Speedup of 'SPICE like solver for PEEC frequency domain problems' using static coupling matrices

Report nr.8, L'Aquila

June 15, 2002

Introduction

A 'full-wave' solution of a frequency domain formulated PEEC problem require the calculation of $(m^2 - m)$ complex capacitive and $(n^2 - n)$ complex inductive elements, for m and n unknown capacitive and inductive cells respectively. The calculation done in the C++ code, using a surface or contour method, involves the usage of Gaussian integration of a specific order. This results in large computation time for large problems, see *Example* section. A common used approach to decrease computation time is to exclude the calculation of partial element couplings that are far apart, ie weakly coupled cells. This decrease the element calculation time and result in more sparse matrices than for a complete model. The solution time can then also be decreased if some 'sparse matrix solver' can be used. In our code, a LU decomposition routine is used to solve the PEEC model and a sparse system matrix is thus not 'needed'. Instead, the code has been altered so different methods with different accuracy, *read order of Gaussian integration*, can be used according to prediefined 'rules'. This has been accomplished through the use of two coupling matrices, inductive and capacitive, and manually set coupling limits and order of Gaussian integration used in the methods.

Scheme

To be able to use different routines for the calculation of the partial elements the C++ code has been changed in the following ways.

1. The program starts with the calculation of static coupling matrices, c_c and i_c, using closed form solutions that are exact and fast. In this test version all the elements are orthogonal with zero thickness. The self elements are also stamped into the 'real' matrices, $pijkc$ and $lpijkc$.

2. Then the frequency loop starts.

3. Before the complex partial elements are calculated, for a specific frequency, the values in the c_c and i_c matrices are used to calculate the coupling coefficients, c_{cap} and c_{ind}, according to eq. 24 and eq. 25 in [1]. Depending on the magnitude of the coupling coefficient a predefined routine can be used for the complex calculation, surface- or contour - routines. *In this way it is possible to control the accuracy in the partial elements and thus the speed of the calculation.* The program calculates the elements and places the 'correct' values in the $pijkc$ and $lpijkc$ matrices. The frequency is stepped and the calculations described in this point is repeated.

4. The complex capacitive and inductive surface and contour routines, $pmnoc$, $lpmnoc$, Pp_cont_c and Lp_cont_c, has also been changed so that the order of Gaussian integration inside the routine is included in the 'calling' of that specific function.

This approach could also be refined by the inclusion of (1) A distance matrix describing the distances between the cells and also (2) A cell dimension matrix. This has not been tested yet.

[1] A. E. Ruehli, H. Heeb, "Circuit Models for Three-Dimensional Geometrices Including dielectrics", IEEE Trans. on Microwave theory & techniques, vol. 40, no.7, july 1992.

Speedup

Some simple tests have been performed using the method described in th eprevious section. The focus, so far, is mainly on the speedup in the calculations and not on the accuracy in the solution. In Figure 1 the calculation time, for seven frequencies, depending on different orders, *8th, 5th 3rd*, of Gaussian integration for the two basic methods are displayed for up to 60 capacitive nodes. It is clearly shown that the surface method is computationally burdensome. In Figure 2 the same 'problems' are solved using two different combinations of computing routines. The first combination, comb1, performes the computation according to

$$
\begin{array}{lll}
coupling \geqslant 0.35 & use & 8th\ order\ surface\ routine \\
0.35 > coupling \geqslant 0.15 & use & 5th\ order\ surface\ routine \\
0.15 > coupling \geqslant 0.10 & use & 3rd\ order\ surface\ routine \\
0.10 > coupling & use & 5th\ order\ contour\ routine
\end{array}
$$

The second combination, comb2, performes calculation according to

$$
\begin{array}{lll}
coupling \geqslant 0.35 & use & 6th\ order\ surface\ routine \\
0.35 > coupling \geqslant 0.15 & use & 5th\ order\ surface\ routine \\
0.15 > coupling \geqslant 0.10 & use & 6th\ order\ contour\ routine \\
0.10 > coupling & use & 5th\ order\ contour\ routine
\end{array}
$$

The results shows that the method gives a considerably speed-up if (1) different order of Gaussian integration can be used and (2) if the surface and contour formulation can be used together.

