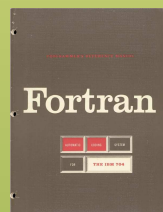


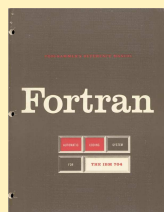
# $\mu$ Curso de Computação Científica

Prof. Werner K. Sauter

7 de outubro de 2016



# Introdução



# Sumário

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

**Introdução**

**Usando o computador**

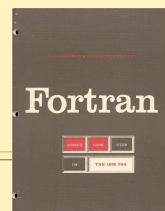
**Dois problemas práticos!**

**Resolvendo com Fortran 90**

**O utilitário Make**

**Breve introdução ao xmgrace**

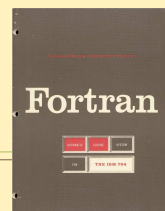
**Considerações Finais**



# Resumindo...

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

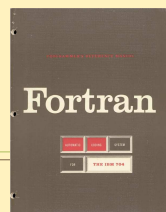
- Introdução muito breve ao tema;
- Usar os programas sugeridos;
- Mexer nos códigos apresentados;
- Pergunte para o(s) “*gurus*”



# Resumindo...

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

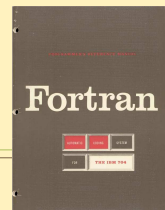
- Introdução muito breve ao tema;
- Usar os programas sugeridos;
- Mexer nos códigos apresentados;
- Pergunte para o(s) “*gurus*”



# Resumindo...

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

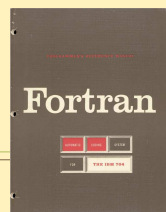
- Introdução muito breve ao tema;
- Usar os programas sugeridos;
- Mexer nos códigos apresentados;
- Prergunte para o(s) “*gurus*”



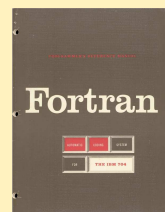
# Resumindo...

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Introdução muito breve ao tema;
- Usar os programas sugeridos;
- Mexer nos códigos apresentados;
- Prergunte para o(s) “*gurus*”



# Usando o computador

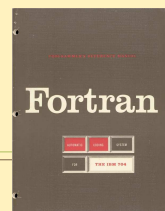




# O que usar?

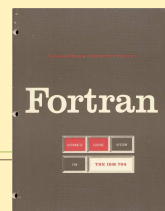
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...
  
- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



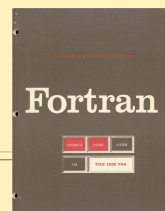
# O que usar?

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...
  
- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



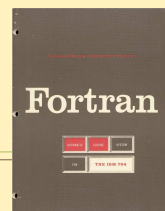
# O que usar?

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...
  
- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



# O que usar?

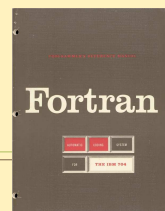
- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...
  
- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



# O que usar?

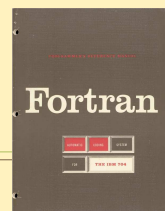
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...
  
- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



# O que usar?

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...
  
- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



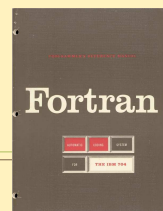
# O que usar?

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...

## Usando Linux:

- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...

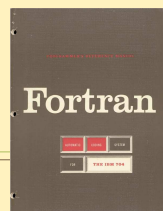


# O que usar?

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...

## Usando Linux:

- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



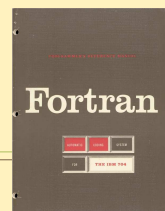


# O que usar?

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...

## Usando Linux:

- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...

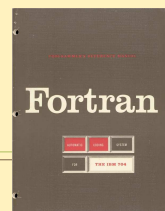


# O que usar?

- Sistema operacional: Windows(\$)(veja Cygwin) vs. **Linux**(Fedora/CentOS) vs. Mac OS
- Programa vs. CAS (Mathematica, Maple, MathLab, (wx)Maxima), FOAM, REDUCE, OCTAVE,...
- Fortran 77 vs. Fortran 90/95/2000/... vs. C/C++ vs. Python
- gfortran(gcc) vs. ifort(\$)
- make
- xmgrace vs. QtiPlot vs. kmplot vs. scidavis vs. veusz vs. ...

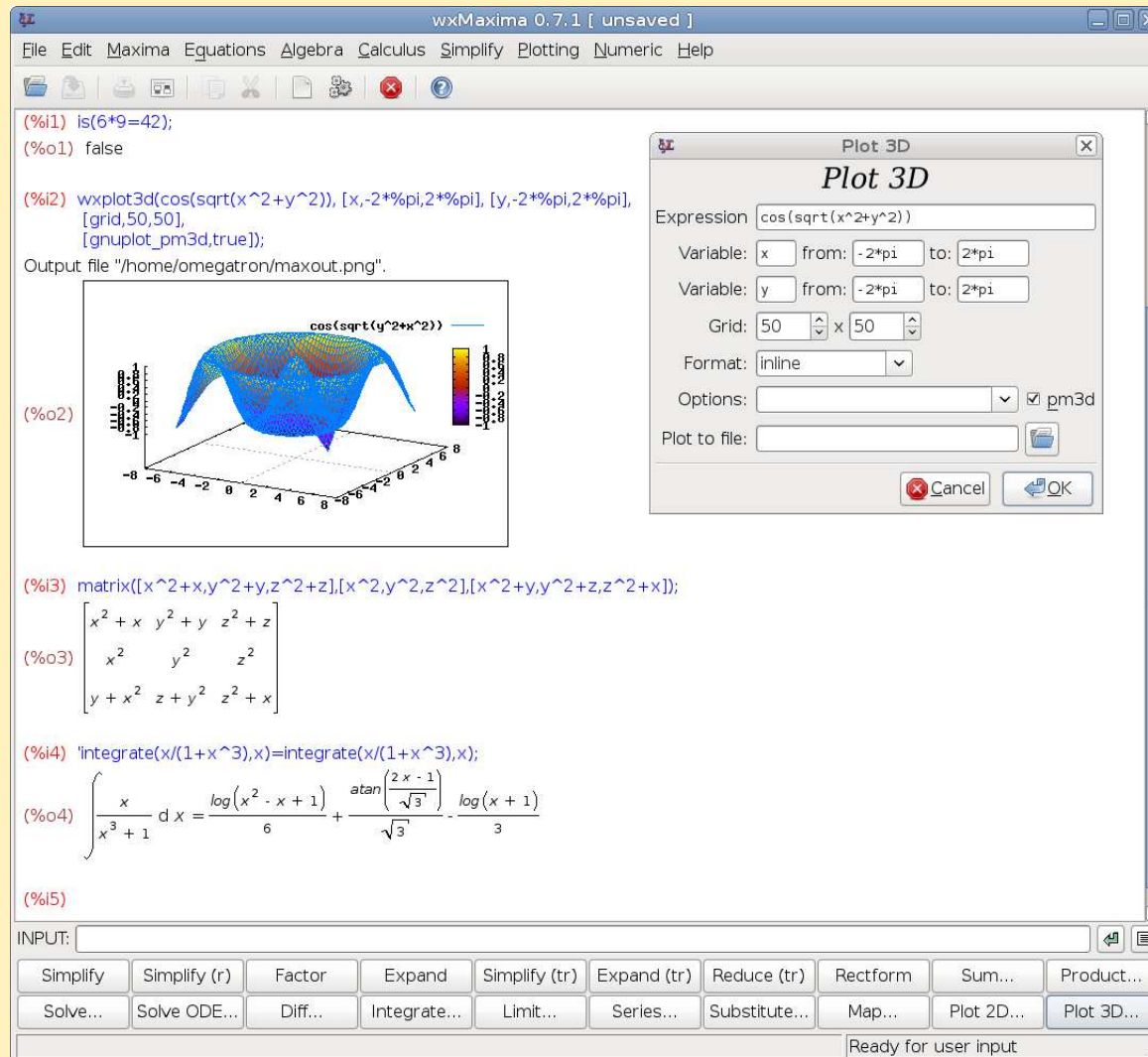
## Usando Linux:

