

## Mètode de Runge-Kutta-Verner

Generat per Doxygen 1.8.17

<b>1 Índex d'Espais de Noms</b>	<b>1</b>
1.1 Llista dels Espais de Noms	1
<b>2 Índex Jeràrquic</b>	<b>3</b>
2.1 Jerarquia de Classes	3
<b>3 Índex de Classes</b>	<b>5</b>
3.1 Llista de Classes	5
<b>4 Documentació de l'Espai de Noms</b>	<b>7</b>
4.1 Referència de l'Espai de Noms rkv78	7
4.1.1 Descripció Detallada	7
4.2 Referència de l'Espai de Noms rkv78_serie	7
4.2.1 Descripció Detallada	7
4.3 Referència de l'Espai de Noms utils	7
4.3.1 Descripció Detallada	8
4.3.2 Documentació de les Variables	8
4.3.2.1 desc	8
<b>5 Documentació de les Classes</b>	<b>9</b>
5.1 Referència de la Classe Circle	9
5.1.1 Descripció Detallada	9
5.1.2 Documentació de les Funcions Membre	9
5.1.2.1 edo()	9
5.2 Referència de la Classe Circle_Serie	10
5.2.1 Descripció Detallada	10
5.2.2 Documentació de les Funcions Membre	10
5.2.2.1 edo()	10
5.3 Referència de la Classe differential_equation	11
5.3.1 Descripció Detallada	11
5.4 Referència de la Classe differential_equation_serie	11
5.4.1 Descripció Detallada	12
5.5 Referència de la Classe Disturbed_Pendulum	12
5.5.1 Descripció Detallada	12
5.5.2 Documentació de les Funcions Membre	12
5.5.2.1 edo()	12
5.6 Referència de la Classe Disturbed_Pendulum_Matrix	13
5.6.1 Descripció Detallada	13
5.6.2 Documentació de les Funcions Membre	13
5.6.2.1 edo()	13
5.7 Referència de la Classe Disturbed_Pendulum_Serie	14
5.7.1 Descripció Detallada	14
5.7.2 Documentació de les Funcions Membre	14
5.7.2.1 edo()	14

5.8 Referència de la Classe rkv78::rkv78	15
5.8.1 Documentació de les Funcions Membre	15
5.8.1.1 rkv78_main()	15
5.9 Referència de la Classe rkv78_serie::rkv78_serie	16
5.9.1 Documentació de les Funcions Membre	16
5.9.1.1 rkv78_main()	16
5.10 Referència de la Classe Serie	17
5.10.1 Descripció Detallada	18
5.10.2 Documentació del Constructor i el Destructor	18
5.10.2.1 Serie() [1/3]	18
5.10.2.2 Serie() [2/3]	18
5.10.2.3 Serie() [3/3]	18
5.10.2.4 ~Serie()	18
5.10.3 Documentació de les Funcions Membre	19
5.10.3.1 addCoeff()	19
5.10.3.2 calculate()	19
5.10.3.3 cos()	19
5.10.3.4 derivate()	20
5.10.3.5 exp()	20
5.10.3.6 getCoeff() [1/2]	20
5.10.3.7 getCoeff() [2/2]	20
5.10.3.8 getDegree()	21
5.10.3.9 log()	21
5.10.3.10 operator!=( ) [1/2]	21
5.10.3.11 operator!=( ) [2/2]	22
5.10.3.12 operator*( ) [1/2]	22
5.10.3.13 operator*( ) [2/2]	22
5.10.3.14 operator*=( ) [1/2]	23
5.10.3.15 operator*=( ) [2/2]	23
5.10.3.16 operator+( ) [1/3]	23
5.10.3.17 operator+( ) [2/3]	24
5.10.3.18 operator+( ) [3/3]	24
5.10.3.19 operator+=( ) [1/2]	24
5.10.3.20 operator+=( ) [2/2]	25
5.10.3.21 operator-( ) [1/3]	25
5.10.3.22 operator-( ) [2/3]	25
5.10.3.23 operator-( ) [3/3]	26
5.10.3.24 operator-=( ) [1/2]	26
5.10.3.25 operator-=( ) [2/2]	26
5.10.3.26 operator/( ) [1/2]	28
5.10.3.27 operator/( ) [2/2]	28
5.10.3.28 operator/=( ) [1/2]	28

5.10.3.29 operator/=( ) [2/2]	30
5.10.3.30 operator=( ) [1/2]	30
5.10.3.31 operator=( ) [2/2]	30
5.10.3.32 operator==( ) [1/2]	32
5.10.3.33 operator==( ) [2/2]	32
5.10.3.34 pow()	32
5.10.3.35 setCoeff()	34
5.10.3.36 sin()	34
5.10.3.37 sqrt()	34
5.11 Referència de la Classe utils::UtilFunctions	35
5.11.1 Documentació de les Funcions Membre	35
5.11.1.1 buscar_punt_fix()	35
5.11.1.2 buscar_vap_vep()	36
5.11.1.3 buscar_varietat()	37
5.11.1.4 evaluate_negative_time()	38
5.11.1.5 evaluate_positive_time()	38
<b>Índex alfabètic</b>	<b>41</b>

