1. INTRODUCCIÓ

Considerem un sistema on un transmissor genera un senyal d’entrada X, el qual pren valors d’una constel·lació de símbols denotada amb una distribució de probabilitat . El transmissor envia símbols amb un rate i un code length , per tant el número de missatges és de .

Ara queda un , on i normalitzant a .

Ara considerem que aquest senyal és enviat a través d’un canal de comunicació que introdueix distorsió en forma de additive noise. En aquest model el noise esta composat per un terme aleatori , que segueix una **complex normal distribution** amb la mitjana a zero i variància unitària. El canal transforma el senyal transmès , generant un senyal de sortida Y que es pot expressar com:



La probabilitat del canal de transmissió la definirem com:

I obtindrem d’output del canal com a missatge on després al receiver/decoder s’estimarà la paraula transmesa com .

Degut a les característiques del canal, ara podem mostrar que es pot transmetre amb una probabilitat d’error , on

Per simplificar, definim que , i per tant, a l’hora d’escriure la probabilitat del canal, aquest queda com , amb un canvi de variable . Ara substituïm a quedant l’equació:

També ho podem escriure com:

Veiem que la integral és respecte z que pertany als números complexos, per tant, podem dividir aquesta integral en dues, una per la part real, i l’altre per la imaginària:

La qual podem aproximar:

On son els pesos de quadratura i son les arrels/nodes de quadratura.

1. OPTIMITZACIÓ DE LA COMPUTACIÓ DE ) AMB L’ÚS DE MATRIUS

El càlcul de la funció ) és computacionalment costós a causa de l'estructura de la fórmula i la dependència de múltiples paràmetres en bucles niats. En implementar directament la fórmula pas a pas l'enfocament iteratiu utilitza múltiples nivells de bucles ‘for’ per a recórrer les quadratures de Gauss-Hermite i avaluar la funció de probabilitat per a cada combinació de variables.

L'optimització del càlcul de mitjançant operacions matricials és fonamental, ja que MATLAB és inherentment més eficient amb aquestes estructures a causa del seu disseny intern basat en llibreries optimitzades com BLAS i LAPACK.

Ara, desenvolupant en la aproximació feta en l’apartat anterior, tenim que:

Ara trobem que

on recordem que és la cardinalitat de la constel·lació. També ens trobem les següents matrius i tal que:

Q té una mida de , i tant com la tenen de .

També mencionar que l’operant es refereix a la multiplicació de Hadamard, mentre que l’operant es refereix a una multiplicació matricial.

1. PRIMERA I SEGONA DERIVADA

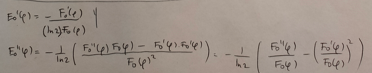
Explicar que hem fet funcions noves per calcular , on

Hem implementat F tant amb fors com amb matrius i hem aprofitat els càlculs d’aquestes per calcular les seva primera i segona derivada.

* Primera derivada:

Explicar que aquestes derivades només funcionen si N es gran, i.e N = 20 (això esta pendent de demostrar)

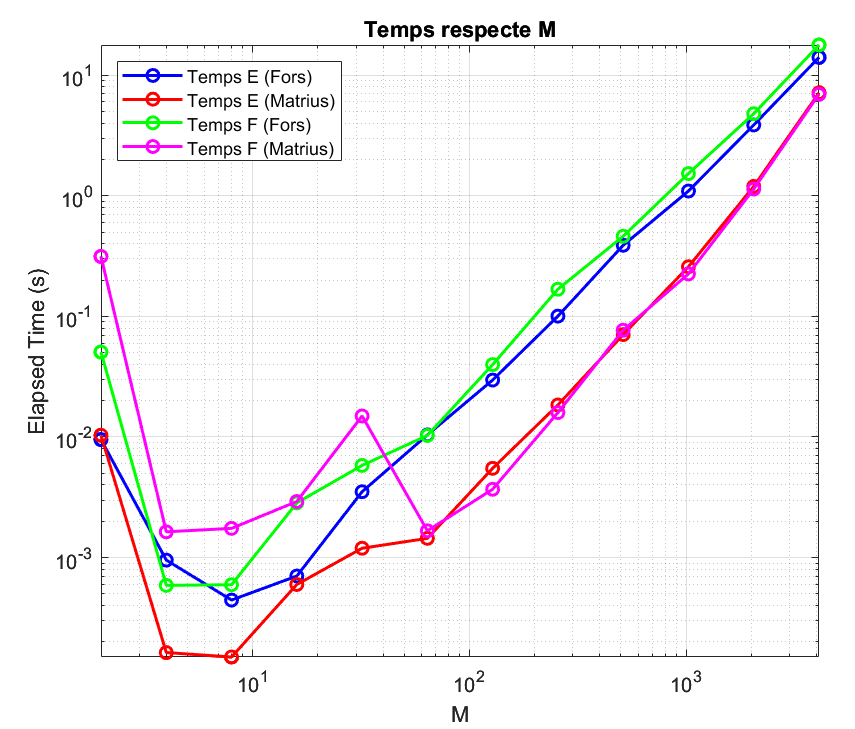
I posar finalment que les derivades de :



