João Vítor Lima de Oliveira - 12694394

Relatório Projeto-1

São Carlos

2025

Tarefa 1

O programa apresentado em Fortran tem como objetivo calcular o valor da parcela mensal fixa de um empréstimo. Inicialmente, o programa solicita ao usuário três valores de entrada: a quantia emprestada Q, o número de parcelas mensais N e a taxa de juros mensais percentuais AJM. Após a leitura desses valores, a taxa de juros AJM é convertida de porcentagem para fração decimal, dividindo-se por 100, de modo que, por exemplo, a entrada 1.1 corresponde a uma taxa de 1.1% ao mês.

Com esses dados, o programa aplica a fórmula de financiamento com juros compostos para determinar o valor da prestação mensal V:

$$V = \frac{AJM \cdot Q \cdot (1 + AJM)^N}{(1 + AJM)^N - 1} \tag{1}$$

Por fim, o programa exibe no terminal o valor calculado de V, formatado com três casas decimais. Dessa forma, o código satisfaz a exigência de calcular e apresentar o valor da parcela fixa de um empréstimo com juros mensais compostos.

```
      program main
            write(*,*) 'Escreva o valor de Q.'
            read(*,*) Q
            write(*,*) 'Escreva o valor de N.'
            read(*,*) N !Inteiro
            write(*,*) 'Escreva o valor de AJM.'
            read(*,*) AJM

            AJM = AJM/100d0
            V = AJM*Q*((1d0+AJM)**N)/(((1d0+AJM)**N) - 1d0)

            write(*,7) V
7           format('O valor mensal pago   : ',F9.3)
      end program main
```

Tarefa 2

Objetivo

Nesse programa será estudado como calcular o volume e a área de um Torus. Na Figura - 2 é mostrada o como os valores do raio externo e interno interferem na estrutura do Torus. Ademais, considerando o caso no qual o raio interno é maior que o raio externo e ambos tem valores não nulos. Podemos utilizar a equação 2 para conseguirmos o valor da Área do Torus.

Calculo

As formulas utilizadas para calcular a área e o volume do torus são respectivamente Equação-2 e Equação-3

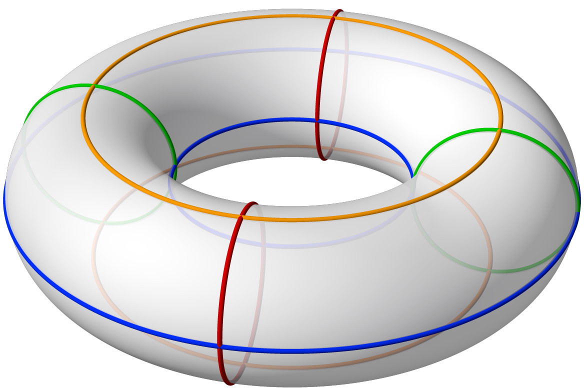
$$A = \int_0^{2\pi} R d\phi \int_0^{2\pi} r d\theta = 4\pi R r \tag{2}$$

$$V = \int_0^{2\pi} R d\phi \int_0^{2\pi} d\theta \int_0^{r} r' dr' = 2\pi R r^2 \tag{3}$$

Torus

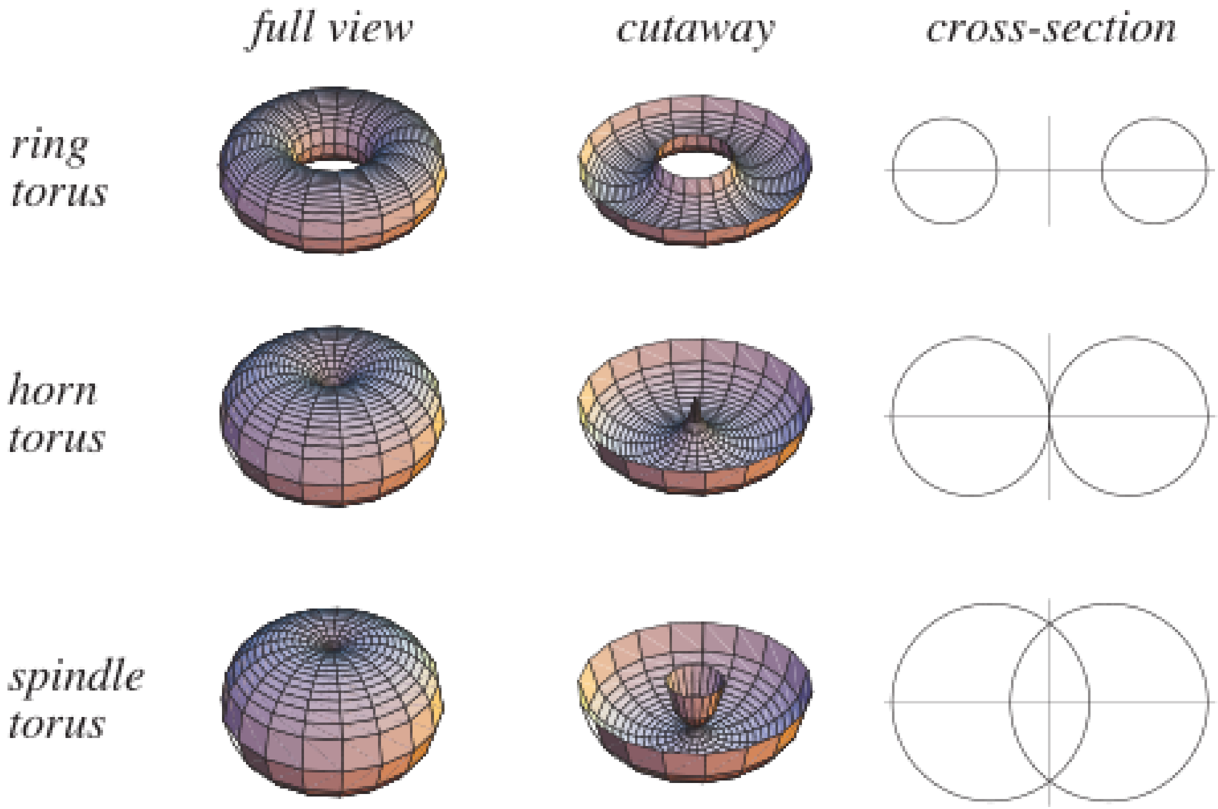
Em geometria, um torus é uma superfície de revolução gerada ao girar um círculo no espaço tridimensional por uma revolução completa em torno de um eixo que é coplanar com o círculo.

