

Método de Newton para Sistemas Não Lineares

Gerado por Doxygen 1.8.16

1 Índice dos Componentes	1
1.1 Lista de Classes	1
2 Índice dos Arquivos	3
2.1 Lista de Arquivos	3
3 Classes	5
3.1 Referência da Classe Jacobiana	5
3.1.1 Descrição detalhada	6
3.1.2 Construtores e Destrutores	6
3.1.2.1 Jacobiana()	6
3.1.2.2 ~Jacobiana()	6
3.1.3 Funções membros	6
3.1.3.1 avaliar()	6
3.1.3.2 calculaDerivada()	7
3.1.3.3 inserirFuncao()	7
3.1.3.4 setH()	8
3.1.4 Atributos	8
3.1.4.1 funcao	8
3.1.4.2 h	8
3.1.4.3 nFuncao	8
3.1.4.4 nIncognita	9
3.2 Referência da Classe SistemaLinear	9
3.2.1 Descrição detalhada	10
3.2.2 Construtores e Destrutores	10
3.2.2.1 SistemaLinear()	10
3.2.2.2 ~SistemaLinear()	11
3.2.3 Funções membros	11
3.2.3.1 calcular() [1/2]	11
3.2.3.2 calcular() [2/2]	11
3.2.3.3 erro()	11
3.2.3.4 inserirMatriz()	12
3.2.3.5 inserirTermoFonte()	12
3.2.3.6 iterar()	12
3.2.3.7 setParadaErro()	13
3.2.3.8 setParadaIteracao()	13
3.2.4 Atributos	13
3.2.4.1 b	13
3.2.4.2 erroMaximo	14
3.2.4.3 m	14
3.2.4.4 nEquacao	14
3.2.4.5 nIteracao	14
3.2.4.6 nMaxIteracao	14

3.2.4.7 xAnt	15
3.2.4.8 xAtual	15
3.3 Referência da Classe SistemaNaoLinear	15
3.3.1 Descrição detalhada	16
3.3.2 Construtores e Destrutores	17
3.3.2.1 SistemaNaoLinear()	17
3.3.2.2 ~SistemaNaoLinear()	17
3.3.3 Funções membros	17
3.3.3.1 atualizaJacobiana()	17
3.3.3.2 atualizaX()	18
3.3.3.3 avaliaFuncoes()	18
3.3.3.4 calcular()	18
3.3.3.5 erro()	18
3.3.3.6 inserirFuncao()	19
3.3.3.7 iterar()	19
3.3.3.8 printX()	19
3.3.3.9 setJacobiana()	19
3.3.3.10 setLinear()	20
3.3.3.11 setParadaErro()	20
3.3.3.12 setParadaIteracao()	20
3.3.3.13 setPrint()	21
3.3.4 Atributos	21
3.3.4.1 delta	21
3.3.4.2 erroMaximo	21
3.3.4.3 funcao	22
3.3.4.4 funcaoPontoN	22
3.3.4.5 jacobiana	22
3.3.4.6 matrizJacobiana	22
3.3.4.7 nFuncao	22
3.3.4.8 nIncognita	23
3.3.4.9 nIteracao	23
3.3.4.10 nMaxIteracao	23
3.3.4.11 print	23
3.3.4.12 sistemaLinear	23
3.3.4.13 xAnterior	24
3.3.4.14 xAtual	24
4 Arquivos	25
4.1 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto↔ SistemaNãoLinear-MS211/jacobiana.cpp	25
4.2 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto↔ SistemaNãoLinear-MS211/jacobiana.h	25
4.2.1 Definições e macros	26

4.2.1.1 tipoFuncao	26
4.2.2 Definições dos tipos	26
4.2.2.1 Funcao	26
4.3 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/pch.cpp	26
4.4 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/pch.h	26
4.5 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211.cpp	26
4.5.1 Funções	27
4.5.1.1 f1()	27
4.5.1.2 f1Teste()	27
4.5.1.3 f2()	27
4.5.1.4 f2Teste()	27
4.5.1.5 main()	28
4.6 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/sistemaLinear.cpp	28
4.7 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/sistemaLinear.h	28
4.8 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/sistemaNaoLinear.cpp	28
4.9 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/sistemaNaoLinear.h	28
4.10 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-↔ SistemaNãoLinear-MS211/testes.cpp	29
Índice Remissivo	31

