ESCOLA POLITÉCNICA DA
UNIVERSIDADE DE SÃO PAULO
ENGENHARIA CIVIL
DEPARTAMENTO DE ENGENHARIA DE
ESTRUTURAS E GEOTÉCNICA

Notas de aula básicas de:

Programação em Linguagem Fortran 77/90

Autor: Prof. Valério S. Almeida

São Paulo
2015

RESUMO

ALMEIDA, V.S. (2015). Notas de aula do curso de: Programação em Linguagem Fortran 77/90*. São Paulo*. Departamento de Engenharia de Estruturas e Geotécnicas, Universidade de São Paulo.

Estas notas de aula têm o objetivo de facilitar o aprendizado da Linguagem Estruturada Fortran 77/90 para os alunos que realizam iniciação cientifica e não tem contato com essa linguagem. Ela traz um pequeno resumo das principais sintaxes e das lógicas de programação necessárias para a execução do código escrito em Fortran. Seu maior objetivo é oferecer exercícios de programação de grau de dificuldade progressivo e ligados as áreas de aplicações empregadas pela disciplina de referência deste curso: Análise Matricial de Estruturas.

SUMÁRIO

1- INTRODUÇÃO: POR QUE FORTRAN?

Muitos me perguntam por que ainda trabalhar com a linguagem Fortran, uma vez que temos hoje em dia muitas outras linguagens mais atraentes, eficientes, adaptadas para os novos processadores e com facilidades de linguagem orientada a objeto com funções imediatas e diversas.

Bem, realmente não tem como competir com essas linguagens nos dias de hoje. Se um programador pedisse minha opinião para desenvolvimento de um aplicativo novo, eu o sugeriria o uso da linguagem C++/C#.

Entretanto, nunca devemos esquecer que a computação cientifica não começou há 10 ou 20 anos, mas sim, têm-se notícias de algoritmos da década de 50 que foram desenvolvidos, por exemplo, na área da mecânica aeronáutica, como algoritmos extremamente eficientes para resolução de sistemas lineares de matrizes esparsas ou não, ou no cálculo de estruturas com o uso do método dos elementos finitos, ou as bibliotecas otimizadas BLAS (Basic Linear Algebra Subprograms) nível I, II e III, que são próprias para a solução numérica de tópicos da álgebra linear.

Esses algoritmos em quase sua totalidade foram desenvolvidos em linguagem Fortran e estão espalhados pelos sites de repositórios, em livros, artigos e nos grupos de pesquisa por todo o mundo.

Assim, o que fazer com esses códigos escritos e já testados pela comunidade cientifica há décadas? Transpô-los para linguagens mais modernas? Se tentar fazer isso, correm-se grandes riscos de cometer erros de sintaxe ou mesmo numérico. Além disso, é valido pontuar que esses algoritmos possuem algumas centenas de milhares de linhas, o que exponência a chance de algo der errado nessa conversão.

Desse modo, não é adequado seguir por esse caminho, o que sugiro para os novos desenvolvedores é aproveitar esses códigos escritos nessa linguagem obsoleta, mas que funciona muito bem. De modo que podem ser montados pelos novos compiladores Fortran, criados em 2014, em forma de DLL que possam ser chamados pelas linguagens mais adequadas as necessidades de hoje.

Seguindo essa ideia, isso demonstra que o Fortran ainda não se extinguiu, pois sempre será necessário dar manutenção nesses códigos antigos, ou mesmo entender ou alterar certas rotinas para facilitar seu uso.

Nesse sentido, essas notas de aula foram desenvolvidas para ajudar o aluno nos primeiros passos nessa linguagem. Ela não tem o intuito de apresentar os conceitos teóricos de programação, ou apresentar uma explicação mais detalhada dos comandos, nem tampouco descrever as centenas de funções internas que o Fortran permite usar. Mas sim, apresentar de forma expedita os principais comandos para o desenvolvimento de um pequeno grupo de problemas da álgebra linear e para a criação de um programa de análise matricial de estruturas reticuladas.

O melhor dessas notas de aula são os exemplos resolvidos, os quais contribuem de forma direta para o entendimento do uso do comando no contexto do programa.

2. CRIANDO PROJETO NO IDE VISUAL STUDIO 2013

Nesse item, são indicados nas figuras de 1 a 10 como criar e construir um projeto usando um compilador fortran que está associado ao IDE do Visual Studio 2013.

A sequência seria ir ao item New Project (figura 2). Caso você tenha instalado o compilador Fortran, aparecerá a linha Intel Visual Fortran e crie a aplicação para console (Figura 3). Se não existir um arquivo com o código fonte já escrito, selecione o empty Project e a pasta de trabalho, conforme tudo indicado na figura 3. Clique com o botão direito no item Source file e adicione um novo item. Escolha um tipo de arquivo de extensão ou “.for” ou “.f90”, atribuindo um nome a esse código fonte - ver figuras 4 e 5.

Nesse código fonte, desenvolva suas rotinas (figuras de 6 a 9), compile, construa e rode o código.