Figura 1 – Ilustração de um Torus.



Fonte: Retirado da Wikipedia.

Figura 2 – Diferentes tipos de Torus.



Fonte: Retirado da Wikipedia.

Em vista disso, foi utilizado o *workflow* apresentado na figura 3 para realizar a tarefa, vale notar que a única constante definida foi o valor de π. Outrossim, para realizar o cálculo da área e volume foram usadas as equações 2 e 3 respectivamente.

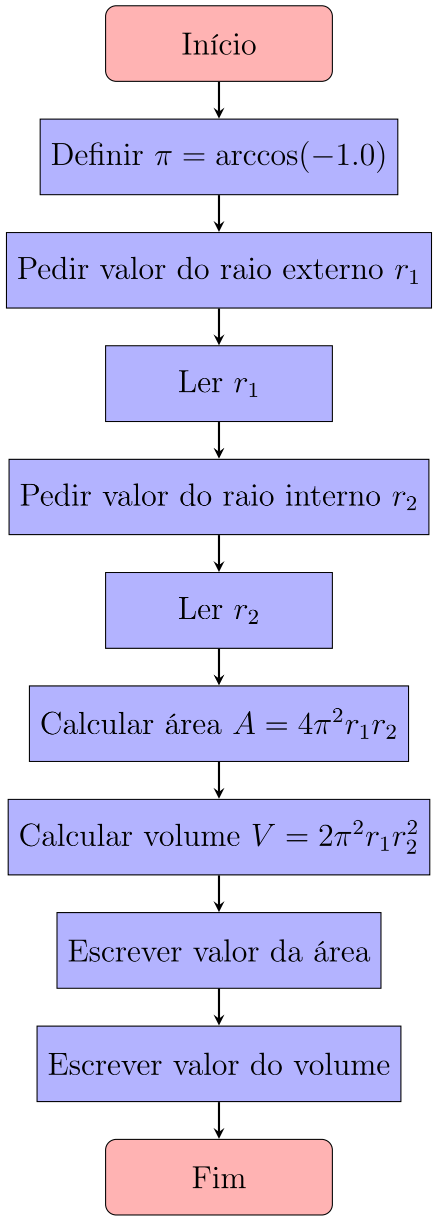


Figura 3 – Fluxograma do cálculo da área e do volume de um toroide

Ademais, o programa em questão está ilustrado abaixo.

Listing 1 – Cálculo da área e volume de um toro

```fortran
program main
    ! define pi
    pi = acos(-1.0)

    ! pede o valor dos raios
    write(*,2)
    read(*,*) r1
    write(*,3)
    read(*,*) r2

    ! realiza os c lculos
    area = 4*(pi**2)*r1*r2
    vol  = 2*(pi**2)*r1*(r2**2)

    ! imprime os resultados
    write(*,7) area
    write(*,8) vol

2   format('Insira␣o␣raio␣externo.')
3   format('Insira␣o␣raio␣interno.')
7   format(' rea :␣', F12.3)
8   format('Volume:␣', F12.3)

end program main
```

Tarefa 3

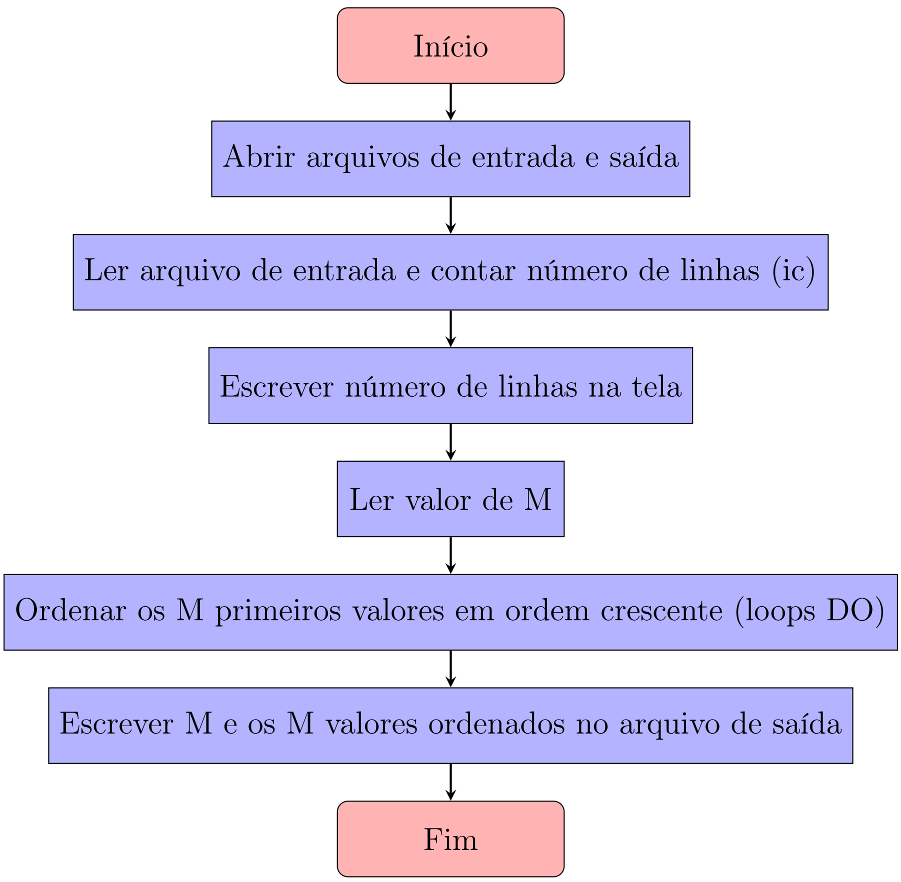
Nesse programa o objetivo é descobrir quantas linhas existem em um arquivo e organizar os M primeiros números em ordem crescente. Para isso, primeiro o arquivo de entrada é aberto com a função *open*. Em seguida, um *loop* percorre o arquivo para contar o número de linhas e guardar os valores em uma lista. Depois disso, a quantidade de linhas é mostrada na tela.