- Terminal (tmux, byobu, mc, ...)
- Editor de texto (Kate, (X)Emacs, ...)
- IDE : Geany, Photran, Force project, Kdevelop, Code::Blocks...



# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais



The screenshot shows the wxMaxima 0.7.1 interface. The main window displays the following content:

```
(%i1) is(6*9=42);
(%o1) false

(%i2) wxplot3d(cos(sqrt(x^2+y^2)), [x,-2*%pi,2*%pi], [y,-2*%pi,2*%pi],
[grid,50,50],
[gnuplot_pm3d,true]);
Output file "/home/omegatron/maxout.png".
```

A 3D plot of the function  $z = \cos(\sqrt{x^2 + y^2})$  is shown, with axes ranging from -8 to 8. The plot is a surface with a central peak and a surrounding valley. A color scale on the right indicates values from -1.000000 to 1.000000.

The Plot 3D dialog box is open, showing the following settings:

- Expression:  $\cos(\sqrt{x^2+y^2})$
- Variable: x from:  $-2\pi$  to:  $2\pi$
- Variable: y from:  $-2\pi$  to:  $2\pi$
- Grid: 50 x 50
- Format: inline
- Options:  pm3d
- Plot to file:

Below the plot, the following commands and results are shown:

```
(%i3) matrix([x^2+x,y^2+y,z^2+z],[x^2,y^2,z^2],[x^2+y,y^2+z,z^2+x]);
(%o3) 
$$\begin{bmatrix} x^2 + x & y^2 + y & z^2 + z \\ x^2 & y^2 & z^2 \\ y + x^2 & z + y^2 & z^2 + x \end{bmatrix}$$

```

```
(%i4) 'integrate(x/(1+x^3),x)=integrate(x/(1+x^3),x);
(%o4) 
$$\int \frac{x}{x^3 + 1} dx = \frac{\log(x^2 - x + 1)}{6} + \frac{\operatorname{atan}\left(\frac{2x - 1}{\sqrt{3}}\right)}{\sqrt{3}} - \frac{\log(x + 1)}{3}$$

```

The interface also shows an INPUT field and a toolbar with buttons for Simplify, Factor, Expand, Integrate, etc.

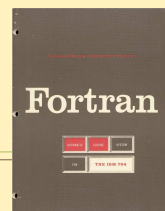
Fortran

Um CAS é um programa de computador que facilita o cálculo na matemática

# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

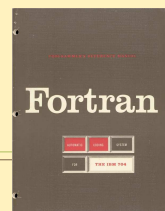
- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um progrma para fazer cálculos numéricos.



# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

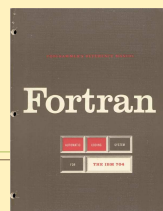
- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um progrma para fazer cálculos numéricos.



# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

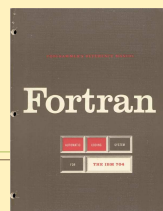
- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um programa para fazer cálculos numéricos.



# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

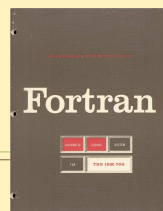
- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um progrma para fazer cálculos numéricos.



# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um progrma para fazer cálculos numéricos.

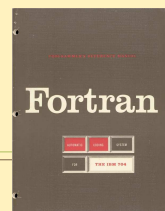




# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

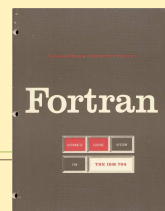
- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um programa para fazer cálculos numéricos.



# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

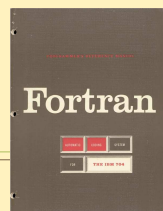
- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um progrma para fazer cálculos numéricos.



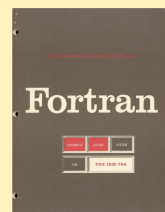
# Sistema de Computação Algébrica (CAS)

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Um CAS é um programa de computador que facilita o cálculo na matemática simbólica.
  1. precisão aritmética arbitrária
  2. manipulação simbólica: simplificar, diferenciar e integrar funções e resolver equações
  3. facilidades gráficas, para produzir gráficos de funções, a duas ou a três dimensões
  4. um subsistema de álgebra linear: cálculo de matrizes e resolver sistemas de equações lineares
  5. uma linguagem de programação de alto nível, permitindo aos utilizadores implementar os seus próprios algoritmos
- (wx)Maxima: gratuito, disponível para Windows, Linux, macOS, Android. (Ver <http://www.neng.usu.edu/cee/faculty/gurro/Maxima.html> e <https://web.csulb.edu/~woollett/>)
- É muito mais lento que um programa para fazer cálculos numéricos.



# Dois problemas práticos!

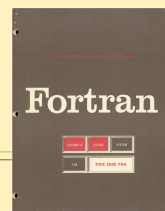


- Gráfico da derivada de uma função dada,

$$g(x) = \frac{df(x)}{dx}$$

- Saída: um arquivo com abscissa e ordenada → `xmgrace`
- Valor de uma integral definida

$$\sigma = \int_{x_{\min}}^{x_{\max}} dx F(x)$$

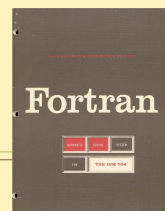


- Gráfico da derivada de uma função dada,

$$g(x) = \frac{df(x)}{dx}$$

- Saída: um arquivo com abscissa e ordenada → xmgrace
- Valor de uma integral definida

$$\sigma = \int_{x_{\min}}^{x_{\max}} dx F(x)$$

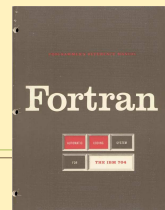


- Gráfico da derivada de uma função dada,

$$g(x) = \frac{df(x)}{dx}$$

- Saída: um arquivo com abscissa e ordenada → `xmgrace`
- Valor de uma integral definida

$$\sigma = \int_{x_{\min}}^{x_{\max}} dx F(x)$$



# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

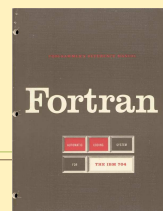
## ■ Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- É fácil fazer *algébricamente* mas *numericamente* não...
- Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

- Não é fácil fazer *algébricamente* mas *numericamente* é...
- Vamos usar rotinas “prontas” → dfridr e QUADPACK
- Existem rotinas para o gradiente: (ndl2)
- Idem para integrais múltiplas: (CUBALib)





# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## ■ Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

## ■ É fácil fazer *algébricamente* mas *numericamente* não...

## ■ Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

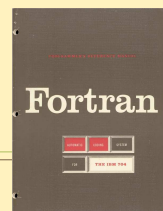
$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

## ■ Não é fácil fazer *algébricamente* mas *numericamente* é...

## ■ Vamos usar rotinas “prontas” → dfridr e QUADPACK

## ■ Existem rotinas para o gradiente: (ndl2)

## ■ Idem para integrais múltiplas: (CUBALib)



# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

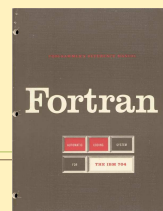
- Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- É fácil fazer *algébricamente* mas *numericamente* não...
- Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

- Não é fácil fazer *algébricamente* mas *numericamente* é...
- Vamos usar rotinas “prontas” → dfridr e QUADPACK
- Existem rotinas para o gradiente: (ndl2)
- Idem para integrais múltiplas: (CUBALib)



# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## ■ Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

## ■ É fácil fazer *algébricamente* mas *numericamente* não...

## ■ Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

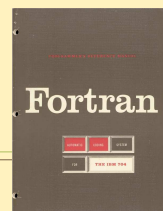
$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

## ■ Não é fácil fazer *algébricamente* mas *numericamente* é...

### ■ Vamos usar rotinas “prontas” → dfridr e QUADPACK

### ■ Existem rotinas para o gradiente: (ndl2)

### ■ Idem para integrais múltiplas: (CUBALib)



# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

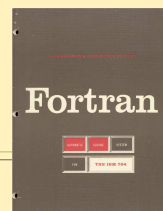
- Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- É fácil fazer *algébricamente* mas *numericamente* não...
- Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

- Não é fácil fazer *algébricamente* mas *numericamente* é...
- Vamos usar rotinas “prontas” → dfridr e QUADPACK
- Exitem rotinas para o gradiente: (nd12)
- Idem para integrais múltiplas: (CUBALib)



# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

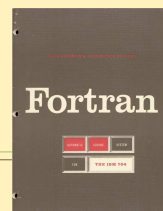
- Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

- É fácil fazer *algébricamente* mas *numericamente* não...
- Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

- Não é fácil fazer *algébricamente* mas *numericamente* é...
- Vamos usar rotinas “prontas” → dfridr e QUADPACK
- Existem rotinas para o gradiente: (nd12)
- Idem para integrais múltiplas: (CUBALib)



# Um pouco de derivada e integral numérica

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

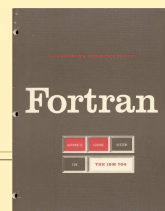
- Definição de derivada:

$$f'(x) = \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

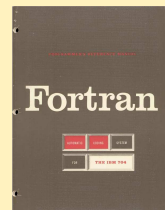
- É fácil fazer *algébricamente* mas *numericamente* não...
- Integral via Teorema Fundamental do Cálculo ou pela definição de *quadratura*

$$\int_{x_{\min}}^{x_{\max}} dx g(x) \approx \sum_{i=1} g(x_i) \Delta x$$

- Não é fácil fazer *algébricamente* mas *numericamente* é...
- Vamos usar rotinas “prontas” → dfridr e QUADPACK
- Existem rotinas para o gradiente: (ndl2)
- Idem para integrais múltiplas: (CUBALib)



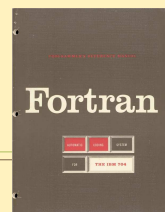
# Resolvendo com Fortran 90



# Interlúdio: objetos, bibliotecas e *linker*

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

Objeto:



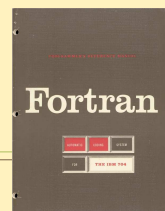


# Interlúdio: objetos, bibliotecas e *linker*

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

Objeto:

- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.

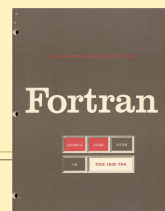


# Interlúdio: objetos, bibliotecas e *linker*

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Objeto:

- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.
- compilador: lê o código-fonte e cria um objeto binário que o SO (ainda) não capaz de executar.



# Interlúdio: objetos, bibliotecas e *linker*

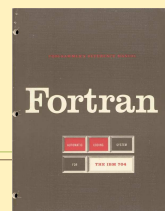
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Objeto:

- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.
- compilador: lê o código-fonte e cria um objeto binário que o SO (ainda) não capaz de executar.

## Biblioteca:

- funções de propósito geral (E/S, matemáticas, etc.) em um tipo de objetos → bibliotecas (library).



# Interlúdio: objetos, bibliotecas e *linker*

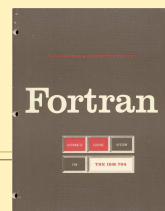
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Objeto:

- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.
- compilador: lê o código-fonte e cria um objeto binário que o SO (ainda) não capaz de executar.

## Biblioteca:

- funções de propósito geral (E/S, matemáticas, etc.) em um tipo de objetos → bibliotecas (library).
- funções específicas também podem ser bibliotecas optidas/criadas pelo usuário (LHAPDF, CubaLib, SLATEC, GSL,...)



# Interlúdio: objetos, bibliotecas e *linker*

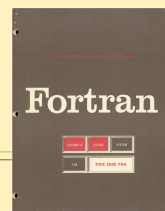
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Objeto:

- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.
- compilador: lê o código-fonte e cria um objeto binário que o SO (ainda) não capaz de executar.

## Biblioteca:

- funções de propósito geral (E/S, matemáticas, etc.) em um tipo de objetos → bibliotecas (library).
- funções específicas também podem ser bibliotecas optidas/criadas pelo usuário (LHAPDF, CubaLib, SLATEC, GSL,...)
- podem ser estáticas ou dinâmicas.



# Interlúdio: objetos, bibliotecas e *linker*

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Objeto:

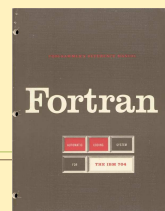
- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.
- compilador: lê o código-fonte e cria um objeto binário que o SO (ainda) não capaz de executar.

## Biblioteca:

- funções de propósito geral (E/S, matemáticas, etc.) em um tipo de objetos → bibliotecas (library).
- funções específicas também podem ser bibliotecas optidas/criadas pelo usuário (LHAPDF, CubaLib, SLATEC, GSL,...)
- podem ser estáticas ou dinâmicas.

## *Linker*:

- o linker (ligador) une o objeto criado pelo compilador e liga (link) com as bibliotecas.



# Interlúdio: objetos, bibliotecas e *linker*

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Objeto:

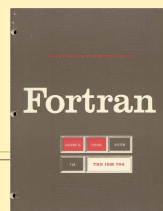
- interpretador: lê o código-fonte e o executa. Ex.: LOGO, Java, Python.
- compilador: lê o código-fonte e cria um objeto binário que o SO (ainda) não capaz de executar.

## Biblioteca:

- funções de propósito geral (E/S, matemáticas, etc.) em um tipo de objetos → bibliotecas (library).
- funções específicas também podem ser bibliotecas optidas/criadas pelo usuário (LHAPDF, CubaLib, SLATEC, GSL,...)
- podem ser estáticas ou dinâmicas.

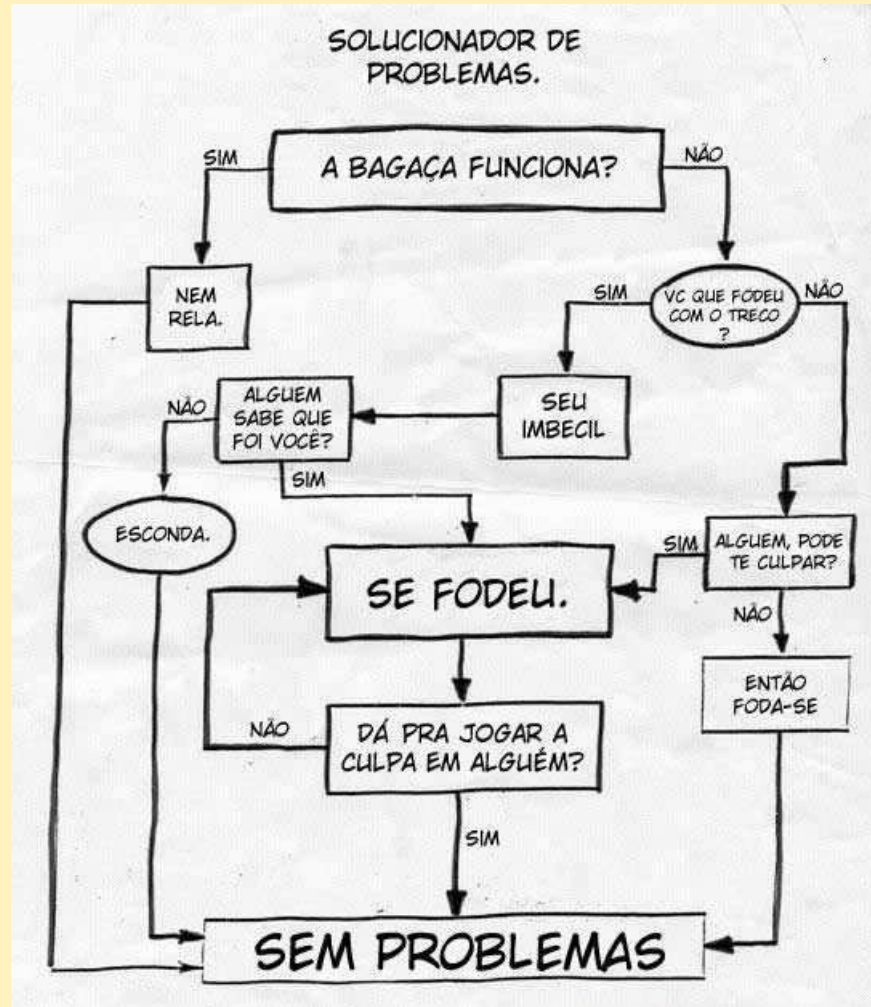
## *Linker*:

- o linker (ligador) une o objeto criado pelo compilador e liga (link) com as bibliotecas.
- gera um binário que o SO é capaz de executar.



# Boas práticas

- fluxograma;



Fortran

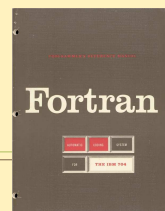




# Boas práticas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

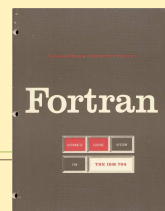
- fluxograma;
- quebrar em problemas menores;



# Boas práticas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

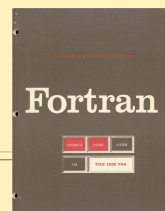
- fluxograma;
- quebrar em problemas menores;
- reutilização de código (vamos ver isso em Makefile);



# Boas práticas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

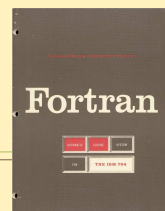
- fluxograma;
- quebrar em problemas menores;
- reutilização de código (vamos ver isso em Makefile);
- esquematização;



# Boas práticas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

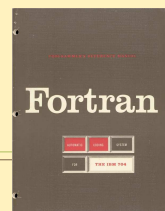
- fluxograma;
- quebrar em problemas menores;
- reutilização de código (vamos ver isso em Makefile);
- esquematização;
- conferências intermediárias;



# Boas práticas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

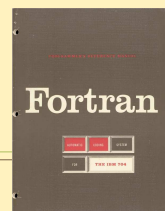
- fluxograma;
- quebrar em problemas menores;
- reutilização de código (vamos ver isso em Makefile);
- esquematização;
- conferências intermediárias;
- comentar muito o código fonte.



# Boas práticas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- fluxograma;
- quebrar em problemas menores;
- reutilização de código (vamos ver isso em Makefile);
- esquematização;
- conferências intermediárias;
- comentar muito o código fonte.
- salvar sempre e ter um (ou mais...) *backups*!!



# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

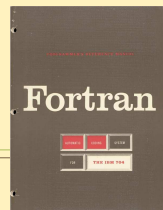
## ■ Aonde consigo as rotinas?

## ■ Existe um grande número de rotinas intrínsecas. Veja em

<https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>

## ■ R.: Na internet, veja nas referências!

## ■ Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



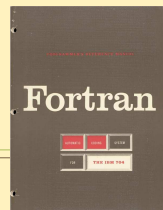
# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

## ■ Aonde consigo as rotinas?

- Existe um grande número de rotinas intrínsecas. Veja em <https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>
- R.: Na internet, veja nas referências!
- Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...





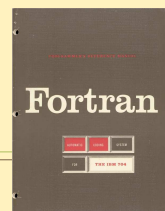
# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

## ■ Aonde consigo as rotinas?

- Existe um grande número de rotinas intrínsecas. Veja em <https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>
- R.: Na internet, veja nas referências!
- Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

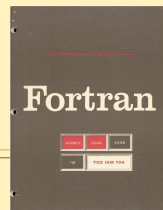
## ■ Aonde consigo as rotinas?

## ■ Existe um grande número de rotinas intrínsecas. Veja em

<https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>

## ■ R.: Na internet, veja nas referências!

## ■ Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

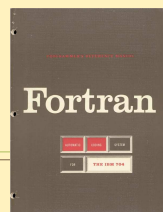
## ■ Aonde consigo as rotinas?

## ■ Existe um grande número de rotinas intrínsecas. Veja em

<https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>

## ■ R.: Na internet, veja nas referências!

## ■ Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



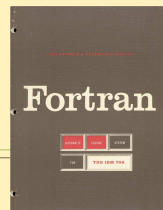
# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

## ■ Aonde consigo as rotinas?

- Existe um grande número de rotinas intrínsecas. Veja em <https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>
- R.: Na internet, veja nas referências!
- Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



# Esquema para o problema

- Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

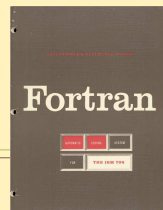
- Aonde consigo as rotinas?

- Existe um grande número de rotinas intrínsecas. Veja em

<https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>

- R.: Na internet, veja nas referências!

- Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



# Esquema para o problema

## ■ Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

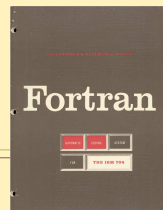
## ■ Aonde consigo as rotinas?

## ■ Existe um grande número de rotinas intrínsecas. Veja em

<https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>

## ■ R.: Na internet, veja nas referências!

- Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...



# Esquema para o problema

- Montando uma seqüência de passos:

1. Definição das variáveis
2. Entrada de parametros
3. Laço para o intervalo de  $x$
4. Cálculo da derivada/integral
5. Salvar o resultado em um arquivo / mostrar na tela

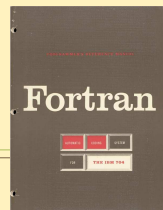
- Aonde consigo as rotinas?

- Existe um grande número de rotinas intrínsecas. Veja em

<https://gcc.gnu.org/onlinedocs/gfortran/Intrinsic-Procedures.html>

- R.: Na internet, veja nas referências!

- Rotinas numéricas para: solução de sistemas lineares, matrizes, funções especiais para diferentes argumentos (Bessel, Legendre, Hermite, Chebyshev, Hipergeométrica, Exponencial, Gama, Poligama, Dilogaritmo,...), solução para equações diferenciais (Runge-Kutta, Numerov), interpolação de pontos, ajuste de curvas,...

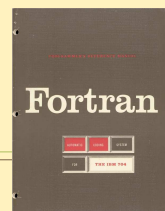


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`



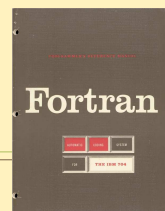


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

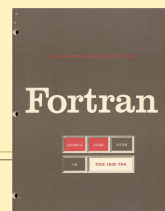


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

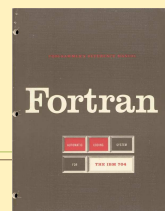


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

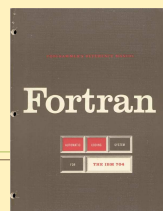


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

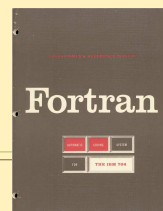


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

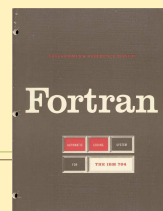


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

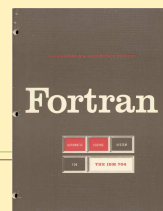


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

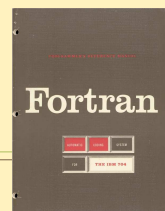


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`



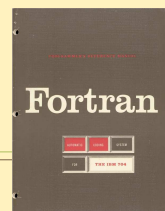


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
  - Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
  - Parâmetros (PARAMETER), vetores (DIMENSION)
  - funções e subrotinas: CONTAINS e INTERFACE
  - laços: DO ... END DO
  - condicionais: IF THEN (ELSE) END IF
  - funções matemáticas
  - bem básico de entrada/saída
  - usando MODULE para passar parâmetros
  - usando uma rotina de cálculo pronta (*montador de Lego*)
- 
- Compilando o programa: `$ ifort [nome do programa].f90`
  - Rodando o programa: `$ ./a.out`

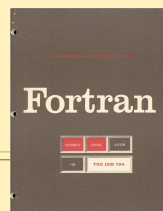


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`

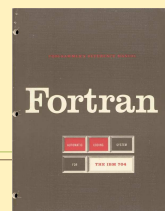


# Um programa simples em Fortran 90

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

## Escrevendo o programa:

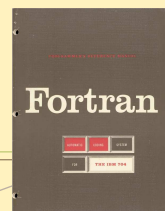
- PROGRAM, formato, quebras de linha, comentários
- Definição de variáveis: REAL, DOUBLE PRECISION, INTEGER
- Parâmetros (PARAMETER), vetores (DIMENSION)
- funções e subrotinas: CONTAINS e INTERFACE
- laços: DO ... END DO
- condicionais: IF THEN (ELSE) END IF
- funções matemáticas
- bem básico de entrada/saída
- usando MODULE para passar parâmetros
- usando uma rotina de cálculo pronta (*montador de Lego*)
  
- Compilando o programa: `$ ifort [nome do programa].f90`
- Rodando o programa: `$ ./a.out`



# E se der errado?

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

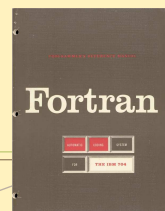
- Erros de compilação;
- Erros de execução: “debuggers” como ddd, idb,...
- Erros numéricos: NAN
- Erros nos resultados: numéricos & físicos
- Não achou? Aplique o “efeito gaveta”



# E se der errado?

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

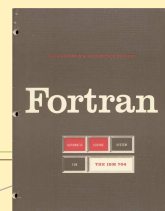
- Erros de compilação;
- Erros de execução: “debuggers” como ddd, idb,...
- Erros numéricos: NAN
- Erros nos resultados: numéricos & físicos
- Não achou? Aplique o “efeito gaveta”



# E se der errado?

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

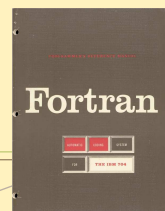
- Erros de compilação;
- Erros de execução: “debuggers” como ddd, idb,...
- Erros numéricos: NAN
- Erros nos resultados: numéricos & físicos
- Não achou? Aplique o “efeito gaveta”



# E se der errado?

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

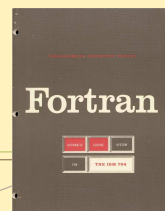
- Erros de compilação;
- Erros de execução: “debuggers” como ddd, idb,...
- Erros numéricos: NAN
- Erros nos resultados: numéricos & físicos
- Não achou? Aplique o “efeito gaveta”



# E se der errado?

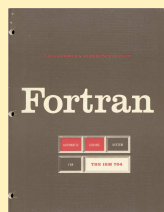
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Erros de compilação;
- Erros de execução: “debuggers” como ddd, idb,...
- Erros numéricos: NAN
- Erros nos resultados: numéricos & físicos
- Não achou? Aplique o “efeito gaveta”





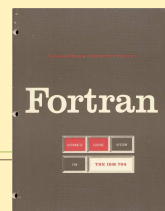
# O utilitário Make



# Introdução

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

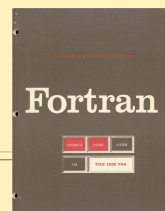
- Utilitário para manter programas
- gerente de comandos de acordo com dependências pré-estabelecidas
- útil para qualquer linguagem



# Introdução

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

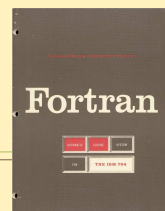
- Utilitário para manter programas
- gerente de comandos de acordo com dependências pré-estabelecidas
- útil para qualquer linguagem



# Introdução

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Utilitário para manter programas
- gerente de comandos de acordo com dependências pré-estabelecidas
- útil para qualquer linguagem



# Um makefile simples

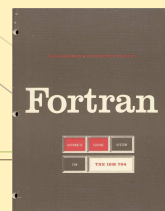
Exemplo: um programa em um único arquivo →

1. gera-se o objeto (.o): `$ f90 -c prog.f90`
2. link e executavel: `$ f90 -o prog.exe prog.o`
3. tudo junto: `$ f90 prog.f90`

Se temos um grande número de programas separados e/ou bibliotecas → longa linha de comando:

```
f90 -o prog.exe -lm -L/opt/lhapdf/lib/ -lcuba -lLHAPDF prog1.f90  
prog2.f90 ...
```

O make gera os objetos, e monta os executáveis *à la carte*. Quando um arquivo fonte é modificado, só o objeto correspondente é construído de novo.



# Um makefile simples: o exemplo

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

Em um arquivo chamado Makefile:

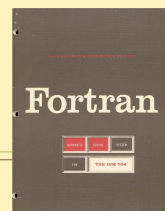
```
# Makefile simples
programa: programa1.o programa2.o
    f90 -o prog.exe programa1.o programa2.o

programa1.o : programa1.f90
    f90 -c programa1.f90

programa2.o : programa2.f90
    f90 -c programa2.f90
```

Linha de comando:

```
$ make programa
$ ./prog.exe
```



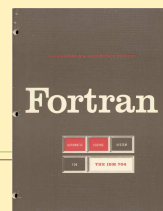
# Macros

Podemos definir *macros* ou definições no início do Makefile:

```
nome = texto qualquer
e chamá-la usando ${nome} :
# Makefile simples
F90C = ifort
F90FLAGS = -O3 -axW
LDFLAGS = -L/opt/lhapdf/lib/
LIBS = -lLHAPDF -lm
EXE = programa
OBJ = programa1.o programa2.o
${EXE}: ${OBJ}
    ${F90C} ${F90FLAGS} ${LDFLAGS} ${LIBS} -o ${EXE} ${OBJ}

programa1.o : programa1.f90
    ${F90C} ${F90FLAGS} -c programa1.f90

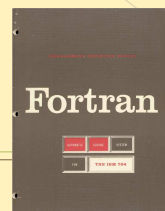
programa2.o : programa2.f90
    ${F90C} ${F90FLAGS} -c programa2.f90
```



# Outras coisas

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

- Alvos: `clean`, `run`;
- Macros dinâmicas;
- Regras de sufixo;





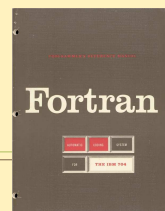
# Exemplo

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

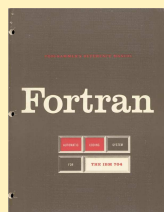
```
# Central resonance diffractive production (Fortran 90)
# definico de macros
F90C = ifort
F90FLAGS = -O3 -axW
FC = ${F90C}
FFLAGS = ${F90FLAGS}
LDFLAGS = -L/opt/lhapdf/lib/
LIBS = -lLHAPDF -lm
OBJG = defaults.o varcom.o parametros.o rotints.o dfridr.o alphas.o dgnit.o nDS.o centralgluon.o
EXEG = gluon.exe
NCLVL = 10
.SUFFIXES: .f90 .f .o
.KEEP_STATE:
#
all: ${EXEG}
#gluon
${EXEG}: ${OBJG}
        ${F90C} -o ${EXEG} ${F90FLAGS} ${LDFLAGS} ${OBJG} ${LIBS}

rung:
        time nice -n ${NCLVL} ./${EXEG}
#####
%.o: %.f90
        ${F90C} ${F90FLAGS} -c -o $@ $<
%.o: %.f
        ${FC} ${FFLAGS} -c -o $@ $<
#####
# limpa executaveis, backups, objetos e modulos
clean:
        -rm -vf *.o *.mod *.*~ *~ *.exe *.sh.o* a.out

veryclean:
        -rm -vf *.exe *.*~ *~ *.o a.out *.mod
```