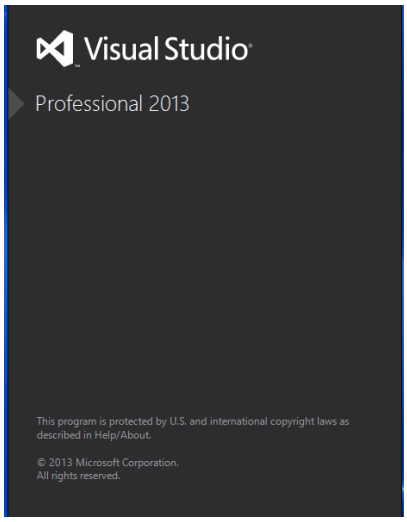


Figura 1

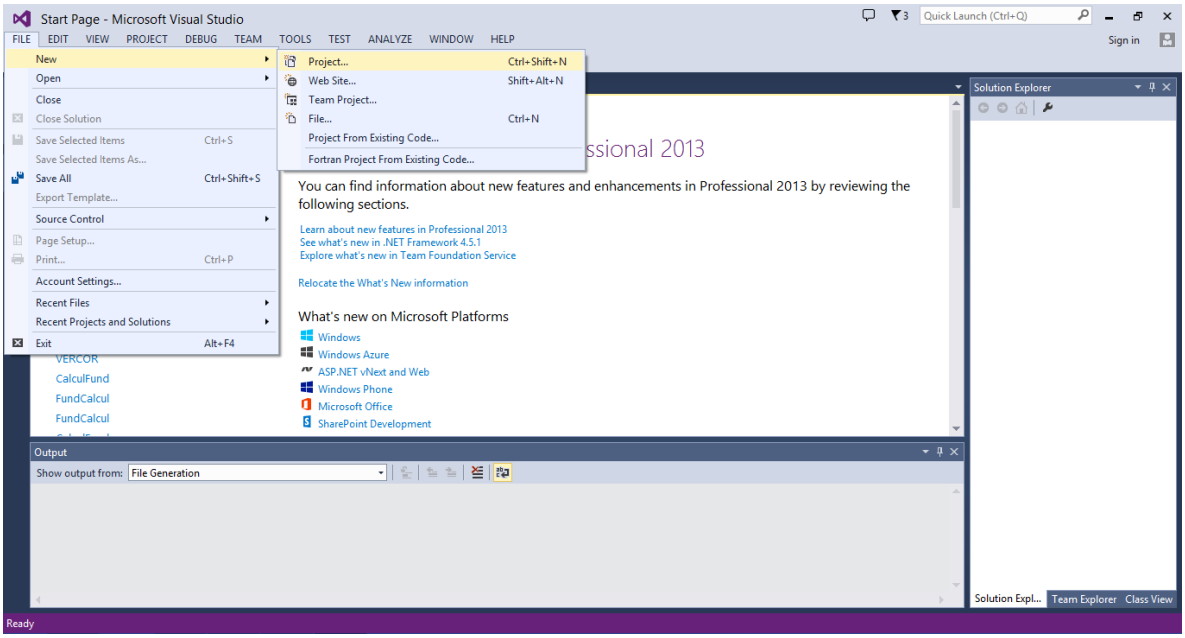


Figura 2

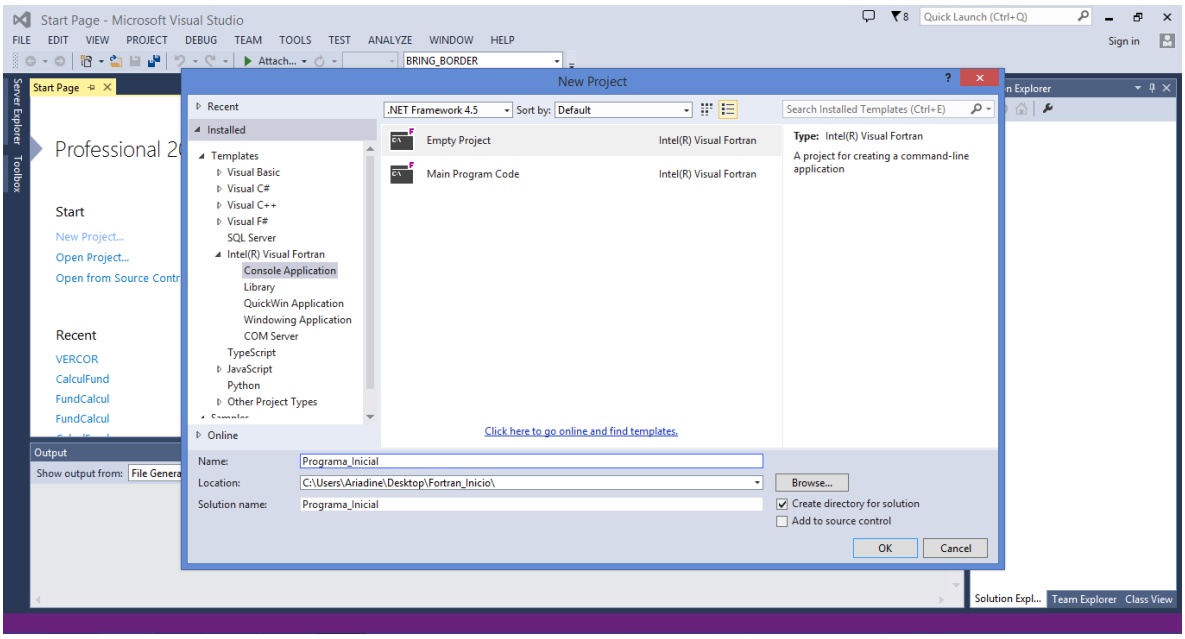


Figura 3

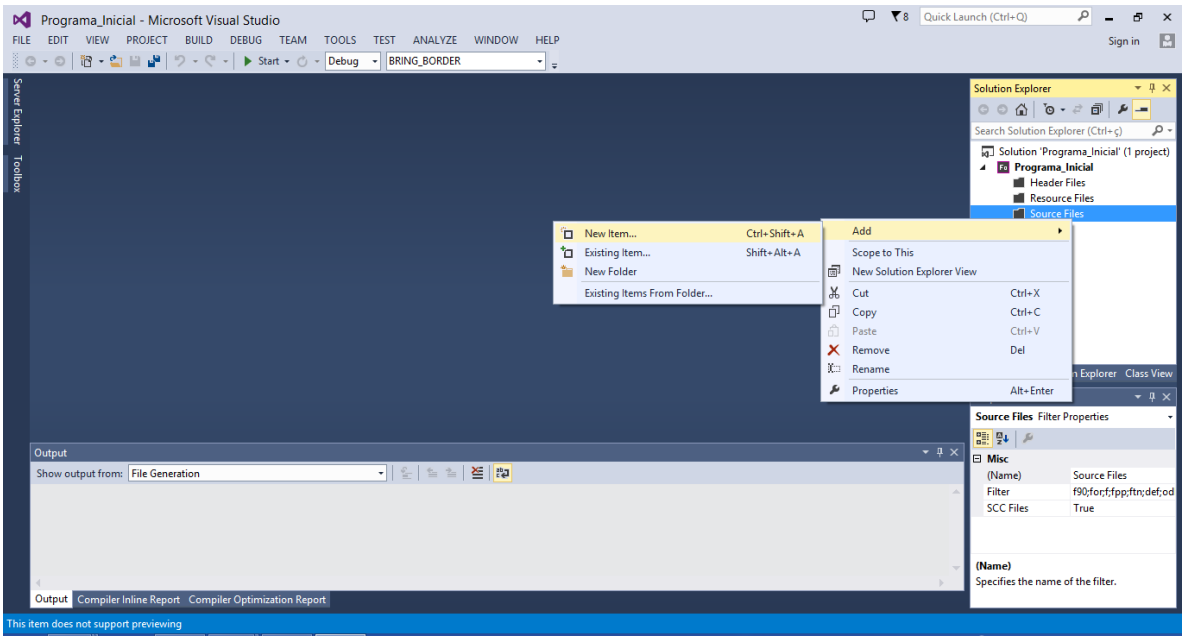


Figura 4

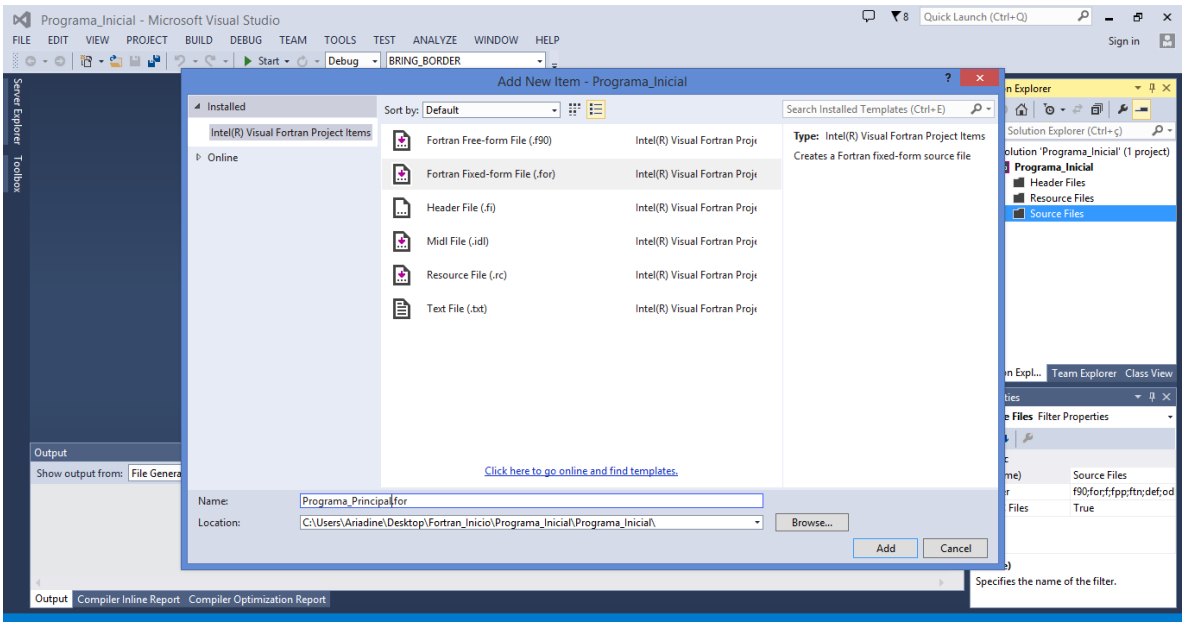


Figura 5

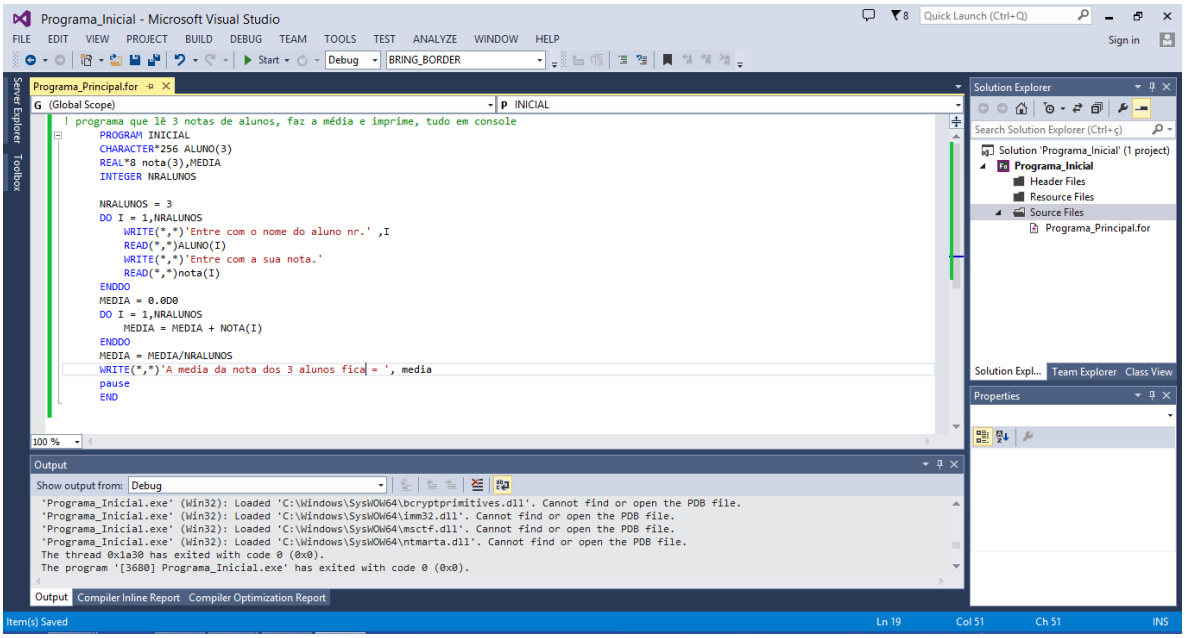


Figura 6

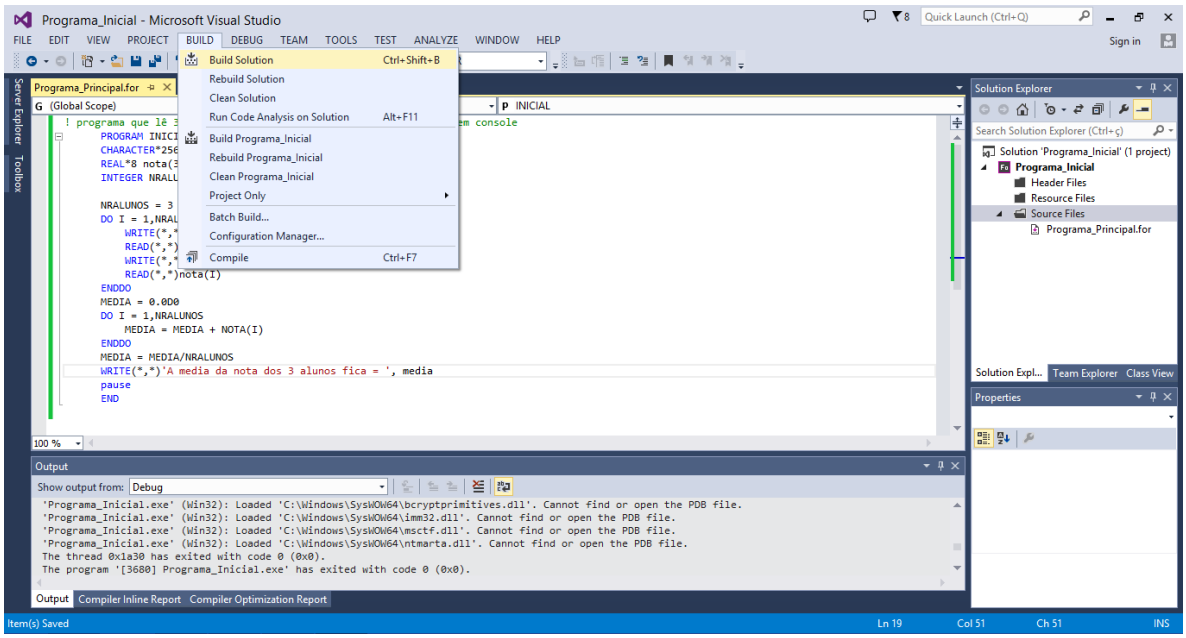


Figura 7

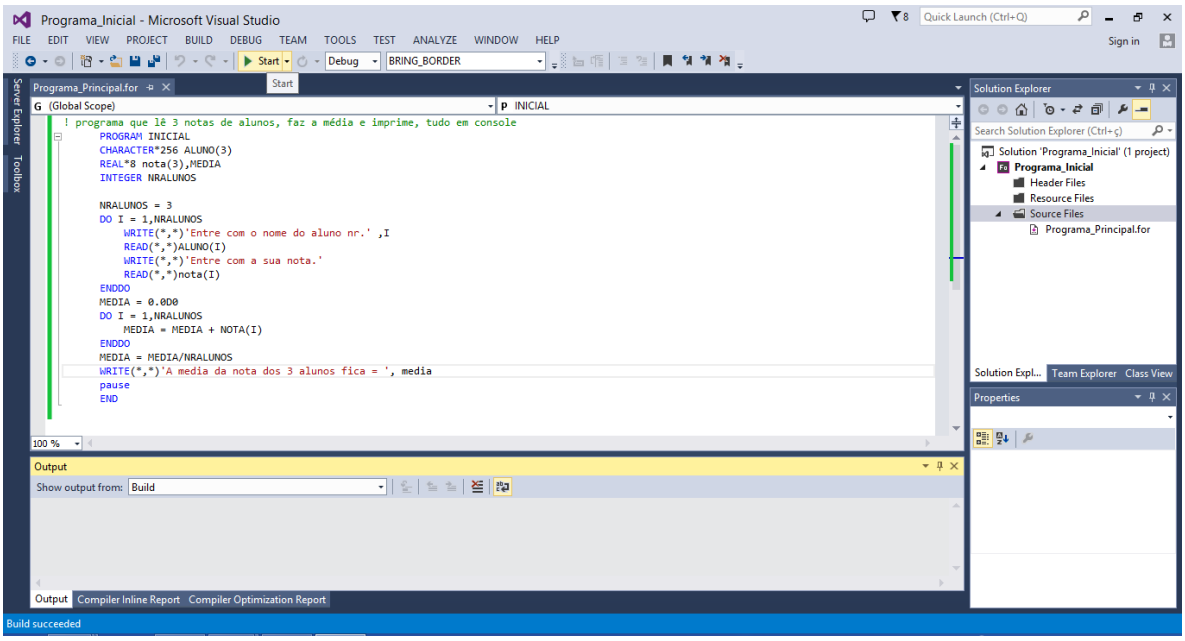


Figura 8

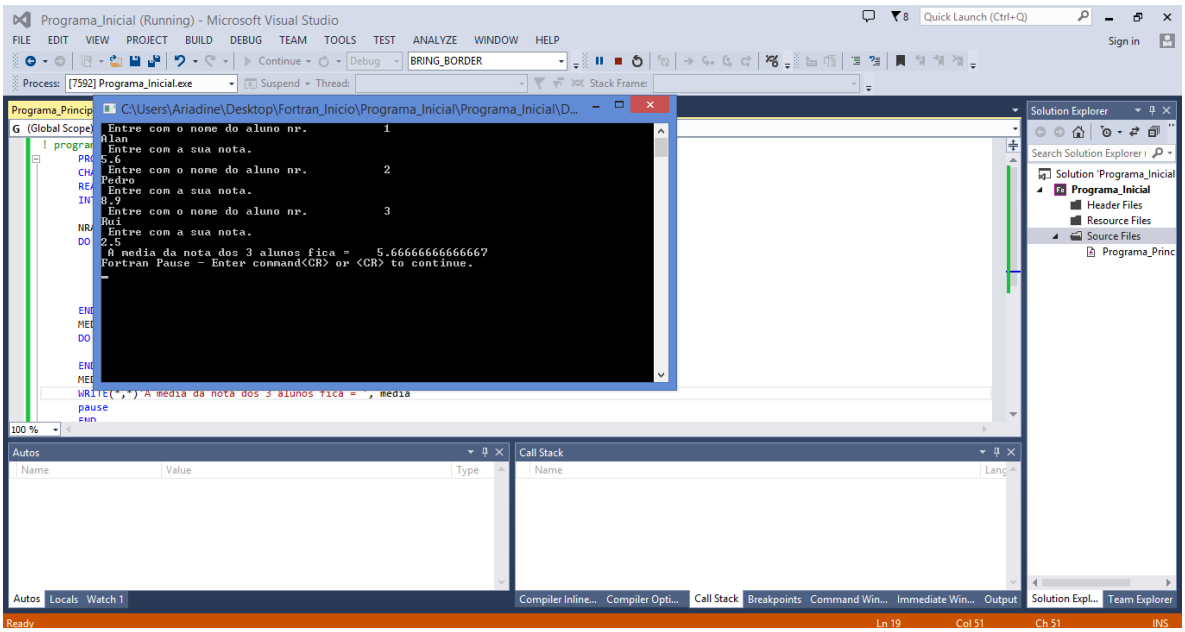


Figura 9

3. DESCRIÇÃO DE COMANDOS BÁSICOS

Leitura e impressão de dados:

WRITE(*,*) READ(*,*)

Exemplo:

```
Write(*,*)'Entre com o numero de nos'
Read(*,*)NNOS
Write(*,*)'Vamos escrever o numero de nos:'
Write(*,*)'O numero de nos e', nnos
```

Finalização do código: **único comando obrigatório do Fortran.**

END

Exercício 2: No editor do Fortran, escreva o código abaixo, compile e execute-o.

```
write(*,*)'Ola Fortran!! '
end
```

4- ARQUITETURA DO CÓDIGO

A forma mais didática e também a mais usada para a seqüência de um código simples.

- Cabeçalho: apresentação do código, comentando o seu objetivo, modelo adotado, autoria(s), versão, etc.
- Declaração de variáveis globais: valores inteiros, reais, parâmetros, caracteres, alocações dinâmicas, etc.
- Núcleo do código: desenvolvimento no código, mediante as tarefas a serem executadas ou as sub-rotinas a serem chamadas.