Na sequência, com a função *read*, o programa recebe o número M que indica quantos valores serão organizados.

A ordenação é feita com dois *loops DO*. O primeiro percorre de 1 até M, e dentro dele o segundo compara os valores da lista. Sempre que um número for menor que o outro, eles trocam de posição, garantindo que os menores valores fiquem nas primeiras posições.

Quando essa etapa termina, os M primeiros valores já estão em ordem crescente. Por fim, esses valores são gravados em um arquivo de saída.

Pseudo código:



Código:

```
program ex3
    parameter(idimax=1e6)
    real*8 rr
    real*8 lista
    dimension lista(idimax)

    iN = idimax
    open(unit=3,file='entrada-1-12694394.txt')
    open(unit=4,file='saida-1-12694394.txt')
    ic = 0
    do i =1,iN
        read(3,*,End=1) lista(i)
        ic = ic + 1
    end do
1   write(*,*) ic

    read(*,*) iM

    do i = 1,iM
    do j = 1,iM
        if (lista(i) .LT. lista(j)) then
            temp = lista(i)
            lista(i) = lista(j)
            lista(j) = temp
        end if
    end do
    end do

    write(4,8) iM
    do i=1,iM
        write(4,7) lista(i)
    end do
7   format(F12.6)
8   format(I8)
end program ex3
```

Tarefa 4

Nessa tarefa é pedido para calcular o *ln* de um valor x com precisão de $\epsilon = 10^{-5}$ e comparar o valor com a função *log* intrínseca do FORTRAN. A série utilizada para calcular o cosseno está descrita na Equação - 4.

$$\ln(x) = -\sum_{k=1}^{n} \frac{(1-x)^k}{k} \tag{4}$$

Isso foi implementado através do programa ilustrado abaixo, ao qual foi utilizado uma *função* para melhor organizar implementação. A principal ideia desenvolvida, foi analisar a precisão da série, isso foi feito sistematicamente subtraindo os valores das iterações i e $i+1$ até que o resultado seja menor ou igual ao erro $\epsilon = 10^{-5}$.

Além disso, é pedido para calcular qual o valor de ϵ é necessário para que a série utilizada seja equivalente a função *dlog* do FORTRAN.

Isso posto, a principal ideia implementada no código foi comparar os valores da série e da função do FORTRAN até que elas sejam iguais. Ademais, observou-se que o valor necessário de ϵ para que isso ocorra é aproximadamente $\epsilon = 10^{-8}$.

```

      program main
            complex*16 meu_ln
            real*8 x,x2,tol

            open(unit=1,file='saida-1-12694394.txt')
            write(1,*) 'x   meu_ln(x)       log(x)'
            tol = 1d-9
            x= 1d0
            do while (x .LT. 10d0)
                  x2 = abs(meu_ln(x,tol))
                  write(1,3) x, x2,dlog(x),abs(x2 - dlog(x))
                  x = x + 0.1d0

            end do
3           format(F16.8,F16.8,F16.8,F16.8)
            close(1)

      end program main


      function meu_ln(x,tol)
      complex*16 meu_ln
      real*8 x,tol,pi
```

```
      real*8 x1,rr,ri,k,slk

      x_ini = x   ! guarda valor original

      pi = acos(-1d0)
      slk  = 1d0
      if (x .LT. 1d0) then
              x = 1d0/x
      endif

      k = 1d0
      x1 = x
      do while (x1 .GE. 2d0)
              k = k * 2d0
              x1 = x ** (1d0/k)
      end do

      meu_ln = 0d0
      rr = 1d0
      i = 1
      do while (abs(rr) .GE. tol)
              ri = i
              rr  = -((1d0 - x1)**i)/ri
              meu_ln = meu_ln + rr
              i = i + 1
      end do

      meu_ln = slk * k*meu_ln

      !caso x seja negativo
      if (x_ini .LT. 0d0) then
      meu_ln = meu_ln + (0d0, 1d0)*pi
      endif

      return
      end function meu_ln
```

Tarefa 6

Nesse problema, o objetivo é calcular o volume de uma esfera em d dimensões, utilizando dois métodos: o **analítico** e o de **Monte Carlo**.