Capítulo 1

Índice dos Componentes

1.1 Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

Jacobiana	
Implementa o cálculo de uma matriz Jacobiana , aproximando as derivadas	5
SistemaLinear	
Implementa a resolução de sistemas lineares	9
SistemaNaoLinear	
Resolve um sistema não linear	15

Capítulo 2

Índice dos Arquivos

2.1 Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[jacobiana.cpp](#)
25

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[jacobiana.h](#)
25

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[pch.cpp](#)
26

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[pch.h](#)
26

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[Projeto-SistemaNãoLinear-M](#)
26

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[sistemaLinear.cpp](#)
28

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[sistemaLinear.h](#)
28

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[sistemaNaoLinear.cpp](#)
28

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[sistemaNaoLinear.h](#)
28

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[testes.cpp](#)
29

Capítulo 3

Classes

3.1 Referência da Classe Jacobiana

Implementa o cálculo de uma matriz [Jacobiana](#), aproximando as derivadas

```
#include <jacobiana.h>
```

Membros Públicos

- [Jacobiana](#) (int [nFuncao](#), int [nIncognita](#))
Construtor
- [~Jacobiana](#) ()
- void [inserirFuncao](#) (int indice, [Funcao funcao](#))
Insere uma função para cálculo
- void [setH](#) (double [h](#))
Define a diferença h para cálculo das derivadas
- double ** [avaliar](#) (double *x)
Avalia a matriz em um ponto

Membros Privados

- double [calculaDerivada](#) (int indiceFuncao, int indiceX, double *x)
Número de variáveis das funções

Atributos Privados

- [Funcao](#) * [funcao](#)
- double [h](#)
Vetor de funções
- int [nFuncao](#)
- int [nIncognita](#)
Número de funções no vetor

3.1.1 Descrição detalhada

Implementa o cálculo de uma matriz [Jacobiana](#), aproximando as derivadas

Aproxima a derivada considerando uma diferença h

Definição na linha 11 do arquivo jacobiana.h.

3.1.2 Construtores e Destrutores

3.1.2.1 Jacobiana()

```
Jacobiana::Jacobiana (
    int nFuncao,
    int nIncognita )
```

Construtor

Parâmetros

<i>nFuncao</i>	Número de funções
<i>nIncognita</i>	Número de incógnitas

Definição na linha 6 do arquivo jacobiana.cpp.

3.1.2.2 ~Jacobiana()

```
Jacobiana::~~Jacobiana ( )
```

Definição na linha 15 do arquivo jacobiana.cpp.

3.1.3 Funções membros

3.1.3.1 avaliar()

```
double ** Jacobiana::avaliar (
    double * x )
```

Avalia a matriz em um ponto

Parâmetros

<i>x</i>	Vetor em análise
----------	------------------

Retorna

A matriz calculada

Definição na linha 50 do arquivo jacobiana.cpp.

3.1.3.2 calculaDerivada()

```
double Jacobiana::calculaDerivada (
    int indiceFuncao,
    int indiceX,
    double * x ) [private]
```

Número de variáveis das funções

Calcula um valor de derivada

Parâmetros

<i>indiceFuncao</i>	Índice da função a ser avaliada
<i>indiceX</i>	Índice da variável em análise
<i>x</i>	Vetor das variáveis

Retorna

Valor da derivada calculada

Definição na linha 24 do arquivo jacobiana.cpp.

3.1.3.3 inserirFuncao()

```
void Jacobiana::inserirFuncao (
    int indice,
    Funcao funcao )
```

Insere uma função para cálculo

Parâmetros

<i>indice</i>	Índice da função sendo analisada
<i>funcao</i>	Ponteiro para a função

Definição na linha 19 do arquivo jacobiana.cpp.

3.1.3.4 setH()

```
void Jacobiana::setH (
    double h )
```

Define a diferença h para cálculo das derivadas

Parâmetros

<i>h</i>	Diferença h sendo inserida
----------	----------------------------

Definição na linha 45 do arquivo jacobiana.cpp.

3.1.4 Atributos

3.1.4.1 funcao

```
Funcao* Jacobiana::funcao [private]
```

Definição na linha 15 do arquivo jacobiana.h.

3.1.4.2 h

```
double Jacobiana::h [private]
```

Vetor de funções

Diferença a ser utilizada

Definição na linha 18 do arquivo jacobiana.h.

