

Bolhas em Cilindro

Inputs:

- Número de Bolhas
- Raio do Cilindro
- Altura do Cilindro

u = time of arrival

t = número de entrada (número da interação)

```
In[1]:= Cilindro[r_, h_] := ParametricPlot3D[{r Cos[\theta], r Sin[\theta], z}, {\theta, 0, \pi}, {z, 0, h}, Mesh -> None, PlotStyle -> Directive[Opacity[1], Red], BoundaryStyle -> Directive[Black, Thin], ViewPoint -> {0, -\infty, 0}, Lighting -> {"Ambient", Red}]
```

```
In[2]:= Cilindro2[r_, h_] := ParametricPlot3D[{r Cos[\theta], r Sin[\theta], z}, {\theta, 0, 2 \pi}, {z, 0, h}, Mesh -> None, PlotStyle -> {Opacity[0.5], LightGray}]
```

```
Bolhas[n_, raio_, h_] :=
  bubbles = {{}, {}, {}, {}};

  numero = RandomVariate[PoissonDistribution[n]];
  For[j = 0, j < numero, j++;
    Rzone = raio * Sqrt[RandomReal[1]];
    \theta = RandomReal[2 \pi];
    xvalue = Rzone Cos[\theta];
    yvalue = Rzone Sin[\theta];
    zvalue = RandomReal[0, h];
    rvalue = RandomReal[{0.01, 1.01}];
    u = RandomReal[{0, 1}];
    t = j;

    If[Length[bubbles[[2]]] == 0,
      AppendTo[bubbles[[1]], {xvalue, yvalue, zvalue}]; AppendTo[bubbles[[2]], rvalue];
    ]
```

```

AppendTo[bubbles[[3]], u]; AppendTo[bubbles[[4]], t]; j = j + 1];
[adiciona a]

touch = {};
For[k = 0, k < Length[bubbles[[1]]], k++;
[..] [comprimento]
    l = k + 1;
    u1 = bubbles[[3, k]]; u2 = bubbles[[3, l]]; t1 = bubbles[[4, k]]; t2 = bubbles[[4, l]];
    pbubble1 = bubbles[[1, k]]; rbubble1 = bubbles[[2, k]]; pbubble2 = bubbles[[1, l]]; rbubble2 = bubbles[[2, l]]; Δx = pbubble1[[1]] - pbubble2[[1]]; Δy = pbubble1[[2]] - pbubble2[[2]]; Δz = pbubble1[[3]] - pbubble2[[3]]; If[
[se] √(Δx^2 + Δy^2 + Δz^2) > (rbubble1 + rbubble2),
[adiciona a]

If[FreeQ[touch[k], 0], Delete[bubbles[[1]], k]; Delete[bubbles[[2]], k];
[... [livre de?] [deleta] [deleta]
    Delete[bubbles[[3]], k];
[deleta]
    AppendTo[burbujas[[4]], t1], ]];
[adiciona a]

If[FreeQ[touch, 0], AppendTo[burbujas[[1]]
[livre de?] [adiciona a]
    x
For[
[para cada]

AppendTo[bubbles,
[adiciona a]

$$\left( \text{Total} \left[ \text{Table} \left[ \frac{4}{3} \pi (\text{bubbles}[[2, i]])^3, \{i, 1, \text{Length}[\text{bubbles}[[2]]]\} \right] \right] \right)$$

[total] [tabela] [comprimento]

bubbles]

```

```

In[2]:= VistaBolhas2D[n_, r_, h_] := (bolhas = Bolhas[n, r, h];
Show[Graphics3D[{Opacity[1], Black,
[representação] [opacidade] [preto]
Sphere[bolhas[[1]], bolhas[[2]]], ViewPoint -> {0, -Infinity, 0}}],
[esfera] [ponto de vista] [infinito]
Cilindro[r, h], Lighting -> {"Ambient", Red}, ViewPoint -> {0, -∞, 0}]])
[iluminação] [vermelho] [ponto de vista]

```

```
In[1]:= VistaBolhas3D[n_, r_, h_] := (bolhas = Bolhas[n, r, h];
  Show[Graphics3D[{Opacity[0.5], Sphere[bolhas[[1]], bolhas[[2]]]}],
    |representação| opacidade |esfera
    Cilindro2[r, h]])
```

```
In[2]:= grafico [n0_, r0_, h0_] :=
  (img =
    ColorSeparate[Image[VistaBolhas2D[n0, r0, h0], "Byte", ImageResolution -> 300]];
    |separação de cores| imagem |byte| resolução da imagem
    myLevels = {ImageLevels[img[[1]], 2], ImageLevels[img[[4]], 2]};
    |níveis de imagem| |níveis de imagem|
    vazio = 1 - N[myLevels[[1, 2, 2]] / myLevels[[2, 2, 2]]];
    |valor numérico|
    areareal = ((2 * r0 * h0) * vazio) / (2 * r0)^2;
    AppendTo[listaproj, {bubbles[[3]] / ((2 * r0)^3), areareal}];
    |adiciona a|
    listaproj);
```

As duas funções a seguir servem para exportar para um .csv e depois de aí exportar para o python - pandas para fazer o gráfico

```
In[3]:= GraficoExport[n0_, r0_, h0_] :=
  (img =
    ColorSeparate[Image[VistaBolhas2D[n0, r0, h0], "Byte", ImageResolution -> 300]];
    |separação de cores| imagem |byte| resolução da imagem
    myLevels = {ImageLevels[img[[1]], 2], ImageLevels[img[[4]], 2]};
    |níveis de imagem| |níveis de imagem|
    vazio = 1 - N[myLevels[[1, 2, 2]] / myLevels[[2, 2, 2]]];
    |valor numérico|
    areareal = ((2 * r0 * h0) * vazio) / (2 * r0)^2;
    AppendTo[listaproj, {bubbles[[3]] / ((2 * r0)^3), areareal}];
    |adiciona a|
    listaproj);
```

```
In[4]:= PlotExport[n_, r_, h_, v_] :=
  (lista3 = {};
   listaproj = {};

   Do[lista3 = GraficoExport[n, r, h], {v}];
   |repete|
   Export["1200b_4x_vertical", lista3, "CSV"]
   |exporta|
  );
```