O método de Monte Carlo, de forma geral, consiste em realizar amostragens aleatórias para obter aproximações numéricas. Um exemplo clássico de aplicação é a estimativa do valor de π, onde se sorteiam pontos dentro de um quadrado e se conta quantos caem dentro de um círculo inscrito. A proporção entre pontos dentro e fora fornece uma aproximação para o valor procurado.

Seguindo essa lógica, o programa em Fortran implementa os seguintes passos para calcular o volume da n-esfera:

Descrição do código

O programa principal define o número de tentativas do método de Monte Carlo, $M = 10^7$, o raio da esfera ($r = 1$).

Para cada dimensão:

1. Calcula-se o volume **analítico** da esfera chamando a função `vol(r,d)`.
2. Calcula-se o volume por **Monte Carlo** chamando a função `carlo(r,d,M)`.
3. Os resultados são exibidos na tela, junto com o erro absoluto entre os dois métodos.

Tabela 1 – Comparação entre volumes analítico e Monte Carlo de esferas unitárias.

Dimensão d	**Volume Analítico**	**Monte Carlo**	**Diferença**
2	3.14159265	3.14285600	0.00126335
3	4.18879015	4.18788800	0.00090215
4	4.93480220	4.93496960	0.00016740

Fonte: Elaborado pelo Autor.

Função `f(x)`

Essa função implementa a generalização do fatorial, isto é, a **função gama** $\Gamma(x)$.

- Para valores inteiros, retorna o fatorial tradicional.
- Para valores não inteiros, utiliza a relação com a raiz quadrada de π, conforme a definição da função gama.

```
real*8 function f(x)
        real*8 x,pi
        pi = acos(-1.e0)
        f = 1e0
        if (mod(x,1e0) .NE. 0e0) then
                do while (x .GT. 0.5e0)
                        x = x-1e0
                        f = f * x
                end do
                        f = f * sqrt(pi)
        else
                i = x
                do j = 1,i-1
                        f = f * j
                end do
        end if
return
end function f
```

Função vol(r,d)

Implementa a fórmula analítica do volume da esfera em d dimensões:

$$V_d(r) = \frac{\pi^{d/2} \cdot r^d}{\Gamma\left(1 + \frac{d}{2}\right)}$$

Nessa função, a chamada `f(1d0 + d/2d0)` corresponde exatamente ao cálculo da função gama.

```
real*8 function vol(r,d)
        real*8 r,d,pi,f
        pi = acos(-1d0)
        vol = (pi**(d/2d0))*(r**d)/f(1d0+(d/2d0))
return
end function vol
```

Função carlo(r,d,M)

Realiza o cálculo do volume por amostragem Monte Carlo:

- Gera M pontos aleatórios uniformemente distribuídos no hipercubo $[-r, r]^d$.

- Para cada ponto, calcula a distância ao centro utilizando a soma dos quadrados das coordenadas.
- Se a distância for menor que o raio, o ponto é contado como "dentro" da esfera.
- Ao final, o volume é estimado pela razão entre pontos dentro e o total de pontos, multiplicada pelo volume do hipercubo:

$$V_d(r) \approx \frac{N_{\text{dentro}}}{M} \cdot (2r)^d$$

```
real*8 function carlo(r,id,M)
        real*8 c,r,p,rM
        c = 0
        rM=M
        do i = 1,M
                p = 0
                do j = 1,id
                        p = p + (2d0*rand() -1d0)**2
                end do

                if (p .LT. r*r) then
                        c = c + 1d0
                end if
        end do

        carlo = (c/rM)*((2d0*r)**id)
end function carlo
```

Código completo

```
            program main
                parameter(M=1e7)
                real*8 d,r,vol,carlo,x1,x2
                r = 1d0

                do i =2,4
                d = i
                x1= vol(r,d)
                x2 = carlo(r,i,M)
                        write(*,*) i,x1,x2,abs(x1-x2)
                end do
7               format(I3,F12.8,F12.8)

```

```
        end program main

        real*8 function f(x)
                real*8 x,pi
                pi = acos(-1.e0)
                f = 1e0
                if (mod(x,1e0) .NE. 0e0) then
                        do while (x .GT. 0.5e0)
                                x = x-1e0
                                f = f * x
                        end do
                                f = f * sqrt(pi)
                else
                        i = x
                        do j = 1,i-1
                                f = f * j
                        end do
                end if
        return
        end function f

        real*8 function vol(r,d)
                real*8 r,d,pi,f
                pi = acos(-1d0)
                vol = (pi**(d/2d0))*(r**d)/f(1d0+(d/2d0))
        return
        end function vol


        real*8 function carlo(r,id,M)
                real*8 c,r,p,rM
                c = 0
                rM=M
                do i = 1,M
                        p = 0
                        do j = 1,id
                                p = p + (2d0*rand() -1d0)**2
                        end do

                        if (p .LT. r*r) then
                                c = c + 1d0
                        end if
                end do

                carlo = (c/rM)*((2d0*r)**id)
        end function carlo
```

Tarefa 7

Nesse programa é pedido para utilizar a forma analítica do volume de uma esfera, Equação-5, para criar um gráfico que mostra o volume por dimensão.

$$V_d = \frac{\pi^{d/2}}{\Gamma(1 + d/2)} R^d \tag{5}$$

Funcionamento do programa

O maior desafio para implementar o programa é definir a função Gamma, por isso esse será o foco dessa discussão.

