You must complete the following task by 5pm on Friday 02/07/14.

Your solutions to the homework must be committed to your Subversion repository in a sub-directory HW01.

You may work in a group of one, two, three, or four students.

All source code, header files, \LaTeX files, Makefiles etc should be committed to your own repository in a directory named HW01. You are required to include a Makefile that creates an executable called hw01 in that directory.

Use \LaTeX to write and typeset your report (saved as HW01/report.pdf in your Subversion repository).

If you are working in a team you may discuss the effort with the other member of your team. Otherwise you may only consult the instructor or graders for verbal assistance. You are encouraged to use textbooks and internet resources. You must cite all resources used via footnotes or a bibliography.

100 points will be awarded for a successful completion.

Extra credit will be awarded as appropriate.
Q1 Build a sparse matrix representation of a resistor network. Use iterative methods to solve for the circuit loop currents. Compute the currents in each wire. Determine how large a circuit you can compute and describe factors limiting the possible size of computations.

Problem description: Build a matrix representation for the loop currents in a circuit composed of $N \times N$ closed loops as illustrated in the case of $N = 9$. The black lines correspond to lossless wires. The black circles correspond to lossless junctions. Assume all wires in the network include a resistor of 1 Ohm resistance. The circuit is powered by a 1 Volt DC battery represented by a blue rectangle.


Tasks: Your task is to create a serial C++ code that builds a circuit matrix (representing the Kirchhoff constraints on the loop currents) and load vector (representing the voltage source). Your code should then solve for the currents in each indivisible loop and finally compute the current in each wire of the resistor network. You should write a flexible code that can handle an $N \times N$ grid circuit for arbitrary, user supplied, integer $N$.

Method of loop currents You should choose loop currents as the unknowns in your linear system. In this representation you will associate a current with each indivisible quadrilateral element. After computing the loop currents you will reconstruct the current in each wire by combining the loop currents of the one or two elements that have the wire as an edge.

You are expected to use a C++ sparse matrix class to store the circuit matrix. The load vector and solution vector should be stored with a dense storage convention.

The details are up to you.

Linear System Solver: each member of the team will implement a different iterative solver from the following list to solve the linear system to find the unknowns:

A. Jacobi method [wiki]
B. Gauss-Seidel method [wiki]
C. Successive over relaxation [wiki]. You will have to determine a reasonable relaxation parameter.
D. Conjugate gradient method [wiki]
Compare the number of iterations and number of operations that your chosen methods require.

[Ninja extra credit:] The convergence for A-C is likely to be slow, taking 1000s of iterations to drop the equation residual a few decimal places. Adventurous students can investigate if a multigrid solver yields better performance.

**Test case:** For a small circuit (say with $N = 2$) compute the solution by hand and make sure that your solver yields the correct wire currents for this case.

**Scale up:** Try running circuits composed of $N \times N$ loops for $N = 1E1, 1E2, 1E3, 1E4, 1E5, 1E6$. Is your circuit solver able to handle these sized circuits? What are the limiting factors? Give timing results and estimate memory requirements.