# Capítol 1

## Índex d'Espais de Noms

### 1.1 Llista dels Espais de Noms

Aquests són tots els espais de noms documentats amb breus descripcions:

<a href="#">rkv78</a>	7
<a href="#">rkv78_serie</a>	7
<a href="#">utils</a>	7

## Capítol 2

# Índex Jeràrquic

### 2.1 Jerarquia de Classes

Aquesta llista d'herència està ordenada toscament, però no completa, de forma alfabètica:

differential_equation . . . . .	11
Circle . . . . .	9
Disturbed_Pendulum . . . . .	12
Disturbed_Pendulum_Matrix . . . . .	13
differential_equation_serie . . . . .	11
Circle_Serie . . . . .	10
Disturbed_Pendulum_Serie . . . . .	14
rkv78::rkv78 . . . . .	15
rkv78_serie::rkv78_serie . . . . .	16
Serie . . . . .	17
utils::UtilFunctions . . . . .	35

## Capítol 3

# Índex de Classes

### 3.1 Llista de Classes

Aquestes són les classes, estructures, unions i interfícies acompanyades amb breus descripcions:

Circle	9
Circle_Serie	10
differential_equation	11
differential_equation_serie	11
Disturbed_Pendulum	12
Disturbed_Pendulum_Matrix	13
Disturbed_Pendulum_Serie	14
rkv78::rkv78	15
rkv78_serie::rkv78_serie	16
Serie	17
utils::UtilFunctions	35

## Capítol 4

# Documentació de l'Espai de Noms

### 4.1 Referència de l'Espai de Noms rkv78

#### Classes

- class [rkv78](#)

#### 4.1.1 Descripció Detallada

Package que conté les funcions necessàries per a realitzar el mètode Runge-Kutta 7(8)

### 4.2 Referència de l'Espai de Noms rkv78\_serie

#### Classes

- class [rkv78\\_serie](#)

#### 4.2.1 Descripció Detallada

Package que conté les funcions necessàries per a realitzar el mètode Runge-Kutta 7(8) treballant amb objectes sèrie

### 4.3 Referència de l'Espai de Noms utils

#### Classes

- class [UtilFunctions](#)



## Variables

- double `desc`
- double `module`
- double `l1`
- double `l2`
- double `v1` [2]
- double `v2` [2]
- double `result` [2]
- double `initial` [2]
- double `z1`
- double `z2`
- double `initial_time`
- double `ah_copy`

### 4.3.1 Descripció Detallada

Paquet que conté funcions útils predeterminades.

### 4.3.2 Documentació de les Variables

#### 4.3.2.1 `desc`

```
double utils::desc
```

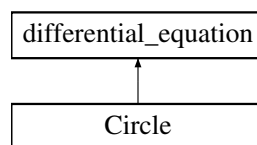
variables auxiliars per al càlcul de les funcions, compartides entre elles.

## Capítol 5

# Documentació de les Classes

### 5.1 Referència de la Classe Circle

Diagrama d'Herència per a Circle:



#### Mètodes Públics Estàtics

- static void `edo` (double t, double x[], int n, double result[])

#### 5.1.1 Descripció Detallada

Classe cercle que hereta de la classe abstracta `differential_equation`. Equació diferencial que defineix un cercle a  $R^2$

#### 5.1.2 Documentació de les Funcions Membre

##### 5.1.2.1 `edo()`

```
static void Circle::edo (  
    double t,  
    double x[],  
    int n,  
    double result[] ) [inline], [static]
```

Equació diferencial.

## Paràmetres

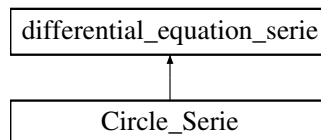
<i>t</i>	temps d'avaluació
<i>x</i>	punt d'avaluació
<i>n</i>	dimensió del punt
<i>result</i>	diferencial resultat

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- `clion_project/src/differential_equation.cpp`

## 5.2 Referència de la Classe Circle\_Serie

Diagrama d'Herència per a Circle\_Serie:



### Mètodes Públics Estàtics

- static void `edo` (double t, `Serie` x[], int n, `Serie` \*\*result)

#### 5.2.1 Descripció Detallada

Classe cercle que hereta de la classe abstracta `differential_equation_serie`. Equació diferencial que defineix un cercle a R2 treballant amb objectes sèries

#### 5.2.2 Documentació de les Funcions Membre

##### 5.2.2.1 edo()

```

static void Circle_Serie::edo (
    double t,
    Serie x[],
    int n,
    Serie ** result ) [inline], [static]
  
```

Equació diferencial.