- Fechamento do código.

5- SINTAXES

a) *Declaração de variáveis escalares.*

- Explícita
- Implícita
- Default no Fortran

- Explícita

```
Integer  NO, NELEM, A
Real*8  comp, pi, neper
Double precision comp,pi, neper
Character*10 arq
```

- Implícita

```
Implicit   real*8(D,H)   ! D e H são reais de precisão dupla
Implicit   real*8(D-H)  ! D a H   são reais " "
Implicit   character*30 (A,B)
```

- *Default* do Fortran

Tipo inteiro: I, J, K, L, M, e N
Reais: para as demais letras

Obs.: Tenha cuidado ao se trabalhar com as variáveis de *Default* do tipo real, pois elas são definidas com precisão simples, o que - em geral - não é aconselhável utilizar nos nossos códigos.

Exercício 3: Faça um programa que leia dois valores e realize as 4 operações básicas (+,-, •, /) sobre estes valores. Entre os dois valores pelo teclado e imprima-os na tela. E se o quociente for zero, o que acontece na divisão?

Exercício 4: Calcule "π", pelos três modos de declaração, sabendo-se que

$$\pi = \text{acos}(-1.0)$$

b) Operações

Em Fortran tem-se que tomar cuidado ao se efetuar as operações com variáveis inteiras e reais. Pois, mesmo sendo definido uma variável (d) como real, na operação

$$d = 1/3 + 1/3 + 1/3$$

