

# Performance of OCaml program of FinMatrix by extraction

---

date: 2024.04.29

## Introduction

---

This is a test for ocaml program of FinMatrix by Coq extraction, which contain matrix multiplication, inversion by GE and inversion by AM.

We use randomly generated matrices of floating point numbers.

The source code are located in [Here](#)

## Environment

---

PC: Thinkbook 14 G2-ITL

OS: Debian 11.8

## Usage

---

- there are two version programs
  - `./matrix.byte`, byte-code version
  - `./matrix.opt`, opt version
- options
  - n=10, size of input matrix
  - mmul=true, enable multiplication
  - GE=true, enable matrix inversion by GE
  - AM=true, enable matrix inversion by AM
- examples
  - opt version, matrix multiplication with two 100\*100 matrices  
`./matrix.opt -mmul=true -n=10`
  - byte-code version, matrix inversion by GE with one 5\*5 matrix  
`./matrix.byte -GE=true -n=5`
  - opt-code version, matrix inversion by AM with one 5\*5 matrix  
`./matrix.byte -AM=true -n=5`

## Result

---

- matrix multiplication: `A(float, n, n) * B(float, n, n)`  
time `./matrix.byte -mmul=true -n=64`  
time `./matrix.opt -mmul=true -n=64`

1	n	time(s)-byte	time(s)-OPT
2	64	0.057	0.027
3	128	0.295	0.074
4	256	1.985	0.430
5	512	16.0	3.0
6	1024		28.0

time =  $O(n^3)$

- matrix inversion by GE `invGE (A(float,n,n))`

time ./matrix.byte -GE=true -n=5

time ./matrix.opt -GE=true -n=5

1	n	time(s)-byte	time(s)-OPT
2	5	0.009	0.004
3	6	0.017	0.006
4	7	0.030	0.015
5	8	0.088	0.031
6	9	0.360	0.085
7	10	1.660	0.377
8	11	8.150	1.723

- matrix inversion by AM `invAM (A(float,n,n))`

time ./matrix.byte -AM=true -n=5

time ./matrix.opt -AM=true -n=5

1	n	time(s)-byte	time(s)-OPT
2	5	0.013	0.005
3	6	0.025	0.010
4	7	0.103	0.030
5	8	0.940	0.150
6	9	10.45	1.550
7	10		18.95

## Conclusion

- The performance of matrix multiplication is same as "Dependent Record by List" model.
- The performance of matrix inversion is too bad, due to the "function" model.

## Future work

- We plan to use a better model, such as axiomized way of "Array.array",  
Note that, this way have strong similarity of FinMatrix, we are doing it...