## Paràmetres

<i>t</i>	temps d'avaluació
<i>x</i>	polinomi d'avaluació
<i>n</i>	dimensió del punt
<i>result</i>	polinomi resultat

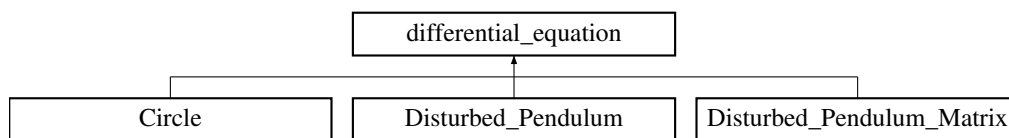
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- clion\_project/src/differential\_equation.cpp

## 5.3 Referència de la Classe differential\_equation

```
#include <differential_equation.h>
```

Diagrama d'Herència per a differential\_equation:



### Mètodes Públics Estàtics

- static void **edo** (double t, double x[], int n, double result[])

#### 5.3.1 Descripció Detallada

Classe abstracta que defineix classes d'equacions diferencials.

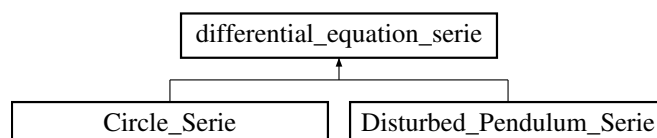
La documentació d'aquesta classe es va generar a partir del següent fitxer:

- clion\_project/src/differential\_equation.h

## 5.4 Referència de la Classe differential\_equation\_serie

```
#include <differential_equation.h>
```

Diagrama d'Herència per a differential\_equation\_serie:



## Mètodes Públics Estàtics

- static void **edo** (double t, [Serie](#) \*\*x, int n, [Serie](#) \*\*result)

### 5.4.1 Descripció Detallada

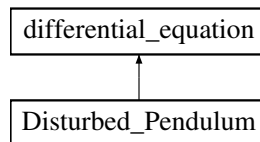
Classe abstracta que defineix classes d'equacions diferencials per a treballar amb sèries.

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- `clion_project/src/differential_equation.h`

## 5.5 Referència de la Classe Disturbed\_Pendulum

Diagrama d'Herència per a Disturbed\_Pendulum:



## Mètodes Públics Estàtics

- static void **edo** (double t, double x[], int n, double result[])

### 5.5.1 Descripció Detallada

Classe pèndol pertorbat que hereta de la classe abstracta [differential\\_equation](#). Equació diferencial que defineix un pèndol pertorbat per un camp magnètic constant

### 5.5.2 Documentació de les Funcions Membre

#### 5.5.2.1 edo()

```

static void Disturbed_Pendulum::edo (
    double t,
    double x[],
    int n,
    double result[] ) [inline], [static]
  
```

Equació diferencial.

## Paràmetres

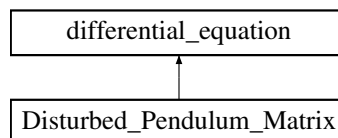
<i>t</i>	temps d'avaluació
<i>x</i>	punt d'avaluació
<i>n</i>	dimensió del punt
<i>result</i>	diferencial resultat

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- `clion_project/src/differential_equation.cpp`

## 5.6 Referència de la Classe Disturbed\_Pendulum\_Matrix

Diagrama d'Herència per a Disturbed\_Pendulum\_Matrix:



### Mètodes Públics Estàtics

- static void `edo` (double t, double x[], int n, double result[])

#### 5.6.1 Descripció Detallada

Classe matriu pèndol pertorbat que hereta de la classe abstracta `differential_equation`. Equació diferencial i matriu diferencial que defineix un pèndol pertorbat per un camp magnètic constant

#### 5.6.2 Documentació de les Funcions Membre

##### 5.6.2.1 `edo()`

```

static void Disturbed_Pendulum_Matrix::edo (
    double t,
    double x[],
    int n,
    double result[] ) [inline], [static]
  
```

Equació diferencial.