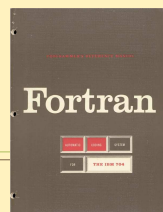
# Breve introdução ao xmgrace



# Introdução

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao `xmgrace` Considerações Finais

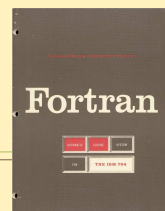
- Programa para fazer gráficos
- opções: QtiPlot, LabPlot, SciDavis,...
- Como invocar: `$ xmgrace`



# Coisas simples

Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

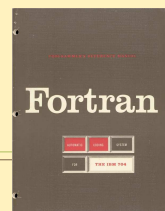
1. abrindo arquivo
2. definindo eixos
3. colocando rótulos: sintaxe
  - `\x`: símbolos
  - `\O`: fonte original
  - `\s`: subescrito
  - `\S`: superescrito
  - `\N`: texto normal
  - `\o`: sobrelinha
  - `\O`: fim da sobrelinha
  - `\u`: sublinhado
  - `\U`: fim do sublinhado



# Coisas simples

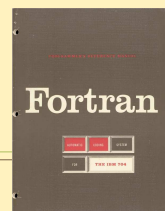
Introdução Usando o computador Dois problemas práticos! Resolvendo com Fortran 90 O utilitário Make Breve introdução ao xmgrace Considerações Finais

1. abrindo arquivo
2. definindo eixos
3. colocando rótulos: sintaxe
  - `\x`: símbolos
  - `\O`: fonte original
  - `\s`: subescrito
  - `\S`: superescrito
  - `\N`: texto normal
  - `\o`: sobrelinha
  - `\O`: fim da sobrelinha
  - `\u`: sublinhado
  - `\U`: fim do sublinhado



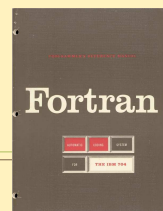
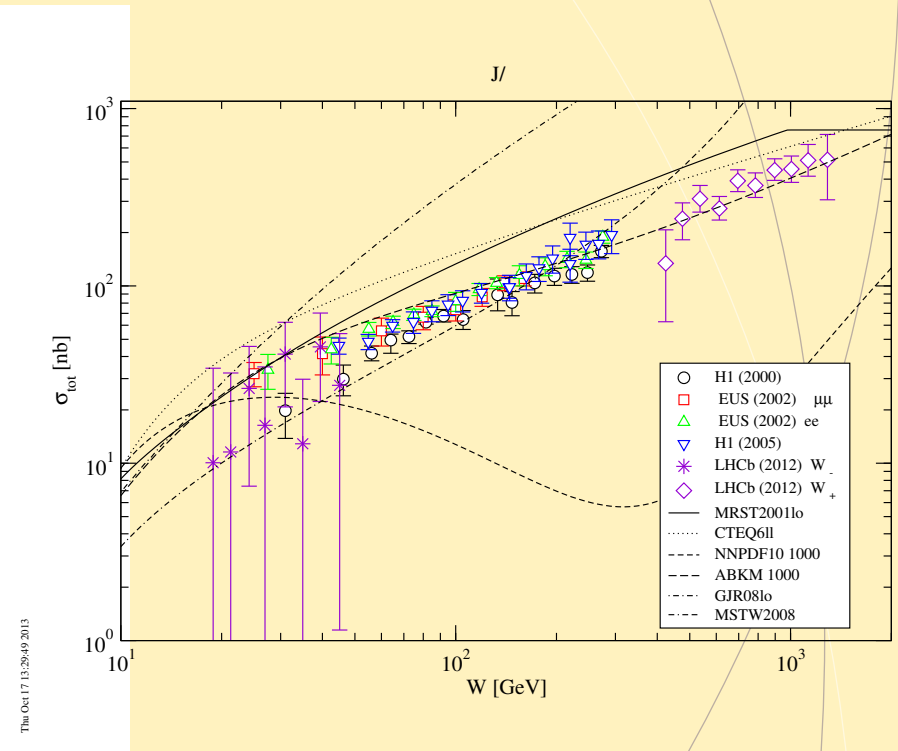
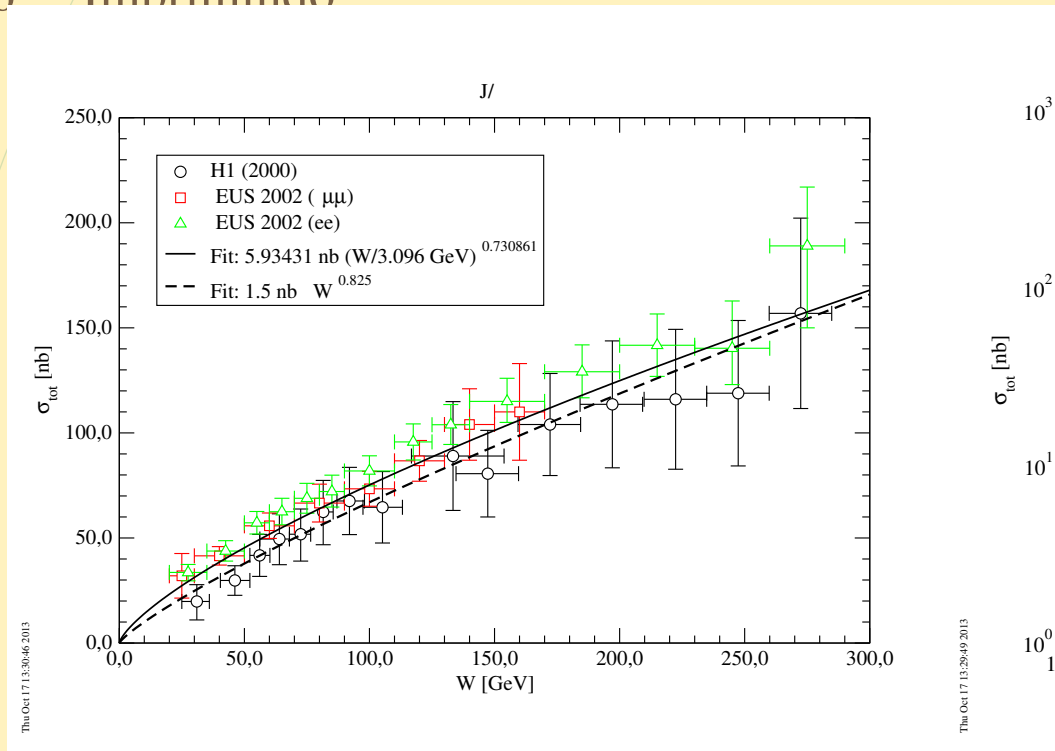
# Coisas simples