3.1.4.3 nFuncao

```
int Jacobiana::nFuncao [private]
```

Definição na linha 19 do arquivo jacobiana.h.

3.1.4.4 nIncognita

```
int Jacobiana::nIncognita [private]
```

Número de funções no vetor

Definição na linha 19 do arquivo jacobiana.h.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[jacobiana.h](#)
- C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/[jacobiana.cpp](#)

3.2 Referência da Classe SistemaLinear

Implementa a resolução de sistemas lineares

```
#include <sistemaLinear.h>
```

Membros Públicos

- [SistemaLinear](#) (int [nEquacao](#))
Construtor da classe
- [~SistemaLinear](#) ()
- void [setParadaIteracao](#) (int [nMaxIteracao](#))
Define o número máximo de iterações
- void [setParadaErro](#) (double [erroMaximo](#))
Define o erro máximo para parada
- void [inserirMatriz](#) (double **matrizCoeficientes)
Inserir a matriz de coeficientes
- void [inserirTermoFonte](#) (double *fonte)
Inserir o vetor de termos independentes
- double * [calcular](#) ()
Calcula a solução do sistema
- double * [calcular](#) (double *inicial)
Calcula a solução do sistema

Membros Privados

- void [iterar](#) ()
Solução atual
- double [erro](#) ()
Calcula o erro máximo nas funções

Atributos Privados

- int `nIteracao`
- int `nMaxIteracao`
Índice da iteração atual
- double `erroMaximo`
Número máximo de iterações
- int `nEquacao`
Erro máximo nas funções
- double `** m`
O número de equações do sistema
- double `* b`
A matriz de coeficientes
- double `* xAnt`
O vetor dos termos independentes
- double `* xAtual`
Solução anterior

3.2.1 Descrição detalhada

Implementa a resolução de sistemas lineares

Utiliza o método de Gauss-Jacobi

Definição na linha 5 do arquivo `sistemaLinear.h`.

3.2.2 Construtores e Destrutores

3.2.2.1 SistemaLinear()

```
SistemaLinear::SistemaLinear (
    int nEquacao )
```

Construtor da classe

Parâmetros

<code>nEquacao</code>	O número de equações do sistema
-----------------------	---------------------------------

Definição na linha 6 do arquivo `sistemaLinear.cpp`.

3.2.2.2 ~SistemaLinear()

```
SistemaLinear::~~SistemaLinear ( )
```

Definição na linha 19 do arquivo sistemaLinear.cpp.

3.2.3 Funções membros

3.2.3.1 calcular() [1/2]

```
double * SistemaLinear::calcular ( )
```

Calcula a solução do sistema

Retorna

O vetor solução

Definição na linha 43 do arquivo sistemaLinear.cpp.

3.2.3.2 calcular() [2/2]

```
double * SistemaLinear::calcular (
    double * inicial )
```

Calcula a solução do sistema

Parâmetros

<i>inicial</i>	Valor inicial
----------------	---------------

Retorna

O vetor solução

Definição na linha 68 do arquivo sistemaLinear.cpp.

3.2.3.3 erro()

```
double SistemaLinear::erro ( ) [private]
```

Calcula o erro máximo nas funções

Retorna

O erro máximo

Definição na linha 118 do arquivo sistemaLinear.cpp.

3.2.3.4 inserirMatriz()

```
void SistemaLinear::inserirMatriz (
    double ** matrizCoeficientes )
```

Insere a matriz de coeficientes

Parâmetros

<i>matrizCoeficientes</i>	A matriz de coeficientes
---------------------------	--------------------------

Definição na linha 33 do arquivo sistemaLinear.cpp.

3.2.3.5 inserirTermoFonte()

```
void SistemaLinear::inserirTermoFonte (
    double * fonte )
```

Insere o vetor de termos independentes

Parâmetros

<i>fonte</i>	Vetor
--------------	-------

Definição na linha 38 do arquivo sistemaLinear.cpp.

3.2.3.6 iterar()

```
void SistemaLinear::iterar ( ) [private]
```

Solução atual

Realiza uma iteração

Definição na linha 93 do arquivo sistemaLinear.cpp.

