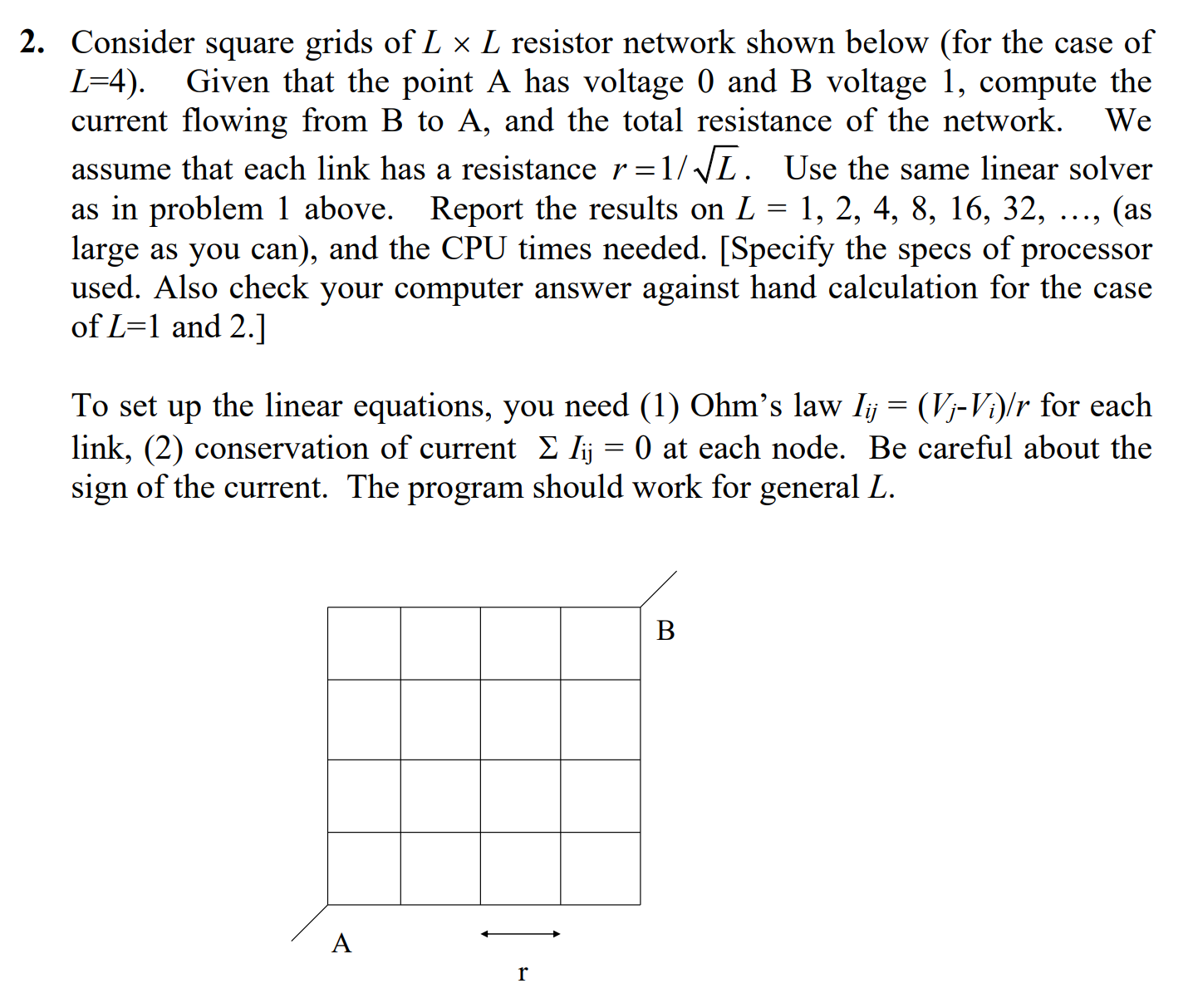
**Using LU decomposition to solve for the node voltages and currents in a square grid of resistors**

This report details the procedure for calculating the grid resistance of a grid of resistors. The question posed is:



To begin solving this, a matrix representing the grid must be constructed. In this, the node voltages were obtained using simultaneous KCL equations involving all the node voltage variables and the coefficients were taken as the matrix **coeff**.