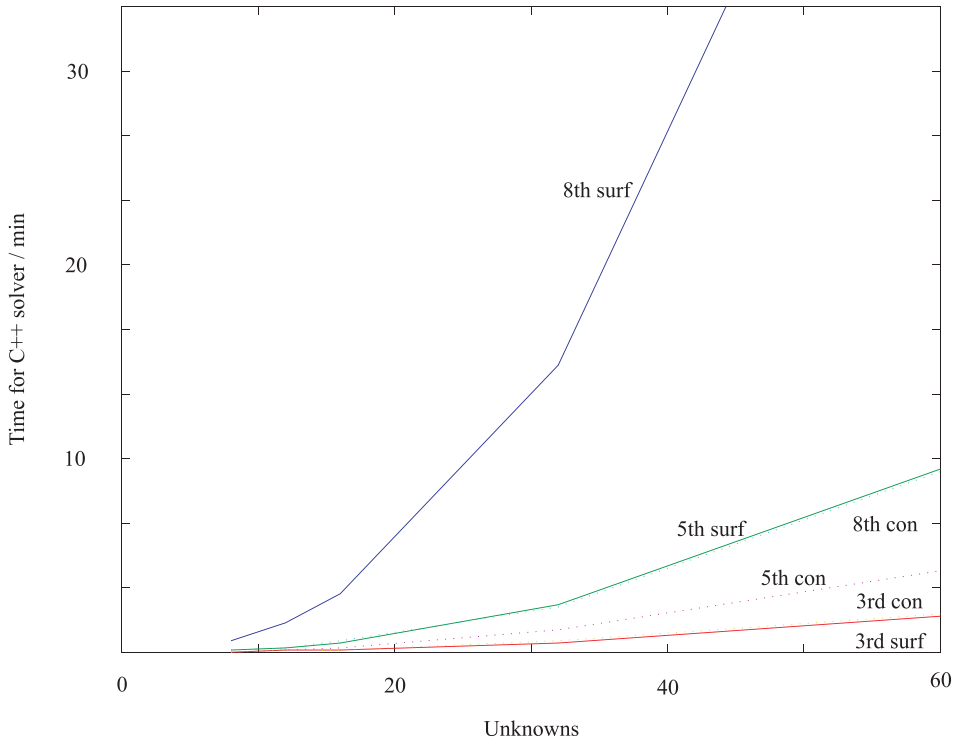

Figure 1: Calculation speed for surface and contour formulation

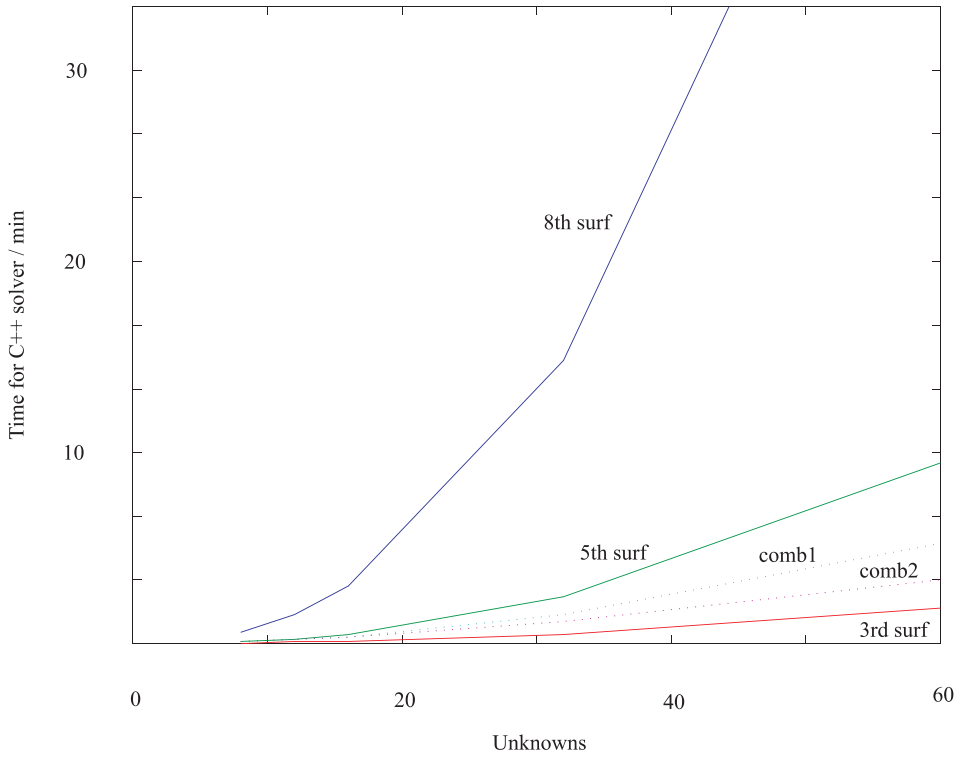

Figure 2: Calculation speed uning surface form. or the 'conditioned solver'