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} \, \mathrm{d}t, \qquad \Re(z) \tag{6}$$

A função gamma é a generalização do fatorial, sua definição é dada pela Fórmula-6, algumas de suas propriedades são $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(1) = 1$ e $\Gamma(x + 1/2) = x\Gamma(x)$. Desse modo, com essas informações, e sabendo que a função Gamma deve se comportar como o fatorial para números inteiros. O código para essa função foi implementado através dos seguintes passos:

- Recebe o valor d, que representa o número de dimensões.
- Utilizando um *IF* vê se a soma de x é um numero inteiro.
- Se o resultado for um numero inteiro, utiliza a definição do fatorial para realizar o cálculo.
- Caso o resultado não seja um numero inteiro utilizamos a propriedade $\Gamma(x + 1/2) = x\Gamma(x)$ para resolver o problema.

Número inteiro (d é múltiplo de 2)

Nesse caso, estamos utilizando a seguinte ideia, definimos uma constante como sendo igual a 1, no casso essa variável representa o zero fatorial, e utilizando um *loop* multiplicamos o numero da iteração i pela variável, realizamos esse processo $1 + d/2$ vezes. Desse modo, conseguimos $(1 + d/2)!$, a Figura-**??**, demostra o código aplicado nesse caso.

```fortran
real*8 function f(x)
        real*8 x,pi
        pi = acos(-1.e0)
        f = 1e0
        if (mod(x,1e0) .NE. 0e0) then
                do while (x .GT. 0.5e0)
                        x = x-1e0
                        f = f * x
                end do
                        f = f * sqrt(pi)
        else
                i = x
                do j = 1,i-1
                        f = f * j
                end do
        end if
return
end function f
```

Número não inteiro (d não é múltiplo de 2)

Nesse caso, é critico utilizar a propriedade $\Gamma(x+1/2) = x\Gamma(x)$ para reduzirmos o problema até $\Gamma(1/2) = \sqrt{\pi}$, ademais, é necessário notar que os casos no qual é preciso implementar a função gamma ocorrem apenas em dimensões impares. Isso posto, a Equação -7 representa o algoritimo que devemos implementar no código.

$$\Gamma(1+3/2) = \frac{3}{2}\Gamma(3/2) = \frac{3}{2}\Gamma(1+1/2) = \frac{3}{2}\cdot\frac{1}{2}\Gamma(1/2) = \frac{3}{2}\cdot\frac{1}{2}\cdot\sqrt{\pi} \tag{7}$$

Isso foi feito da seguinte forma, sabendo que o maior coeficiente que aparece multiplicando a função gamma é o valor de dimensão, d, e como demostrado na Equação-7, só iremos utilizar números impares. Portanto, foi implementado um *loop* que realiza d interações onde uma variável com valor $\sqrt{\pi}$ é multiplicada pelo numero da i da interação dividido por 2, o valor dado para essa variável *dummy* representa o caso onde $\Gamma(1/2)$.

Ademais, foi apos definir a função f ela foi utilizada para criar uma outra função *vol* que calcula o volume de uma n-esfera de raio arbitrário.

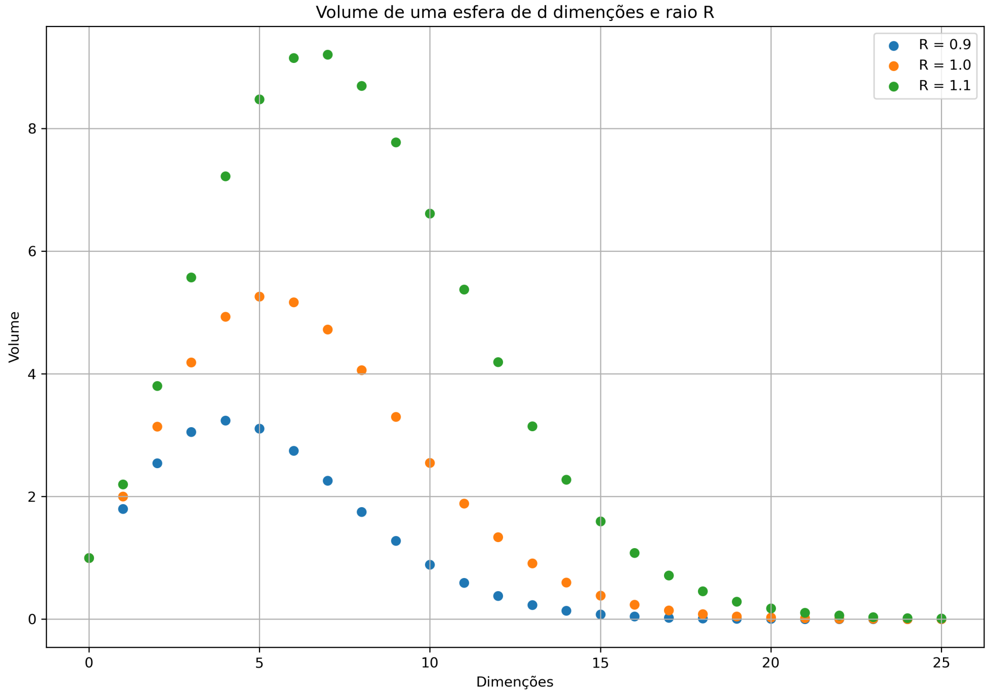
```fortran
real*8 function vol(r,d)
        real*8 r,d,pi,f
        pi = acos(-1d0)
        vol = (pi**(d/2d0))*(r**d)/f(1d0+(d/2d0))
return
end function vol
```

