MATERIAL COMPLEMENTAR

PRODUTO EDUCACIONAL DA DISSERTAÇÃO "SIMULAÇÕES COMPUTACIONAIS 3D NO VPYTHON: UM AMBIENTE DE APRENDIZAGEM PARA A FÍSICA DOS FLUIDOS"

ALUNA: BRUNA CAMARGO SOARES DE ASSIS

ORIENTADOR: FABRIZIO MYAKI ALVES

A FÍSICA DOS FLUIDOS E O PRODUTO EDUCACIONAL

Este material destacado da dissertação é um complemento que visa descrever os principais aspectos do nosso produto educacional constituído por três animações computacionais que simulam algumas situações típicas decorrentes da física dos fluidos. O conteúdo desse texto servirá como um guia para o uso do nosso produto e é constituído por três partes:

A) uma breve introdução a física dos fluidos que permitirá auxiliar o usuário no melhor entendimento do produto educacional;

B) a descrição das nossas animações mostradas através de ilustrações e acompanhadas das suas principais características pedagógicas;

C) o código das animações escritas na linguagem de programação python.

I. A física dos fluidos: informações preliminares

Embora exista uma certa dificuldade em classificá-lo, um fluido é normalmente definido como uma substância que pode escoar e portanto não possui uma forma própria como acontece com os sólidos. O escoamento é uma tendência promovida pela ação de forças tangenciais que atuam sobre as “camadas” do fluido de modo a deslizar umas sobre as outras. Impedir ou barrar esse escoamento somente por forças perpendiculares na sua superfície, é o que ocorre quando um líquido está dentro de um recipiente. Assim, é mais apropriado tratar a física do fluido por meio dos conceitos de pressão, que está associado a uma força exercida perpendicularmente numa superfície, e massa específica (ou densidade) conveniente para descrever distribuições macroscópicas com aspecto contínuo. Vale lembrar que a física de fluidos é o alicerce para explicar o comportamento de líquidos e gases, no entanto faremos daqui em diante a nossa discussão voltada exclusivamente para os líquidos que é o propósito desse trabalho.

A densidade ρ de um líquido é definida como a razão entre a sua massa e o volume que ocupa. De maneira geral, utilizamos esta definição considerando uma pequena porção do líquido com massa Δm ocupando um volume ΔV, ou seja,

$$\rho = \frac{\Delta m}{\Delta V} \quad . (1)$$

Se a massa do líquido estiver distribuída uniformemente, a densidade será a mesma em todas as suas partes, então,

$$\rho=\frac{m}{V} \quad , (2)$$

sendo m e V, a massa e o volume totais do líquido respectivamente.

Já a pressão mede a força ΔF aplicada a uma pequena superfície de área ΔA, de modo que,

$$p=\frac{\Delta F}{\Delta A} \quad . (3)$$

Teoricamente, a pressão em qualquer ponto no líquido é o limite dessa razão quando a área ΔA com o centro nesse ponto tende a zero. Entretanto, se a força F é uniforme em uma superfície plana de área A podemos escrever a expressão anterior como,

$$p=\frac{F}{A} \quad . (4)$$

A unidade de pressão no S.I. é o Newton por metro quadrado (N/m^2), também designada por Pascal (Pa). A relação entre o Pascal e outras unidades é a seguinte:

$$1\ atm = 1{,}01 \times 10^5\ Pa = 14{,}7\ lb/in^2.$$

Quando um corpo é imerso num líquido de densidade ρ confinado num recipiente aberto verifica-se que a pressão p do líquido experimentado pelo corpo varia linearmente com a sua profundidade h, a partir da pressão atmosférica p_0, isto é,

$$p=p_0+\rho g h \quad , (5)$$

em que g é a aceleração da gravidade local. Verifica-se também que essa pressão é a mesma em todos os pontos do líquido sob o mesmo h.

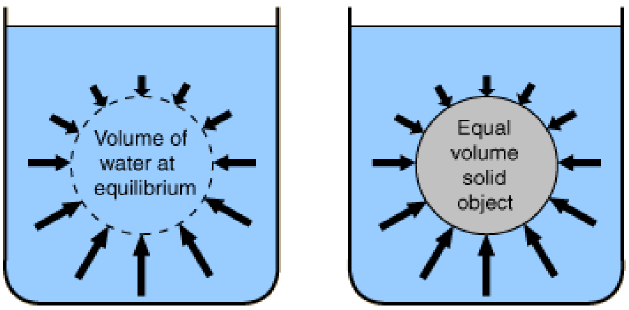

Figura 4: Surgimento da força empuxo. Fonte: http://hyperphysics.phyastr.gsu.edu/hbase/pbuoy.html.

Se lembrarmos que esse corpo é extenso, ele sofrerá pressão do líquido ao redor, de acordo com (5), em todos os pontos em sua superfície. Numa outra perspectiva podemos interpretar esse problema como forças atuando perpendicularmente nos pontos da superfície, cuja intensidade aumenta com a profundidade, conforme exemplificado na figura 4. Essas forças seriam as mesmas se no volume ocupado pelo corpo preenchêssemos com o mesmo líquido que envolve o corpo, também ilustrado na figura 4. Quando somamos todas as forças, para isso dividindo-as nas suas componentes horizontal e vertical, identificamos que para cada componente horizontal existe uma simétrica oposta que a anula. Isso porém não acontece para as componentes verticais. Esse processo então resulta numa força, denominada de empuxo, atuando no centro de gravidade do corpo e com direção vertical para cima. Em uma situação de equilíbrio verifica-se que o empuxo é igual em módulo a força peso do líquido deslocado pelo corpo,

$$F_e = m_L g \quad , (6)$$

de modo que $m_L = \rho_L V_C$ é a massa do líquido de densidade ρ_L , abrangendo o corpo, e V_C o volume do corpo. Assim, a expressão (6) torna-se,

$$F_e = \rho_L V_C g \quad . (7)$$

Essa discussão está embasada no Princípio de Arquimedes que pode ser enunciado do seguinte modo:

> *Todo corpo total ou parcialmente imerso em um fluido recebe deste um empuxo vertical dirigido para cima, de módulo igual ao peso do fluido deslocado pelo corpo.*

II. Corpo em movimento num líquido

Com o intuito de propor uma explicação e o funcionamento das nossas animações computacionais vamos agora descrever a dinâmica de um corpo em queda quando mergulhado em um líquido. A descrição desse problema está baseada na referência, www.unicamp.br/fea/ortega/aulas/aula17_MovimentoParticulas.ppt, e que pode ser consultada a fim de saber mais detalhes.

Consideremos um corpo de massa *m* movimentando-se verticalmente em um líquido incompressível em repouso de densidade ρ_L constante, utilizaremos o eixo *y* para descrever o seu movimento. O corpo sofre essencialmente a ação de três forças responsáveis pela sua dinâmica, o peso P devido a interação do corpo com a Terra, o empuxo F_e como consequência da interação do corpo com o líquido, discutido anteriormente, e uma força de resistência (arraste) F_R que surge com o movimento do corpo no meio líquido, tal força sempre se opondo ao sentido do movimento. Esta última é representada por,

$$F_R = \frac{1}{2}\rho_L A_c C_d v_y^2 \quad , (8)$$

em que, ρ_L é a densidade do meio líquido, A_c é a área projetada do corpo relativo a sua direção do movimento, C_d é o coeficiente de arraste e v_y é a velocidade do corpo em um instante *t*.

As forças peso e empuxo estão definidas por grandezas que não mudam no tempo, logo elas também são constantes no tempo. Já a força de arraste em que o corpo está submetido é uma função da velocidade que por sua vez depende do tempo.

De acordo com a 2ª lei de Newton temos,

$$P - F_e - F_R = m\frac{dv_y}{dt} \quad . (9)$$

Será útil reescrever a equação (9) fazendo, $k = \frac{P - F_e}{m}$ e $\alpha = \frac{\rho_L A_c C_d}{2m}$, que são constantes na equação, sendo sempre $\alpha > 0$. Temos então,

$$k - \alpha v_y^2 = \frac{dv_y}{dt} \quad , (10)$$

mas para uma investigação completa do problema requer a sua divisão em três casos citados na tabela 1.

$k>0$	$P>F_e$
$k=0$	$P=F_e$
$k<0$	$P<F_e$

Tabela 1: Condição entre as forças P e F_e para cada caso de *k*.

O significado do sinal na constante *k* fica evidente quando substituímos na sua expressão os termos de P e de F_e de modo a obter após algumas simplificações:

$$k = \left(1 - \frac{\rho_L}{\rho_c}\right) g \quad . (11)$$

Além do mais, com embasamento na referência, admitindo que um corpo extenso tenha um "raio" r_p o qual define o seu tamanho característico e que depende da sua forma geométrica, podemos representar α da seguinte maneira,

$$\alpha = \frac{3}{8}\frac{C_d \rho_L}{r_p \rho_c} \quad . (12)$$

Na perspectiva da razão entre as densidades do líquido e do corpo, temos a seguir uma tabela mostrando cada ocorrência de *k*.

$k > 0$	$0 \leq \frac{\rho_L}{\rho_c} < 1$
$k = 0$	$\frac{\rho_L}{\rho_c} = 1$
$k < 0$	$\frac{\rho_L}{\rho_c} > 1$

Tabela 2: Razão entre as densidades do líquido e do corpo para cada caso de *k*.

Resolver portanto esse problema requer a integração da expressão (10) para cada situação *k*,

$$\int_{v_{0y}}^{v_y(t)} \frac{dv_y}{k - \alpha v_y^2} = \int_{t_0}^{t} dt \quad , (13)$$

o instante t_0 refere-se as condições iniciais do corpo, localizado em $y(t_0) = y_0$ e com velocidade $v_y(t_0) = v_{0y}$. Resolvendo as integrais anteriores vamos obter a velocidade $v_y(t)$ em qualquer instante $t > t_0$. Prosseguindo a linha desse cálculo podemos determinar a equação horária que exprime o deslocamento do corpo num líquido,

$$v_y(t) = \frac{dy}{dt} \rightarrow \int_{y_0}^{y(t)} dy = \int_{t_0}^{t} v_y(t)\, dt \quad , (14)$$

e finalmente,

$$y(t) = y_0 + \int_{t_0}^{t} v_y(t)\, dt \quad . (15)$$

A solução analítica final desse problema para, $y(t)$ e $v_y(t)$ válida para $t > t_0$, está apresentada nas tabelas 3, 4 e 5, conforme o domínio de validade da constante *k*:

$0 \le \frac{\rho_L}{\rho_c} < 1$	$y(t) = y_0 + \frac{1}{\alpha} \ln\left[\frac{e^{\sqrt{k\alpha}(t-t_0)} - \gamma e^{-\sqrt{k\alpha}(t-t_0)}}{1-\gamma}\right] \quad , \quad \gamma = \left(\frac{v_{0y} - \sqrt{k/\alpha}}{v_{0y} + \sqrt{k/\alpha}}\right)$
	$v_y(t) = \sqrt{\frac{k}{\alpha}}\left[\frac{1+\gamma e^{-2\sqrt{k\alpha}(t-t_0)}}{1-\gamma e^{-2\sqrt{k\alpha}(t-t_0)}}\right] \quad , \quad \gamma = \left(\frac{v_{0y} - \sqrt{k/\alpha}}{v_{0y} + \sqrt{k/\alpha}}\right)$

Tabela 3: Equações $y(t)$ e $v_y(t)$ referentes a *k>0*.

$\frac{\rho_L}{\rho_c} = 1$	$y(t) = y_0 + \frac{1}{\alpha} \ln\left[1 + v_{0y}\alpha(t-t_0)\right]$
	$v_y(t) = \frac{v_{0y}}{1 + v_{0y}\alpha(t-t_0)}$

Tabela 4: Equações $y(t)$ e $v_y(t)$ referentes a *k=0*.

$\frac{\rho_L}{\rho_c} > 1$	$y(t) = y_0 + \frac{1}{\alpha} \ln\left\{\frac{\cos\left[\arctan\left(\sqrt{\frac{\alpha}{\lvert k\rvert}} v_{0y}\right) - \sqrt{\lvert k\rvert\alpha}(t-t_0)\right]}{\cos\left[\arctan\left(\sqrt{\frac{\alpha}{\lvert k\rvert}} v_{0y}\right)\right]}\right\}$
	$v_y(t) = \sqrt{\frac{\lvert k\rvert}{\alpha}} \tan\left[\arctan\left(\sqrt{\frac{\alpha}{\lvert k\rvert}} v_{0y}\right) - \sqrt{\lvert k\rvert\alpha}(t-t_0)\right]$

Tabela 5: Equações $y(t)$ e $v_y(t)$ referentes a *k<0*.

Queremos agora interpretar o resultado de cada uma dessas equações, começando com as da tabela 3, referindo-se a um corpo em queda mergulhado em um líquido em que $\rho_L < \rho_c$. Nesse caso o corpo sendo mais denso que o líquido desloca-se desacelerado atingindo após um tempo suficiente uma velocidade limite (terminal) não nula, mudando-se nesse instante para um movimento uniforme. Uma situação que evidencia esse caso seria uma bolinha de chumbo liberada para ser mergulhada em um recipiente com água.