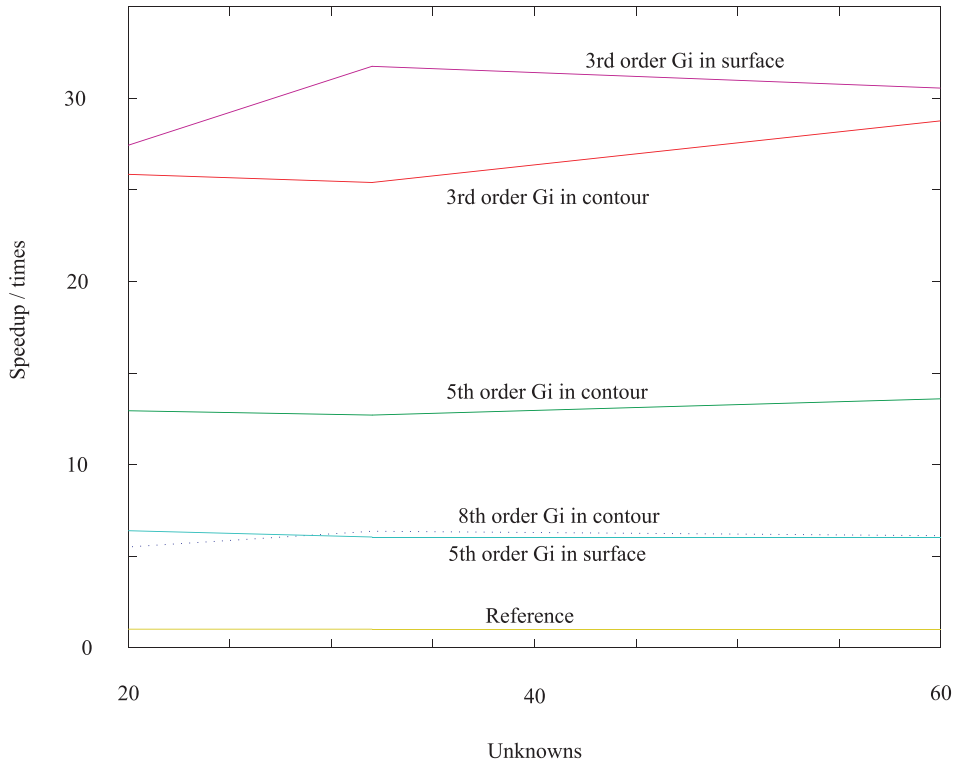

Figure 3: Speedup for different order of Gaussian integration

Speedup vs accuracy 1/2

In this section the accuracy, in term of the dipole input impedance, versus speedup is adressed for a simple 20 cm half wavelength dipole. The dipole is discretized into 20 cells per 10 cm 'arm'. This offers 20 cells per wavelength at 3GHz, wich is the upper frequency limit for this test. For the reference case, Ref in the figures, the mutual partial elements are calculated using 5th order Gaussian integration in the surface routine, indicated with the notation *5 ths* in the table below. The test cases, test1-test4 is performed using (1) different combinations of the surface and contour routine and (2) different accuracy in the Gaussion integration in the routines, as displayed in the tables below. For example the notation *7th c* indicates that the *c*ontour routine has been used with *7th* order of Gaussian integration. For the tests, the computation time is also displayed.