3.2.3.7 setParadaErro()

```
void SistemaLinear::setParadaErro (
    double erroMaximo )
```

Define o erro máximo para parada

Também determina que o algoritmo utilizará esse critério

Parâmetros

<i>erroMaximo</i>	O erro máximo a ser definido
-------------------	------------------------------

"0" para não utilizar esse critério

Definição na linha 28 do arquivo sistemaLinear.cpp.

3.2.3.8 setParadaIteracao()

```
void SistemaLinear::setParadaIteracao (
    int nMaxIteracao )
```

Define o número máximo de iterações

Parâmetros

<i>nMaxIteracao</i>	O número máximo de iterações
---------------------	------------------------------

Definição na linha 23 do arquivo sistemaLinear.cpp.

3.2.4 Atributos

3.2.4.1 b

```
double* SistemaLinear::b [private]
```

A matriz de coeficientes

Definição na linha 15 do arquivo sistemaLinear.h.

3.2.4.2 erroMaximo

```
double SistemaLinear::erroMaximo [private]
```

Número máximo de iterações

Definição na linha 10 do arquivo sistemaLinear.h.

3.2.4.3 m

```
double** SistemaLinear::m [private]
```

O número de equações do sistema

Definição na linha 14 do arquivo sistemaLinear.h.

3.2.4.4 nEquacao

```
int SistemaLinear::nEquacao [private]
```

Erro máximo nas funções

Definição na linha 12 do arquivo sistemaLinear.h.

3.2.4.5 nIteracao

```
int SistemaLinear::nIteracao [private]
```

Definição na linha 8 do arquivo sistemaLinear.h.

3.2.4.6 nMaxIteracao

```
int SistemaLinear::nMaxIteracao [private]
```

Índice da iteração atual

Definição na linha 9 do arquivo sistemaLinear.h.

3.2.4.7 xAnt

```
double* SistemaLinear::xAnt [private]
```

O vetor dos termos independentes

Definição na linha 17 do arquivo sistemaLinear.h.

3.2.4.8 xAtual

```
double* SistemaLinear::xAtual [private]
```

Solução anterior

Definição na linha 18 do arquivo sistemaLinear.h.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/[sistemaLinear.h](#)
- C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/[sistemaLinear.cpp](#)

3.3 Referência da Classe SistemaNaoLinear

Resolve um sistema não linear

```
#include <sistemaNaoLinear.h>
```

Membros Públicos

- [SistemaNaoLinear](#) (int nFuncao, int nIncognita)
Construtor
- [~SistemaNaoLinear](#) ()
- void [inserirFuncao](#) (int indice, [Funcao funcao](#))
Define uma das funções do sistema
- void [setParadaIteracao](#) (int nMaxIteracao)
Define o número máximo de iterações
- void [setParadaErro](#) (double erroMaximo)
Define o erro máximo para parada
- double * [calcular](#) (double *inicial)
Calcula a solução do sistema
- void [setJacobiana](#) (double h)
Define a diferença h para cálculo das derivadas
- void [setLinear](#) (int nIteracaoMax, double erroMax)
Define os parâmetros para cálculo do sistema linear
- void [setPrint](#) (bool print)
Define se deverá imprimir os valores das iterações

Membros Privados

- void `iterar` ()
Calcula a matriz `Jacobiana`
- double `erro` ()
Calcula o erro máximo nas funções
- void `avaliaFuncoes` ()
Avalia as funções no ponto atual
- void `atualizaX` ()
Atualiza o vetor `xAtual`
- void `atualizaJacobiana` ()
Atualiza a matriz `Jacobiana`
- void `printX` ()
Imprime o valor da iteração atual

Atributos Privados

- double * `xAtual`
- double * `xAnterior`
Solução atual do sistema
- double * `funcaoPontoN`
Solução anterior do sistema
- double * `delta`
Valor da função no ponto atual
- int `nFuncao`
Diferença entre `xAtual` e `xAnterior`
- int `nIncognita`
Número de funções do sistema
- `Funcao` * `funcao`
Número de incógnitas do sistema
- int `nIteracao`
- int `nMaxIteracao`
Índice da iteração atual
- double `erroMaximo`
Número máximo de iterações
- double ** `matrizJacobiana`
Erro máximo nas funções
- bool `print`
Matriz `Jacobiana` atual
- `SistemaLinear` `sistemaLinear`
Armazena a opção de imprimir os resultados
- `Jacobiana` `jacobiana`
Resolve o sistema linear

3.3.1 Descrição detalhada

Resolve um sistema não linear

Definição na linha 13 do arquivo `sistemaNaoLinear.h`.