Isso posto, o código completo para a definição da função Gamma é apresentado abaixo:

```
        program main
                real*8 d,r,vol
                r = 1.1d0

                open(unit=1,file='saida-3-12694394.txt')
                do i =0,25
                d = i
                        write(1,7) i,vol(r,d)
                end do
                close(1)
7               format(I3,F12.8)

        end program main

        real*8 function f(x)
                real*8 x,pi
                pi = acos(-1.e0)
                f = 1e0
                if (mod(x,1e0) .NE. 0e0) then
                        do while (x .GT. 0.5e0)
                                x = x-1e0
                                f = f * x
                        end do
                                f = f * sqrt(pi)
                else
                        i = x
                        do j = 1,i-1
                                f = f * j
                        end do
                end if
        return
        end function f

        real*8 function vol(r,d)
                real*8 r,d,pi,f
                pi = acos(-1d0)
                vol = (pi**(d/2d0))*(r**d)/f(1d0+(d/2d0))
        return
        end function vol
```

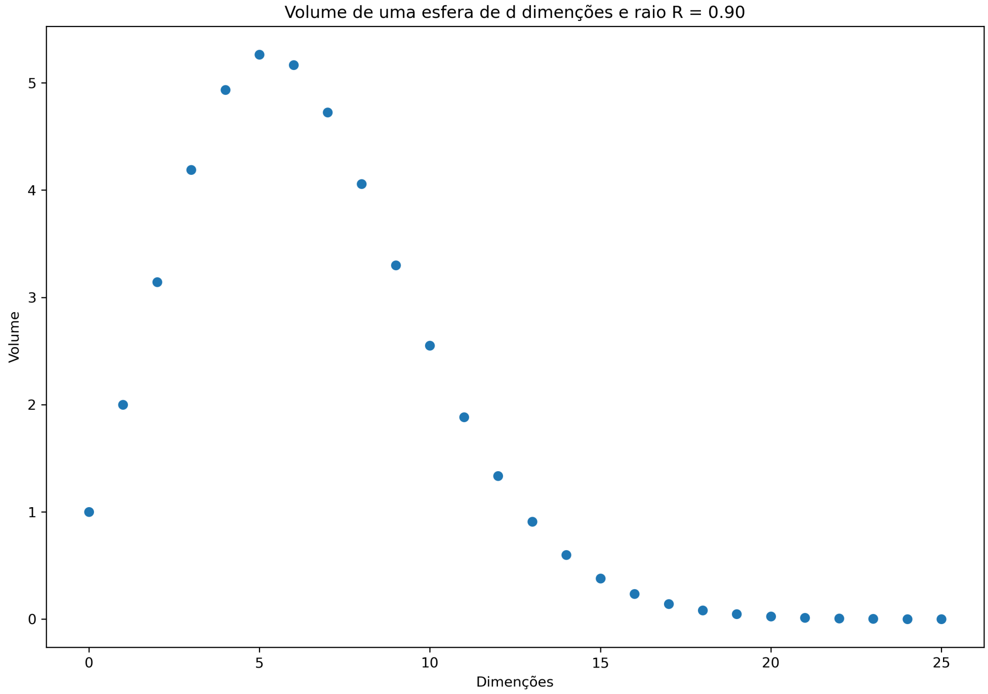
Por fim, nos gráficos abaixo são mostrados os resultados obtidos.

Figura 4 – Gráfico do Volume × Número de dimensões



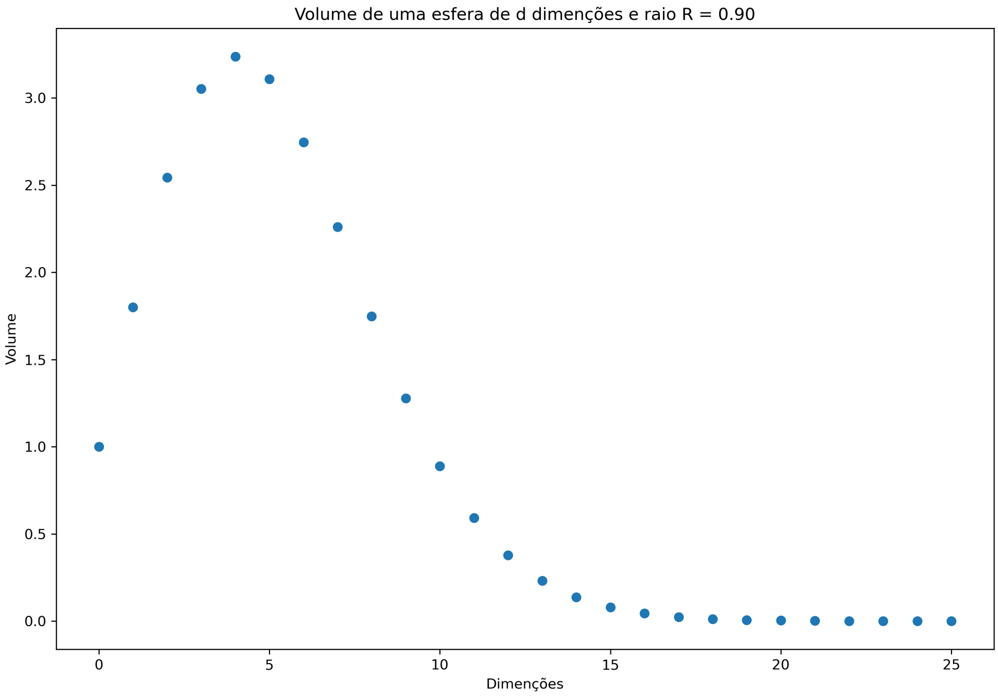
Fonte: Elaborado pelo autor.