O segundo caso está relacionado com as equações da tabela 4, tratando-se de um corpo cuja densidade é a mesma do líquido. As equações descrevem um corpo movendo-se desacelerado até atingir a velocidade nula após algum tempo permanecendo o tempo restante em repouso no interior do líquido. É o que acontece quando um objeto solto num recipiente com água possui a mesma densidade que a água ou bem próxima.

O movimento de um corpo no terceiro caso é bastante interessante e traz novidade revelada pelas equações da tabela 5. O resultado agora refere-se a um corpo, com densidade menor que a do líquido, $\rho_L > \rho_c$, inicialmente movendo-se verticalmente desacelerado e para baixo até alcançar a velocidade zero. Após esse instante o corpo inverte o sentido do seu movimento, mudando para acelerado até retornar a superfície do líquido. Como exemplo, podemos imaginar um bloco de isopor ou de madeira quando mergulhado com velocidade (impulso) inicial na água.

A descrição feita para os 3 casos anteriores está exemplificada pelos gráficos de $y(t)$ na figura 5 e de $v_y(t)$ na figura 6, considerando as seguintes condições iniciais: $y_0 = 2.84\,m$ e $v_{0y} = 2.0\,m/s$.

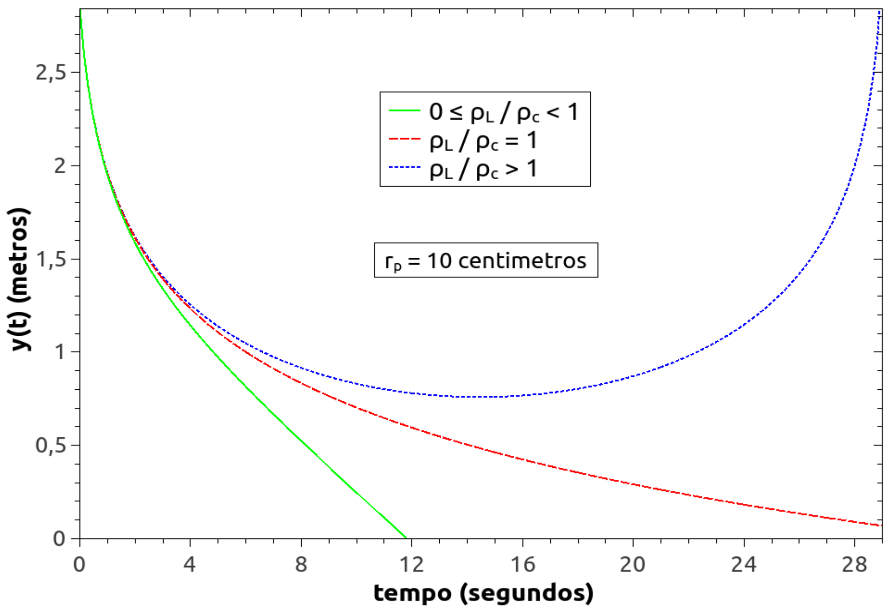

Figura 5: gráfico *y(t) versus t.*

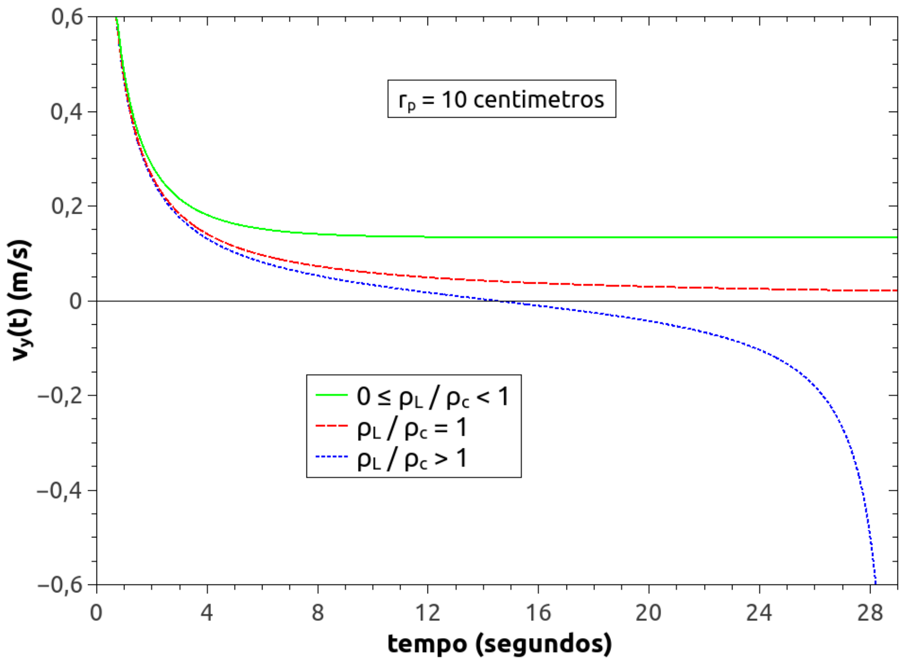

Figura 6: gráfico $v_y(t)$ *versus t.*

III. Escoamento de líquido de um recipiente

Outro tópico importante que nos fornecerá detalhes físicos para o entendimento de uma das nossas animações provém da equação de Bernoulli, em particular do escoamento de um filete líquido por um orifício num recipiente. Consideraremos esse escoamento como estacionário, ou seja, a velocidade do líquido em qualquer ponto fixo não muda com o tempo. Isso significa que a velocidade de um elemento de volume do líquido pode variar quando ele muda de posição, mas a velocidade do líquido em cada ponto fixo no espaço permanece constante no tempo.

A equação de Bernoulli trata de um líquido incompressível escoando em regime estacionário por um recipiente ou tubulação, podendo ocorrer durante esse processo variação de pressão, variação de altura (se houver desnível vertical), e também variação da sua velocidade. Se o líquido ainda for do tipo não-viscoso, situação que será considerada na nossa animação, o problema conserva-se a energia mecânica e a equação de Bernoulli será derivada desse princípio.

Nosso propósito aqui é descrever a dinâmica de uma pequena porção de um líquido de massa m , de volume V , e com densidade ρ_L , todos esses constantes, obedecendo as condições declaradas no parágrafo anterior. Conforme ilustrado na figura

4, a porção líquida em evidência desloca-se por um tubo, vindo da região 1 com velocidade v_1 e altura y_1 para a região 2 com velocidade v_2 e altura y_2 .