3.3.2 Construtores e Destrutores

3.3.2.1 SistemaNaoLinear()

```
SistemaNaoLinear::SistemaNaoLinear (
    int nFuncao,
    int nIncognita )
```

Construtor

Calcula o valor absoluto de um nmero

Parâmetros

<i>nFuncao</i>	Número de funções do sistema
<i>nIncognita</i>	Número de incógnitas do sistema

param name="x"> Nmero

returns> Valor absoluto do nmero

Definição na linha 24 do arquivo sistemaNaoLinear.cpp.

3.3.2.2 ~SistemaNaoLinear()

```
SistemaNaoLinear::~~SistemaNaoLinear ( )
```

Definição na linha 41 do arquivo sistemaNaoLinear.cpp.

3.3.3 Funções membros

3.3.3.1 atualizaJacobiana()

```
void SistemaNaoLinear::atualizaJacobiana ( ) [private]
```

Atualiza a matriz [Jacobiana](#)

Definição na linha 158 do arquivo sistemaNaoLinear.cpp.

3.3.3.2 atualizaX()

```
void SistemaNaoLinear::atualizaX ( ) [private]
```

Atualiza o vetor xAtual

Definição na linha 149 do arquivo sistemaNaoLinear.cpp.

3.3.3.3 avaliaFuncoes()

```
void SistemaNaoLinear::avaliaFuncoes ( ) [private]
```

Avalia as funções no ponto atual

Definição na linha 140 do arquivo sistemaNaoLinear.cpp.

3.3.3.4 calcular()

```
double * SistemaNaoLinear::calcular (
    double * inicial )
```

Calcula a solução do sistema

Parâmetros

<i>inicial</i>	O valor inicial para o sistema
----------------	--------------------------------

Retorna

O vetor solução

Definição na linha 62 do arquivo sistemaNaoLinear.cpp.

3.3.3.5 erro()

```
double SistemaNaoLinear::erro ( ) [private]
```

Calcula o erro máximo nas funções

Retorna

O erro máximo

Definição na linha 125 do arquivo sistemaNaoLinear.cpp.

3.3.3.6 inserirFuncao()

```
void SistemaNaoLinear::inserirFuncao (
    int indice,
    Funcao funcao )
```

Define uma das funções do sistema

A função deve retornar "0" no vetor solução

Parâmetros

<i>indice</i>	Índice da função
<i>funcao</i>	Ponteiro para a função

Definição na linha 45 do arquivo sistemaNaoLinear.cpp.

3.3.3.7 iterar()

```
void SistemaNaoLinear::iterar ( ) [private]
```

Calcula a matriz [Jacobiana](#)

Realiza uma iteração

Definição na linha 110 do arquivo sistemaNaoLinear.cpp.

3.3.3.8 printX()

```
void SistemaNaoLinear::printX ( ) [private]
```

Imprime o valor da iteração atual

Definição na linha 163 do arquivo sistemaNaoLinear.cpp.

3.3.3.9 setJacobiana()

```
void SistemaNaoLinear::setJacobiana (
    double h )
```

Define a diferença h para cálculo das derivadas

Parâmetros

<i>h</i>	Diferença
----------	-----------

Definição na linha 99 do arquivo sistemaNaoLinear.cpp.

3.3.3.10 setLinear()

```
void SistemaNaoLinear::setLinear (
    int nIteracaoMax,
    double erroMax )
```

Define os parâmetros para cálculo do sistema linear

Parâmetros

<i>nIteracaoMax</i>	O número máximo de iterações
<i>erroMax</i>	O erro máximo do sistema, 0 para não usar

Definição na linha 104 do arquivo sistemaNaoLinear.cpp.

3.3.3.11 setParadaErro()

```
void SistemaNaoLinear::setParadaErro (
    double erroMaximo )
```

Define o erro máximo para parada

Também determina que o algoritmo utilizará esse critério

Parâmetros

<i>erroMaximo</i>	O erro máximo a ser definido
-------------------	------------------------------

"0" para não utilizar esse critério

Definição na linha 57 do arquivo sistemaNaoLinear.cpp.