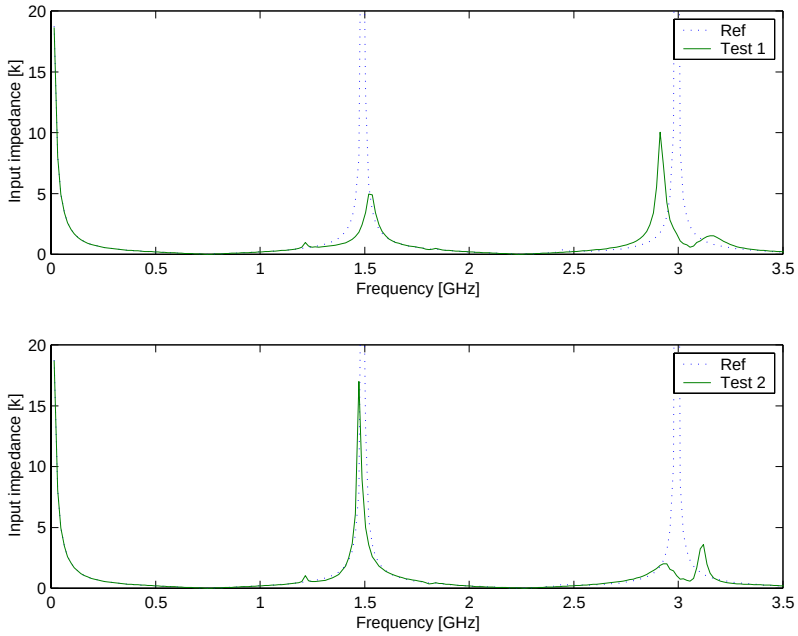

Figure 4: Input impedance for dipole for test1 and 2

Threshold	*Ref*	*Test1*	*Test2*
$coupling \geqslant 0.25$	5th s	3rd s	5th s
$0.25 > coupling \geqslant 0.15$	5th s	3rd s	4th s
$0.15 > coupling \geqslant 0.05$	5th s	3rd s	3rd s
$0.05 > coupling \geqslant 0.01$	5th s	3rd s	3rd s
$0.01 > coupling$	5th s	3rd s	2nd s
	Time = 15625 s.	Time = 3750 s.	Time = 3125 s.

To be noted from figure 4

To use 3rd order of Gaussian integration in the surface routine offer no good accuracy, compared to 5th order Gi(reference case).

For test2, the agreement compared to the reference case is 'good' almost up to the upper frequency limit, ie 3GHz, and the speedup is a factor 5.

Test1 and test2 shows the importance of keeping the accuracy in the strong coupled cells, ie a high order of Gaussian integration. For weakly coupled cells it is possible yo use a low order of Gi, in test2 2nd order.

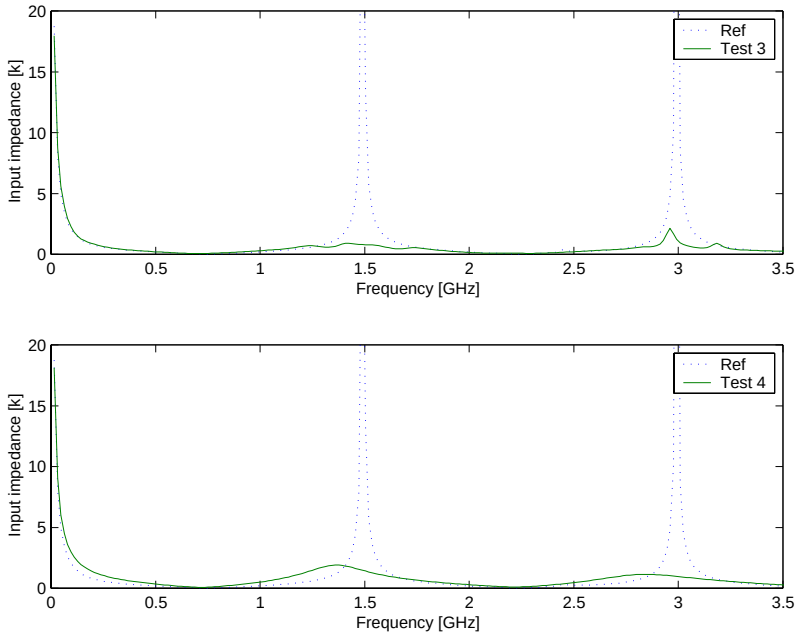

Figure 5: Input impedance for dipole for test3 and 4

Threshold	*Ref*	*Test3*	*Test4*
$coupling \geqslant 0.25$	5th s	4th s	6th c
$0.25 > coupling \geqslant 0.15$	5th s	4th s	6th c
$0.15 > coupling \geqslant 0.05$	5th s	3rd s	5th c
$0.05 > coupling \geqslant 0.01$	5th s	7th c	5th c
$0.01 > coupling$	5th s	6th c	4th c
	Time = 15625 s.	Time = 9375 s.	Time = 9500 s.