Figura 5 – Gráfico do Volume $\times$ Número de dimensões com raio R = 0.90



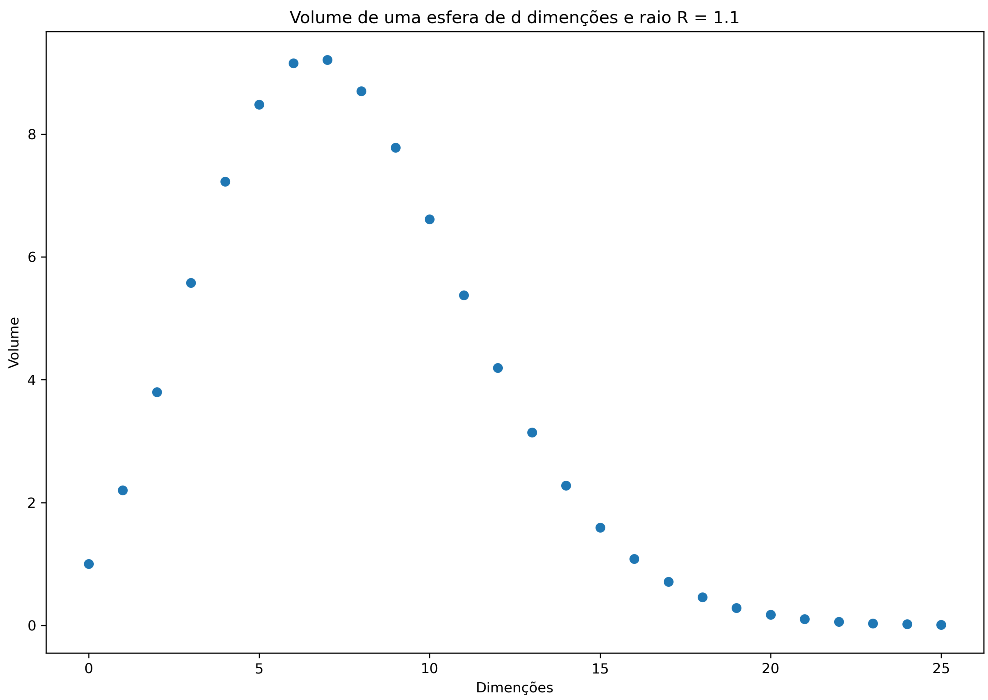
Fonte: Elaborado pelo autor.

Figura 6 – Gráfico do Volume × Número de dimensões com raio R = 1.0



Fonte: Elaborado pelo autor.

Figura 7 – Gráfico do Volume × Número de dimensões com raio R = 1.1



Fonte: Elaborado pelo autor.

Tarefa 8

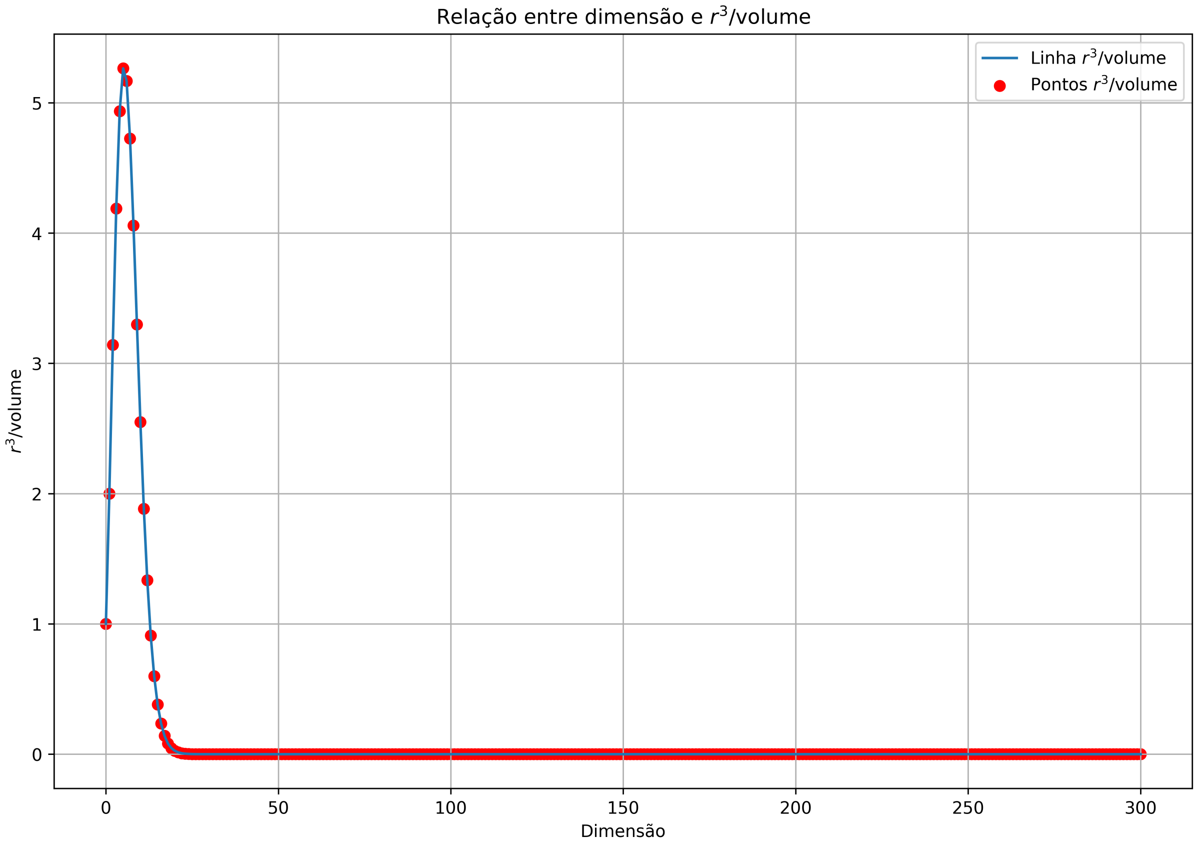
Nesse problema foi utilizado o código da Tarefa 7, no qual a única modificação foi subtrair o valor do volume do cubo em d dimensões. Por considerarmos uma esfera de raio unitário e um cubo com arestas de valor $1[m^d]$ foi necessário apenas realizar uma simples subtração, entre o volume do cubo e da esfera. Isso posto, a modificação realizada em comparação ao código da Tarefa 7, é ilustrada na Figura **??**

```
program main
        real*8 d,r,vol
        r = 1d0

        open(unit=1,file='saida-1-12694394.txt')
        write(1,3)
        do i =0,300
        d = i
        write(1,7) i,vol(r,d)/(r**3),abs((r**3)-vol(r,d))
        end do
        close(1)
7               format(I3,F12.8,F12.8)
3               format('dimensao  r**3/vol  abs(r**3-vol)')
end program main
```