3.3.3.12 setParadaIteracao()

```
void SistemaNaoLinear::setParadaIteracao (
    int nMaxIteracao )
```

Define o número máximo de iterações

Parâmetros

<i>nMaxIteracao</i>	O número máximo de iterações
---------------------	------------------------------

Definição na linha 52 do arquivo sistemaNaoLinear.cpp.

3.3.3.13 setPrint()

```
void SistemaNaoLinear::setPrint (
    bool print )
```

Define se deverá printar os valores das iterações

Parâmetros

<i>print</i>	True se deverá printar, false caso contrário
--------------	--

Definição na linha 172 do arquivo sistemaNaoLinear.cpp.

3.3.4 Atributos**3.3.4.1 delta**

```
double* SistemaNaoLinear::delta [private]
```

Valor da função no ponto atual

Definição na linha 19 do arquivo sistemaNaoLinear.h.

3.3.4.2 erroMaximo

```
double SistemaNaoLinear::erroMaximo [private]
```

Número máximo de iterações

Definição na linha 30 do arquivo sistemaNaoLinear.h.

3.3.4.3 funcao

```
Funcao* SistemaNaoLinear::funcao [private]
```

Número de incógnitas do sistema

Vetor de funções

Devem retorna "0" no vetor solução

Definição na linha 26 do arquivo sistemaNaoLinear.h.

3.3.4.4 funcaoPontoN

```
double* SistemaNaoLinear::funcaoPontoN [private]
```

Solução anterior do sistema

Definição na linha 18 do arquivo sistemaNaoLinear.h.

3.3.4.5 jacobiana

```
Jacobiana SistemaNaoLinear::jacobiana [private]
```

Resolve o sistema linear

Definição na linha 35 do arquivo sistemaNaoLinear.h.

3.3.4.6 matrizJacobiana

```
double** SistemaNaoLinear::matrizJacobiana [private]
```

Erro máximo nas funções

Definição na linha 31 do arquivo sistemaNaoLinear.h.

3.3.4.7 nFuncao

```
int SistemaNaoLinear::nFuncao [private]
```

Diferença entre xAtual e xAnterior

Definição na linha 21 do arquivo sistemaNaoLinear.h.

3.3.4.8 nIncognita

```
int SistemaNaoLinear::nIncognita [private]
```

Número de funções do sistema

Definição na linha 21 do arquivo sistemaNaoLinear.h.

3.3.4.9 nIteracao

```
int SistemaNaoLinear::nIteracao [private]
```

Definição na linha 28 do arquivo sistemaNaoLinear.h.

3.3.4.10 nMaxIteracao

```
int SistemaNaoLinear::nMaxIteracao [private]
```

Índice da iteração atual

Definição na linha 29 do arquivo sistemaNaoLinear.h.

3.3.4.11 print

```
bool SistemaNaoLinear::print [private]
```

Matriz [Jacobiana](#) atual

Definição na linha 32 do arquivo sistemaNaoLinear.h.

3.3.4.12 sistemaLinear

```
SistemaLinear SistemaNaoLinear::sistemaLinear [private]
```

Armazena a opção de printar os resultados

Definição na linha 34 do arquivo sistemaNaoLinear.h.

3.3.4.13 xAnterior

```
double* SistemaNaoLinear::xAnterior [private]
```

Solução atual do sistema

Definição na linha 17 do arquivo sistemaNaoLinear.h.

3.3.4.14 xAtual

```
double* SistemaNaoLinear::xAtual [private]
```

Definição na linha 16 do arquivo sistemaNaoLinear.h.

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/[sistemaNaoLinear.h](#)
- C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/[sistemaNaoLinear.cpp](#)

Capítulo 4

Arquivos

4.1 Referência do Arquivo

**C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/↵
Projeto-SistemaNãoLinear-MS211/jacobiana.cpp**

```
#include "pch.h"  
#include "jacobiana.h"
```

4.2 Referência do Arquivo

**C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/↵
Projeto-SistemaNãoLinear-MS211/jacobiana.h**

```
#include <stdlib.h>
```

Componentes

- class [Jacobiana](#)
Implementa o cálculo de uma matriz [Jacobiana](#), aproximando as derivadas

Definições e Macros

- #define [tipoFuncao](#)

Definições de Tipos

- typedef double(* [Funcao](#)) (double *)

4.2.1 Definições e macros

4.2.1.1 tipoFuncao

```
#define tipoFuncao
```

Definição na linha 5 do arquivo jacobiana.h.

