“Per ogni ambiente di programmazione (MATLAB + open source diversa da Octave a vostra scelta) una breve descrizione della libreria usata mettendo in evidenza le sue caratteristiche e se `e documentata e mantenuta;”

Libreria EJML

Il primo progetto implementato per la risoluzione di sistema lineari associati ad una matrice sparsa, simmetrica e definita positiva è basato sulla libreria open source EJML. Efficient Java Matrix Library (EJML) è una libreria scritta interamente in Java e rilasciata con licenza Apache v2, è stata progettata per essere facile da usare e offre una sintassi chiara e concisa per la manipolazione di vettori e matrici reali, complesse, dense oppure sparse. Più formalmente fornisce un'ampia gamma di funzionalità matematiche, tra cui la risoluzione di sistemi lineari, il calcolo di autovalori e autovettori, la decomposizione QR, la fattorizzazione LU, la decomposizione di Cholesky, la fattorizzazione SVD.

Gli obbiettivi della libreria EJML sono:

* Essere il più efficiente possibile dal punto di vista computazionale e della memoria per matrici di piccole e grandi dimensioni
* Essere accessibile sia a principianti che agli esperti

Questi obbiettivi sono raggiunti grazie alla selezione dinamica dei migliori algoritmi da utilizzare in fase di esecuzione, a un’API pulita e interfacce multiple.

EJML ha tre interfacce distinte con cui poter interagire:

1) Procedural, fornisce pieno accesso a tutte le funzionalità e capacità di EJML e un controllo quasi completo sulla creazione della memoria, sulla velocità e su algoritmi specifici con un'API procedurale

2) SimpleMatrix, è l’interfaccia utilizzata all’interno del progetto, fornisce un sottoinsieme semplificato delle funzionalità principali tramite un'API orientata alla programmazione ad oggetti, ispirata a Jama.

3) Equations, è un'interfaccia simbolica, concepita per la manipolazione di matrici da parte degli utenti in modo simile a Matlab e ad altri CAS, fornendo un modo compatto per scrivere equazioni.

Di seguito sono riportate tutte le funzionalità per l’algebra lineare offerte dalla libreria EJML

Immagine che contiene testo, numero, schermata, Carattere

Descrizione generata automaticamente

Documentazione di EJML

Sul sito della libreria disponibile al link: <http://ejml.org/wiki/index.php?title=Manual> è presente un breve manuale. Esso si divide in 4 sezioni che forniscono una panoramica della libreria:

* The Basics (nozioni base)
* Tutorials
* Example Code (esempi di codice)
* External References (riferimenti esterni).

La sezione del manuale riguardante le nozioni base fornisce un'introduzione a EJML, compresa la lista delle operazioni standard e le funzionalità principali della libreria. Descrive inoltre come utilizzare e sviluppare un'applicazione utilizzando EJML fornendo in aggiunta una lista di domande frequenti con la possibilità di scrivere eventuali domande relative alla libreria sfruttando una bacheca.

La sezione Tutorials del manuale fornisce delle guide per vari problemi di algebra lineare, come ad esempio operare con matrici a valori complessi, risoluzione di sistemi lineari, decomposizioni di matrici ed esecuzione di unit test. Questi tutorial sono progettati per presentare diversi aspetti di EJML e aiutare gli utenti a capire come risolvere diversi problemi di algebra lineare utilizzando la libreria.

La sezione Example Code del manuale fornisce una tabella contenente vari esempi di codice per i più comuni problemi di algebra lineare risolvibili utilizzando la libreria EJML.

Nella sezione External References del manuale viene consigliato materiale aggiuntivo per tutti gli utenti che desiderano approfondire il funzionamento della libreria EJML per poter sviluppare tecniche avanzate di programmazione.

Infine, sempre sul sito internet della libreria, è presente una pagina dedicata in cui viene spiegato come accedere al codice sorgente di EJML sfruttando la pagina Github ufficiale <https://github.com/lessthanoptimal/ejml> e tutti i comandi necessari per importare ed utilizzare tutti i componenti della libreria tramite file JAR, utilizzando Gradle oppure Maven.

Libreria HEBI

//todo parlare della libreria hebi utilizzata per importare le matrici in formato .mat all’interno del programma

//todo listato del codice (poi da sistemare in latex)

//todo descrivere le classi utilizzate