Isso posto, na Figura-9 é ilustrado o resultado, quando a dimensão da esfera tende a infinito o volume tende a zero, portanto o resultado da subtração tende ao volume do cubo.

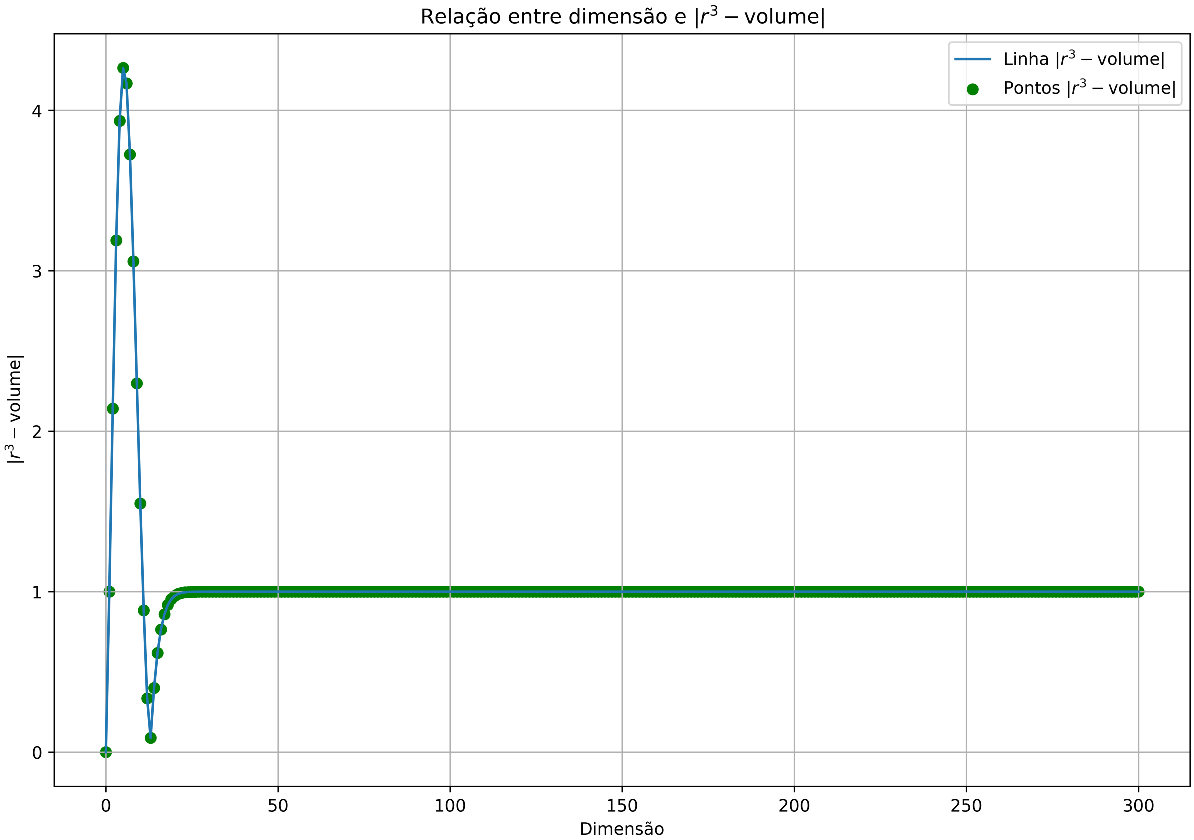
Ademais, em um universo com d dimensões 1 mol corresponderia a equação da Figura - 10. Vale notar, que no caso tridimensional o valor correspondente é aproximadamente o número de Avogadro.

Figura 8 – Gráfico mostrando o resultado da razão do volume do cubo e da esfera quando d tende a infinito.



Fonte: Elaborado pelo autor.

Figura 9 – Gráfico mostrando o resultado da subtração do volume do cubo e da esfera quando d tende a infinito.



Fonte: Elaborado pelo autor.

Figura 10 – Número de Avogadro em d dimensões

$$1 \text{ mol} = \frac{10^{-6}\ m^{D}}{1 \cdot (10^{-10}\ m)^{D}} = 10^{(D \cdot 10 - 6)}$$

Fonte: Elaborado Pelo autor.