4.2.2 Definições dos tipos

4.2.2.1 Funcao

```
typedef double(* Funcao) (double *)
```

Definição na linha 6 do arquivo jacobiana.h.

4.3 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-↵ SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/pch.cpp

```
#include "pch.h"
```

4.4 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-↵ SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/pch.h

4.5 Referência do Arquivo C:/Users/Elton/source/Repos/Projeto-↵ SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211.cpp

```
#include "pch.h"  
#include <iostream>  
#include "sistemaNaoLinear.h"  
#include <math.h>
```


Funções

- double `f1` (double *x)
- double `f2` (double *x)
- double `f1Teste` (double *x)
- double `f2Teste` (double *x)
- int `main` ()

4.5.1 Funções

4.5.1.1 `f1()`

```
double f1 (  
    double * x )
```

Definição na linha 12 do arquivo Projeto-SistemaNãoLinear-MS211.cpp.

4.5.1.2 `f1Teste()`

```
double f1Teste (  
    double * x )
```

Definição na linha 23 do arquivo Projeto-SistemaNãoLinear-MS211.cpp.

4.5.1.3 `f2()`

```
double f2 (  
    double * x )
```

Definição na linha 17 do arquivo Projeto-SistemaNãoLinear-MS211.cpp.

4.5.1.4 `f2Teste()`

```
double f2Teste (  
    double * x )
```

Definição na linha 28 do arquivo Projeto-SistemaNãoLinear-MS211.cpp.

4.5.1.5 main()

```
int main ( )
```

Definição na linha 33 do arquivo Projeto-SistemaNãoLinear-MS211.cpp.

4.6 Referência do Arquivo

**C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/↵
Projeto-SistemaNãoLinear-MS211/sistemaLinear.cpp**

```
#include "pch.h"  
#include "sistemaLinear.h"  
#include <stdlib.h>
```

4.7 Referência do Arquivo

**C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/↵
Projeto-SistemaNãoLinear-MS211/sistemaLinear.h**

Componentes

- class [SistemaLinear](#)
Implementa a resolução de sistemas lineares

4.8 Referência do Arquivo

**C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/↵
Projeto-SistemaNãoLinear-MS211/sistemaNaoLinear.cpp**

```
#include "pch.h"  
#include "sistemaNaoLinear.h"  
#include <iostream>
```

4.9 Referência do Arquivo

**C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/↵
Projeto-SistemaNãoLinear-MS211/sistemaNaoLinear.h**

```
#include "sistemaLinear.h"  
#include "jacobiana.h"
```

Componentes

- class [SistemaNaoLinear](#)
Resolve um sistema não linear

4.10 Referência do Arquivo

C:/Users/Elton/source/Repos/Projeto-SistemaNãoLinear-MS211/Projeto-SistemaNãoLinear-MS211/testes.cpp

```
#include <iostream>
#include <stdlib.h>
#include "pch.h"
#include "sistemaLinear.h"
```


Índice Remissivo

~Jacobiana
 Jacobiana, 6
~SistemaLinear
 SistemaLinear, 10
~SistemaNaoLinear
 SistemaNaoLinear, 17
atualizaJacobiana
 SistemaNaoLinear, 17
atualizaX
 SistemaNaoLinear, 17
avaliaFuncoes
 SistemaNaoLinear, 18
avaliar
 Jacobiana, 6
b
 SistemaLinear, 13
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/jacobiana.cpp, 25
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/jacobiana.h, 25
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/pch.cpp, 26
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/pch.h, 26
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211.cpp, 26
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/sistemaLinear.cpp, 28
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/sistemaLinear.h, 28
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/sistemaNaoLinear.cpp, 28
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/sistemaNaoLinear.h, 28
C:/Users/Elton/source/Repos/Projeto-SistemaNaoLinear-MS211/Projeto-SistemaNaoLinear-MS211/testes.cpp, 29
calculaDerivada
 Jacobiana, 7
 calcular
 SistemaLinear, 11
 SistemaNaoLinear, 18
 delta
 SistemaNaoLinear, 21
 erro
 SistemaLinear, 11
 SistemaNaoLinear, 18
 erroMaximo
 SistemaLinear, 13
 SistemaNaoLinear, 21
 f1
 Projeto-SistemaNaoLinear-MS211.cpp, 27
 f1Teste
 Projeto-SistemaNaoLinear-MS211.cpp, 27
 f2
 Projeto-SistemaNaoLinear-MS211.cpp, 27
 f2Teste
 Projeto-SistemaNaoLinear-MS211.cpp, 27
 funcao
 jacobiana.h, 26
 funcao
 Jacobiana, 8
 SistemaNaoLinear, 21
 funcaoPontoN
 SistemaNaoLinear, 22
 h
 Jacobiana, 8
 inserirFuncao
 Jacobiana, 7
 SistemaNaoLinear, 18
 inserirMatriz
 SistemaLinear, 12
 inserirTermoFonte
 SistemaLinear, 12
 SistemaNaoLinear.cpp, 12
 SistemaLinear, 12
 SistemaNaoLinear, 19
 Jacobiana, 5
 ~Jacobiana, 6
 avaliar, 6
 calculaDerivada, 7
 funcao, 8

