

# Implementation av ett effektivt matrisbibliotek i C

Redaktör: Martin Söderén

Version 0.1

## Status

|          |                 |   |
|----------|-----------------|---|
| Granskad | Martin Söderén  | - |
| Godkänd  | Andreas Runfalk | - |

## Innehåll

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Inledning</b>  | <b>1</b> |
| 1.1      | Syfte . . . . .   | 1        |
| 1.2      | Frågeställning . . . . .  | 1        |
| 1.3      | Avgränsningar . . . . .   | 1        |
| <b>2</b> | <b>Bakgrund</b>   | <b>1</b> |
| <b>3</b> | <b>Teori</b>  | <b>2</b> |
| 3.1      | Tidskomplexitet på nuvarande implementationer operationer . . . . . | 2        |
| 3.2      | Eventuella förbättringar . . . . .                                  | 2        |
| <b>4</b> | <b>Metod</b>  | <b>2</b> |
| <b>5</b> | <b>Resultat</b>   | <b>2</b> |
| <b>6</b> | <b>Diskussion</b>   | <b>2</b> |
| 6.1      | Resultat . . . . .  | 2        |
| 6.2      | Metod . . . . .   | 2        |
| <b>7</b> | <b>Slutsatser</b>   | <b>2</b> |

## 1 Inledning

Denna rapport går igenom eventuella optimeringar av en befintlig implementation av ett matrisbibliotek. I slutändan så implementerades optimeringarna och jämförelser gjordes mellan den gamla och den nya versionen av biblioteket.

### 1.1 Syfte

Optimera av ett matrisbibliotek som en lösare av kvadratiske konvexa optimeringsproblem ska använda.

### 1.2 Frågeställning

Finns det möjlighet till förbättringar vad gäller prestanda i den befintliga implementationen av matrisbiblioteket matLib som har tagits fram under kandidatprojektet?

### 1.3 Avgränsningar

Matrisbiblioteket kan körs på alla plattformar som har en c kompilare men i denna rapport så körs det endast x86-64.

Koden kompileras utan optimeringsflaggor så att alla optimeringar sker i koden.

## 2 Bakgrund

Under kandidatprojektet Prediktionsreglering så skulle en lösare av kvadratiske komplexa problem tas fram. För detta behövdes ett matrisbibliotek väljas men inget bibliotek uppfyllde kraven. Dessa var:

1. Lättanvänt api
2. Bra prestanda
3. Platformsberoende
4. Lätt att kompilera
5. Tar upp lite minne
6. Bra dokumenterad kod så man själv kan implementera förbättringar

Inget bibliotek uppfyllde alla dessa krav. De som undersöktes var:

- GNU Scientific library
- LAPACK
- ATLAS
- NAG

Då togs beslutet att ta fram ett eget bibliotek som döptes till matLib. De matrisoperationer som behövdes var:

- addition
- subtraktion
- multiplikation

- beräkna determinat
- beräkna invers
- lösa linjära ekvationssystem
- gausselimination
- transponering
- skalärmultiplikation

Efter implementation ska nu eventuella optimeringar undersökas.

### 3 Teori

#### 3.1 Tidskomplexitet på nuvarande implementationer operationer

Alla tidskomplexiteter beräknas på  $n \times n$  matriser.

Addition:  $\mathcal{O}(n^2)$

Subtraktion:  $\mathcal{O}(n^2)$

Multiplikation:  $\mathcal{O}(n^3)$

Invers (crout och sedan lösa  $n$  ekvationssystem):  $\mathcal{O}(n^3)$

#### 3.2 Eventuella förbättringar

Multiplikation:

Istället för den naive algoritmen kan strassen algoritmen implementeras vilket reducerar tidskomplexiteten från  $\mathcal{O}(n^3)$  till  $\mathcal{O}(n^{2.807})$  (källa).

Inverse:

Istället för crout så kan inversen beräknas med strassen algoritmen och på så sätt så sänks tidskomplexiteten från  $\mathcal{O}(n^3)$  till  $\mathcal{O}(n^{2.807})$ .

### 4 Metod

### 5 Resultat

### 6 Diskussion

#### 6.1 Resultat

#### 6.2 Metod

### 7 Slutsatser