

Oscillateur harmonique quantique 1D

1

Generated by Doxygen 1.9.1

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 psiSolution Class Reference	7
4.1.1 Member Function Documentation	7
4.1.1.1 calculeSolution()	7
4.1.1.2 derivee_seconde()	8
4.1.1.3 energyMat()	8
4.1.1.4 orthoMat()	9
4.2 TestHermiteSuite Class Reference	9
4.2.1 Member Function Documentation	10
4.2.1.1 testHermite1()	10
4.2.1.2 testHermite2()	10
4.3 TestProprietes Class Reference	11
4.3.1 Member Function Documentation	11
4.3.1.1 testOrthonormalite()	12
5 File Documentation	13
5.1 headers/constantes.h File Reference	13
5.1.1 Detailed Description	13
5.1.2 Variable Documentation	13
5.1.2.1 wi	14
5.1.2.2 zi	14
5.2 headers/hermite.h File Reference	14
5.2.1 Detailed Description	15
5.2.2 Function Documentation	15
5.2.2.1 hermiteMat()	16
5.3 src/hermite.cpp File Reference	16
5.3.1 Detailed Description	17
5.3.2 Function Documentation	17
5.3.2.1 hermiteMat()	17
Index	19

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

psiSolution	7
CxxTest::TestSuite	
TestHermiteSuite	9
TestProprietes	11

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

psiSolution	7
TestHermiteSuite	9
TestProprietes	11

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

headers/ constantes.h	
Ce fichier contient les constantes utiles pour le projet	13
headers/ dataExporter.h	??
headers/ hermite.h	
Interface qui donne toutes les fonctions nécessaires pour calculer des polynomes d'Hermite en des points donnés	14
headers/ psiSolution.h	??
src/ hermite.cpp	
Implémentation d' hermite.h	16
tests/ test_hermite.h	??
tests/ test_proprietes.h	??

Chapter 4

Class Documentation

4.1 psiSolution Class Reference

Public Member Functions

- **psiSolution** (int)
- arma::mat **calculeSolution** (const arma::vec &)
Calcule les psi-solutions.
- arma::mat **orthoMat** ()
Vérifie l'orthonormalité des psi-solutions.
- arma::vec **derivee_seconde** (const arma::vec &)
Calcule la dérivée seconde de la solution en des points du vecteur. On peut utiliser l'approximation discrète : https://fr.wikipedia.org/wiki/D%C3%A9riv%C3%A9e_seconde_discr%C3%A8te.
- arma::vec **energyMat** ()
Calcule les niveaux d'énergie pour les solutions psi de l'équation à l'aide de l'équation 1D.

Public Attributes

- int **n_max**
Jusqu'à quel niveau d'énergie max seront calculés les solutions de l'équation.

4.1.1 Member Function Documentation

4.1.1.1 calculeSolution()

```
arma::mat psiSolution::calculeSolution (
    const arma::vec & vecteurZ )
```

Calcule les psi-solutions.

retourne une matrice à 2 dimensions, contenant les valeurs de $\psi_i(z)$; La ième colonne correspond aux valeurs de ψ_i évalué à une valeur de z différente à chaque ligne; Les valeurs de z sont données par le vecteur [vecteurZ] placé en argument

Returns

arma::mat

Parameters

<code>vecteurZ</code>	est le vecteur de réels pour lesquels on possède les valeurs d'évaluation du polynome d'Hermite
-----------------------	---

Returns

`arma::mat` Retourne une matrice de la forme suivante : exemple `n_max=2`

$$\begin{bmatrix} \psi_0(z_0) & \psi_1(z_0) & \psi_2(z_0) \\ \psi_0(z_1) & \psi_1(z_1) & \psi_2(z_1) \\ \psi_0(z_2) & \psi_1(z_2) & \psi_2(z_2) \end{bmatrix}$$

4.1.1.2 `derivee_seconde()`

```
arma::vec psiSolution::derivee_seconde (
    const arma::vec & Z )
```

Calcule la dérivée seconde de la solution en des points du vecteur. On peut utiliser l'approximation discrète : https://fr.wikipedia.org/wiki/D%C3%A9riv%C3%A9e_seconde_discr%C3%A8te.

Returns

`arma::vec` Les résultats de la dérivée seconde en ces points.

Parameters

<code>Z</code>	Les points où seront calculés la dérivée seconde
----------------	--

Returns

`arma::vec` Les résultats de la dérivée seconde

4.1.1.3 `energyMat()`

```
arma::vec psiSolution::energyMat ( )
```

Calcule les niveaux d'énergie pour les solutions ψ de l'équation à l'aide de l'équation 1D.