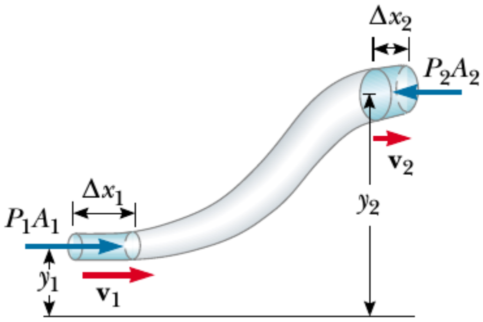

Figura 7: Fluido em escoamento estacionário numa tubulação. Fonte: Fundamentals of Physics (Halliday-Resnick-Walker)

Na região 1 o tubo tem área transversal A_1 e a porção sofre a ação de uma força, $F_1 = p_1 A_1$ no mesmo sentido do movimento, como consequência da pressão p_1 em que está submetida. Na região 2 de seção transversal A_2 , a porção sofre uma força, $F_2 = p_2 A_2$ no sentido oposto a direção do movimento, agora devido a pressão p_2 . Podemos então determinar o trabalho realizado por essas forças,

$$W_p = F_1 \Delta x_1 - F_2 \Delta x_2 = p_1 A_1 \Delta x_1 - p_2 A_2 \Delta x_2 = -(p_2 - p_1) V \quad . (16)$$

Além disso, existe também o trabalho contrário a força peso que promove o deslocamento da porção líquida de y_1 para y_2 , ou seja,

$$W_g = -mg(y_2 - y_1) = -\rho_L V g (y_2 - y_1) \quad . \qquad (17)$$

O ponto de partida da nossa discussão será o teorema do trabalho-energia, que tem a ver com o trabalho total realizado por todas as forças que participam do movimento da porção como igual a variação da energia cinética ΔK dessa porção,

$$W_p + W_g = \Delta K = \frac{1}{2} m (v_2^2 - v_1^2) \quad , \qquad (18)$$

$$-(p_2 - p_1) V - \rho_L V g (y_2 - y_1) = \frac{1}{2} \rho_L V (v_2^2 - v_1^2) \quad , \qquad (19)$$

ou simplesmente,

$$p_1 + \rho_L g y_1 + \frac{1}{2} \rho_L v_1^2 = p_2 + \rho_L g y_2 + \frac{1}{2} \rho_L v_2^2 \quad . \qquad (20)$$

A expressão (20) é conhecida por equação de Bernoulli. Olhando para essa expressão percebemos que a quantidade, $p+\rho_L gy+\frac{1}{2}\rho_L v^2$, se conserva. Outra forma de anunciá-la é a seguinte:

$$p+\rho_L gy+\frac{1}{2}\rho_L v^2=constante \quad . (21)$$

Se fizermos uma análise dimensional nessa equação veremos que todos os termos constituintes têm unidade de pressão. No caso em que um líquido encontra-se em repouso, $v_1=v_2=0$, a equação de Bernoulli reduzirá a (5) que é a equação da pressão para o problema estático.

IV. As animações computacionais

Nesta seção discutiremos sobre o nosso produto educacional, essencialmente constituído por três animações computacionais que simulam algumas situações típicas ocorridas na física dos líquidos. Faremos agora a abordagem dessas animações apresentando suas principais características pedagógicas, referindo-nos a cada uma delas por animações 1, 2 e 3.

Animação 1

Na primeira animação visamos simular uma situação com aspecto original e para tal consideraremos os elementos constituintes do problema com densidades bastante conhecidas da literatura dos fluidos. A animação promoverá o movimento de três corpos esféricos quando mergulhados em um recipiente preenchido por água, ou seja, vamos admitir $\rho_L=1.0\,g/cm^3$. A primeira esfera reproduz o elemento químico alumínio, com densidade igual a 2.7 g/cm^3, a segunda simula a borracha, com densidade aproximadamente igual a 1.0 g/cm^3, e a terceira representa a madeira com densidade igual a 0.43 g/cm^3. Desse modo elas estarão simulando, respectivamente, objetos com densidade maior, igual (ou bem próxima) e menor que a da água. Inicialmente as esferas estão ligeiramente fora da água compartilhando o mesmo valor da posição vertical e da velocidade (impulso). Ao imergirem, os seus movimentos estarão sob a ação de três forças conforme estabelecido na seção II.

A figura 5 ilustra o instante final desse problema, a esfera vermelha que possui densidade maior do que a água terá uma força resultante (soma vetorial das três forças envolvidas) no sentido do peso e por isso o seu movimento estará acelerado nesse mesmo sentido até ser parado pelo fundo do recipiente. A esfera amarela simula o movimento de um corpo com a mesma densidade da água (a força peso então tem módulo igual ao empuxo) que após um tempo suficiente terá a anulação da força de arraste e portanto da força resultante, cessando o seu movimento nesse instante antes mesmo de alcançar o fundo do recipiente. Já a esfera branca descreve o movimento de um corpo com densidade menor que a da água e este é o único caso em que a força resultante mudará de sentido, invertendo portanto o sentido do movimento. Embora essa situação não esteja revelada na figura 5, a animação simula a esfera branca descendo até zerar a sua velocidade e logo após subindo de volta a superfície.

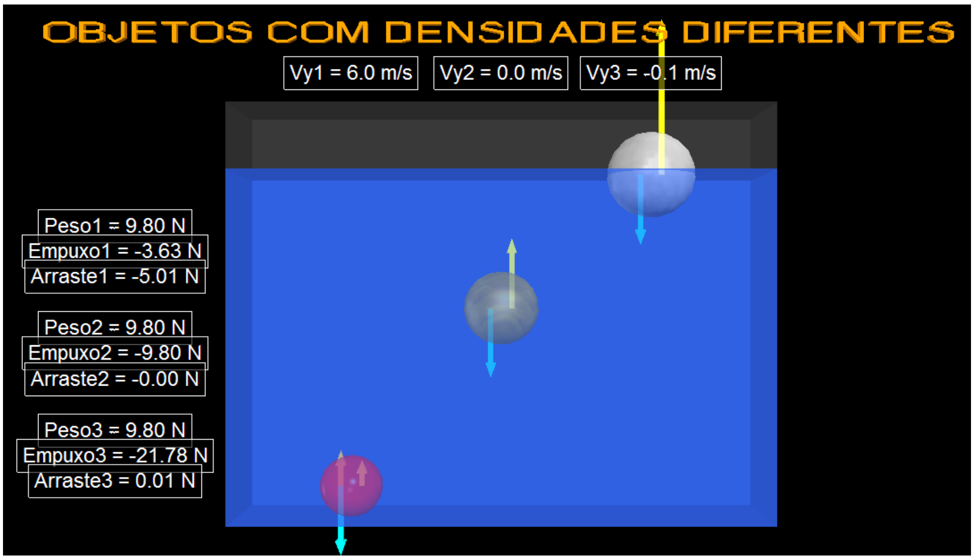

Figura 8: Simulação 1 - objetos de mesmo formato mas com densidades diferentes.

