# Brain Activity and Parallel Computing

Bergman, M.[1], van Albada, G.D.[1], Simon, H.[2], Buchner, H.[2], Sloot, P.M.A.[1], Friedrich, V.[3]

[1]Universiteit van Amsterdam, The Netherlands; [2]Dep. of Neurology RWTH, Aachen, Germany;
[3]Parsytec Computer GmbH, Aachen, Germany;

**Abstract.** The ESPRIT project Cauchy$^{par}$ provides an interesting example of the use of HPC in a real-world medical application. Components of a software system used to derive the electrical activity within the brain from the electric and magnetic field outside the skull have been parallelised to attain realistically usable processing times.

## Introduction

A variety of neurological disorders is characterised by specific patterns of electromagnetic activity within the brain. Successful treatment of these disorders, such as some forms of epilepsy, requires that the location of the affected areas is determined as accurately as possible. One way to realise this, is to measure the electric and/or magnetic activity outside the patient's skull (electro-encephalograms and magneto-encephalograms, EEG and MEG) and to reconstruct the source positions by "inverting" a suitable finite element model of the head.

The program CAUCHY - developed at the RWTH Aachen - uses a FEM method to represent a realistically shaped model of the head. CAUCHY, initially designed to reconstruct sources from EEG data, was extended to the treatment of MEG measurements and to the combined treatment of MEG and EEG data. The processing time required for the CAUCHY code was found, however, to be an obstacle to its more widespread use.

The clinical and experimental use of the CAUCHY code is described elsewhere [1][2][3]. One of the important findings was that the simultaneous use of MEG and EEG data leads to a much more reliable reconstruction of the source distribution than the use of either independently. In this note, we describe our experiences in parallelising the significant sections of the code in the ESPRIT project nr. 26433 Cauchy$^{par}$[1].

1 Cauchy$^{par}$ is a collaboration between Parsytec Computer GmbH, Philips Medizin Systeme, the Rheinisch-Westfälische Technische Hochschule Aachen, and the Universiteit van Amsterdam, funded through the TTN Thuringia

## Parallelising the CAUCHY program

The CAUCHY program executes in two phases:

1. A pre-processing phase in which the EEG/MEG response at the measurement points is computed for a large number of possible source locations, using a high-resolution Finite Element Model (FEM) of the brain and the skull.

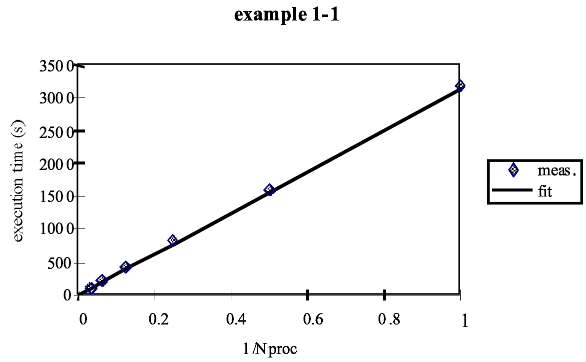
2. A model fitting phase, in which the "lead field matrix" computed in the pre-processing phase is used to reconstruct the actual source distribution from the EEG and/or MEG measurements.

The pre-processing phase is by far the most expensive part of the computation and can take almost a full day on an average workstation. The memory requirements are significant (up to about 200MB), but not unusual. Though various post-processing options exist, the code that we worked on only provided a post-processing method based on simulated annealing.

In order to increase clinical and experimental usability of source reconstruction, it was decided to port the CAUCHY code to a parallel computer, more specifically, a Parsytec CC. The code concerned consists of some 1350 separate modules of FORTRAN 77 code, comprising some 470 000 lines. Obviously, it would not be cost-effective to redesign a significant fraction of this code. In order to analyse the call-structure and performance of the code, we made use of profiling tools on the sequential code. It was found that some 90% of the execution time of the code was spent in a very limited number of routines.

