0)$	0.03	0.13
$\Omega(^3P_1,^1S_0)$	0.09	0.40	$\Omega(^3P_2,^1S_0)$	0.16	0.66
$\Omega(^3P_0,^3P_2)$	0.27	1.11	$\Omega(^3P_0,^3P_1)$	0.54	2.64
$\Omega(^1D,^1S)$	0.62	1.88	$\Omega(^3P_1,^3P_2)$	1.29	5.79