To be noted from figure 5

To mix the surface and contour formulation in the same PEEC model is not good, as can be seen in figure 5(top).

To use a high order of Gi in the contour formulation for fullwave PEEC models offer (1) no good agreement compared to the reference case and (2) a speedup of approx. 1.5.

Speedup vs accuracy 2/2

To conclude the testing from the last section, a refined discretization of the dipole is made, Test 5 in Figure 6. The cells are now 2.5mm long, ie 40 cells per dipole 'arm', and the upper freqency limit is now approx. 6GHz. This makes the unknowns twice as many compared to previous test, this is also noted in the computation time, i.e. 40000 s. In this test the surface formulation is used with the same order of Gaussian integration as in Test2. The results, up to 6GHz, are displayed in Figure 6 together with the results from Test 2(valid up to 3 GHz) and Ref (also valid up to 3GHz).

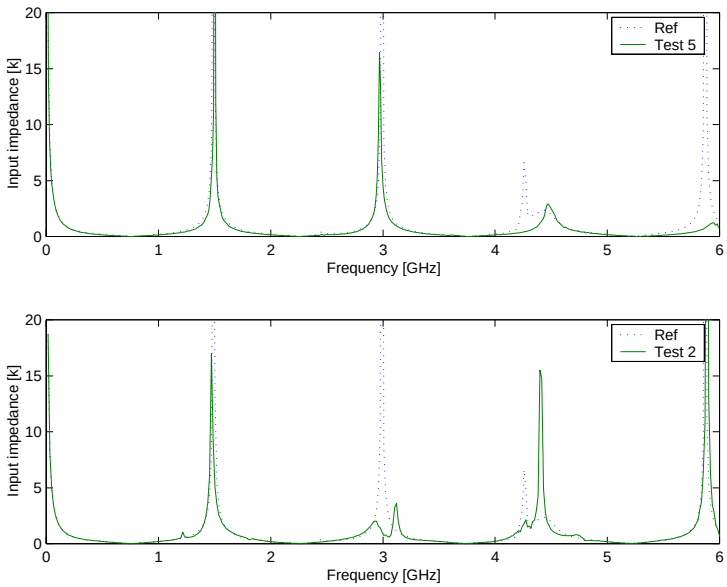

Figure 6: Input impedance for dipole for test2 and 5

This test shows the importance of the discretization, i.e. to use min. 20 cells per shortest wavelength. It also shows the increase in computation time, from 3 125s to 30 000s.

'Clocking' of computer routines

When the speedup is discussed in the previous sestion, we consider the speedup of the complete system. This includes (1) The time it takes to perform the discretization, (2) The time it takes to calculate the partial elements and (3) The time it takes to solve the admittance matrix to obtain the unknown node potentials. For a large system, approx. ¿50 capacitive PEEC nodes, the discretization time becomes less important compared to point (2) and (3). In this section the four different complex routines used in the previous sections(lpmnoc, Lpcontc, pmnoc, PPcontc) and the soution time, for the admittance matrix method, is 'isolated' and clocked/timed. This gives raw data about the different routines used but can it can also be used to estimate the solution time(calculation and solution) for large systems. The results are obtained on an IBM Thinkpad T20, P3-800MHz/256Mb RAM.

In Figure 7 the computation time for the different routines for different order of Gaussian integration are displayed. We can for example see in the figure that it takes 163 ms to calculate one mutual partial inductance using the surface formulation, lpmnoc, with 8th order of Gaussian integration.