To insert the voltages at the start and end points of the grid, an Ohmic contact (of 1e-100 resistance) was introduced at the KCL equations for points A and B, thus allowing us to arbitrarily set the voltages at A and B.

The coefficients follow a very regular pattern, and hence **coeff** could be constructed easily. The geometry of the matrix allows us to find the position of the neighbours of a specific point and hence we need but add or subtract 1 per neighbour to represent their contributions in the KCL equations. Correspondingly, each row in **coeff** represents a KCL equation.

After we have obtained **coeff,** the RHS of the KCL equations is stored as **b**. Here, The KCL equations have been balanaced such that the RHS is 0, as the sum of currents entering any node must be null. However, at A and B, we have to account for the arbitrary voltages and hence the RHS becomes (r\*Vin)/tr and (r\*Vout)/tr respectively, where r is the resistance of each resistor, tr is the Ohmic contact resistance (1e-100) and Vin and Vout are the voltages at A and B.

Lastly to solve for the values of the node voltages, LU decomposition is used to solve **coeff \* v = b**, where v is the column matrix of node voltages.

LU decomposition of **AX=B** works by first finding a lower triangular matrix **L** and upper triangular matrix **U** such that **LU = A**. Then, a matrix **Y** is found such that **LY = B**, and then **X**, such that **UX = Y**, which means that **LUX = AX = LY = B**. LU decomposition is used as it becomes easier to solve problems like **AX=B** if **A** were an upper or lower triangular matrix. To obtain these, we decompose **A** into two such matrices.

When applying LU decomposition to **coeff \* v = b**, we obtain **v**¸ which is then printed out in a grid. Using KCL, we can obtain half of the total current flowing through the grid by looking at the node voltage difference between the first resistor in the grid (between point B and any of its immediate neighbours). The total current is twice of this, Having found the total current, we can find the grid resistance by dividing the difference in the voltages at A and B by the total current.

Algorithm pseudo-steps

1. Calculate the node voltage coefficients of the KCL equation for each node and store it in a matrix **coeff**
   1. To the end points of the grid, insert an additional branch of infinitesimal resistance (Rt = 1e-100 Ohms) to link those points to the endpoint voltages (Vin and Vout)
2. Set the result matrix **b** to zeroes, with the exception of the endpoint node voltages set to (Vin\*R)/Rt and (Vout\*R)/Rt respectively.
3. Use LU decomposition to solve for the node voltages, as a matrix **v**
   1. Use LU decomposition to decompose **coeff** into LU
      1. The appropriately sized (N=n^2, since n\*n grid has n^2 nodes, resulting in an N\*N matrix to solve) upper and lower triangular placeholder matrices are initialised.
      2. Both matrices are then filled according to the LU decomposition algorithm
   2. Use forwards-substitution to solve **Ly** = **b**
   3. Use backwards-substitution to solve **Ux** = **y**, where **v** = **x**
4. The total current through the grid is twice of that between one of the endpoint branches
5. The grid resistance is then found by dividing the voltage difference by the total current

Results:

For n = 1, I = 1A, R = 1 Ohms

For n = 2, I = 1.4142135623730951 A, R = 0.7071067811865475 Ohms

For n = 4, I = 1.076923076923077 A, R = 0.9285714285714284 Ohms

For n = 8, I = 1.036442362909999 A, R = 0.9648389874689408 Ohms

For n = 16, I = 1.108487770502693 A, R = 0.9021299346825501 Ohms

For n = 32, I = 1.2597970218647323A, R = 0.7937786664393088 Ohms

Sample images of program running, with results and grids printed:

