

Métodos Computacionais em Física I (FIW234)

Turmas IFA e IFB

Equações Diferenciais Acopladas

Edivaldo M. Santos e João R. T. de Mello Neto

Aula 5

Tópicos de Equações Diferenciais

- O método de Euler
- O método de Runge-Kutta de ordem 2
- O método de Runge-Kutta de ordem 4

Equações Diferenciais

- São inúmeros os problemas de física, química, biologia, economia, etc, cuja resposta envolvem a solução de equações diferenciais:
 - ① osciladores harmônicos livres com ou sem forças de amortecimento, osciladores forçados;
 - ② lançamento de projéteis;
 - ③ tensões e correntes em circuitos elétricos;
 - ④ concentrações de substâncias em reações químicas;
 - ⑤ fluxo de calor em condutores térmicos;
 - ⑥ propagação de ondas mecânicas e eletromagnéticas;
 - ⑦ dinâmica de populações biológicas;
 - ⑧ evolução do preço de ações no mercado financeiro;

- A ordem de uma equação diferencial é definida pela derivada mais alta aparecendo na equação. Assim, dizemos que a equação:

$$a \frac{d^4 y}{dx^4} + b \frac{d^3 y}{dx^3} + c \frac{d^2 y}{dx^2} + d \frac{dy}{dx} + ex + f = 0 \quad (1)$$

em que y é uma função de x é de ordem 4.

Equações Diferenciais

- A ordem de uma equação diferencial é definida pela derivada mais alta aparecendo na equação. Assim, dizemos que a equação:

$$a \frac{d^4 y}{dx^4} + b \frac{d^3 y}{dx^3} + c \frac{d^2 y}{dx^2} + d \frac{dy}{dx} + ex + f = 0 \quad (1)$$

em que y é uma função de x é de ordem 4.

- Ao longo tempo, desenvolveram-se métodos para a resolução de **sistemas acoplados de equações diferencial de primeira ordem**.
- Como então aplicar tais métodos quando a equação diferencial em questão tem ordem maior do que 1?

Equações Diferenciais

- A ordem de uma equação diferencial é definida pela derivada mais alta aparecendo na equação. Assim, dizemos que a equação:

$$a \frac{d^4 y}{dx^4} + b \frac{d^3 y}{dx^3} + c \frac{d^2 y}{dx^2} + d \frac{dy}{dx} + ex + f = 0 \quad (1)$$

em que y é uma função de x é de ordem 4.

- Ao longo tempo, desenvolveram-se métodos para a resolução de **sistemas acoplados de equações diferencial de primeira ordem**.
- Como então aplicar tais métodos quando a equação diferencial em questão tem ordem maior do que 1?
- Por sorte, é sempre possível transformar uma equação diferencial de ordem n num conjunto de n equações diferenciais acopladas de ordem 1.
- Vejamos como...

Equações Diferenciais

- Tomemos como exemplo o problema de encontrar a solução $x(t)$ para a posição de uma partícula de massa m sujeita a uma força restauradora do tipo lei de Hook ($F(x) = -kx$).

Equações Diferenciais

- Tomemos como exemplo o problema de encontrar a solução $x(t)$ para a posição de uma partícula de massa m sujeita a uma força restauradora do tipo lei de Hook ($F(x) = -kx$).
- A 2ª lei de Newton aplicado à partícula, fornece:

$$m \frac{d^2 x}{dt^2} = -kx \quad \text{ou} \quad m \frac{d^2 x}{dt^2} + kx = 0 \quad \text{ou} \quad \frac{d^2 x}{dt^2} + \frac{k}{m}x = 0, \quad (2)$$

que é uma equação diferencial de segunda ordem para a posição x como função do tempo t .

Equações Diferenciais

- Tomemos como exemplo o problema de encontrar a solução $x(t)$ para a posição de uma partícula de massa m sujeita a uma força restauradora do tipo lei de Hook ($F(x) = -kx$).
- A 2ª lei de Newton aplicado à partícula, fornece:

$$m \frac{d^2 x}{dt^2} = -kx \quad \text{ou} \quad m \frac{d^2 x}{dt^2} + kx = 0 \quad \text{ou} \quad \frac{d^2 x}{dt^2} + \frac{k}{m}x = 0, \quad (2)$$

que é uma equação diferencial de segunda ordem para a posição x como função do tempo t .

- Se definirmos uma variável $y_1 = x$ e outra variável $y_2 = \frac{dx}{dt}$, a solução da equação diferencial de segunda ordem acima é equivalente à do seguinte conjunto de 2 equações acopladas de primeira ordem:

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = -\frac{k}{m}y_1 \end{cases} \quad (3)$$

Equações Diferenciais

- Tomemos como exemplo o problema de encontrar a solução $x(t)$ para a posição de uma partícula de massa m sujeita a uma força restauradora do tipo lei de Hook ($F(x) = -kx$).
- A 2ª lei de Newton aplicado à partícula, fornece:

$$m \frac{d^2 x}{dt^2} = -kx \quad \text{ou} \quad m \frac{d^2 x}{dt^2} + kx = 0 \quad \text{ou} \quad \frac{d^2 x}{dt^2} + \frac{k}{m}x = 0, \quad (2)$$

que é uma equação diferencial de segunda ordem para a posição x como função do tempo t .

- Se definirmos uma variável $y_1 = x$ e outra variável $y_2 = \frac{dx}{dt}$, a solução da equação diferencial de segunda ordem acima é equivalente à do seguinte conjunto de 2 equações acopladas de primeira ordem:

$$\begin{cases} \frac{dy_1}{dt} = y_2 \\ \frac{dy_2}{dt} = -\frac{k}{m}y_1 \end{cases} \quad (3)$$

- E não é difícil ver que o procedimento acima pode ser aplicado a uma equação de ordem arbitrária n para produzir um conjunto de n equações acopladas de ordem 1:

$$\frac{dy_i}{dx} = f(x, y_1, \dots, y_n) \quad i = 1, 2, \dots, n \quad (4)$$

em que a derivada em relação a x da i -ésima variável pode, em princípio, ser uma função da variável independente x , assim como das n variáveis dependentes y_i .

Equações Diferenciais: condições de contorno

- Do ponto de vista formal, a solução de equação diferencial de ordem n , envolve n integrações para sair da n -ésima derivada e chegar na primitiva de ordem 0. A cada integração surge uma constante de integração.
- Assim, a solução completa de uma equação diferencial de ordem n envolve a especificação de n condições de contorno.

Equações Diferenciais: condições de contorno

- Do ponto de vista formal, a solução de equação diferencial de ordem n , envolve n integrações para sair da n -ésima derivada e chegar na primitiva de ordem 0. A cada integração surge uma constante de integração.
- Assim, a solução completa de uma equação diferencial de ordem n envolve a especificação de n condições de contorno.
- Pode-se dividir os problemas de equações diferenciais em 2 grandes categorias dependendo da natureza das condições de contorno envolvidas:
 - ❶ **problemas de valores iniciais:** as variáveis dependentes y_i são dadas para algum valor inicial x_i e propagadas até um valor final x_f .
 - ❷ **problemas de fronteira:** as variáveis dependentes devem satisfazer condições de contorno em mais de um valor de x . Em problemas de fronteira de dois pontos, por exemplo, algumas condições (k) devem ser satisfeitas em x_i e o restante ($n - k$) em x_f .

Equações Diferenciais: condições de contorno

- Do ponto de vista formal, a solução de equação diferencial de ordem n , envolve n integrações para sair da n -ésima derivada e chegar na primitiva de ordem 0. A cada integração surge uma constante de integração.
- Assim, a solução completa de uma equação diferencial de ordem n envolve a especificação de n condições de contorno.
- Pode-se dividir os problemas de equações diferenciais em 2 grandes categorias dependendo da natureza das condições de contorno envolvidas:
 - ❶ **problemas de valores iniciais:** as variáveis dependentes y_i são dadas para algum valor inicial x_i e propagadas até um valor final x_f .
 - ❷ **problemas de fronteira:** as variáveis dependentes devem satisfazer condições de contorno em mais de um valor de x . Em problemas de fronteira de dois pontos, por exemplo, algumas condições (k) devem ser satisfeitas em x_i e o restante ($n - k$) em x_f .
- Nesse curso abordaremos apenas problemas de valores iniciais.
- Problemas de fronteira são mais complicados que os de valores iniciais e exigem algoritmos mais sofisticados dos que os que apresentaremos aqui.

Equações Diferenciais: O método de Euler

- O método mais simples que pode ser utilizado para se avançar a i -ésima variável dependente y_i , após n passos, de x até $x + h$ é o de Euler, cuja fórmula é:

$$y_i^{(n+1)} = y_i^{(n)} + h * f(x_n, y_1, \dots, y_n), \quad (5)$$

ou seja, uma simples extrapolação linear utilizando-se a derivada de y_i com respeito a x , calculada no ponto x_n .

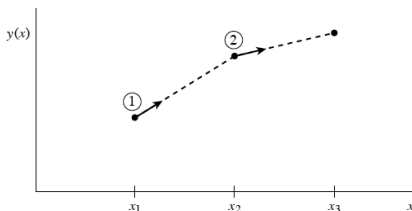
Equações Diferenciais: O método de Euler

- O método mais simples que pode ser utilizado para se avançar a i -ésima variável dependente y_i , após n passos, de x até $x + h$ é o de Euler, cuja fórmula é:

$$y_i^{(n+1)} = y_i^{(n)} + h * f(x_n, y_1, \dots, y_n), \quad (5)$$

ou seja, uma simples extrapolação linear utilizando-se a derivada de y_i com respeito a x , calculada no ponto x_n .

- O erro que se comete com esse tipo de avanço é de ordem h^2 a cada passo.
- O algoritmo de Euler é bastante rudimentar e na maioria dos casos, para um número de passos moderados, deve levar a erros apreciáveis.
- Exploreemos então as limitações de tal algoritmo, para que vocês se convençam disso...



Equações Diferenciais: o método de Euler

- Apliquemos então o algoritmo de Euler para a solução do problema do pêndulo simples.
- Baixe o código fonte `mhs_euler.c`.
- Vamos analisá-lo com cuidado.

Equações Diferenciais: o método de Euler

- Apliquemos então o algoritmo de Euler para a solução do problema do pêndulo simples.
- Baixe o código fonte `mhs_euler.c`.
- Vamos analisá-lo com cuidado.
- Compile e execute o programa no seu terminal.

Equações Diferenciais: o método de Euler

- Apliquemos então o algoritmo de Euler para a solução do problema do pêndulo simples.
- Baixe o código fonte `mhs_euler.c`.
- Vamos analisá-lo com cuidado.
- Compile e execute o programa no seu terminal.
- grafique no Gnuplot a posição $x(t)$.

Equações Diferenciais: o método de Euler

- Apliquemos então o algoritmo de Euler para a solução do problema do pêndulo simples.
- Baixe o código fonte `mhs_euler.c`.
- Vamos analisá-lo com cuidado.
- Compile e execute o programa no seu terminal.
- grafique no Gnuplot a posição $x(t)$.
- superponha a $x(t)$, o gráfico da velocidade $v(t)$.

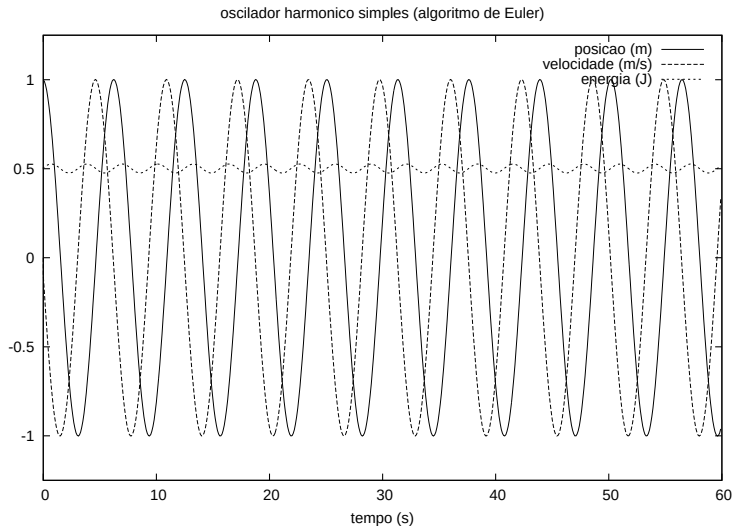
Equações Diferenciais: o método de Euler

- Apliquemos então o algoritmo de Euler para a solução do problema do pêndulo simples.
- Baixe o código fonte `mhs_euler.c`.
- Vamos analisá-lo com cuidado.
- Compile e execute o programa no seu terminal.
- grafique no Gnuplot a posição $x(t)$.
- superponha a $x(t)$, o gráfico da velocidade $v(t)$.
- superponha a $x(t)$ e $v(t)$, o gráfico da energia mecânica $E = K + U$.

