João Vítor Lima de Oliveira - 12694394

Relatório Projeto-1

São Carlos

2025

Sumário

Teoria

$$(P)_{k=0}^{k=n} = (P_0, P_1, \dots, P_n) \tag{1}$$

onde P_k é definido como sendo o resto do valor inicial ao fazer k pagamentos. toda vez que um mês é pago o interesse diminui, logo,

$$\begin{aligned} P_{k+1} &= P_k - (A - rP_k) \\ &= P_k(1+r) - A \end{aligned} \tag{2}$$

Mas, eu quero derivar uma formula geral, para A que não seja recursiva,

$$P_1 = P_0(1+r) - A \tag{3}$$

$$\begin{aligned} P_2 &= P_1(1+r) - A \\ &= \big[P_0(1+r) - A\big](1+r) - A \\ &= P_0(1+r)^2 - A(1+r) - A \end{aligned} \tag{4}$$

$$\begin{aligned} P_3 &= P_2(1+r) - A \\ &= \big[P_0(1+r)^2 - A(1+r) - A\big](1+r) - A \\ &= P_0(1+r)^3 - A(1+r)^2 - A(1+r) - A \end{aligned} \tag{5}$$

Tarefa 2

Objetivo

Nesse programa será estudado como calcular o volume e a área de um Torus. Na Figura - **??** 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 **??** para conseguirmos o valor da Área do Torus.

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. Os principais tipos de torus incluem.

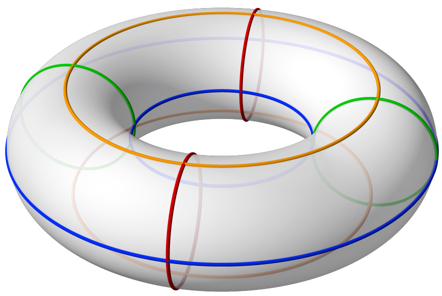
Calculo

As formulas utilizadas para calcular a área e o volume do torus são respectivamente Equação-6 e Equação-7

$$A = \int_0^{2\pi} R d\phi \int_0^{2\pi} r d\theta = 4\pi R r \tag{6}$$

$$V = \int_0^{2\pi} R d\phi \int_0^{2\pi} d\theta \int_0^{r} r' dr' = 2\pi R r^2 \tag{7}$$

Figura 1 – Ilustração de um Torus.



Fonte: Retirado da Wikipedia.

Em vista disso, foi utilizado o *workflow* apresentado na figura 2 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 **??** e **??** respectivamente.

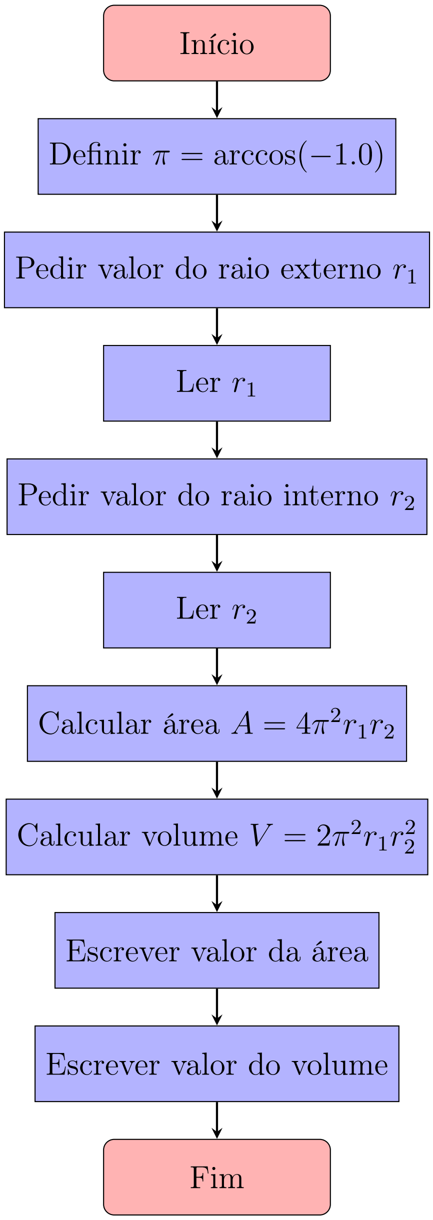


Figura 2 – Fluxograma do cálculo da área e do volume de um toroide

Ademais, o programa em questão está ilustrado na Figura - **??**.