Returns

`arma::vec` les niveaux d'énergies sous forme d'un vecteur colonne

$$Z = \begin{bmatrix} E_0 \\ E_1 \\ \dots \\ E_{n_{max}} \end{bmatrix}$$

4.1.1.4 orthoMat()

```
arma::mat psiSolution::orthoMat ( )
```

Vérifie l'orthonormalité des psi-solutions.

Verifie l'orthonormalité des psi-solutions. On vérifie pour tous les couples $n, m \leq n_max$. On retourne les résultats dans une matrice dont les numéros de lignes et les colonnes correspondent aux valeurs de n et m .

Returns

arma::mat une matrice avec les résultats des différents produits scalaires des solutions

arma::mat Retourne une matrice de la forme suivante : exemple $n_max=1$

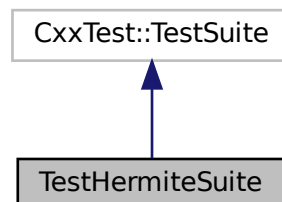
$$\begin{bmatrix} \int \psi_0^*(z) \psi_0(z) dz & \int \psi_0^*(z) \psi_1(z) dz \\ \int \psi_1^*(z) \psi_0(z) dz & \int \psi_1^*(z) \psi_1(z) dz \end{bmatrix}$$

The documentation for this class was generated from the following files:

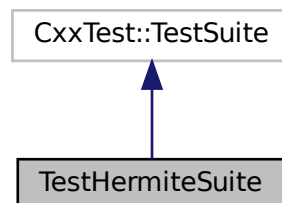
- headers/psiSolution.h
- src/psiSolution.cpp

4.2 TestHermiteSuite Class Reference

Inheritance diagram for TestHermiteSuite:



Collaboration diagram for TestHermiteSuite:



Public Member Functions

- void [testHermite1](#) (void)
Test du calcul de la matrice Hermite avec n_max=5 et.
- void [testHermite2](#) (void)
Test du calcul de la matrice Hermite avec n_max=1 et.

4.2.1 Member Function Documentation

4.2.1.1 testHermite1()

```
void TestHermiteSuite::testHermite1 (
    void ) [inline]
```

Test du calcul de la matrice Hermite avec n_max=5 et.

$$Z = \begin{bmatrix} 1 \\ 5 \end{bmatrix}$$

Ce calcul normalement donne le résultat suivant :

$$\begin{bmatrix} 1 & 2 & 2 & -4 & -20 \\ 1 & 10 & 98 & 940 & 8812 \end{bmatrix}$$

4.2.1.2 testHermite2()

```
void TestHermiteSuite::testHermite2 (
    void ) [inline]
```

Test du calcul de la matrice Hermite avec n_max=1 et.

$$Z = \begin{bmatrix} 1 \\ 5 \\ 2 \\ 8 \end{bmatrix}$$

Ce calcul normalement donne le résultat suivant :

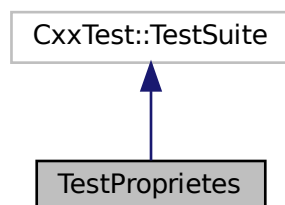
$$\begin{bmatrix} 1 & 2 \\ 1 & 10 \\ 1 & 4 \\ 1 & 16 \end{bmatrix}$$

The documentation for this class was generated from the following file:

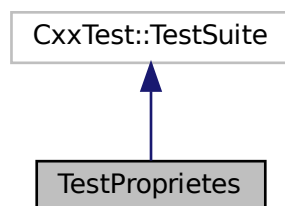
- tests/test_hermite.h

4.3 TestProprietes Class Reference

Inheritance diagram for TestProprietes:



Collaboration diagram for TestProprietes:



Public Member Functions

- void `testOrthonormalite` (void)
Test de l'orthonormalité des solutions de l'équation.
- void `testEnergies` (void)
Test des énergies déduits des solutions de l'équation.

4.3.1 Member Function Documentation

4.3.1.1 testOrthonormalite()

```
void TestProprietes::testOrthonormalite (  
    void ) [inline]
```

Test de l'orthonormalité des solutions de l'équation.

On génère la matrice des produits scalaires des différentes solutions et on compare avec la matrice identité.

The documentation for this class was generated from the following file:

- tests/test_proprietes.h

Chapter 5

File Documentation

5.1 headers/constantes.h File Reference

Ce fichier contient les constantes utiles pour le projet.