Equações Diferenciais: o método de Euler

- Apliquemos então o algoritmo de Euler para a solução do problema do pêndulo simples.
- Baixe o código fonte `mhs_euler.c`.
- Vamos analisá-lo com cuidado.
- Compile e execute o programa no seu terminal.
- grafique no Gnuplot a posição $x(t)$.
- superponha a $x(t)$, o gráfico da velocidade $v(t)$.
- superponha a $x(t)$ e $v(t)$, o gráfico da energia mecânica $E = K + U$.
- que conclusões tirar dessas tarefas?

Equações Diferenciais: o método de Euler



Equações Diferenciais: Runge-Kutta de ordem 2 (RK2)

- Vimos que um método cujo erro no avanço a cada passo é de ordem 2, como o de Euler, leva a erros grosseiros, mesmo para sistemas de equações simples como um oscilador harmônico livre.
- o próximo método que analisaremos é o de Runge-Kutta de ordem 2, conhecido pela sigla RK2.

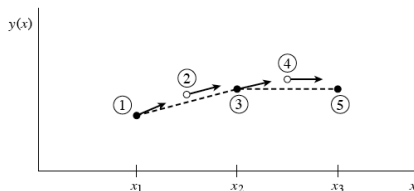
Equações Diferenciais: Runge-Kutta de ordem 2 (RK2)

- Vimos que um método cujo erro no avanço a cada passo é de ordem 2, como o de Euler, leva a erros grosseiros, mesmo para sistemas de equações simples como um oscilador harmônico livre.
- o próximo método que analisaremos é o de Runge-Kutta de ordem 2, conhecido pela sigla RK2.
- A diferença principal do RK2 em relação a Euler é a utilização de um passo tentativa no meio do intervalo de avanço, como representado esquematicamente na figura abaixo. Matematicamente, o algoritmo é caracterizado pelas seguintes equações (para o caso de uma única variável):

$$k_1 = hf(x^{(n)}, y^{(n)})$$

$$k_2 = hf\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_1}{2}\right)$$

$$y^{(n+1)} = y^{(n)} + k_2 + O(h^3)$$



Equações Diferenciais: Runge-Kutta de ordem 2 (RK2)

- O método de Runge-Kutta de ordem 2 simula a acurácia de uma expansão em série de Taylor até ordem 2, e o erro a cada passo é de ordem 3. Vejamos como...

Equações Diferenciais: Runge-Kutta de ordem 2 (RK2)

- O método de Runge-Kutta de ordem 2 simula a acurácia de uma expansão em série de Taylor até ordem 2, e o erro a cada passo é de ordem 3. Vejamos como...
- Se tomarmos f como uma função de duas variáveis: x e y , com $y = y(t)$, a regra da cadeia para funções de muitas variáveis implica que até ordem h :

$$f\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_1}{2}\right) = f(x^{(n)}, y^{(n)}) + \frac{h}{2} \left[\frac{df}{dt} \right]_{x^{(n)}, y^{(n)}} + \dots = \frac{dy}{dx} \Big|_{x^{(n)}, y^{(n)}} + \frac{h}{2} \frac{d^2 y}{dx^2} \Big|_{x^{(n)}, y^{(n)}} + \dots \quad (6)$$

Equações Diferenciais: Runge-Kutta de ordem 2 (RK2)

- O método de Runge-Kutta de ordem 2 simula a acurácia de uma expansão em série de Taylor até ordem 2, e o erro a cada passo é de ordem 3. Vejamos como...
- Se tomarmos f como uma função de duas variáveis: x e y , com $y = y(t)$, a regra da cadeia para funções de muitas variáveis implica que até ordem h :

$$f\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_1}{2}\right) = f(x^{(n)}, y^{(n)}) + \frac{h}{2} \left[\frac{df}{dt} \right]_{x^{(n)}, y^{(n)}} + \dots = \frac{dy}{dx} \Big|_{x^{(n)}, y^{(n)}} + \frac{h}{2} \frac{d^2 y}{dx^2} \Big|_{x^{(n)}, y^{(n)}} + \dots \quad (6)$$

- Logo:

$$y^{(n+1)} = y(x^{(n)}) + h \frac{dy}{dx} \Big|_{x^{(n)}, y^{(n)}} + \frac{h^2}{2} \frac{d^2 y}{dx^2} \Big|_{x^{(n)}, y^{(n)}} + \dots \quad (7)$$

que é a expansão de Taylor de y até ordem 2 com erro de ordem 3.