The pre-processing phase can be seen as a collection of calculations of the response at all MEG/EEG sensors due to each of a thousand or more sources of EM activity in the brain. Each calculation essentially generated a single column in the lead-field matrix by iteratively solving a large linear problem described by the FEM and the source parameters. We found that these calculations could be treated independently, leading to an embarrassingly parallel code. Implementing this code meant distributing the entire FEM to all nodes, plus the storage space required for the lead-field matrix. For portability, we chose to use MPI 1.0.13 as the communication library. Though MPI is widely accepted, it has certain drawbacks. One of these is that it enforces the use of the SPMD formalism for the program. For FORTRAN, this implies (barring non-standard tricks) that all nodes need to assign the maximum amount of memory needed by any one node.

Figure 1 shows the performance attained for two realistic test problems with this code.



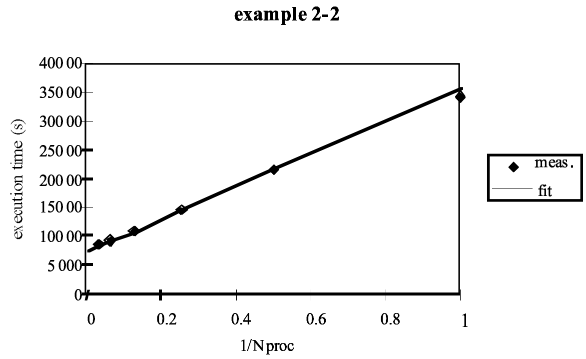




Fig. 1 – The execution times for two test cases run on 1, 2, 4, 8, 16 and 32 nodes plotted against the inverse of the number of processor nodes. The first case concerns a relatively small problem with no post-processing, the latter, a large problem with significant post-processing. The plots clearly show the expected behaviour for embarrassingly parallel codes with a fixed start-up/post-processing time. The fitted lines have been derived from the execution times for more than one node. Some extra overhead for the parallel code is evident in the second case, as the single node point lies 4 minutes below the extrapolated line.

The increased performance of the parallelised code led to the desire to run higher-resolution, and thus larger models, and we immediately ran into memory problems. A number of options were considered; in the end, we chose to completely separate the pre-processing and post-processing code. The latter had not been parallelised anyway, and its memory requirements only were a burden to the parallel code. And, as more than one post-processing method had to be supported, this also led to a cleaner code structure. It also meant that the lead-field matrix was never needed in its entirety in the pre-processing code. We chose to write it to disk, using a direct-access file to maintain the correct order of the columns.

Presumably, we could have avoided the memory problems by parallelising each of the iterative FEM solutions. Had the memory problems been much more severe, this had been the way to go, but it would have involved a much larger programming effort and would thus have been more error prone. Also, the attainable speed-up would not have been as good.

## Conclusions

After parallelisation, the software package CAUCHY is now well suited to meet clinical requirements. It can be run on MPP machines as well as on clusters of workstations. The resulting, highly scaleable system can reduce the calculation time for the localisation of electrical activity from some days to a few hours or even less. An additional advantage is the improvement of the underlying model in terms of resolution.

The price-performance of the Cauchy$^{par}$ software together with the parallel system and the EEG hardware, which mostly is already available at a neurological department of a hospital, is better than competing techniques requiring e.g. invasive operation methods and/or more expensive diagnostic hardware. The decrease of the patient's operational risk is an additional and probably the most important advantage.

## References


1. Buchner H, Knoll G, Fuchs M, Rienäcker A, Beckmann R, Wagner M, Silny J, Pesch J: *Inverse localization of electric current sources in finite element models of the human head*. Electroenceph. clin. Neurophysiol., 102: 267-278, 1997.
2. Pohlmeier R, Buchner H, Knoll G, Rienäcker A, Beckmann R, Pesch J: *The influence of skull-conductivity misspecification on inverse source localization in realistically shaped finfite element head models*. Brain Topography, 9(3): 157-162, 1997.
3. Buchner H, Kaiser S, Sloot P.M.A, Fuchs M, Wassmuth K: *EEG/MEG Quellen Rekonstruktion - Die Funktionen von CURRY und CAUCHY$^{par}$*. In T. Lehmann, V. Metzler, K. Spitzer, T. Tolxdorff, editors, *Bildverarbeitung für die Medizin 1998: Algorithmen-Systeme-Anwendungen*: 455-459, 1998. ISBN 3-540-63885-7.