o valor de "d" será zero, já que para o Fortran esta é uma operação com valores inteiros. Assim, é **obrigatório** diferenciar os valores digitados com a inclusão de um ponto no número, ou seja para se obter d=1, na equação acima, tem-se que escrever

$$d = 1./3 + 1./3 + 1./3.$$

Exercício 5 : Declarar uma variável (d) como real*8 e fazer

$$d = 1/3 + 1/3$$

imprimir na tela e depois acrescer o ponto de decimal nas parcelas somadas e verificar o resultado. Concatenar também colocando o comando para dupla precisão.

Exercício 6 : Faça um programa que leia, via teclado, o valor de um ângulo em graus, converta-o para radianos e imprima o resultado da relação trigonométrica:

$$[\cos(x)]^2 + [\text{sen}(x)]^2$$

c) Estrutura de controle

- Estrutura condicional;
- Estrutura de laços.

c.1) Estrutura condicional

- IF aritmético

  Exemplo:

```
        If (M) 16,17,18
16      write(*,*) 'valor negativo'
        goto 20
17      write(*,*) 'valor nulo'
        goto 20
18      write(*,*) 'valor positivo'
```

- IF lógico (só permite um comando)

  Exemplo:

```
        If (m.gt.0) write(*,*) 'valor positivo'
```

- IF BLOCO

  If-then-else

  Forma geral:

```
        If (condicao) then
                BLOCO1
        Else
                BLOCO2
        Endif  ! comando obrigatório
```

Obs: Dentro destas condições pode-se colocar outros critérios condicionantes.