Listing 1 – Cálculo da área e volume de um toro

```
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

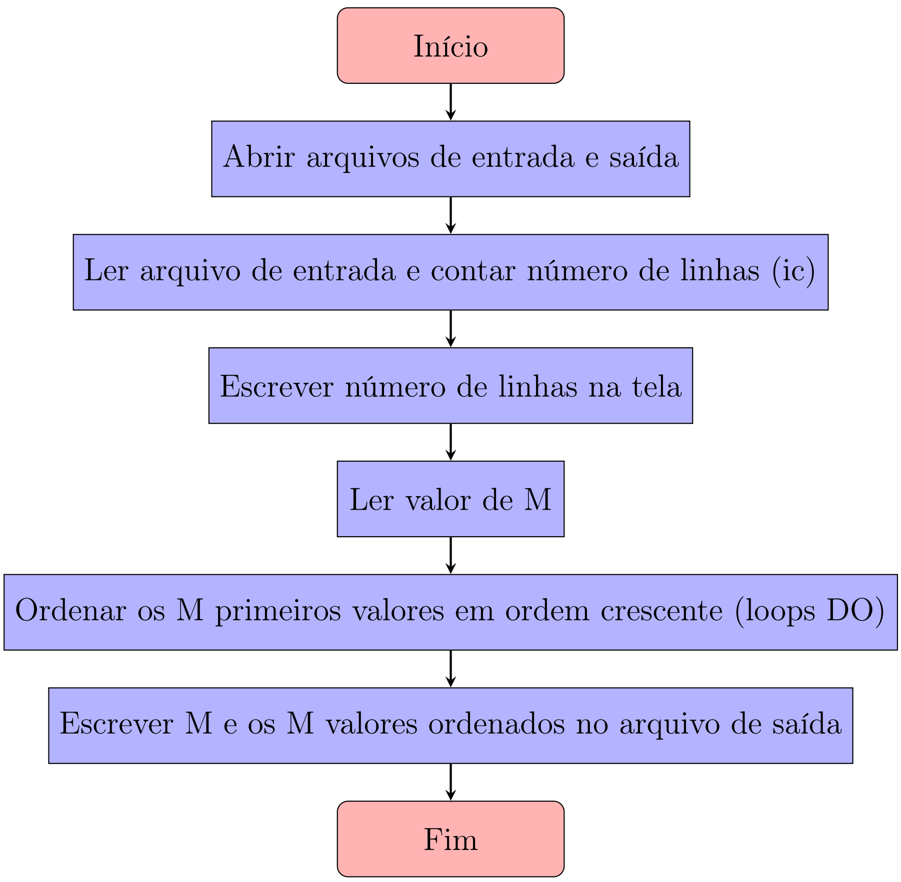
Nesse programa o objetivo é descobrir quantas linhas existem em um arquivo e organizar os M primeiros números em ordem crescente. Para isso, primeiro o arquivo de entrada é aberto com a função *open*. Em seguida, um *loop* percorre o arquivo para contar o número de linhas e guardar os valores em uma 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 cosseno de um valor x com precisão de $\epsilon = 10^{-5}$ e comparar o valor com a função *cos* intrínseca do FORTRAN. A série utilizada para calcular o cosseno está descrita na Equação - 8.

$$\cos x = \sum_{k=1}^{n} \frac{(-1)^k x^{2k}}{(2k)!} \tag{8}$$

Isso foi implementado através do programa ilustrado na Figura - 3, ao qual foi utilizado uma *subrotina* 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 $\epsilon = 10^{-5}$.

Figura 3 – *Subrotina* utilizada para implementar primeira parte da Tarefa.

```fortran
      subroutine rcalc_1(x,iN)
        implicit real*8 (a-h,o-z)

        rcos = 1

        open(UNIT=1, file="saida-4-1-12694394.data")

        do i =1,iN
        rdum = rcos
        rcos = rcos + ((-1)**i)*(x**(2*i))/fac(2*i)
        IF (abs(rcos - rdum) .LE. 1e-5) THEN
            exit
        END IF
        write(1,7) cos(x),rcos,abs(cos(x)-rcos),abs(rdum-rcos)
7       format(F12.8, " , ", F12.8, " , ", F12.8, " , ", F12.8)
        end do
        close(1)
        RETURN
      end subroutine rcalc_1
```

Fonte: Elaborado Pelo autor.

Além disso, é pedido para calcular qual o valor de ϵ é necessário para que a série utilizada seja equivalente a função *dcos* do FORTRAN. Isso foi feito utilizando a *subortina*, ilustrada na Figura - 4.

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 $\epsilon = 10^{-6}$.

Figura 4 – *Subrotina* utilizada para implementar segunda parte da Tarefa.