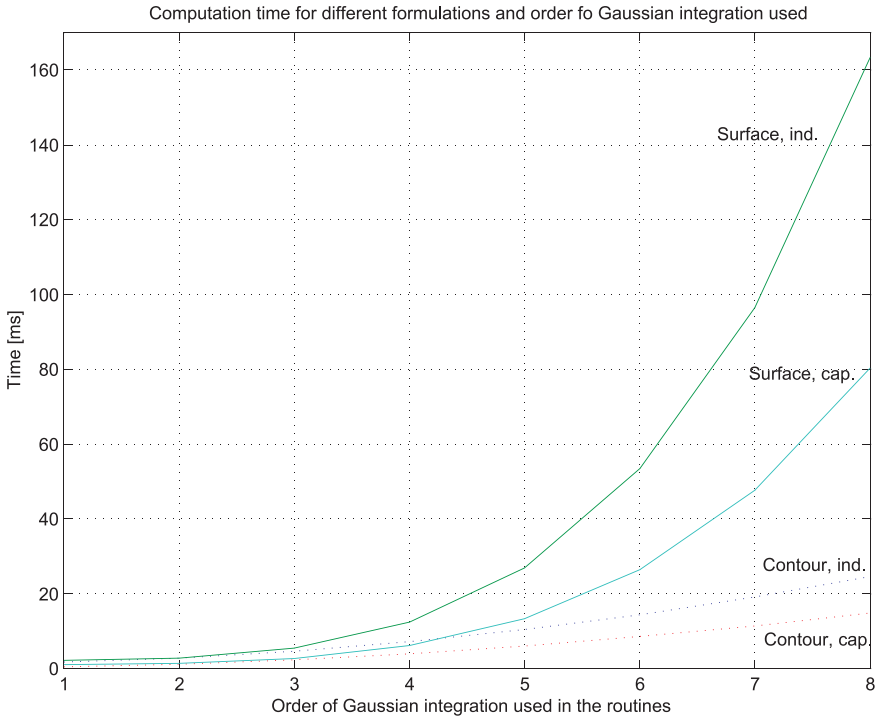

Figure 7: Computation times for the different routines

In Figure 8 the solution time for the admittance matrix is displayed for different number of unknowns. In the upper figure the solution time vs. nbr of unknowns are plotted using lin-lin scale while the lower figure shows the same for log-log scale. In the lower figure the solution time for larger systems has been estimated, using a dashed line. From the figure we can estimate the solution time, for one frequency, for a system with 10 000 unknown PEEC nodes to 2e7 seconds (231 days).

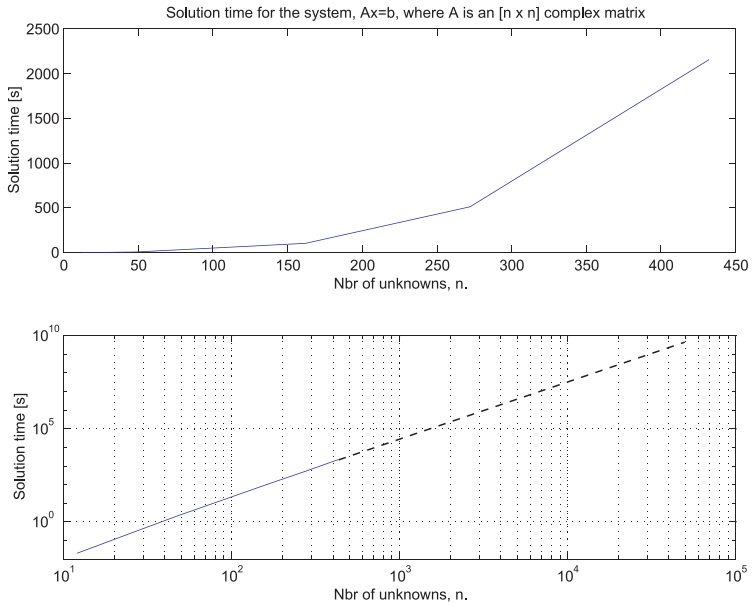

Figure 8: Solution time for the Y-matrix method

The data in Figure 7 and 8 has been used to estimate the solution time for a system of 60 unknowns when the surface routine has been used (5th order of Gaussian integration) to calculate the partial elements. The solution time is calculated as $13.3 \times 10^{-3} \times \frac{60^2}{2} + 26.9 \times 10^{-3} \times \frac{58^2}{2} + 3.5 \approx 73s$. for one frequency. This can be compared with $\approx$ 510s. for seven frequencies, or 72s/freq, from Figure 1.

Concluding remarks

This report decribes the first tests performed using this approach and the results should be read with care. However, the speedup is considerably and the calculations are still done with a good accuracy. For futher work a (1) refined version , discussed in the Scheme section, and (2) a non orthogonal version will be implemented.

This report also shows on the large computation times, for the solution, using the Techsoft Matrix library. It is clear that iterative solvers must be used for large systems.