Exercício 7 : Faça um programa que leia 3 valores naturais e que possam ser impressos da seguinte forma:

i) ordem crescente;

ii) ordem decrescente;

iii) na ordem em que foram entrados via teclado;

c.2) Estrutura de Laços

- DO WHILE
- DO – ENDDO

- DO WHILE: executa um bloco de comandos repetidas vezes enquanto a condição

```
DO WHILE                    DO 13 WHILE
    ....        ou              ....
    ....                        ....
ENDDO                13     CONTINUE
```

lógica permaneça verdadeira.

Exercício 8 : Sabendo que a igualdade abaixo é valida para -1<x<+1. Escreva um código que calcule este quociente com uma precisão fornecida pelo usuário.

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots + x^n$$

- DO-ENDDO/CONTINUE

Forma geral:

```
      DO n   I=n1,n2,incr          DO  I=n1,n2,incr
      ...                 ou       ...
      ...                          ...
n     CONTINUE                     ENDDO
```

Onde:

n: abragência do laço (só é usado com “continue”);

I: variável inteira do laço;

n_1: valor inicial para a contagem;

n_2: valor final para a contagem;
incr: incremento do laço de n_1 a n_2.

Obs: Pode-se haver laços dentro de laços, desde que sejam de forma coerente dentro da estrutura de cada um, sem cruzamento entre eles.

Exercício 9 : Escreva um algoritmo que calcule o fatorial dos vinte primeiros naturais, imprima-os na tela.

Exercício 10 : Calcule "π" mediante a série.

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - ....$$

Estipule uma precisão.

d) Entrada e saída de dados

Imprimindo dados:

- WRITE(i, n)A,B,A+C

Lendo dados:

- READ(i, n)

onde

i: refere-se ao arquivo a ser escrito , se * ou 0 indica impressão na tela;
n: formato das variáveis a serem escritas.

e) FORMAT

Este comando permite para os dados de impressão:

- pular linhas (/);
- atribuir espaçamento entre os parâmetros (X);
- escrever textos (‘ ‘);
- especificar tamanho e forma de impressão das variáveis.

e.1) Formato para inteiro

aIw

onde

a: n° de vezes que a especificação I deve ser obedecida;

w: constante inteira e sem sinal que indica o comprimento do inteiro.

Exemplo:

```
WRITE(*,19) N_NO,N_ELEM,N_FORCAS,N_RESTR
 19     FORMAT(//,X,I4,X,2(X,I3),X,I5)
```

e.2) Formato para reais

aFw.d

d: constante inteira e sem sinal que indica o n° de casas decimais

Exemplos:

e.2.1)

126.85 → F6.2

-2.1 → F4.1

-53. → F4.0

e.2.2)

Sabendo-se que:

H1= 1,2356 H2= -0,00339 H3= 123,4159

```
WRITE(*,901)H1,H2,H3
901    FORMAT(//,X,3(X,F8.4))
```

Para notação científica, usar

a**Ew.d**: 0,00123e-2

a**ENw.d**: 12,3e-6

a**ESw.d**: 1,23e-5

procurando respeitar $w \geq d+7$.

f) Vetores e matrizes

Declaração “estática”:

```
REAL*8 A(20,20), B(3,2), V(40), ROT(6,6)
INTEGER CONECT(10,3)
```

Alocação dinâmica:

```
REAL*8    CORD[ALLOCATABLE](:,:), Iyp[ALLOCATABLE](:)
INTEGER CODI[ALLOCATABLE](:)
Ou para compiladores unix usar
REAL*8,DIMENSION(:,:),ALLOCATABLE::CORD
INTEGER,DIMENSION(:),ALLOCATABLE::CODI
```

```
...
...
READ(4,*)NO NELE
ALLOCATE(CORD(NO,3),ELEM(NELE,3),CODI(3*NO))
...
...
DEALLOCATE(CODI,ELEM)
ALLOCATE(CODI(NO))
...
```

No código:

Exercício 11 : Escreva um código em que se entre com a dimensão da matriz quadrada (A_{nxn}) e de um vetor (B_n). Aloque-os dinamicamente e atribua valores unitários em todas as posições de A e B. Então, faça o produto entre A e B e imprima o vetor resultante na tela. Finalmente, desaloque A e B.

$A_{ij} = 1,0 \qquad B_i = 1,0 \qquad (i = 1,n)$

$C_i = A_{ij} \bullet B_j \quad (i,j = 1,n)$

Exercício 12: Generalize o código supra para o caso de A ser uma matriz de dimensão (nxk) e B outra matriz (kxm). Aloque-os dinamicamente e atribua valores unitários em todas as posições de A e B. Então, faça o produto entre A e B e imprima a matriz resultante na tela. Finalmente, desaloque A e B.

$$A_{nxk} = B_{kxm} = 1,0$$

$$C_{nm} = A_{nk} \bullet B_{km}$$

g) Arquivos

Forma geral:

```
OPEN(n,FILE=ARQUIVO,STATUS='UNKNOWN')
```

Exemplo:

Listagem do código

```
3     FORMAT (A30)
      WRITE(*,1)
1     FORMAT(//,' Nome do arquivo de dados:',\)
      READ(*,3) ARQ
      OPEN (2, FILE=ARQ , STATUS='unknown')

!     leitura dos titulos

      READ(2,3) TIT1
      READ(2,3) TIT2
      READ(2,3)
      READ(2,3)

! leitura do n. de nós, pontos internos, n. de
! elementos, nos duplos e ptos de Gauss
      READ(2,*)NN,NPI,NE,NND,NG
      READ(2,3)
      READ(2,3)
      READ(2,3)
      READ(2,*)MAT(3),MAT(4),MAT(5),MAT(6)
      READ(2,3)
      READ(2,3)
! alocando variaveis necessarias...
     ALLOCATE( CORD(nn+nnd,2),CONECT(ne,2),L(ne,3) )

! lendo coordenadas dos nós..
     DO I=1,NN
           READ(2,*)J,CORD(J,1),CORD(J,2)
     ENDDO

    READ(2,3)
    READ(2,3)

! lendo no duplo e no de origem..
! le-se no arquivo: N# no duplo  No origem
    DO I=1,NND
           READ(2,*)J,K
          CORD(J,1)=CORD(K,1)
          CORD(J,2)=CORD(K,2)
   ENDDO
```

Listagem do arquivo de dados

```
ELASTICO BIDIMENSIONAL
EXEMPLO 3: Membrana de Cook

Nos  Nos int.    Elementos   Nos duplos  Ptos de Gauss
14        67          14          14             12

epd=1/ pt fora =1/ dist. pt fora (dist)/ dist. no elem (kp)
EPD      PF        DIST              KP
 1.       0.         0.5              0.25

COORDENADAS DE C/ NO
1  0.0,0.0
2  12.,11.0
4  24.0,22.0
6  36.,33.0
8  48.0,44.0
10 48.,52.0
12 48.0,60.0
14 36.0,56.0
16 24.0,52.0
18 12.0,48.0
20 0.0,44.0
22 0.0,33.0
24 0.0,22.0
26 0.0,11.0

NO DUPLO/ NO ORIGEM
  3              2
  5              4
  7              6
  9              8
  11             10
  13             12
  15             14
  17             16
  19             18
  21             20
  23             22
  25             24
  27             26
```