```fortran
      subroutine rcalc_2(x,iN)
          implicit real*8 (a-h,o-z)

          rcos = 1

          open(UNIT=9, file="saida-4-2-12694394.data")

          do i =1,iN
          rdum = rcos
          rcos = rcos + ((-1)**i)*(x**(2*i))/fac(2*i)

          write(9,7) dcos(x),rcos,abs(rdum-rcos)
7         format(F12.8, " , ", F12.8, " , ", F12.8)

          IF (abs(dcos(x) - rcos) .EQ. 0) THEN
              exit
          END IF

          end do
          close(9)
          RETURN
      end subroutine rcalc_2
```

Fonte: Elaborado Pelo autor.

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.

Funções do código

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:

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

```fortran
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

Nesse

```
            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-9, para criar um gráfico que mostra o volume por dimensão.

$$V_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)} R^d \tag{9}$$

Funcionamento do programa

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

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, \mathrm{d}t, \qquad \Re(z) \tag{10}$$

A função gamma é a generalização do fatorial, sua definição é dada pela Fórmula-10, 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.

```
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 -11 representa o algoritimo que devemos implementar no código.

$$\Gamma(1+3/2) = \frac{3}{2}\Gamma(3/2) = \frac{3}{2}\Gamma(1+1/2) = \frac{3}{2}\cdot\frac{1}{2}\Gamma(1/2) = \frac{3}{2}\cdot\frac{1}{2}\cdot\sqrt{\pi} \tag{11}$$

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-11, 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 $\Gamma(1/2)$.

```
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 na Figura-??.

```
        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
```

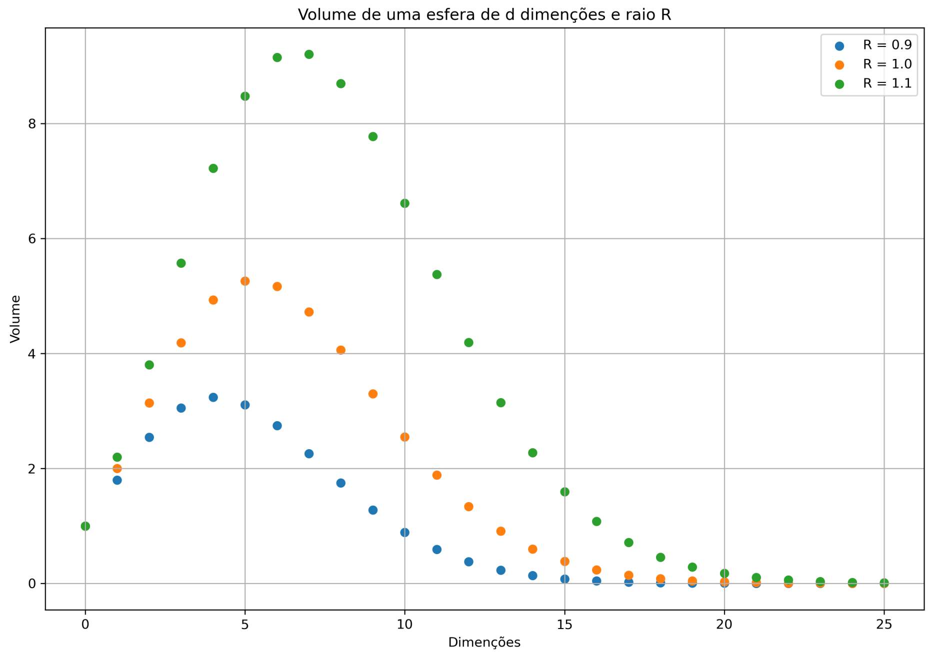
Por fim, utilizando uma *subrotina* e um simples *loop* os volumes referentes as d dimensões foram salvos em um arquivo de saída. Um gráfico utilizando esses dados é mostrado na Figura-9.

Figura 5 – *Subrotina* que implementa a equação analítica.

```
        subroutine rcalc(id,rrad)
            implicit real*8 (a-h,o-z)
            parameter (pi = acos(-1.0))

            open(UNIT=1,FILE="dados.data")
            do i = 0, id
                rvol = ((sqrt(pi)*rrad)**(i))/rgamma(i)
                write(1,7) rvol      , i
7               format(F10.3,",", I6)
            end do