Também podemos notar na figura 8 duas formas de acompanhar e descrever a dinâmica dos três corpos proporcionada pelo ambiente vpython. A primeira é pelo valor das forças e das velocidades das esferas que podem ser visualizados na animação em tempo real. Algumas grandezas possuem valor positivo e outras o valor é negativo, o sinal assume o papel de indicar os dois sentidos possíveis (para baixo ou para cima) existentes no sistema. Este é um recurso importante que favorece a imaginação correta do problema em questão, até porque algumas grandezas como a força de arraste e a velocidade mudam no tempo. Em se tratando de um problema unidimensional poderia ser descrito unicamente pelos valores e sinais das grandezas envolvidas na figura 5.

Entretanto também optamos em utilizar na animação o recurso vetores. O vetor é uma ferramenta visual essencial pois mostra, mesmo que numa visão qualitativa, a ocorrência de equilíbrio ou desequilíbrio no sistema físico. Nesse problema os vetores azuis representam o peso, os amarelos indicam o empuxo e os laranjas simbolizam o arraste. Os seus respectivos módulos estão mostrados no lado esquerdo da animação.

Diante disso, o aluno poderá interagir com a animação analisando amplamente a situação individual de cada corpo e depois compará-los. Poderá fazer especulações como somar as forças que atuam para baixo e fazer o mesmo para as forças que agem para cima e então verificar como que cada força sobre um corpo influencia esse resultado. Como as forças citadas depende da densidade e da área projetada do corpo no líquido e da densidade do líquido, o estudante poderá ser estimulado a alterar valores dessas grandezas e assim investigar o impacto dessas mudanças na força resultante e consequentemente na dinâmica de cada corpo dentro do líquido.

Animação 2

Nesta animação temos novamente o intuito de simular o movimento de três objetos no interior de um meio líquido. As condições estabelecidas são as mesmas consideradas para a animação anterior, exceto de que agora os objetos assumem formas diferentes mas dispõem da mesma densidade. Para essa segunda animação optamos em assumir o caso em que a densidade dos objetos é menor que a do líquido.

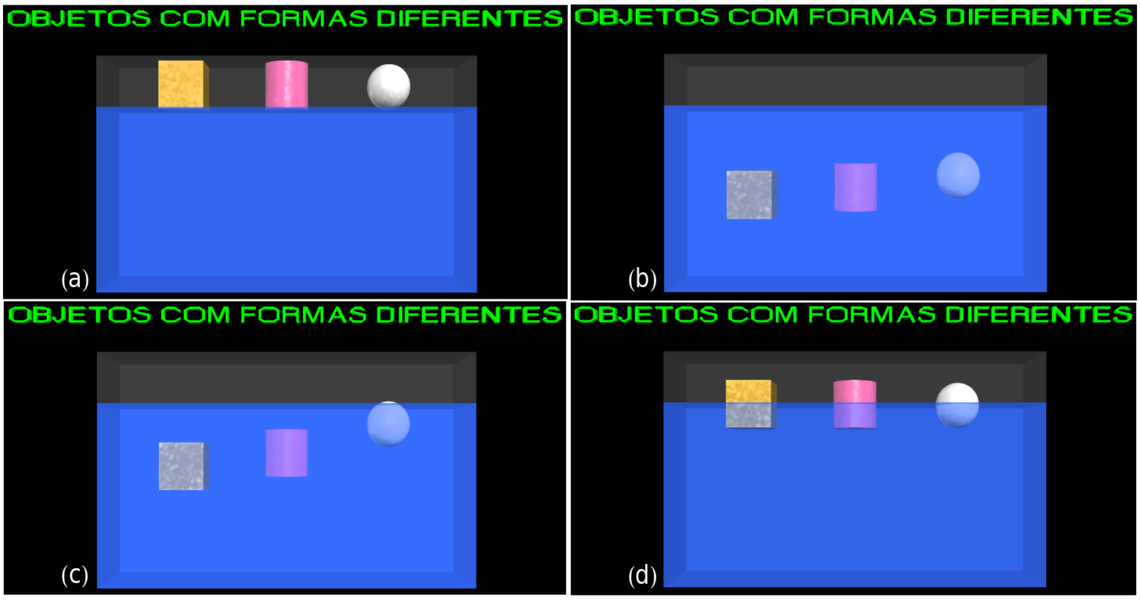

Figura 9: Simulação 2 - objetos com densidades iguais mas com formas diferentes.

A figura 9 mostra quatro instantes diferentes em que a animação descreve o movimento de três objetos com as seguintes formas: cúbica, cilíndrica e esférica. O tamanho dos objetos foram escolhidos de modo que a “largura” e a “altura” de cada um permanecessem iguais. Embora eles são mergulhados no líquido no mesmo instante e com a mesma velocidade iniciais (figura 9a), a inversão do sentido de sentido dos seus movimentos (figuras 9b e 9c) e a chegada na superfície (figura 9d) acontece em instantes diferentes. Nessa simulação não houve uma interpretação do problema por meio de valores numéricos e nem pela utilização de vetores. Embora o aluno tenha a opção de modificar os valores dos parâmetros envolvidos no código da animação, a intenção principal é apenas observar sem interagir, como acontece em um vídeo. A ideia aqui, após a análise da simulação, é provocar discussões e reflexões do porque os três objetos contendo a mesma densidade, mas menor que a do líquido, não retornam juntos à superfície.

Animação 3

A terceira e última simulação é uma aplicação da equação de Bernoulli discutida na seção III, representando um grande recipiente aberto (poderia ser um reservatório ou uma represa) preenchido por água sendo escoada por pequenos orifícios na parede do recipiente, conforme mostrados na figura 10. O intuito é compreender o alcance horizontal do filete de água escoado de acordo com a posição (profundidade) de cada orifício. Esse problema de escoamento é bastante interessante mas requer alguns cuidados na sua interpretação, já foi motivo inclusive de uma análise crítica em prova do ENEM (veja mais em: http://www.if.ufrgs.br/~lang/Textos/Ancritquestfis_INEP.pdf).