## Paràmetres

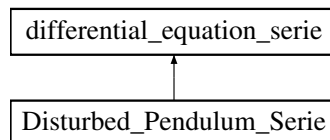
<i>t</i>	temps d'avaluació
<i>x</i>	punt d'avaluació
<i>n</i>	dimensió del punt
<i>result</i>	diferencial resultat

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- `clion_project/src/differential_equation.cpp`

## 5.7 Referència de la Classe Disturbed\_Pendulum\_Serie

Diagrama d'Herència per a Disturbed\_Pendulum\_Serie:



### Mètodes Públics Estàtics

- static void `edo` (double `t`, `Serie` \*\*`x`, int `n`, `Serie` \*\*`result`)

#### 5.7.1 Descripció Detallada

Classe pèndol pertorbat que hereta de la classe abstracta `differential_equation_serie` Equació diferencial que defineix un pèndol pertorbat per un camp magnètic constant treballant amb objectes sèrie

#### 5.7.2 Documentació de les Funcions Membre

##### 5.7.2.1 `edo()`

```

static void Disturbed_Pendulum_Serie::edo (
    double t,
    Serie ** x,
    int n,
    Serie ** result ) [inline], [static]
  
```

Equació diferencial.

## Paràmetres

<i>t</i>	temps d'avaluació
<i>x</i>	polinomi d'avaluació
<i>n</i>	dimensió del punt
<i>result</i>	polinomi resultat

La documentació d'aquesta classe es va generar a partir del següent fitxer:

- `clion_project/src/differential_equation.cpp`

## 5.8 Referència de la Classe rk78::rk78

### Mètodes Públics Estàtics

- static int [rk78\\_main](#) (double \*at, double \*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*edo)(double, double \*, int, double \*))

### 5.8.1 Documentació de les Funcions Membre

#### 5.8.1.1 rk78\_main()

```
int rk78::rk78::rk78_main (
    double * at,
    double * x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, double *, int, double *) edo ) [static]
```

Càlcul d'un pas d'integració.

## Paràmetres

<i>at</i>	temps actual d'avaluació
<i>x</i>	punt d'avaluació
<i>n</i>	dimensió del punt
<i>ah</i>	següent pas temporal
<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>edo</i>	punter a l'equació a avaluar



**Retorna**

flag segons si hem acabat (1) o no (0)

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- clion\_project/src/rkv78.h
- clion\_project/src/rkv78.cpp

## 5.9 Referència de la Classe rkv78\_serie::rkv78\_serie

### Mètodes Públics Estàtics

- static int `rkv78_main` (double \*at, `Serie` \*\*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*edo)(double, `Serie` \*\*, int, `Serie` \*\*))

### 5.9.1 Documentació de les Funcions Membre

#### 5.9.1.1 rkv78\_main()

```
int rkv78_serie::rkv78_serie::rkv78_main (
    double * at,
    Serie ** x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, Serie **, int, Serie **) ode ) [static]
```

Càlcul d'un pas d'integració.

#### Paràmetres

<i>at</i>	temps actual d'avaluació
<i>x</i>	polinomi d'avaluació
<i>n</i>	dimensió del punt
<i>ah</i>	següent pas temporal
<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>edo</i>	punter a l'equació a avaluar

**Retorna**

flag segons si hem acabat (1) o no (0)

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- `clion_project/src/rkv78_serie.h`
- `clion_project/src/rkv78_serie.cpp`

## 5.10 Referència de la Classe Serie

```
#include <Serie.h>
```

**Mètodes públics**

- `Serie` (int degree)
- `Serie` ()
- `Serie` (double \*in\_coeff, int degree)
- `~Serie` ()
- int `getDegree` () const
- double \* `getCoeff` () const
- void `setCoeff` (double value, int position)
- void `addCoeff` (double value, int position)
- double `getCoeff` (int position) const
- `Serie & operator=` (const `Serie` serie)
- `Serie & operator=` (const `Serie` \*serie)
- `Serie operator+` (const `Serie` serie)
- `Serie operator+` (const `Serie` \*serie)
- `Serie operator-` (const `Serie` serie)
- `Serie operator*` (const `Serie` serie)
- `Serie operator/` (const `Serie` serie)
- `Serie operator+` (double value)
- `Serie operator-` (double value)
- `Serie operator-` ()
- `Serie operator*` (double value)
- `Serie operator/` (double value)
- `Serie operator+=` (const `Serie` serie)
- `Serie operator-=` (const `Serie` serie)
- `Serie operator*=` (const `Serie` serie)
- `Serie operator/=` (const `Serie` serie)
- `Serie operator+=` (double value)
- `Serie operator-=` (double value)
- `Serie operator*=` (double value)
- `Serie operator/=` (double value)
- bool `operator==` (`Serie` serie)
- bool `operator!=` (`Serie` serie)
- bool `operator==` (double value)
- bool `operator!=` (double value)
- `Serie derivate` ()
- `Serie sqrt` ()
- `Serie log` ()
- `Serie exp` ()
- `Serie pow` (double a)
- `Serie sin` ()
- `Serie cos` ()
- double `calculate` (double s)
- void `empty_coef` ()

### 5.10.1 Descripció Detallada

Classe per a operar amb sèries. Aquesta classe permet crear polinomis amb un grau determinat i operar amb ells