Exercício 13 : Escreva um código em que se leia em um arquivo os seguintes dados de um pórtico plano:

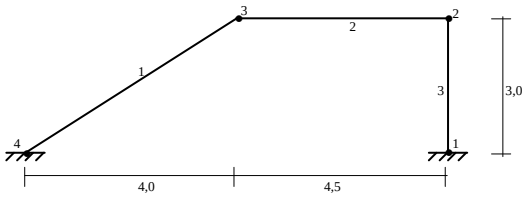
N° de nós = NN

N° de elementos = NE

Coordenadas dos nós = CORD(NN,2)

Conectividade dos elementos = CONECT(NE,2)

Com estes valores já armazenados, calcule e imprima em outro arquivo o comprimento e os cossenos diretores de cada barra. Verifique no exemplo a seguir seu código.



h) Subprogramas

Exercício 14 : Escreva um código onde o usuário entre com 2 valores inteiros pelo teclado e em duas sub-rotinas estes números sejam, respectivamente, somados e multiplicados. Imprima-os a seguir no programa principal.

Exercício 15 : Transforme os exercícios 11 e 12 em sub-rotinas de um programa principal. Então, no programa principal leia de um arquivo duas matrizes distintas - que foram alocadas dinamicamente - e um vetor. Chame, em seguida, a sub-rotina que faça o produto das matrizes. Multiplique a matriz resultante pelo vetor e imprima o resultado na tela.

i) Resolução de sistemas lineares: matriz cheia e em banda

Exercício 16 : Crie um arquivo em que estão definidos o tamanho (N=9), os valores da matriz de rigidez R_{9x9} e do vetor de forças F_9. Leia estes valores no programa principal e chame a sub-rotina para resolução de sistemas lineares com a matriz armazenada cheia. Em seguida, imprima os valores do vetor resultante em outro arquivo.

$$[R] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 & -9{,}6 & 8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & -9{,}6 & 0 & 15{,}36 & 0 & 0 & 0 & 9{,}6 \\ 0 & 0 & 8 & 0 & 0 & 32 & 0 & 0 & 8 \\ 0 & 0 & 0 & -4 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 9{,}6 & 8 & 0 & 0 & 16 \end{bmatrix} \qquad \{f\} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -10 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

Exercício 17 : Desenvolva um código computacional que leia a matriz de rigidez e o vetor de forças do exercício anterior e o comprimento de banda desta matriz (calcule!!). Então, crie uma outra matriz de rigidez a qual receberá apenas os valores da matriz R que estejam localizados acima da diagonal principal, incluindo esta. Utilizando esta nova matriz em banda e com o vetor {f}, resolva o sistema linear mediante o método de resolução em banda. Verifique o resultado.