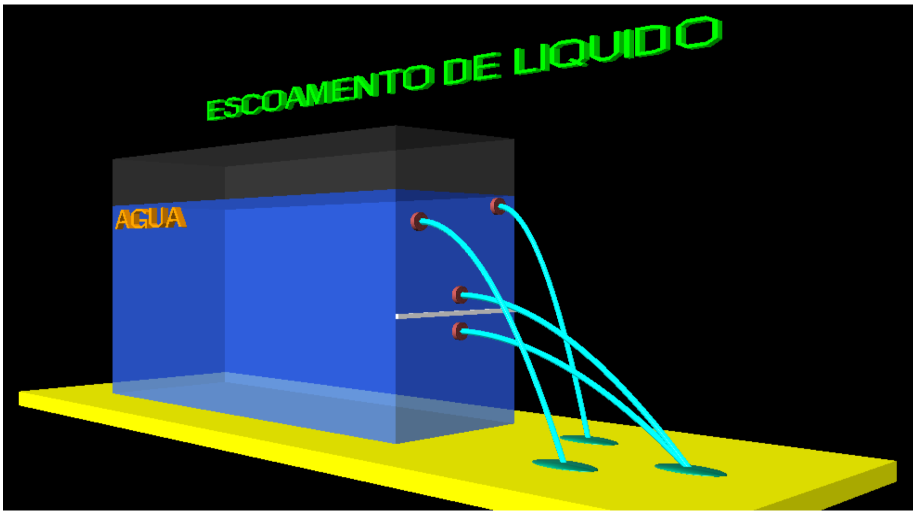

Figura 10: Simulação 3 - escoamento de um fluido através de pequenos orifícios contidos na parede de um recipiente.

Como estamos tratando o problema contendo um grande volume de água, a velocidade de esvaziamento do recipiente v_1 (ou de diminuição da coluna de água) nesse caso é muito menor que a velocidade do filete escoado v_2 e assim, com boa aproximação, podemos considerar $v_1=0$. Levando em consideração essas informações e que as pressões na superfície da água e no orifício são iguais a pressão atmosférica, $p_1=p_2=p_0$, a equação de Bernoulli (20) nos conduz então a expressão,

$$v \equiv v_2=\sqrt{2g\Delta H} \quad , (22)$$

sendo v a velocidade do filete de água no instante em que sai do orifício, g a aceleração da gravidade e ΔH a diferença das profundidades da coluna de água e do orifício, ambas medidas à partir do fundo do recipiente. Vemos que a expressão anterior na realidade não depende da densidade do líquido escoado de modo que, sob as mesmas condições impostas para esse problema, a velocidade do filete será a mesma para todos os líquidos que obedecem a nossa aproximação.

A primeira observação que podemos fazer de acordo com a animação é que o alcance horizontal de um filete de água muda em função de ΔH , de maneira que orifícios próximos a superfície escoam o filete com alcance pequeno enquanto que orifícios localizados mais abaixo no recipiente tem um alcance maior. Podemos observar também que orifícios próximos a superfície têm uma coluna de água acima menor que a coluna dos orifícios mais abaixo. A partir daí surge algumas questões:

(a) seria essa coluna que estaria provocando as mudanças no alcance horizontal do filete escoado? E o que isso tem a ver com o conceito de pressão?

(b) podemos concluir que quanto mais distante estiver o orifício da superfície maior também será o alcance do filete de água?

(c) existe alguma altura em que esse alcance é máximo?

Entretanto, vemos algo diferente ocorrendo do que foi comentado na observação anterior, quando olhamos na animação para os dois orifícios equidistantes da meia altura da parede do recipiente (marcada por um risco branco). Os filetes escoados desses orifícios têm o mesmo alcance, embora estejam localizados em profundidades diferentes. O que diz então a equação de Bernoulli a respeito dessa suposta “anomalia”?

É preciso esclarecer que a intenção desses comentários é promover novamente discussões e reflexões dos alunos acerca do fenômeno e das grandezas físicas envolvidas, mediadas pelo professor e com a ajuda das animações, antes mesmo de propor uma solução ou explicação final para o problema. Dessa forma, acreditamos manter o aluno na condição participativa essencial para o processo de aprendizagem.

V. Códigos das animações computacionais

Nesta seção apresentaremos os códigos das animações construídas na linguagem python. Para a sua utilização é necessário que o compilador python e também o módulo gráfico 3D vpython esteja instalado no computador. Os endereços à seguir mostram a seção de *downloads*.no *site* do vpython que fornecem informações de como baixá-los e instalá-los de acordo com o sistema operacional utilizado na máquina:

1) http://vpython.org/contents/download_windows.html (versão windows)
2) http://vpython.org/contents/download_linux.html (versão linux)
3) http://vpython.org/contents/download_mac.html (versão macintosh)

Mais detalhes sobre instalação do vpython também podem ser obtidos consultando o *site* do *youtube*.