### 5.10.2 Documentació del Constructor i el Destructor

#### 5.10.2.1 Serie() [1/3]

```
Serie::Serie (
    int degree ) [inline]
```

Constructor de Sèrie. Crea un polinomi de grau degree amb coeficients zero

##### Paràmetres

<i>degree</i>	grau del polinomi a crear
---------------	---------------------------

#### 5.10.2.2 Serie() [2/3]

```
Serie::Serie ( ) [inline]
```

Constructor de Sèrie. Crea un polinomi de grau DEGREE (constant global triada a main.cpp) amb coeficients zero

#### 5.10.2.3 Serie() [3/3]

```
Serie::Serie (
    double * in_coeff,
    int degree ) [inline]
```

Constructor de Sèrie. Crea un polinomi de grau degree amb coeficients segons in\_coeff

##### Paràmetres

<i>in_coeff</i>	valors dels coeficients
<i>degree</i>	grau del polinomi a crear

#### 5.10.2.4 ~Serie()

```
Serie::~Serie ( ) [inline]
```

Destructor de Sèrie.

### 5.10.3 Documentació de les Funcions Membre

#### 5.10.3.1 addCoeff()

```
void Serie::addCoeff (
    double value,
    int position ) [inline]
```

Suma un valor al coeficient de grau position.

Paràmetres

<i>value</i>	valor a sumar
<i>position</i>	grau del valor a sumar

#### 5.10.3.2 calculate()

```
double Serie::calculate (
    double s ) [inline]
```

Avaluació del polinomi en un punt

Paràmetres

<i>s</i>	punt a avaluar
----------	----------------

Retorna

valor d'avaluació

#### 5.10.3.3 cos()

```
Serie Serie::cos ( ) [inline]
```

Funció de cosinus d'un polinomi

Retorna

cosinus del polinomi

#### 5.10.3.4 `derivate()`

```
Serie Serie::derivate ( ) [inline]
```

Funció de derivació de polinomi

**Retorna**

polinomi derivat

#### 5.10.3.5 `exp()`

```
Serie Serie::exp ( ) [inline]
```

Funció d'exponenciar un polinomi

**Retorna**

exponencial del polinomi

#### 5.10.3.6 `getCoeff()` [1/2]

```
double * Serie::getCoeff ( ) const [inline]
```

Getter dels coeficients.

**Retorna**

punter a l'array de coeficients

#### 5.10.3.7 `getCoeff()` [2/2]

```
double Serie::getCoeff (
    int position ) const [inline]
```

Getter del coeficient en posició position.

**Paràmetres**

<i>position</i>	grau del coeficient a retornar
-----------------	--------------------------------

**Retorna**

coeficient

**5.10.3.8 getDegree()**

```
int Serie::getDegree ( ) const [inline]
```

Getter del grau del polinomi.

**Retorna**

grau del polinomi

**5.10.3.9 log()**

```
Serie Serie::log ( ) [inline]
```

Funció de logaritme d'un polinomi

**Retorna**

logaritme del polinomi

**5.10.3.10 operator!=( ) [1/2]**

```
bool Serie::operator!= (
    double value ) [inline]
```

Sobrecàrrega de l'operador !=

**Paràmetres**

<i>value</i>	número a comparar
--------------	-------------------

**Retorna**

booleà si el polinomi i el valor són diferents

### 5.10.3.11 operator!=() [2/2]

```
bool Serie::operator!= (
    Serie serie ) [inline]
```

Sobrecàrrega de l'operador !=

#### Paràmetres

<i>serie</i>	sèrie a comparar
--------------	------------------

#### Retorna

booleà si els dos polinomis són diferents

### 5.10.3.12 operator\*() [1/2]

```
Serie Serie::operator* (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador \*

#### Paràmetres

<i>serie</i>	<b>Serie</b> a multiplicar
--------------	----------------------------

#### Retorna

polinomi multiplicat

### 5.10.3.13 operator\*() [2/2]

```
Serie Serie::operator* (
    double value ) [inline]
```

Sobrecàrrega de l'operador \*

#### Paràmetres

<i>value</i>	Número a multiplicar
--------------	----------------------

#### Retorna

polinomi multiplicat

#### 5.10.3.14 operator\*=( ) [1/2]

```
Serie Serie::operator*= (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador \*=

Paràmetres

<i>serie</i>	sèrie a multiplicar
--------------	---------------------

Retorna

this

#### 5.10.3.15 operator\*=( ) [2/2]

```
Serie Serie::operator*= (
    double value ) [inline]
```

Sobrecàrrega de l'operador \*=

Paràmetres

<i>value</i>	número a multiplicar
--------------	----------------------

Retorna

this

#### 5.10.3.16 operator+( ) [1/3]

```
Serie Serie::operator+ (
    const Serie * serie ) [inline]
```

Sobrecàrrega de l'operador +

Paràmetres

<i>serie</i>	punter al polinomi a sumar
--------------	----------------------------