6 – LISTAGEM DOS EXERCÍCIOS

```
! Exemplos
! EXE.2
      write(*,*)' Ola Fortran!! '
      end
! EXE.3
            real*8 a,b
            write(*,*)'Entre com o numerador'
            read(*,*)a
            write(*,*)'Entre com o denominador'
            read(*,*)b
! soma:
            write(*,*) a+b
! subtracao:
            write(*,*) a-b
! multiplicação:
            write(*,*) a*b
! divisão:
            write(*,*) a/b
            end

! EXE.4
! Declaração explicita:
            real*8 pi
            pi=acos(-1.0)
            write(*,*)pi
            end
!-----------------------------------------
! Declaração implicita:
            implicit real*8 (p)
            pi=acos(-1.0)
            write(*,*)pi
            end
!-----------------------------------------
! Declaração por default:
            pi=acos(-1.0)
            write(*,*)pi
            end
!-----------------------------------------

! EXE.5
            real*8 d
            d=1/3+1.d0/3
            write(*,*)d
            end

! EXE.6
            real*8 angulo,s,c
            write(*,*)'Entre com o ângulo em graus:'
            read(*,*)angulo
            angulo=cos(-1.0d0)*angulo/180.d0
            s=(sin(angulo))**2
            c=(cos(angulo))**2
            write(*,*)s+c
            end

! EXE.7
! Declaracao implicita:
            write(*,*)'Entre com 3 numeros naturais'
            read(*,*)i,j,k
            if( i.le.j)then
                  if(i.le.k) then
                        menor=i
                        if(j.le.k) then
                              maior=k
```

```
                                        medio=j
                                else
                                        maior=j
                                        medio=k
                                endif
                        else
                                medio=i
                                menor=k
                                maior=j
                        endif
                else
                        if(i.ge.k) then
                                maior=i
                                if(j.ge.k) then
                                        menor=k
                                        medio=j
                                else
                                        menor=j
                                        medio=k
                                endif
                        else
                                medio=i
                                maior=k
                                menor=j
                        endif
                endif
                write(*,*)'i) ordem crescente:'
                write(*,*)menor,medio,maior
                write(*,*)
                write(*,*)'ii) ordem decrescente:'
                write(*,*)maior,medio,menor
                write(*,*)
                write(*,*)'iii) ordem de entrada:'
                write(*,*)i,j,k
        end

! EXE.8
        real*8 x,precisao,dif,soma

        write(*,*)'Entre com a variavel do denominador:'
        read(*,*)x
        write(*,*)'Entre com a precisao querida:'
        read(*,*)precisao
        dif=2*precisao
        i=2
        soma=x+1.d0
        do while(dif.gt.dabs(precisao))
                dif=soma
                soma=soma+x**i
                dif=dabs(soma-dif)
                i=i+1
        enddo
        write(*,*)'O quociente e',soma
!..
        end

! EXE.9
        real*8 fat
!..
        fat=1.d0
        write(*,2)
2       format(3(//),' Numero!',5x,'Valor')
        do i=1,20
                fat=i*fat
                write(*,1)i,fat
```

```
1               format(X,I2,5X,En12.4)
        enddo
!..
        end

! EXE.10
        real*8 pi,precisao,dif
!
        write(*,*)'Entre com a precisao querida:'
        read(*,*)precisao
        dif=1
        i=1
        j=1
        pi=0.d0
        do while(dif.gt.precisao)
                dif=pi
                pi= 4.d0/(i*(-1.d0)**(j-1))+pi
                i=i+2
                j=j+1
                dif=dabs(pi-dif)
        enddo
        print*,'O valor de pi fica',pi
        end

! EXE.11
       REAL*8,DIMENSION(:,:),ALLOCATABLE :: A
       REAL*8,DIMENSION(:),ALLOCATABLE :: B
       REAL*8,DIMENSION(:),ALLOCATABLE :: C
        INTEGER N

        WRITE(*,*)'Entre com a dimensao da matriz e do vetor '
        READ(*,*)N
! alocando A,B e C
        ALLOCATE(A(N,N),B(N),C(N))
! atribuindo valores unitarios
        DO I=1,N
                B(I)=1.D0
                        DO J=1,N
                                A(I,J)=1.D0
                        ENDDO
        ENDDO
        DO I=1,N
                C(I)=0.D0
                        DO J=1,N
                                C(I)=A(I,J)*B(J)+C(I)
                        ENDDO
        ENDDO
! imprimindo C
        WRITE(*,*)'O vetor resultante fica:'
        DO I=1,N
                WRITE(*,*)'C(',I,')=',C(I)
        ENDDO
! desalocando A, B e C
        DEALLOCATE(A,B,C)
        END

! EXE.12
       REAL*8,DIMENSION(:,:),ALLOCATABLE :: A
       REAL*8,DIMENSION(:,:),ALLOCATABLE :: B
       REAL*8,DIMENSION(:,:),ALLOCATABLE :: C
       INTEGER N,K,M

        WRITE(*,*)'Entre com a dimensao (n) da matriz A:'
        READ(*,*)N
        WRITE(*,*)'Entre com a dimensao (k) da matriz A:'
```

```
        READ(*,*)K
        WRITE(*,*)'Entre com a dimensao (m) da matriz B:'
        READ(*,*)M

! alocando A,B e C
        ALLOCATE(A(N,K),B(K,M),C(N,M))
! atribuindo valores unitarios
        A=1.D0
        B=1.D0
        DO I=1,N
                DO L=1,M
                        C(I,L)=0.D0
                        DO J=1,K
                                C(I,L)=A(I,J)*B(J,L)+C(I,L)
                        ENDDO
                ENDDO
        ENDDO
! imprimindo C
        WRITE(*,*)'A matriz resultante fica:'
        DO I=1,N
                DO J=1,M
                        WRITE(*,*)'C(',I,',',J,')=',C(I,J)
                ENDDO
        ENDDO
! desalocando A, B e C
        DEALLOCATE(A,B,C)
        PAUSE
        END

C       EXE.13
         REAL*8,DIMENSION(:,:),ALLOCATABLE :: CORD
         INTEGER,DIMENSION(:,:),ALLOCATABLE :: CONECT
         REAL*8 L,X1,X2,Y1,Y2,SENO,COS
         INTEGER NN,NE
         CHARACTER*30 ARQENT,ARQSAI

           WRITE(*,*)'ENTRE COM O NOME DO ARQUIVO DE ENTRADA:'
           READ(*,*)ARQENT
           WRITE(*,*)'ENTRE COM O NOME DO ARQUIVO DE SAIDA:'
           READ(*,*)ARQSAI
! abrindo arquivo de entrada...
           OPEN(2,FILE=ARQENT,STATUS='UNKNOWN')

           READ(2,*)
           READ(2,*)
           READ(2,*)
           READ(2,*)
           READ(2,*) NN
           READ(2,*)
           READ(2,*)
           READ(2,*) NE
           READ(2,*)
           READ(2,*)
           READ(2,*)
! alocando arrays necessarias...
           ALLOCATE( CORD( NN,2), CONECT(NE,2) )
           DO I=1,NN
                   READ(2,*)J,CORD(J,1),CORD(J,2)
           ENDDO
           READ(2,*)
           READ(2,*)
           READ(2,*)
           DO I=1,NE
                   READ(2,*)J,CONECT(J,1),CONECT(J,2)
           ENDDO
```

```
! abrindo arquivo de saida...
          OPEN(3,FILE=ARQSAI,STATUS='UNKNOWN')
          WRITE(3,*)' ARQUIVO DE SAIDA:'
          WRITE(3,*)
! imprimindo no arquivo de saida...
          WRITE(3,*)'ELEMENTO   COMPRIMENTO  SENO  COSSENO'
          WRITE(3,*)
          DO J=1,NE
                    X1= CORD( CONECT(J,1),1 )
                    X2= CORD( CONECT(J,2),1 )
                    Y1= CORD( CONECT(J,1),2 )
                    Y2= CORD( CONECT(J,2),2 )
                    L=DSQRT( (X2-X1)**2 + (Y2-Y1)**2 )
                    SENO= (Y2-Y1)/L
                    COS=  (X2-X1)/L
                    WRITE(3,1)J,L,SENO,COS
1                   FORMAT(I4,3X,F9.4,3X,F9.4,3X,F9.4)
          ENDDO
          END
!-------------------------------------------------------------------------------
!----------------------  ARQUIVO ------------------------------------------------
Exercicio 13 da apostila de Fortran
PORTICO PLANO, 4 nos e 3 elementos

NUMERO DE NOS
4

NUMERO DE ELEMENTOS
3

COORDENADAS
NO        X         Y
1          8.5      0
2          8.5      3.0
3          4.0      3.0
4          0        0

CONECTIVIDADE
