Otto-von-Guericke-Universität Magdeburg
Max-Planck-Institut for Dynamics of Complex Technical Systeme
Computational Methods in Systems and Control Theory

Dr. Jens Saak, Dipl.-Math. Martin Köhler
Website: http://www.mpi-magdeburg.mpg.de/mpcsc/lehre/2013_SS_SC/

summer term 2013

Scientific Computing 2
Tutorial 1

Exercise 1:

Given is a set of algorithms which have the following properties depending on an input of size n:

	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
Number of Flops	$815n$	$2n^2 + n$	$6n^3$	$15n^3$
Percentage of parallelizeable code	90%	99%	15%	80%

a.) Compute the upper bounds for the speedup of all three algorithms.

b.) Consider an input of size $n = 2000$ and a CPU with a performance of 5 GFlop/s. Compute the runtime and the speedup if you employ 1, 2, 4 or 8 of these CPUs in parallel.

Exercise 2:

Consider the following pseudo code algorithms. Compute the number of performed flops in dependence of the size of the input. Think about which parts of them are parallelizeable. Use Amdahl's Law to determine the maximum theoretical speedup.

a.) The scalar (dot) product:

Input: $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$
Output: $s = (x, y) = x^T y$

```
s := 0
for i := 1, ..., n do
    s := s + x_i y_i
end for
```

b.) The general matrix-matrix product (GEMM):

Input: $A \in \mathbb{R}^{n \times k}$, $B \in \mathbb{R}^{k \times m}$, $C \in \mathbb{R}^{n \times m}$
Output: $C := C + AB$

```
for i := 1, ..., n do
    for j := 1, ..., m do
        for l := 1, ..., k do
            C_ij := C_ij + A_il B_lj
        end for
    end for
end for
```

c.) The LU decomposition without pivoting:

Input: $A \in \mathbb{R}^{n \times n}$
Output: $LU := A$, A is overwritten by L and U

for $k = 1, \dots, n-1$ **do**
  $A_{k+1:n,k} := A_{k+1:n,k} / A_{k,k}$
  $A_{k+1:n,k+1:n} := A_{k+1:n,k+1:n} - A_{k+1:n,k} A_{k,k+1:n}$
end for

d.) Forward substitution $Lx = b$ (Backward substitution is the same):

Input: $L \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$
Output: $x \in \mathbb{R}^n$, $x = L^{-1} b$, b is overwritten by x

$b_1 := b_1 / L_{1,1}$
for $i := 2, \dots, n$ **do**
  $b_i := b_i - L_{i,1:i-1} b_{1:i-1}$
  $b_i := \frac{b_i}{L_{i,i}}$
end for

e.) Solving a linear system $Ax = b$, $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^{n \times n}$. Use the LU-decomposition from c.) and the forward/backward substitution from d.).

Exercise 3:

Consider the so called vector-triad benchmark:

Input: $a, b, c, d \in \mathbb{R}^n$
Output: $a_i = b_i + c_i d_i \quad \forall\, i = 1 \dots, n$

for $i = 1, \dots, n$ **do**
  $a_i := b_i + c_i d_i$
end for

a.) Compute the performance (in GFlops/s) when we execute the vector-triad sequentially on a CPU with a performance of 12.8 GFlops/s for different vector lengths n:

Length n of a, b, c, d	Runtime in s
$n = 10\,000$	
$n = 100\,000$	
$n = 1\,000\,000$	
$n = 100\,000\,000$	

b.) Explain why the practical results are much different from the theoretical ones. Why does even employing a second CPU-core with an additional performance of 12.8 GFlops/s not result in a performance gain?