Variables

- double `pas_der` = 1e-7
Constante utilisée comme pas pour la dérivée seconde.
- double `hbar` = 1.0
Constante hbar.
- double `omega` = 1.0
Constante omega.
- double `m` = 1.0
Constante m.
- double `pi` = 3.141592653589793
Contante pi.
- int `degQuadrature` = 50
Degré utilisé pour la quadrature.
- arma::vec `zi`
Vecteur contenant les valeurs des z_i pour un degré de 50 Contient 50 valeurs $zi[i] = z_i$.
- arma::vec `wi`
Vecteur contenant la valeurs des w_i pour un degré de 50 Contient 50 valeurs $wi[i] = w_i$.

5.1.1 Detailed Description

Ce fichier contient les constantes utiles pour le projet.

5.1.2 Variable Documentation

5.1.2.1 wi

```
arma::vec wi
```

Initial value:

```
= {1.83379405e-37, 1.67380167e-32, 1.21524412e-28, 2.13765831e-25,
  1.41709360e-22, 4.47098437e-20, 7.74238296e-18, 8.09426189e-16,
  5.46594403e-14, 2.50665552e-12, 8.11187736e-11, 1.90904054e-09,
  3.34679340e-08, 4.45702997e-07, 4.58168271e-06, 3.68401905e-05,
  2.34269892e-04, 1.18901178e-03, 4.85326383e-03, 1.60319411e-02,
  4.30791592e-02, 9.45489355e-02, 1.70032456e-01, 2.51130856e-01,
  3.05085129e-01, 3.05085129e-01, 2.51130856e-01, 1.70032456e-01,
  9.45489355e-02, 4.30791592e-02, 1.60319411e-02, 4.85326383e-03,
  1.18901178e-03, 2.34269892e-04, 3.68401905e-05, 4.58168271e-06,
  4.45702997e-07, 3.34679340e-08, 1.90904054e-09, 8.11187736e-11,
  2.50665552e-12, 5.46594403e-14, 8.09426189e-16, 7.74238296e-18,
  4.47098437e-20, 1.41709360e-22, 2.13765831e-25, 1.21524412e-28,
  1.67380167e-32, 1.83379405e-37
}
```

Vecteur contenant la valeurs des w_i pour un degré de 50 Contient 50 valeurs $w[i] = w_i$.

Pour obtenir ces valeurs, on a utilisé la bibliothèque numpy de Python via la fonction : `numpy.polynomial.hermite.↔
hermgauss(50)`

5.1.2.2 zi

```
arma::vec zi
```

Initial value:

```
= {-9.18240696, -8.52277103, -7.97562237, -7.48640943, -7.03432351,
  -6.60864797, -6.20295252, -5.81299468, -5.43578609, -5.06911758,
  -4.71129367, -4.36097316, -4.01706817, -3.67867706, -3.34503831,
  -3.01549777, -2.6894847, -2.3664939, -2.04607197, -1.72780655,
  -1.41131775, -1.09625113, -0.78227173, -0.46905906, -0.15630255,
  0.15630255, 0.46905906, 0.78227173, 1.09625113, 1.41131775,
  1.72780655, 2.04607197, 2.3664939, 2.6894847, 3.01549777,
  3.34503831, 3.67867706, 4.01706817, 4.36097316, 4.71129367,
  5.06911758, 5.43578609, 5.81299468, 6.20295252, 6.60864797,
  7.03432351, 7.48640943, 7.97562237, 8.52277103, 9.18240696
}
```

Vecteur contenant les valeurs des z_i pour un degré de 50 Contient 50 valeurs $z[i] = z_i$.

Pour obtenir ces valeurs, on a utilisé la bibliothèque numpy de Python via la fonction : `numpy.polynomial.hermite.↔
hermgauss(50)`

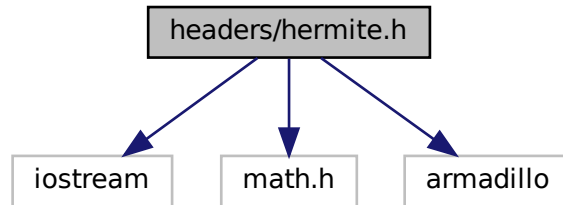
5.2 headers/hermite.h File Reference

Interface qui donne toutes les fonctions nécessaires pour calculer des polynomes d'Hermite en des points donnés.

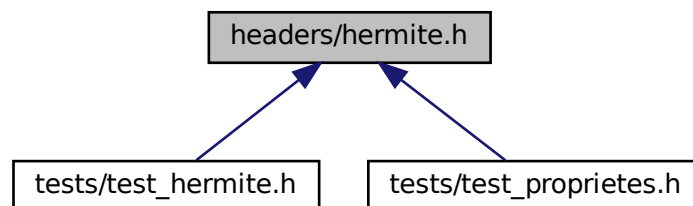
```
#include <iostream>
#include <math.h>
```

```
#include <armadillo>
```

Include dependency graph for hermite.h:



This graph shows which files directly or indirectly include this file:



Functions

- arma::mat [hermiteMat](#) (int, arma::vec)

Renvoie les résultats du polynome d'Hermite d'indice $i=0$ à un entier donné n_{max} appliqué à un vecteur colonne Z , sous forme d'une matrice.