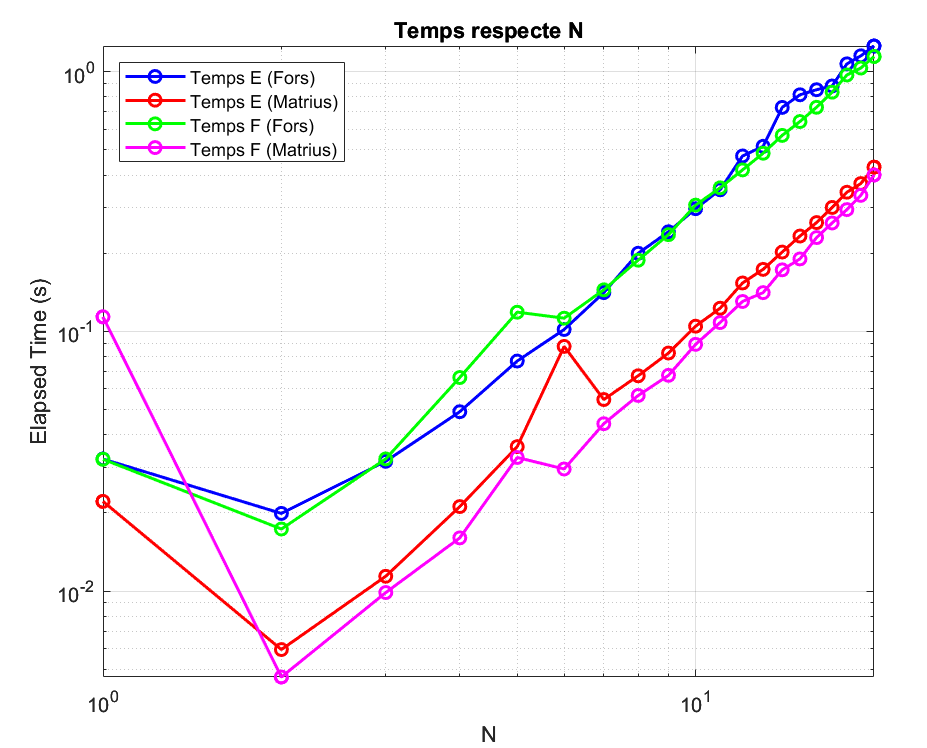
1. TEMPS D’EXECUCIÓ

Fer comparació dels temps d’execució i treure algunes conclusions.

Temps d’execució per ( + derivades) respecte el tamany de la modulació. Fins M = 4096 (plot logarítmic):



Temps execució respecte N (fins N = 20) i per una M = 128:



S’ha de tenir en compte que depèn del Hardware del ordinador, el temps d’execució pot variar degut a factors com el processador, la memòria RAM, etc. Encara així, el ratio entre els fors i matrius no deu variar gaire.

1. INTENT DE PARALEL·LITAZIÓ

He intentant paralel·litzar el codi de fors amb parfor (OpenMP de MATLAB). He provat en els diferents bucles que hi ha a la funció però el mes òptim de pararel·litzar és l’extern el qual al executar té un temps d’execució casi igual al de sense fors. Per tant no val la pena fer-ho.

1. MÈTODES D’OPTIMITZACIÓ

Hem implementat dos mètodes principals d’òptimització per trobar la rho que dona una A màxima:

* Gradient ascent
* Newton method

Estaria bé fer gràfiques dels temps d’execució (al excel n’he penjat) i doncs s’arriba a la conclusió de que Newton va molt més ràpid encara que a partir de 1024 (inclòs), em quedo out of memory.

Per tant estaria guay mirar altre mètodes que no requereixin tanta memòria.