- Em geral, designa-se um método como sendo de ordem n se o erro a cada passo é de ordem $n + 1$.

Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)

- Um refinamento do método RK2 é feito pelo algoritmo RK4. Nesse método, 4 avaliações da função f devem ser realizadas. As equações para avanço de x até $x + h$ são:

$$k_1 = hf(x^{(n)}, y^{(n)})$$

$$k_2 = hf\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_2}{2}\right)$$

$$k_4 = hf(x^{(n)} + h, y^{(n)} + k_3)$$

$$y^{(n+1)} = y^{(n)} + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)

- Um refinamento do método RK2 é feito pelo algoritmo RK4. Nesse método, 4 avaliações da função f devem ser realizadas. As equações para avanço de x até $x + h$ são:

$$k_1 = hf(x^{(n)}, y^{(n)})$$

$$k_2 = hf\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_1}{2}\right)$$

$$k_3 = hf\left(x^{(n)} + \frac{h}{2}, y^{(n)} + \frac{k_2}{2}\right)$$

$$k_4 = hf(x^{(n)} + h, y^{(n)} + k_3)$$

$$y^{(n+1)} = y^{(n)} + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

- Veja então que o erro a cada passo em RK4 é de ordem 5. É possível mostrar, de maneira similar a RK2, que o RK4 simula a acurácia de uma expansão em série de Taylor até ordem 4.
- RK4 é um dos métodos mais utilizados na resolução de equações diferenciais. Acoplado a um bom algoritmo de ajuste de passo, esse método oferece boa precisão para uma grande variedade de problemas envolvendo eqs. diferenciais.
- Ele não é, obviamente, a palavra final quando se fala de eqs. diferenciais. Dentre outros métodos podemos destacar: extrapolação de Richard e preditor-corretor.
- Para os problemas abordados nesse curso, RK4 será o método mais apropriado, pela rapidez e acurácia dos resultados.

Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)

- Baixe o código `mhs_rk4.c` que implementa o método RK4 para resolver as equações do oscilador harmônico simples.
- Vamos analisá-lo com calma.

Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)

- Baixe o código `mhs_rk4.c` que implementa o método RK4 para resolver as equações do oscilador harmônico simples.
- Vamos analisá-lo com calma.
- Compile e rode esse código.

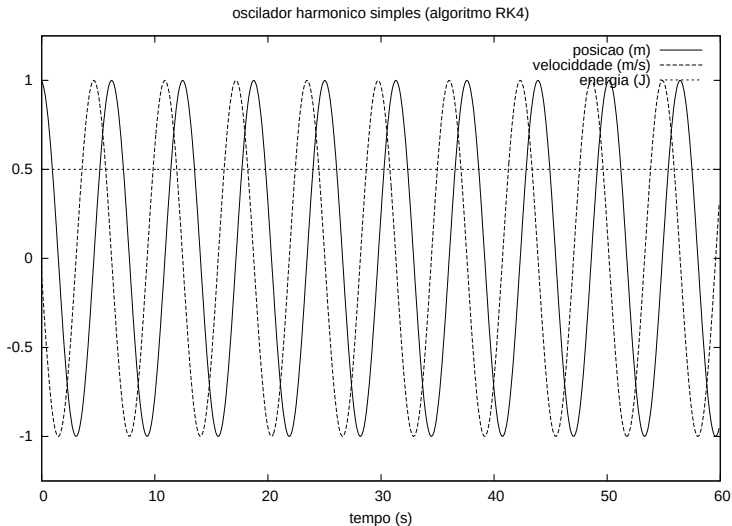
Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)

- Baixe o código `mhs_rk4.c` que implementa o método RK4 para resolver as equações do oscilador harmônico simples.
- Vamos analisá-lo com calma.
- Compile e rode esse código.
- Grafique novamente $x(t)$, $v(t)$ e $E(t) = K + U$.

Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)

- Baixe o código `mhs_rk4.c` que implementa o método RK4 para resolver as equações do oscilador harmônico simples.
- Vamos analisá-lo com calma.
- Compile e rode esse código.
- Grafique novamente $x(t)$, $v(t)$ e $E(t) = K + U$.
- grafique o espaço de fase $v(t) \times x(t)$.

Equações Diferenciais: Runge-Kutta de ordem 4 (RK4)



PARA FAZER EM SALA DE AULA

- Modifique o programa mhs_rk4.c para integrar as equações de movimento de um oscilador harmônico amortecido por uma força proporcional à velocidade $F_a = -bv$, com $b > 0$. Aplicando-se a 2ª lei de Newton para esse sistema, chega-se à seguinte equação diferencial de segunda ordem:

$$\frac{d^2x}{dt^2} + \frac{b}{m} \frac{dx}{dt} + \frac{k}{m}x = 0, \quad (8)$$

onde k é a constante elástica da mola e m a massa presa a ela.

- Grafique no gnuplot a posição $x(t)$, $v(t)$ e a energia mecânica $E(t) = K(t) + U(t)$ do oscilador para a situação $b = 0.2$, $k = 2$. e $m = 0.5$.
- Grafique o espaço de fase do sistema $v(t) \times x(t)$.
- mantendo k e m fixos nos valores acima, mude o parâmetro b para 2 e veja o acontece com a solução.
- mantendo k e m fixos nos valores acima, mude o parâmetro b para 3 e veja o acontece com a solução.

Variáveis externas

- Vimos que variáveis declaradas no interior de funções vivem somente no corpo dessa. Ao ser invocada, a sequência de instruções no interior da função são executadas. Após a execução da instrução de retorno ou da última instrução de funções do tipo void, o fluxo de instrução volta à unidade de origem e todas as variáveis criadas no interior da função são destruídas.
- Como não vestígio de tais variáveis locais entre uma chamada e outra da função, variáveis locais, também conhecidas como *automáticas* devem sempre ser inicializadas no corpo da função, pois não guardam informação de chamadas anteriores.
- Uma maneira de tornar certas variáveis visíveis a mais de uma função é declará-las como sendo do tipo *external*. Esse tipo de variável não existe dentro de um corpo de função em específico, de forma que não são destruídas por chamadas de funções que porventura venham a acessá-las.
- Variáveis externas podem então ser utilizadas para comunicar dados entre várias funções.
- Variáveis externas devem ser declaradas **UMA ÚNICA VEZ**, fora de qualquer função. Além disso, toda função que deseja acessá-las deve também declará-las com auxílio da palavra-chave *extern*.

Variáveis externas: exemplo

```
#include <stdio.h>
#define MAXLINE 1000 /* maximum input line size */
int max; /* maximum length seen so far */
char line[MAXLINE]; /* current input line */
char longest[MAXLINE]; /* longest line saved here */
int getline(void);
void copy(void);
/* print longest input line; specialized version */
int main()
{
    int len;
    extern int max;
    extern char longest[];
    max = 0;
    while ((len = getline()) > 0)
    {
        if (len > max) {
            max = len;
            copy();
        }
        if (max > 0) /* there was a line */
            printf("%s", longest);
        return 0;
    }
}
```

Variáveis externas e Header Files

Num arquivo com extensão .h (header.h por exemplo) coloca-se:

```
#include <stdio.h>
#define MAXLINE 1000 /* maximum input line size */
int max; /* maximum length seen so far */
char line[MAXLINE]; /* current input line */
char longest[MAXLINE]; /* longest line saved here */
int getline(void);
void copy(void);
```

Variáveis externas e Header Files

Num arquivo com extensão .h (header.h por exemplo) coloca-se:

```
#include <stdio.h>
#define MAXLINE 1000 /* maximum input line size */
int max; /* maximum length seen so far */
char line[MAXLINE]; /* current input line */
char longest[MAXLINE]; /* longest line saved here */
int getline(void);
void copy(void);
```

No arquivo .c coloca-se então uma diretiva do tipo #include:

```
#include "header.h"
/* print longest input line; specialized version */
int main()
{
    int len;
    extern int max;
    extern char longest[];
    max = 0;
    while ((len = getline()) > 0)
        if (len > max) {
            max = len;
            copy();
        }
    if (max > 0) /* there was a line */
        printf("%s", longest);
    return 0;
}
```