            close(1)
            end subroutine rcalc
```

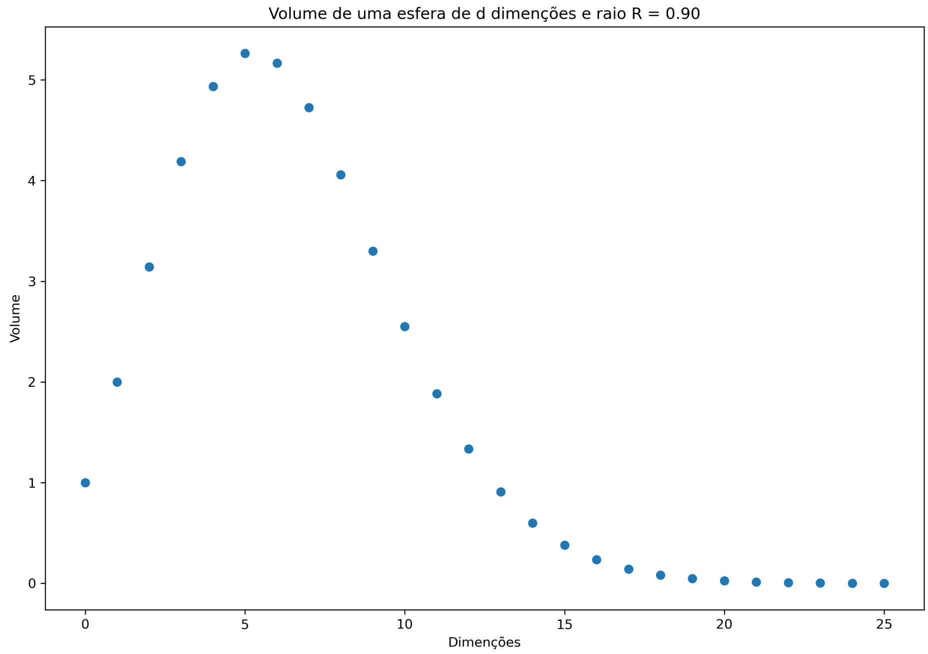
Fonte: Elaborado pelo autor.

Figura 6 – Gráfico do Volume × Número de dimensões



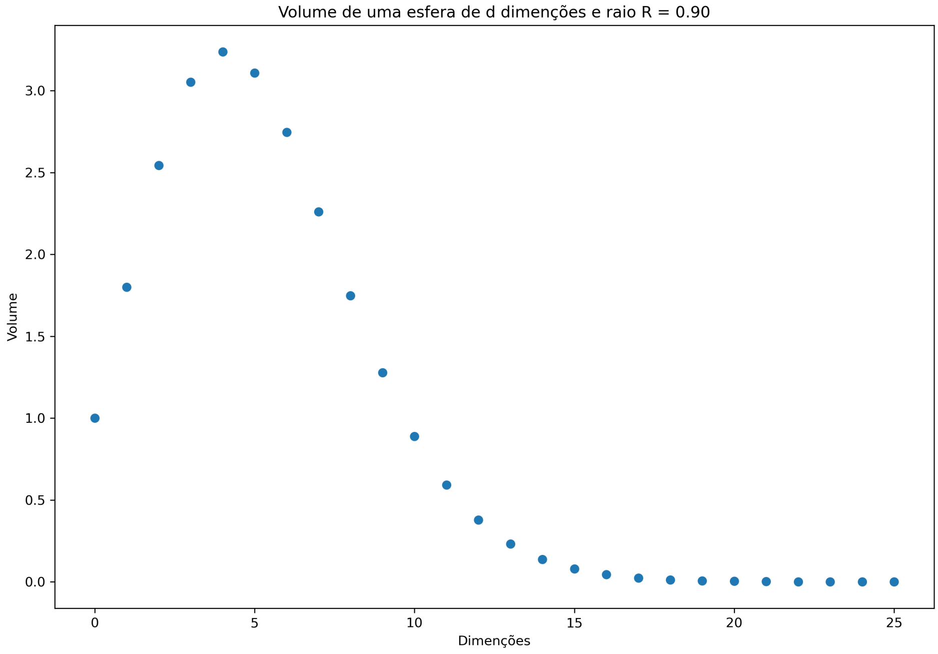
Fonte: Elaborado pelo autor.

Figura 7 – Gráfico do Volume × Número de dimensões com raio R = 0.90



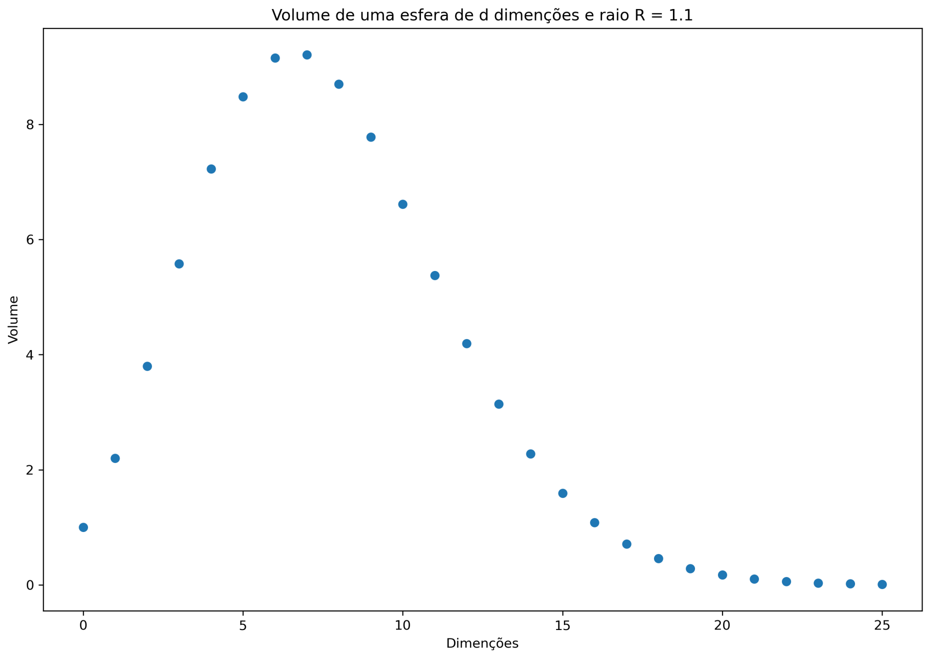
Fonte: Elaborado pelo autor.

Figura 8 – Gráfico do Volume × Número de dimensões com raio R = 1.0



Fonte: Elaborado pelo autor.

Figura 9 – Gráfico do Volume $\times$ 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 10

Figura 10 – Código da Tarefa-8