5.2.1 Detailed Description

Interface qui donne toutes les fonctions nécessaires pour calculer des polynomes d'Hermite en des points donnés.

5.2.2 Function Documentation

5.2.2.1 hermiteMat()

```
arma::mat hermiteMat (
    int n_max,
    arma::vec Z )
```

Renvoie les résultats du polynome d'Hermite d'indice $i=0$ à un entier donné n_max appliqué à un vecteur colonne Z , sous forme d'une matrice.

Returns

arma::mat La matrice avec les résultats du polynome d'Hermite

Parameters

n_max	Indice max des polynomes d'Hermite utilisés dans la matrice
Z	Vecteur colonne contenant les points à appliquer aux polynomes d'Hermite

Returns

arma::mat Retourne une matrice de la forme suivante : exemple $n_{max} = 2$ et $Z = [z_0 \quad \dots \quad z_m]$

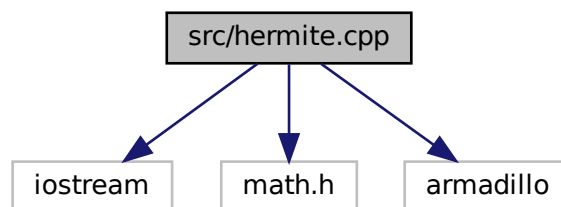
$$\begin{bmatrix} H_0(z_0) & \dots & H_{n_{max}}(z_0) \\ \vdots & \ddots & \vdots \\ H_0(z_m) & \dots & H_{n_{max}}(z_m) \end{bmatrix}$$

5.3 src/hermite.cpp File Reference

Implémentation d'[hermite.h](#).

```
#include <iostream>
#include <math.h>
#include <armadillo>
```

Include dependency graph for hermite.cpp:



Functions

- arma::mat [hermiteMat](#) (int n_max , arma::vec Z)

Renvoie les résultats du polynome d'Hermite d'indice $i=0$ à un entier donné n_max appliqué à un vecteur colonne Z , sous forme d'une matrice.

5.3.1 Detailed Description

Implémentation d'[hermite.h](#).

5.3.2 Function Documentation

5.3.2.1 hermiteMat()

```
arma::mat hermiteMat (
    int n_max,
    arma::vec Z )
```

Renvoie les résultats du polynome d'Hermite d'indice i=0 à un entier donné n_max appliqué à un vecteur colonne Z, sous forme d'une matrice.

Parameters

<i>n_max</i>	Indice max des polynomes d'Hermite utilisés dans la matrice
<i>Z</i>	Vecteur colonne contenant les points à appliquer aux polynomes d'Hermite

Returns

arma::mat Retourne une matrice de la forme suivante : exemple $n_{max} = 2$ et $Z = [z_0 \quad \dots \quad z_m]$

$$\begin{bmatrix} H_0(z_0) & \dots & H_{n_{max}}(z_0) \\ \vdots & \ddots & \vdots \\ H_0(z_m) & \dots & H_{n_{max}}(z_m) \end{bmatrix}$$

Index

- calculerSolution
 - psiSolution, [7](#)
- constantes.h
 - wi, [13](#)
 - zi, [14](#)
- derivee_seconde
 - psiSolution, [8](#)
- energyMat
 - psiSolution, [8](#)
- headers/constantes.h, [13](#)
- headers/hermite.h, [14](#)
- hermite.cpp
 - hermiteMat, [17](#)
- hermite.h
 - hermiteMat, [15](#)
- hermiteMat
 - hermite.cpp, [17](#)
 - hermite.h, [15](#)
- orthoMat
 - psiSolution, [8](#)
- psiSolution, [7](#)
 - calculerSolution, [7](#)
 - derivee_seconde, [8](#)
 - energyMat, [8](#)
 - orthoMat, [8](#)
- src/hermite.cpp, [16](#)
- testHermite1
 - TestHermiteSuite, [10](#)
- testHermite2
 - TestHermiteSuite, [10](#)
- TestHermiteSuite, [9](#)
 - testHermite1, [10](#)
 - testHermite2, [10](#)
- testOrthonormalite
 - TestProprietes, [11](#)
- TestProprietes, [11](#)
 - testOrthonormalite, [11](#)
- wi
 - constantes.h, [13](#)
- zi
 - constantes.h, [14](#)