



The Graduate School of Management and Economics

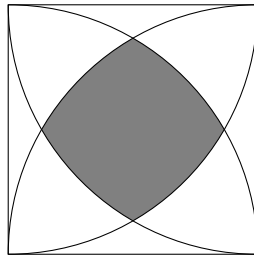
HPC in Julia

SIMD

1.
 - a) write code in julia with `@simd` for computing $tr(AB)$ (your function will get matrix A and B and return $tr(AB)$). (your code must be efficient!)
 - b) Are you use inbound macro ? why?
 - c) use trace function in *LinearAlgebra* package for computing trace and compare performance of them with each other in one plot (for different dimension of A and B. in this part you can assume that A,B are square and have same dimension)

Multi Threading

- 2) write function in julia that get two matrix A,B and compute their multiplication you must use multi threading approach (use 8 thread) and compare your function with julia Matrix multiplication for different dimension in one plot.
- 3) calculate area of colored region in unit square with monte carlo method and use multi threading approach. (write code!)



Distributed Computing

4) Given a needle of length ℓ dropped on a plane ruled with parallel lines t ($\ell < t$) units apart, the probability that the needle will lie across a line upon landing is $\frac{2\ell}{\pi t}$. from this fact we can estimate π . use monte carlo simulation with distributed implementation. compare performance of distributed implementation with serial implementation. How much distributed implementation is faster? [link](#))

Static Arrays

5) At first install *StaticArrays* package in julia.

a) when and why we used *StaticArrays*?

b) compare matrix multiplication performance of *StaticArrays* with julia own multiplication for different dimension in one plot.

Yeppp!

6) at first install *Yeppp* package in julia. a) which mathematical operation has optimized in *Yeppp*!? give some intuition that how they optimize this operation b) compare performance one operation of *Yeppp* with julia own operation. (you can do this operation on array)