ALGORITMO DA ANIMAÇÃO 1

```
# coding: iso-8859-1
from visual import *
from visual.text import *
import math
import time

janela = display(title = 'Fisica dos Fluidos')
janela.fullscreen = True # tela cheia
janela.forward = (0.0, 0.0, -0.6)
janela.range = 11.5 # altera o zoom da janela
janela.ambient = 0.4 # altera do luminosidade do ambiente

####### FLUIDO  E RECIPIENTE!!!
lado = 10.0 # lado da caixa
verm = 50; verd = 100; azul = 255.0
norma = 255.0 # serve para normalizar os valores do código RGB
cola = frame() # esse recurso permite colar (unir) os objetos que se deseja!!
fluido = box(frame = cola, pos = (0.0, 0.0, 0.0), length = lado, height = lado/1.5, width = lado/5,
          color = (verm/norma,verd/norma,azul/norma), opacity = 0.5) # o fluido
recip = box(frame = cola, pos = (0.0, fluido.height/2 + lado/15.99, 0.0), length = lado, height = lado/8,
         width = fluido.width, color = color.white, opacity = 0.1) # o recipiente
cola.pos = (0.0, -1.0, 0.0)
###############
alt = 0.7
text(pos = (0.0, recip.pos.y + recip.height + cola.pos.y + alt, 0.0), axis = (0.1,0,0), length = 1, height = 0.4, width = 0.6, string =
    'OBJETOS COM DENSIDADES DIFERENTES', color = color.orange, depth = 0.2, justify = 'center')
raio = 0.6 # raio das esferas
bola1 = sphere (pos = (-fluido.length/3.5, fluido.pos.y + fluido.height/2.0 + cola.pos.y, 0.0), radius = raio,
            color = color.red, material = materials.plastic, opacity = 0.7)
```

```
bola2 = sphere (pos = (0.0, bola1.pos.y, 0.0), radius = 1.2 * bola1.radius, color = color.yellow,
            material = materials.wood, opacity = 0.7)
bola3 = sphere (pos = (fluido.length/3.5, bola1.pos.y, 0.0), radius = 1.4 * bola1.radius,
            material = materials.rough, opacity = 0.7)
g = 9.8 # aceleração da gravidade
cd = 0.44 # coeficiente de arraste
denf = 1.0 # densidade do fluido
################ Parâmetros: k e alfa
#razao_dens1 = 0.9997 # razão das densidades do fluido e do corpo 1.
denc1 = 2.7 # densidade alumínio ou bolinha de gude
razao_dens1 = denf / denc1
k1 = (1.0 - razao_dens1) * g # k1 deve ser > 0!!
dp1 = (6.0 / (pi * denc1))**(1.0/3.0) # diâmetro do corpo 1
alfa1 = 3.0 * razao_dens1 * cd / dp1 / 4.0 # ATENÇÃO!! Essa expressão é válida apenas para um corpo esférico.

#razao_dens2 = 1.0 # razão das densidades do fluido e do corpo 2.
denc2 = 1.0 # simulando a densidade da borracha que é 0.93 (nesse caso podemos tratá-la como aproximadamente 1.0 e assim considerarmos k2 = 0)
razao_dens2 = denf / denc2
k2 = (1.0 - razao_dens2) * g # k2 deve ser = 0
dp2 = (6.0 / (pi * denc2))**(1.0/3.0) # diâmetro do corpo 2
alfa2 = 3.0 * razao_dens2 * cd / dp2 / 4.0

#razao_dens3 = 1.0003 # razão das densidades do fluido e do corpo 3.
denc3 = 0.45 # densidade da madeira
razao_dens3 = denf / denc3
k3 = (1.0 - razao_dens3) * g # k3 deve ser < 0
dp3 = (6.0 / (pi * denc3))**(1.0/3.0) # diâmetro do corpo 3
alfa3 = 3.0 * razao_dens3 * cd / dp3 / 4.0
################
v0y1 = 0.1; v0y2 = v0y1; v0y3 = v0y1
vy1 = v0y1; vy2 = v0y2; vy3 = v0y3
gama = (v0y1 - math.sqrt(k1 / alfa1)) / (v0y1 + math.sqrt(k1 / alfa1))
arg = atan(math.sqrt(alfa3 / abs(k3)) * v0y3)
y0 = bola1.pos.y
t0 = 0.0; t = t0; dt = 0.01
delta = - 1.0
############### VETORES FORÇA ###############
amplia = 0.14 # fator essencial para ampliar ou reduzir a visualização dos vetores força!!
deslocax = vector(0.2, 0.0, 0.0)

# bola 1
peso1 = arrow(pos = bola1.pos - deslocax, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.cyan)
empuxo1 = arrow(pos = bola1.pos + deslocax, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.yellow)
arraste1 = arrow(pos = bola1.pos, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.orange)
# bola 2
peso2 = arrow(pos = bola2.pos - deslocax, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.cyan)
empuxo2 = arrow(pos = bola2.pos + deslocax, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.yellow)
arraste2 = arrow(pos = bola2.pos, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.orange)
# bola 3
peso3 = arrow(pos = bola3.pos - deslocax, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.cyan)
empuxo3 = arrow(pos = bola3.pos + deslocax, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.yellow)
arraste3 = arrow(pos = bola3.pos, axis = (0.0, 0.0, 0.0), shaftwidth = 0.1, color = color.orange)
##########################################
deslocarot = vector(0.0, 2.0, 0.0)
rotv1 = label(pos = bola1.pos + deslocarot, text = u'Vy1 = %1.1f m/s' % (vy1), height = 20)
rot_peso1 = label(pos = bola1.pos - (4.5, 1.0, 0.0), text = u'Peso1 = %1.2f N' % (-peso1.axis.y / amplia),
          height = 20)
rot_emp1 = label(pos = bola1.pos - (4.5, 1.5, 0.0), text = u'Empuxo1 = %1.2f N' % (empuxo1.axis.y),
          height = 20)
rot_arr1 = label(pos = bola1.pos - (4.5, 2.0, 0.0), text = u'Arraste1 = %1.2f N' % (arraste1.axis.y),
          height = 20)

rotv2 = label(pos = bola2.pos + deslocarot, text = u'Vy2 = %1.1f m/s' % (vy2), height = 20)
rot_peso2 = label(pos = bola1.pos - (4.5, 3.0, 0.0), text = u'Peso2 = %1.2f N' % (-peso2.axis.y / amplia),
          height = 20)
rot_emp2 = label(pos = bola1.pos - (4.5, 3.5, 0.0), text = u'Empuxo2 = %1.2f N' % (empuxo2.axis.y),
```

```
              height = 20)
rot_arr2 = label(pos = bola1.pos - (4.5, 4.0, 0.0), text = u'Arraste2 = %1.2f N' % (arraste2.axis.y),
              height = 20)

rotv3 = label(pos = bola3.pos + deslocarot, text = u'Vy3 = %1.1f m/s' % (vy3), height = 20)
rot_peso3 = label(pos = bola1.pos - (4.5, 5.0, 0.0), text = u'Peso3 = %1.2f N' % (-peso3.axis.y / amplia),
              height = 20)
rot_emp3 = label(pos = bola1.pos - (4.5, 5.5, 0.0), text = u'Empuxo3 = %1.2f N' % (empuxo3.axis.y),
              height = 20)
rot_arr3 = label(pos = bola1.pos - (4.5, 6.0, 0.0), text = u'Arraste3 = %1.2f N' % (arraste3.axis.y),
              height = 20)

time.sleep(4.0)

while True:

    rate(150)
    temp = bola3.pos.y
    t += dt # isso serve para atualizar o tempo t


    if (bola1.pos.y > fluido.pos.y - fluido.height/2.0 + cola.pos.y + bola1.radius):
        bola1.pos.y = y0 - (1.0 / alfa1) * log((math.exp(math.sqrt(k1 * alfa1) * (t - t0)) - gama * math.exp(-math.sqrt(k1 * alfa1) * (t - t0))) / (1.0
- gama))
        vy1 = math.sqrt(k1 / alfa1) * ((1.0 + gama * math.exp(-2.0 * math.sqrt(k1 * alfa1) * (t - t0))) /
                               (1.0 - gama * math.exp(-2.0 * math.sqrt(k1 * alfa1) * (t - t0))))

    if (bola2.pos.y > fluido.pos.y - fluido.height/2.0 + cola.pos.y + bola2.radius and vy2 >= 5.e-2):
        bola2.pos.y = y0 - (1.0 / alfa2) * log(1.0 + v0y2 * alfa2 * (t - t0))
        vy2 = v0y2 / (1.0 + v0y2 * alfa2 * (t - t0))

    if(bola3.pos.y < recip.pos.y - recip.height/2.0 + cola.pos.y or delta < 0.0):
        bola3.pos.y = y0 - (1.0 / alfa3) * log(cos(arg - math.sqrt(abs(k3) * alfa3) * (t - t0)) / cos(arg))
        vy3 = math.sqrt(abs(k3) / alfa3) * tan(arg - math.sqrt(abs(k3) * alfa3) * (t - t0))
        delta = bola3.pos.y - temp

    else:
        bola3.pos.y = recip.pos.y - recip.height/2.0 + cola.pos.y

    peso1.pos = bola1.pos - deslocax; peso1.axis.y = -amplia * g
    empuxo1.pos = bola1.pos + deslocax; empuxo1.axis.y = amplia * razao_dens1 * g
     arraste1.pos = bola1.pos - (deslocax.x * vy1 / abs(vy1), deslocax.y, deslocax.z); arraste1.axis.y = amplia * alfa1 * vy1 * vy1 * vy1 /
abs(vy1)

    peso2.pos = bola2.pos - deslocax; peso2.axis.y = -amplia * g
    empuxo2.pos = bola2.pos + deslocax; empuxo2.axis.y = amplia * razao_dens2 * g
     arraste2.pos = bola2.pos - (deslocax.x * vy2 / abs(vy2), deslocax.y, deslocax.z); arraste2.axis.y = amplia * alfa2 * vy2 * vy2 * vy2 /
abs(vy2)

    peso3.pos = bola3.pos - deslocax; peso3.axis.y = -amplia * g
    empuxo3.pos = bola3.pos + deslocax; empuxo3.axis.y = amplia * razao_dens3 * g
     arraste3.pos = bola3.pos - (deslocax.x * vy3 / abs(vy3), deslocax.y, deslocax.z); arraste3.axis.y = amplia * alfa3 * vy3 * vy3 * vy3 /
abs(vy3)

    rotv1.text = u'Vy1 = %1.1f m/s' % (vy1); rotv2.text = u'Vy2 = %1.1f m/s' % (vy2);
    rotv3.text = u'Vy3 = %1.1f m/s' % (vy3)

    rot_peso1.text = u'Peso1 = %1.2f N' % (-peso1.axis.y / amplia);
    rot_emp1.text = u'Empuxo1 = %1.2f N' % (-empuxo1.axis.y / amplia);
    rot_arr1.text = u'Arraste1 = %1.2f N' % (-arraste1.axis.y  / amplia);


    rot_peso2.text = u'Peso2 = %1.2f N' % (-peso2.axis.y / amplia);
    rot_emp2.text = u'Empuxo2 = %1.2f N' % (-empuxo2.axis.y / amplia);
    rot_arr2.text = u'Arraste2 = %1.2f N' % (-arraste2.axis.y / amplia);

    rot_peso3.text = u'Peso3 = %1.2f N' % (-peso3.axis.y / amplia);
```

```
    rot_emp3.text = u'Empuxo3 = %1.2f N' % (-empuxo3.axis.y / amplia);
    rot_arr3.text = u'Arraste3 = %1.2f N' % (-arraste3.axis.y / amplia);
```

