Parallel computing for eddy current testing simulation
Laurent Santandrea, Guilhem Savel, Yann Le Bihan, Adel Razek

To cite this version:
Laurent Santandrea, Guilhem Savel, Yann Le Bihan, Adel Razek. Parallel computing for eddy current testing simulation. Sixth International Conference on Computation in Electromagnetics (CEM 2006), Apr 2006, Aachen, Germany. 2006, Proceeding of Sixth International Conference on Computation in Electromagnetics (CEM 2006). <hal-01547496>
Eddy current testing (ECT) numerical simulation may need excessive computational time; e.g., a 3D modelling to detect a thin flaw with an air-coil probe. The computational time along a scan line can reach half a day on an Intel Pentium 4 3.2 GHz processor. It is obvious that computational effort becomes prohibitive if we want to optimise the probe or to reconstruct the flaw (inverse problem).

Dual-processors with hyper-threading technology is became a current architecture, and can be obtained with a very reasonable cost. We will see that OpenMP is a well adapted solution to parallelize easily a code and to exploit this machine architecture.

**Parallel Computing**

The computational technology evolution has involved radical modifications in the applications development. In particular the increase of performing computer park as consequence the decrease of cost and the new powerful processors appearance. The fast Ethernet network with large bande has allowed the computer network group.

**MPI** is a library, which is used on distributed memory systems. For MPI the management of these communications is of the user responsibility. MPI is thus often more delicate to be implemented.

OpenMP is based on compiler directives, and is generally used on shared memory machines. The great advantage of OpenMP is that the communication mode between the threads is managed directly by the compiler. In particulary, OpenMP is well adapted to hyper-threading technology. This technology is developed by Intel corporation allows a single processor to manage data as if it was two processors by executing data instructions from different threads in parallel rather than serially.

**Conclusion**

Computational time saving is quasi-proportional to processors number and can be better with a true multi-processors architecture. The proposed method can be used of course with a single hyper-threaded processor such as Pentium III and Pentium 4 processors to exploit parallelism. This technique is well adapted to the ECT problem which is easily divisible in independent problems.