Retorna

polinomi sumat

#### 5.10.3.17 operator+() [2/3]

```
Serie Serie::operator+ (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador +

Paràmetres

<i>serie</i>	polinomi a sumar amb this
--------------	---------------------------

Retorna

polinomi sumat

#### 5.10.3.18 operator+() [3/3]

```
Serie Serie::operator+ (
    double value ) [inline]
```

Sobrecàrrega de l'operador +

Paràmetres

<i>value</i>	número a sumar
--------------	----------------

Retorna

polinomi sumat

#### 5.10.3.19 operator+=() [1/2]

```
Serie Serie::operator+= (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador +=

## Paràmetres

<i>serie</i>	serie a sumar
--------------	---------------

## Retorna

this

**5.10.3.20 operator+=()** [2/2]

```
Serie Serie::operator+= (
    double value ) [inline]
```

Sobrecàrrega de l'operador +=

## Paràmetres

<i>value</i>	número a restar
--------------	-----------------

## Retorna

this

**5.10.3.21 operator-()** [1/3]

```
Serie Serie::operator- ( ) [inline]
```

Sobrecàrrega de l'operador -

## Retorna

polinomi canviat de signe

**5.10.3.22 operator-()** [2/3]

```
Serie Serie::operator- (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador -

**Paràmetres**

<i>serie</i>	<a href="#">Serie</a> a restar
--------------	--------------------------------

**Retorna**

polinomi restat

**5.10.3.23 operator-() [3/3]**

```
Serie Serie::operator- (
    double value ) [inline]
```

Sobrecàrrega de l'operador -

**Paràmetres**

<i>value</i>	Número a restar
--------------	-----------------

**Retorna**

polinomi restat

**5.10.3.24 operator-=() [1/2]**

```
Serie Serie::operator-= (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador -=

**Paràmetres**

<i>serie</i>	sèrie a restar
--------------	----------------

**Retorna**

this

**5.10.3.25 operator-=() [2/2]**

```
Serie Serie::operator-= (
    double value ) [inline]
```

Sobrecàrrega de l'operador -=

**Paràmetres**

<i>value</i>	número a restar
--------------	-----------------

**Retorna**

this

**5.10.3.26 operator/() [1/2]**

```
Serie Serie::operator/ (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador /

**Paràmetres**

<i>serie</i>	Sèrie a dividir
--------------	-----------------

**Retorna**

polinomi dividit

**5.10.3.27 operator/() [2/2]**

```
Serie Serie::operator/ (
    double value ) [inline]
```

Sobrecàrrega de l'operador /

**Paràmetres**

<i>value</i>	Número a dividir
--------------	------------------

**Retorna**

polinomi dividit

**5.10.3.28 operator/=( ) [1/2]**

```
Serie Serie::operator/=(
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador -=

**Paràmetres**

<i>serie</i>	sèrie a restar
--------------	----------------

**Retorna**

this

**5.10.3.29 operator/=() [2/2]**

```
Serie Serie::operator/= (
    double value ) [inline]
```

Sobrecàrrega de l'operador /=

**Paràmetres**

<i>value</i>	número a restar
--------------	-----------------

**Retorna**

this

**5.10.3.30 operator=() [1/2]**

```
Serie & Serie::operator= (
    const Serie * serie ) [inline]
```

Sobrecàrrega de l'operador =

**Paràmetres**

<i>serie</i>	punter a la sèrie a posar a this
--------------	----------------------------------

**Retorna**

this

**5.10.3.31 operator=() [2/2]**

```
Serie & Serie::operator= (
    const Serie serie ) [inline]
```

Sobrecàrrega de l'operador =



**Paràmetres**

<i>serie</i>	serie a posar a this
--------------	----------------------

**Retorna**

this

**5.10.3.32 operator==( ) [1/2]**

```
bool Serie::operator==(
    double value ) [inline]
```

Sobrecàrrega de l'operador ==

**Paràmetres**

<i>value</i>	número a comparar
--------------	-------------------

**Retorna**

booleà si el polinomi i el valor són iguals

**5.10.3.33 operator==( ) [2/2]**

```
bool Serie::operator==(
    Serie serie ) [inline]
```

Sobrecàrrega de l'operador ==

**Paràmetres**

<i>serie</i>	sèrie a comparar
--------------	------------------

**Retorna**

booleà si els dos polinomis són iguals

**5.10.3.34 pow()**

```
Serie Serie::pow (
    double a ) [inline]
```

Funció d'eleva un polinomi

**Paràmetres**

<i>a</i>	número al que elevem el polinomi
----------	----------------------------------

**Retorna**

elevació del polinomi

**5.10.3.35 setCoeff()**

```
void Serie::setCoeff (
    double value,
    int position ) [inline]
```

Setter del coeficient en posició position.