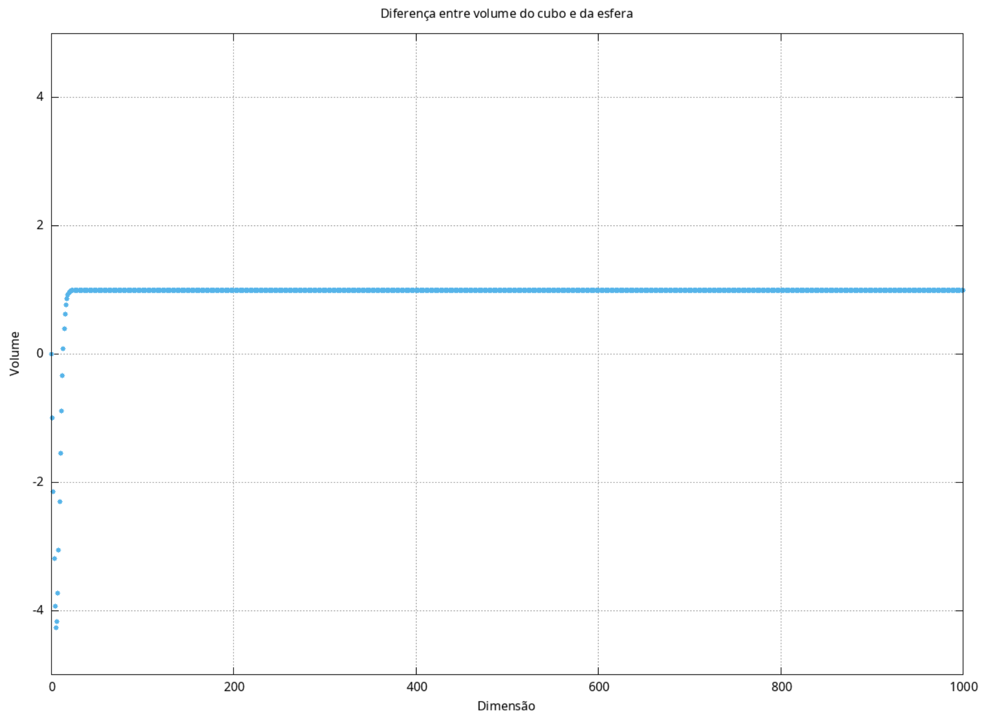
```
open(UNIT=1,FILE="dados.data")
do i = 0, id
    rvol = ((sqrt(pi)*rrad)**(i))/rgamma(i)
    write(1,7) i,rvol_cube - rvol
7   format(I6,",", F12.6)
end do

close(1)
```

Fonte: Elaborado pelo autor.

Isso posto, na Figura-11 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 - 12. Vale notar, que no caso tridimensional o valor correspondente é aproximadamente o número de Avogadro.

Figura 11 – Gráfico mostrando o resultado da subtração do volume do cubo e da esfera quando d tende a infinito.



Fonte: Elaborado Pelo autor.

Figura 12 – 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.