ELEMENTO      NO1              NO2
1                  4                   3
2                  3                   2
3                  1                   2

! EXE.14
! Declaracao implicita...

! lendo valores pelo console...
          write(*,*)'Entre com o primeiro valor inteiro:'
          read(*,*) n
          write(*,*)'Entre com o segundo valor inteiro:'
          read(*,*) m
! chamando sub-rotinas...
          CALL SOMA(N,M,NMSOMA)
          CALL MULTIPLICA(N,M,NMPROD)
! imprimindo valores...
          write(*,*)'O valor de',n,'      +',m,'=',nmsoma
          write(*,*)
          write(*,*)'O valor de',n,'      *',m,'=',nmprod
!
          end
!-------------------------------------------------------------------------
          SUBROUTINE SOMA(I,J,K)
          K=I+J
          END
!-------------------------------------------------------------------------
```

```
      SUBROUTINE MULTIPLICA(L,K,I)
      I=L*K
      END
!-----------------------------------------------------------------------

! EXE.15
      REAL*8,DIMENSION(:,:),ALLOCATABLE :: A
      REAL*8,DIMENSION(:,:),ALLOCATABLE :: B
      REAL*8,DIMENSION(:,:),ALLOCATABLE :: C
      REAL*8,DIMENSION(:),ALLOCATABLE :: D
      REAL*8,DIMENSION(:),ALLOCATABLE :: E
        INTEGER N,K,M
        CHARACTER*20 ARQ

      WRITE(*,*)'ENTRE COM O NOME DO ARQUIVO DE ENTRADA'
      READ(*,*)ARQ
      OPEN(2,FILE=ARQ,STATUS='UNKNOWN')

      DO 3 I=1,5
            READ(2,*)
3     CONTINUE
! lendo as dimensões da 1a. matriz...
      READ(2,*)N,K
      READ(2,*)
      READ(2,*)
! lendo a dimensão da coluna da 2a. matriz...
      READ(2,*)M
      READ(2,*)
      READ(2,*)
! alocando arrays...
      ALLOCATE( A(N,K), B(K,M), C(N,M), D(M), E(N) )
! lendo os valores das arrays do arquivo...
      DO I=1,N
            READ(2,*)
            DO 2 J=1,K
                  READ(2,*)A(I,J)
2           CONTINUE
      ENDDO
      READ(2,*)
      DO I=1,K
            READ(2,*)
            DO 5 J=1,M
                  READ(2,*)B(I,J)
5           CONTINUE
      ENDDO
      READ(2,*)
      DO I=1,M
            READ(2,*)D(I)
      ENDDO
!  fechando arquivo de entrada...
      CLOSE(2)
!   fazendo o produto de [A]*[B]=[C]...
      CALL MULTIMA(A,B,C,N,K,M)
!  desalocando [A] e [B]...
      DEALLOCATE(A,B)
! fazendo o produto de [C]*{D}={E}...
      CALL MULTIVE(C,D,E,N,M)
!  desalocando [C]e {D}...
      DEALLOCATE(C,D)
! imprimindo valores de {E} no console...
      WRITE(*,*)'O vetor resultante fica:'
      DO I=1,N
            WRITE(*,*)'E(',I,')=',E(I)
      ENDDO
! desalocando {E}...
```

```
        DEALLOCATE(E)
! finalizando código
        END
!-------------------------------------------------------------------------
        SUBROUTINE MULTIMA(H,T,P,LH,CH,CT)
        INTEGER LH,CH,CT
        REAL*8 H(LH,CH),T(CH,CT),P(LH,CT)
        DO I=1,LH
                DO L=1,CT
                        P(I,L)=0.D0
                        DO J=1,CH
                                P(I,L)=H(I,J)*T(J,L)+P(I,L)
                        ENDDO
                ENDDO
        ENDDO
        END
!-------------------------------------------------------------------------------
        SUBROUTINE MULTIVE(H,T,P,LH,CH)
        INTEGER LH,CH
        REAL*8 H(LH,CH),T(CH),P(LH)
        DO I=1,LH
                P(I)=0.D0
                        DO J=1,CH
                                P(I)=H(I,J)*T(J)+P(I)
                        ENDDO
        ENDDO
        END
!-------------------------------------------------------------------------------
!---------- ARQUIVO DE DADOS --------------------------------
Arquivo do exercicio 15 da apostila de Notas de aula do Fortran
Dados de duas matrizes e um vetor

Dimensao da primeira matriz:
Linha    Coluna
  3       4
Dimensao da segunda matriz:
Coluna
  2

 Valores da primeira matriz
 Linha 1
 3
 -4
 1.5
 -1
 Linha 2
 2
 -3
 7
 6.2
 Linha 3
 1.2
 -9
 8
 0.5
 Valores da segunda matriz
 Linha 1
 -1
 0.01
 Linha 2
 2
 7
 Linha 3
 -4
 6.5
```

```
 Linha 4
 -5
 -1.5
Valores do vetor
12
-13

! EXE.16
          REAL*8,DIMENSION(:,:),ALLOCATABLE :: R
          REAL*8,DIMENSION(:),ALLOCATABLE :: F
          REAL*8,DIMENSION(:),ALLOCATABLE :: X
            INTEGER N
            CHARACTER*30 ARQENT,ARQSAI
!   lendo nomes dos arquivos de entrada e saida:
            WRITE(*,*)'ENTRE COM O NOME DO ARQUIVO DE ENTRADA'
            READ(*,*)ARQENT
            WRITE(*,*)'ENTRE COM O NOME DO ARQUIVO DE SAIDA'
            READ(*,*)ARQSAI
!   abrindo arquivo de entrada
            OPEN(2,FILE=ARQENT,STATUS='UNKNOWN')

            READ(2,*)
            READ(2,*)
            READ(2,*)
            READ(2,*)
            READ(2,*) N
            READ(2,*)
            READ(2,*)
!  alocando as arrays...
            ALLOCATE( R(N,N),F(N),X(N) )
!  lendo os valores das arrays do arquivo...
            DO I=1,N
                        READ(2,*)
                        DO J=1,N
                                    READ(2,*)R(I,J)
                        ENDDO
            ENDDO

            READ(2,*)
            READ(2,*)

            DO I=1,N
                        READ(2,*)F(I)
            ENDDO
!  fechando o arquivo de entrada...
            CLOSE(2)
! chamando rotina para resolver sistema...
          CALL GAUSS(X,R,F,N)
! desalocando matriz e vetor...
            DEALLOCATE(R,F)
! imprimindo {X}...
            OPEN(3,FILE=ARQSAI,STATUS='UNKNOWN')
            WRITE(3,*)' RESULTADOS DE {X}: '
            WRITE(3,*)
            DO I=1,N
                        WRITE(3,*)'{X',I,'}=',X(I)
            ENDDO
!  desalocando vetor resposta...
            DEALLOCATE(X)
!  fechando arquivo...
            CLOSE(3)
!   finalizando código...
            END
!-------------------------------------------------------------------------------
 ! Sub-rotina para resoluca de sistema linear via método direto
```

```
 ! considerando a matriz cheia.
 ! Valores de entrada: [R] e {F} e nnos
 ! Valores de saida: {U}

           SUBROUTINE GAUSS(U,RG,F,NNOS)
!
           INTEGER GL,SOL,NNOS,NMAIOR
!
         REAL*8  U(NNOS),RG(NNOS,NNOS),F(NNOS),AUX
!
           SOL=1
           I=1
           GL=NNOS
!
!       pivoteando...
!
           DO WHILE (SOL.EQ.1.AND.I.LE.GL)
           NMAIOR=I
             DO 120 J=I+1,GL
               IF (ABS(RG(J,I)).GT.ABS(RG(NMAIOR,I))) THEN
                 NMAIOR=J
               END IF
120          CONTINUE
             IF (NMAIOR.NE.I) THEN
               DO 130 J=I,GL
                 AUX=RG(I,J)
                 RG(I,J)=RG(NMAIOR,J)
                 RG(NMAIOR,J)=AUX
130          CONTINUE
          AUX=F(I)
               F(I)=F(NMAIOR)
               F(NMAIOR)=AUX
             ENDIF
!            Fim do pivoteamento...
!
!            Triangularizacao da matriz...
!
       IF (RG(I,I).EQ.0.0) THEN
             WRITE(*,*) 'SISTEMA SEM SOLUÇAO !!!!'
             SOL=0
             STOP
            ELSE
              DO 180 J=I+1,GL
                AUX=RG(J,I)/RG(I,I)
                       RG(J,I)=0.0
                       DO  K=I+1,GL
                         RG(J,K)=RG(J,K)-RG(I,K)*AUX
                       END DO
                       F(J)=F(J)-F(I)*AUX
180           CONTINUE
             ENDIF
           I=I+1
             END DO
!            Fim da Triangularizacao...
!
!                       Retrosubstituicao...
!
             IF (SOL.EQ.1) THEN
               U(GL)=F(GL)/RG(GL,GL)
               DO 200 I=GL-1,1,-1
                 AUX=0.0
                 DO 190 J=I+1,GL
                       AUX=AUX+RG(I,J)*U(J)
190                    CONTINUE
               U(I)=(F(I)-AUX)/RG(I,I)
```

```
200          CONTINUE
          END IF
!
          RETURN
            END
!-------------------------------------------------------------------------------
!---------------------- ARQUIVO DE DADOS  --------------------------
EXERCICIO 16
RESOLUCAO DE SISTEMA, 07/06

Dimensao da matriz
9

Valores da matriz
Linha 1
1
0
0
0
0
0
0
0
0
Linha 2
0
1
0
0
0
0
0
0
0
Linha 3
0
0
16
0
-9.6
8
0
0
0
Linha 4
0
0
0
8
0
0
-4
0
0
Linha 5
0
0
-9.6
0
15.36
0
0
0
9.6
Linha 6
```

```
0
0
8
0
0
32
0
0
8
Linha 7
0
0
0
-4
0
0
4
0
0
Linha 8
0
0
0
0
0
0
0
1
0
Linha 9
0
0
0
0
9.6
8
0
0
16

Vetor f
0
0
0
0
-10
0
0
0
0
```