Produzindo gráficos

```
In[4]:= PlotGeral[n_, r_, h_, v_, colour_] :=
(
  lista2 = {};
  listaproj = {};

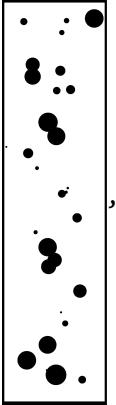
  Do[lista2 = grafico[n, r], {v}];
  |repete

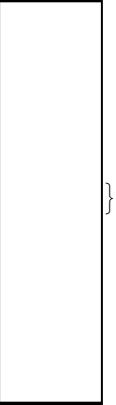
  ListPlot[lista2, PlotTheme -> "Web", FrameLabel ->
    |gráfico de uma lista d... tema do gráfico |legenda do quadro
    {{HoldForm[ $\frac{\text{HoldForm[Area]}}{D^2}$ ], None}, {HoldForm[ $\frac{\text{HoldForm[Volume]}}{D^3}$ ], None}},
    |forma sem avaliação |nenhum |forma sem avaliação |nenhum

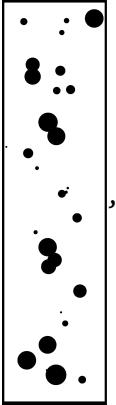
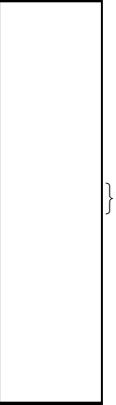
    PlotLabel -> None, LabelStyle -> {GrayLevel[0]}, PlotStyle -> Hue[colour]]
  |etiqueta de gr...|nenh...|estilo de etiqueta |tons de cinzas |estilo do gráfico|matiz
)
```

O processo de “tratamento” da imagem e contagem de pixels ocorre da seguinte forma:

```
In[5]:= img1 = ColorSeparate[Image[VistaBolhas2D[30, 9.5, 76], "Byte", ImageResolution -> 300]];
|separação de co...|imagem |byte |resolução da imagem
```



```
Out[5]= {, , , }
```



```
In[6]:= myLevels = {ImageLevels[img1[[1]], 2], ImageLevels[img1[[4]], 2]};
|níveis de imagem |níveis de imagem
```

```
Out[6]= {{0., 128037}, {0.5, 683763}, {{0., 60878}, {0.5, 750922}}}
```

```
N[myLevels[[1, 2, 2]] / myLevels[[2, 2, 2]]]
|valor numérico
```

```
0.925969
```

O processo de criação de gráfico é o da função PlotGeral.

n = número de bolhas

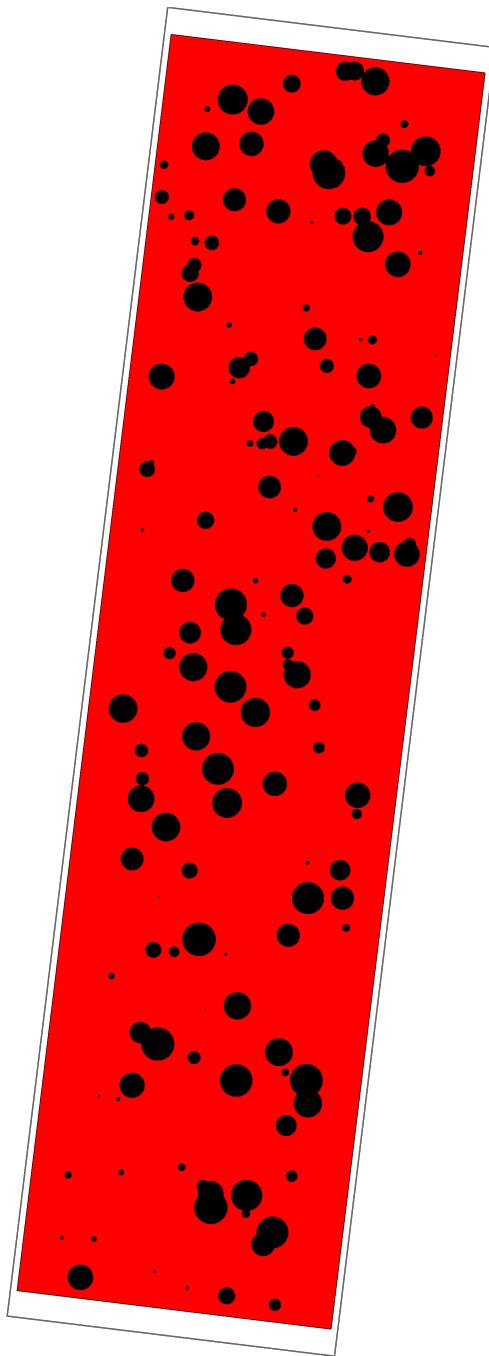
r = raio do cilindro

h = altura do cilindro

v = vezes de repetição da ação com o Do

Ao criar x cilindros com as mesmas especificações [n, raio, h], chegamos ao seguinte tipo de gráfico e processo:

In[\circ]:= **VistaBolhas2D[158, 9.5, 76]**



Out[\circ]:=



Out[\circ]= { {0., 11729}, {0.5, 3271} }

In[\circ]:= **VistaBolhas3D**[100, 9.5, 76]



