***Institute for Computer Science, Friedrich Schiller University Jena*** ***fmi.uni-jena.de***
***H. Martin Bücker, Ralf Seidler*** {***martin.buecker***,***ralf.seidler***}***@uni-jena.de***

***Parallel Computing 2, Summer 2015***

# Exercise 4

**Problem 4.1** (Monte Carlo Approach for the Computation of $\pi$ Using OpenMP)**.** Implement a simple algorithm for the computation of $\pi$ by a Monte Carlo approach. To this end, consider the unit square and choose $N$ points in that square at random. The area of the unit square is 1 and the area enclosed by a quarter of a cricle centered at the origin and with radius 1 is $\pi/4$. Therefore, an approximate to $\pi$ is the ratio of the number of randomly chosen points that fall into the quarter of the unit circle and all chosen points, $N$. A corresponding algorithm is given by the following pseudocode:

```
points_within_circle = 0;
for (N points) {
    Choose x and y coordinates of the point randomly
    if (abs(x,y) <= 1){
       points_within_circle += 1;
    }
}
mypi = 4*points_within_circle/N
```

Implement a parallel algorithm using OpenMP.

Conclusion: *Pay attention to synchronization of threads.*

**Problem 4.2** (Compute the Leadfield Matrix Using OpenMP)**.** In the neuromagnetic inverse problem, one is interested in determining the current density inside the human brain from measurements of the magnetic field recorded outside the head. The solution of this inverse problem involves a function

```
basis(S,R,nm,L,nd,G,ng)
```

that computes the Leadfield matrix `S` from a given `nm` $\times$ 6 matrix `R`, a given `nd` $\times$ 3 matrix `L`, and a given `ng` $\times$ 2 matrix `G`. Details of the mathematical model are available in the following article:

- H. M. Bücker, R. Beucker, and A. Rupp. Parallel minimum $p$-norm solution of the neuromagnetic inverse problem for realistic signals using exact Hessian-vector products. *SIAM Journal on Scientific Computing*, 30(6):2905–2921, 2008. `http://epubs.siam.org/doi/abs/10.1137/07069198X`

Parallelize the given program for the computation of the Leadfield matrix using OpenMP.

Conclusion: *Think about data scoping when parallelizing with OpenMP.*