**Paràmetres**

<i>value</i>	valor a settejar
<i>position</i>	grau del coeficient a canviar

**5.10.3.36 sin()**

```
Serie Serie::sin ( ) [inline]
```

Funció de sinus d'un polinomi

**Retorna**

sinus del polinomi

**5.10.3.37 sqrt()**

```
Serie Serie::sqrt ( ) [inline]
```

Funció d'arrel d'un polinomi

**Retorna**

arrel del polinomi

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- clion\_project/src/Serie.h
- clion\_project/src/Serie.cpp

## 5.11 Referència de la Classe `utils::UtilFunctions`

### Mètodes Públics Estàtics

- static void `buscar_punt_fix` (double \*at, double \*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*dif)(double, double \*, int, double \*))
- static void `evaluate_positive_time` (double \*at, double \*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*dif)(double, double \*, int, double \*))
- static void `evaluate_negative_time` (double \*at, double \*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*dif)(double, double \*, int, double \*)) static void `buscar_vap_vep`(double \*at
- static void `buscar_vap_vep` (double \*at, double \*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*dif)(double, double \*, int, double \*))
- static void `buscar_varietat` (double \*at, double \*x, int n, double \*ah, int sc, double tol, double \*atf, double \*aer, void(\*dif)(double, double \*, int, double \*), int deg, int vap\_a\_usar, void(\*dif\_serie)(double, `Serie` \*\*, int, `Serie` \*\*))

### Atributs Públics

- static void double \* **x**
- static void double int **n**
- static void double int double \* **ah**
- static void double int double int **sc**
- static void double int double int double **tol**
- static void double int double int double double \* **atf**
- static void double int double int double double double \* **aer**
- static void double int double int double double double void(\* **dif** )(double, double \*, int, double \*)
- static void double int double int double double double void(\*) double **vap1** )
- static void double int double int double double double void(\*) double double **vap2** )
- static void double int double int double double double void(\*) double double doubl **vep1** )[]
- static void double int double int double double double void(\*) double double double doubl **vep2** )[]

### 5.11.1 Documentació de les Funcions Membre

#### 5.11.1.1 `buscar_punt_fix()`

```
void utils::UtilFunctions::buscar_punt_fix (
    double * at,
    double * x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, double *, int, double *) dif ) [static]
```

Funció que a partir d'una bona aproximació troba un punt fix amb el mètode de Newton

## Paràmetres

<i>at</i>	temps inicial
<i>x</i>	punt inicial
<i>n</i>	dimensió punt
<i>ah</i>	pas d'integració
<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>dif</i>	punter a l'equació a avaluar

## 5.11.1.2 buscar\_vap\_vep()

```
void utils::UtilFunctions::buscar_vap_vep (
    double * at,
    double * x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, double *, int, double *) dif ) [static]
```

Funció que a partir d'un punt fix troba els valors i vectors propis usant el mètode de Cramer. Aquest algoritme troba els valors propis només si són reals!

## Paràmetres

<i>at</i>	temps inicial
<i>x</i>	punt inicial
<i>n</i>	dimensió punt
<i>ah</i>	pas d'integració
<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>dif</i>	punter a l'equació a avaluar

Funció que a partir d'un punt fix troba els valors i vectors propis usant el mètode de Cramer

## Paràmetres

<i>at</i>	temps inicial
<i>x</i>	punt inicial
<i>n</i>	dimensió punt
<i>ah</i>	pas d'integració

## Paràmetres

<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>dif</i>	punter a l'equació a avaluar
<i>vap1</i>	punter al que s'introdueix el vap1
<i>vap2</i>	punter al que s'introdueix el vap2
<i>vep1</i>	punter al que s'introdueix el vep1
<i>vep2</i>	punter al que s'introdueix el vep2

5.11.1.3 `buscar_varietat()`

```
void utils::UtilFunctions::buscar_varietat (
    double * at,
    double * x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, double *, int, double *) dif,
    int deg,
    int vap_a_usar,
    void(*) (double, Serie **, int, Serie **) dif_serie ) [static]
```

Funció que busca la parametrització de les varietats estable o inestable d'un punt fix

## Paràmetres

<i>at</i>	temps inicial
<i>x</i>	punt inicial
<i>n</i>	dimensió punt
<i>ah</i>	pas d'integració
<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>dif</i>	punter a l'equació a avaluar
<i>deg</i>	grau màxim del polinomi
<i>vap_a_usar</i>	flag que indica si trobem la varietat estable (0) / inestable (1)
<i>dif_serie</i>	punter a l'equació a avaluar amb sèries

#### 5.11.1.4 evaluate\_negative\_time()

```
void utils::UtilFunctions::evaluate_negative_time (
    double * at,
    double * x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, double *, int, double *) dif ) [static]
```

Funció que avalua l'algoritme de Runge-Kutta amb els valors demanats. Aquesta funció s'ha de cridar si volem que el temps final sigui anterior al temps inicial.

