

LDLT FACTORIZATION IN CUDA

for systems solving

Sunday 22nd March, 2020

Maxime DARRIN Pierre GUETSCHÉL

M2A - Sorbonne Université

OUR PROJECT

The project

We build our project in two parts:

- 1 The factorization algorithm
- 2 The solver (using a factorized form)

Hardware

Our experiments have been conducted on a GTX 1060 for laptop.

DATA STORAGE

Storage of n matrix of size $d * d$

Matrices L and D :

$D_{1,1}^1$	$D_{2,2}^1$	\dots	$D_{d,d}^1$	$L_{1,1}^1$	$L_{2,1}^1$	\dots	$L_{d,1}^1$	\dots	$L_{d,d}^1$	\dots
$\dots D_{1,1}^2$	\dots	$L_{d,d}^n$								

Matrix A :

$A_{1,1}^1$	$A_{2,2}^1$	\dots	$A_{d,d}^1$	\emptyset	$A_{2,1}^1$	\dots	$A_{d,1}^1$	\dots	\emptyset	\dots
$\dots A_{1,1}^2$	\dots	\emptyset								

with $M_{i,j}^k$ being the element (i,j) of the k^{th} matrix M .

We choosed to store the diagonal elements of L to simplify our code.

This configuration allows us to compute the factorization in place.

THE FACTORIZATION

	Max Col	Max k (row)	row + shared memor
Execution time	1.489760 ms	1.487296 m ms	0.514624

Figure 1: Comparison on small matrices. (100 matrices of size 32x32)

	Max Col	Max k (row)
Execution time	1106.73 ms ms	1108.9 ms

Figure 2: Comparison on large matrices. (100 matrices of size 512x512)

Error propagation

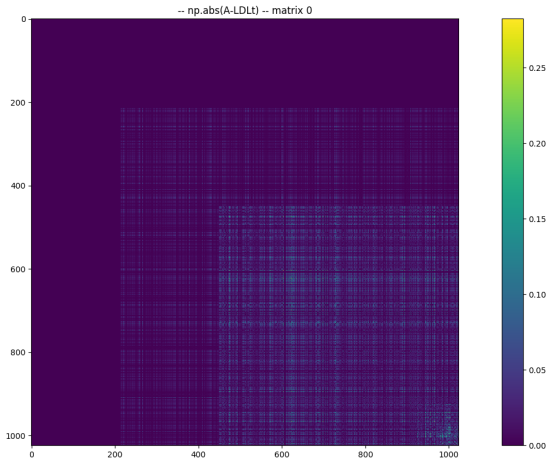


Figure 3: Error propagation on a big matrixe

THE SOLVER

d =	16	128	512
Execution time	0.084 ms	0.960 ms	12.50 ms

Figure 4: Comparison with 128 threads and 100 matrices (on per block)

Behavior

We have a gain of time which is linear in the number of threads.

THE FULL PIPELINE

	Max Col	Max k (row)	row + shared memory
Execution time	1108.7ms	1163.1 ms	0.0091 ms
Solving time	13.9 ms	13.9 ms	13.9 ms

Figure 5: Comparison on large matrices. (100 matrices of size 512x512)

The end



Figure 6: A pangolin, probably the source of our current sorrows.