- h, 8
- inserirFuncao, 7
- Jacobiana, 6
- nFuncao, 8
- nIncognita, 8
- setH, 8
- jacobiana
 - SistemaNaoLinear, 22
- jacobiana.h
 - Funcao, 26
 - tipoFuncao, 26
- m
 - SistemaLinear, 14
- main
 - Projeto-SistemaNaoLinear-MS211.cpp, 27
- matrizJacobiana
 - SistemaNaoLinear, 22
- nEquacao
 - SistemaLinear, 14
- nFuncao
 - Jacobiana, 8
 - SistemaNaoLinear, 22
- nIncognita
 - Jacobiana, 8
 - SistemaNaoLinear, 22
- nIteracao
 - SistemaLinear, 14
 - SistemaNaoLinear, 23
- nMaxIteracao
 - SistemaLinear, 14
 - SistemaNaoLinear, 23
- print
 - SistemaNaoLinear, 23
- printX
 - SistemaNaoLinear, 19
- Projeto-SistemaNaoLinear-MS211.cpp
 - f1, 27
 - f1Teste, 27
 - f2, 27
 - f2Teste, 27
 - main, 27
- setH
 - Jacobiana, 8
- setJacobiana
 - SistemaNaoLinear, 19
- setLinear
 - SistemaNaoLinear, 20
- setParadaErro
 - SistemaLinear, 12
 - SistemaNaoLinear, 20
- setParadaIteracao
 - SistemaLinear, 13
 - SistemaNaoLinear, 20
- setPrint
 - SistemaNaoLinear, 21
- SistemaLinear, 9
 - ~SistemaLinear, 10
 - b, 13
 - calcular, 11
 - erro, 11
 - erroMaximo, 13
 - inserirMatriz, 12
 - inserirTermoFonte, 12
 - iterar, 12
 - m, 14
 - nEquacao, 14
 - nIteracao, 14
 - nMaxIteracao, 14
 - setParadaErro, 12
 - setParadaIteracao, 13
 - SistemaLinear, 10
 - xAnt, 14
 - xAtual, 15
- sistemaLinear
 - SistemaNaoLinear, 23
- SistemaNaoLinear, 15
 - ~SistemaNaoLinear, 17
 - atualizaJacobiana, 17
 - atualizaX, 17
 - avaliaFuncoes, 18
 - calcular, 18
 - delta, 21
 - erro, 18
 - erroMaximo, 21
 - funcao, 21
 - funcaoPontoN, 22
 - inserirFuncao, 18
 - iterar, 19
 - jacobiana, 22
 - matrizJacobiana, 22
 - nFuncao, 22
 - nIncognita, 22
 - nIteracao, 23
 - nMaxIteracao, 23
 - print, 23
 - printX, 19
 - setJacobiana, 19
 - setLinear, 20
 - setParadaErro, 20
 - setParadaIteracao, 20
 - setPrint, 21
 - sistemaLinear, 23
 - SistemaNaoLinear, 17
 - xAnterior, 23
 - xAtual, 24
- tipoFuncao
 - jacobiana.h, 26
- xAnt
 - SistemaLinear, 14
- xAnterior
 - SistemaNaoLinear, 23
- xAtual

SistemaLinear, [15](#)
SistemaNaoLinear, [24](#)