1. abrindo arquivo
2. definindo eixos
3. colocando rótulos: sintaxe
  - `\x`: símbolos
  - `\O`: fonte original
  - `\s`: subescrito
  - `\S`: superescrito
  - `\N`: texto normal
  - `\o`: sobrelinha
  - `\O`: fim da sobrelinha
  - `\u`: sublinhado
  - `\U`: fim do sublinhado

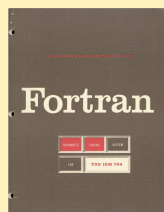


# Mais coisas simples

4. Legendas
5. Títulos
6. Imprimindo

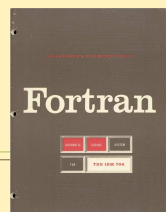


# Considerações Finais

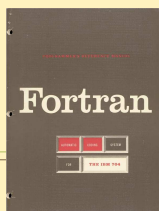




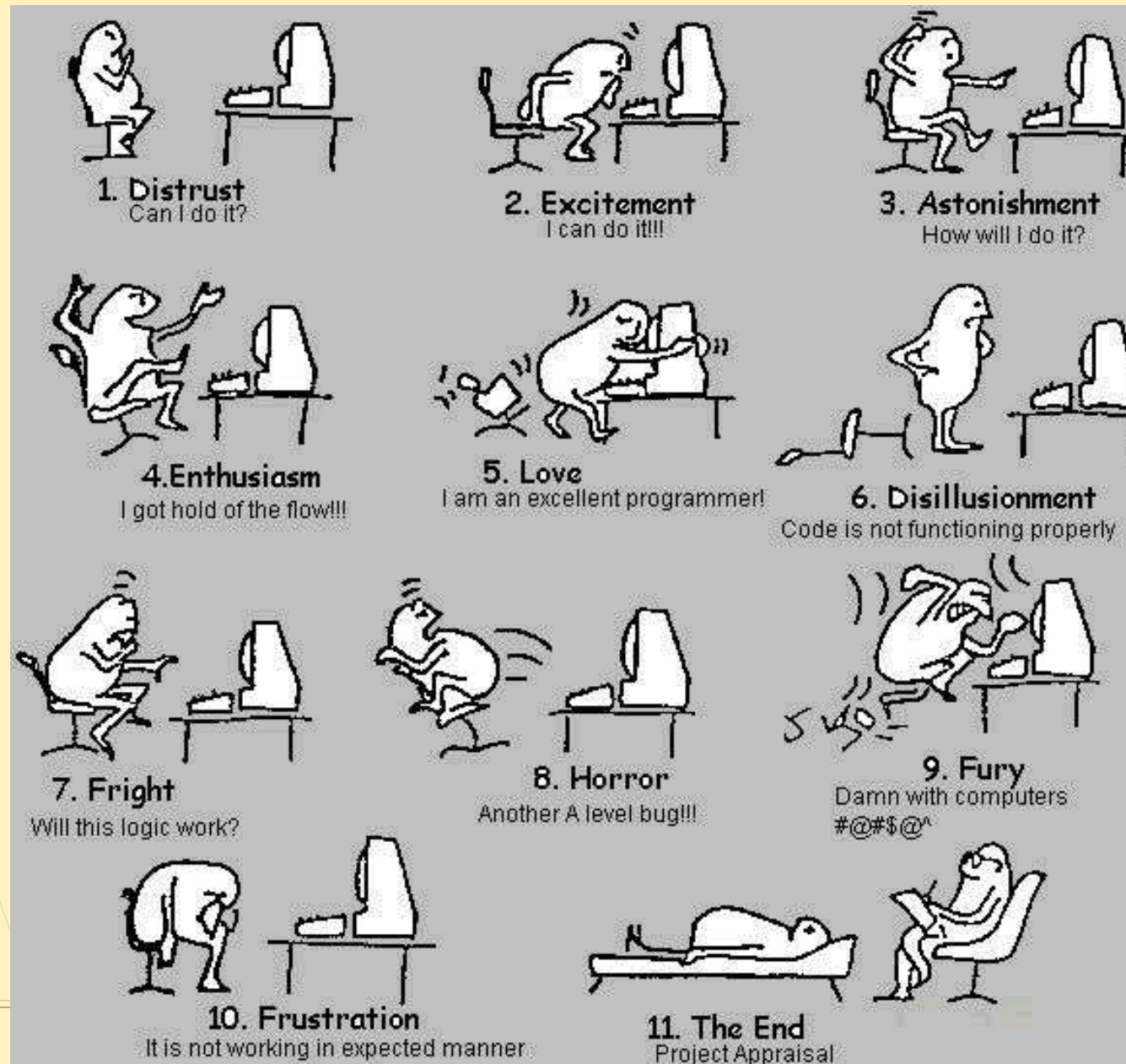
## Mão na massa!




## Permaneça calmo...



# Dicas finais



# Referências

- [1] W. Press *et al.*, “*Numerical Recipes in Fortran*”, Cambridge.
- [2] R. Gaelzer, Introdução ao Fortran 90/95 (apostila).  
[http://minerva.ufpel.edu.br/~rudi/grad/ModComp/Apostila/Apostila\\_links.pdf](http://minerva.ufpel.edu.br/~rudi/grad/ModComp/Apostila/Apostila_links.pdf)
- [3] R. Dias da Cunha. Introdução à linguagem de programação Fortran 90, EDUFRGS.
- [4] F90 Course Development, Liverpool University.  
<http://www.liv.ac.uk/HPC/F90page.html>.
- [5] Durham University. <http://www.dur.ac.uk/cis/docs/category/?id=26>.
- [6] J. Burkardt. <http://people.sc.fsu.edu/~jburkardt/>
- [7] Computer Physics Communications Program Library.  
<http://cpc.cs.qub.ac.uk/>
- [8] ITS Durham University Guide 176 “*Using the make command to maintain your software*”; Red Hat Linux 5: Dicas e ; Dominando o Linux.