##### Paràmetres

<i>at</i>	temps inicial
<i>x</i>	punt inicial
<i>n</i>	dimensió punt
<i>ah</i>	pas d'integració (ha de ser negatiu!)
<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>dif</i>	punter a l'equació a avaluar

#### 5.11.1.5 evaluate\_positive\_time()

```
void utils::UtilFunctions::evaluate_positive_time (
    double * at,
    double * x,
    int n,
    double * ah,
    int sc,
    double tol,
    double * atf,
    double * aer,
    void(*) (double, double *, int, double *) dif ) [static]
```

Funció que avalua l'algoritme de Runge-Kutta amb els valors demanats. Aquesta funció s'ha de cridar si volem que el temps final sigui posterior al temps inicial.

##### Paràmetres

<i>at</i>	temps inicial
<i>x</i>	punt inicial
<i>n</i>	dimensió punt
<i>ah</i>	pas d'integració (ha de ser positiu!)

## Paràmetres

<i>sc</i>	ús (1)/ no ús (0) del control de pas
<i>tol</i>	tolerància màxima si usem control de pas
<i>atf</i>	temps final d'integració
<i>aer</i>	aturada (1)/ no aturada (0) del programa si es supera la tolerància
<i>dif</i>	punter a l'equació a avaluar

La documentació d'aquesta classe es va generar a partir dels següents fitxers:

- `clion_project/src/utils.h`
- `clion_project/src/utils.cpp`





# Índex alfabètic

~Serie  
Serie, 18

addCoeff  
Serie, 19

buscar\_punt\_fix  
utils::UtilFunctions, 35

buscar\_vap\_vep  
utils::UtilFunctions, 36

buscar\_varietat  
utils::UtilFunctions, 37

calculate  
Serie, 19

Circle, 9  
edo, 9

Circle\_Serie, 10  
edo, 10

cos  
Serie, 19

derivate  
Serie, 19

desc  
utils, 8

differential\_equation, 11

differential\_equation\_serie, 11

Disturbed\_Pendulum, 12  
edo, 12

Disturbed\_Pendulum\_Matrix, 13  
edo, 13

Disturbed\_Pendulum\_Serie, 14  
edo, 14

edo  
Circle, 9  
Circle\_Serie, 10  
Disturbed\_Pendulum, 12  
Disturbed\_Pendulum\_Matrix, 13  
Disturbed\_Pendulum\_Serie, 14

evaluate\_negative\_time  
utils::UtilFunctions, 37

evaluate\_positive\_time  
utils::UtilFunctions, 38

exp  
Serie, 20

getCoeff  
Serie, 20

getDegree  
Serie, 21

log  
Serie, 21

operator!=  
Serie, 21

operator\*  
Serie, 22

operator\*=  
Serie, 23

operator+  
Serie, 23, 24

operator+=  
Serie, 24, 25

operator-  
Serie, 25, 26

operator-=  
Serie, 26

operator/  
Serie, 28

operator/=  
Serie, 28, 30

operator=  
Serie, 30

operator==  
Serie, 32

pow  
Serie, 32

rkv78, 7

rkv78::rkv78, 15  
rkv78\_main, 15

rkv78\_main  
rkv78::rkv78, 15  
rkv78\_serie::rkv78\_serie, 16

rkv78\_serie, 7

rkv78\_serie::rkv78\_serie, 16  
rkv78\_main, 16

Serie, 17  
~Serie, 18  
addCoeff, 19  
calculate, 19  
cos, 19  
derivate, 19  
exp, 20  
getCoeff, 20

- getDegree, [21](#)
- log, [21](#)
- operator!=, [21](#)
- operator\*, [22](#)
- operator\*==, [23](#)
- operator+, [23](#), [24](#)
- operator+==, [24](#), [25](#)
- operator-, [25](#), [26](#)
- operator-=, [26](#)
- operator/, [28](#)
- operator/==, [28](#), [30](#)
- operator=, [30](#)
- operator==, [32](#)
- pow, [32](#)
- Serie, [18](#)
- setCoeff, [34](#)
- sin, [34](#)
- sqrt, [34](#)
- setCoeff
  - Serie, [34](#)
- sin
  - Serie, [34](#)
- sqrt
  - Serie, [34](#)
- utils, [7](#)
  - desc, [8](#)
- utils::UtilFunctions, [35](#)
  - buscar\_punt\_fix, [35](#)
  - buscar\_vap\_vep, [36](#)
  - buscar\_varietat, [37](#)
  - evaluate\_negative\_time, [37](#)
  - evaluate\_positive\_time, [38](#)