ALGORITMO DA ANIMAÇÃO 2

```
# coding: iso-8859-1

from visual import *
from visual.text import *
import math
import time

janela = display(title = 'Fisica dos Fluidos')
janela.fullscreen = True # tela cheia
#janela.width = janela.height = 1000. # tela personalizada
#janela.width, janela.height, janela.range = 1900, 1200, 300 # tela personalizada
janela.forward = (0.0, 0.0, -0.6)
janela.range = 9 # altera o zoom da janela
janela.ambient = 0.4 # altera do luminosidade do ambiente

####### FLUIDO  E RECIPIENTE!!!
lado = 10.0 # lado da caixa
verm = 50; verd = 100; azul = 255.0 # os valores dos parâmetros "verd", "verm" e "azul" correspondem respectivamente aos do código RGB de cores (consultar tabela na internet)
norma = 255.0 # serve para normalizar os valores do código RGB
cola = frame() # esse recurso permite colar (unir) os objetos que se deseja!!
fluido = box(frame = cola, pos = (0.0, 0.0, 0.0), length = lado, height = lado/2.2, width = lado/5,
             color = (verm/norma,verd/norma,azul/norma), opacity = 0.5) # o fluido
recip = box(frame = cola, pos = (0.0, fluido.height/2 + lado/15.99, 0.0), length = lado, height = lado/8,
            width = fluido.width, color = color.white, opacity = 0.1) # o recipiente
cola.pos = (0.0, -1.0, 0.0)
################

alt = 0.3 # altura do texto 3D à partir do topo do recipiente
text(pos = (0.0, recip.pos.y + recip.height + cola.pos.y + alt, 0.0), axis = (0.1,0,0), length = 1, height = 0.4, width = 0.6, string = 'OBJETOS IMERSOS EM UM FLUIDO',
     color = color.green, depth = 0.2, justify = 'center') # o recurso "text" permite inserir textos (sem acentos) em 3D

raio = 0.6 # raio da esfera
cubo = box(pos = (-2.5, recip.pos.y - recip.height/2.0 + cola.pos.y + raio, 0.0), length = 2*raio, height = 2*raio, width = 2*raio,
           color = (184/norma,134/norma,11/norma), material=materials.rough)
cilindro = cylinder (pos = (0.0, cubo.pos.y - raio, 0.0), axis = (0.0, 2*raio, 0.0), radius = raio,
                     color = (255/norma,105/norma,180/norma), material=materials.rough)
bola = sphere (pos = (2.5, cubo.pos.y, 0.0), radius = raio, material=materials.rough)

g = 9.8 # aceleração da gravidade
cd = 0.44 # coeficiente de arraste

dp3 = 0.2 # diâmetro do corpo esférico
dp1 = dp3 * (6.0 / pi)**(1.0/3.0) # diâmetro que representa o corpo cúbico no cálculo
dp2 = dp3 * (1.5)**(1.0/3.0) # diâmetro que representa o corpo cilíndrico no cálculo

# print dp1, dp2

#massa = 0.7
#densidadefluido = 1.0
#Ac = (pi / 4.0) * dp**2
#volume = (pi / 6.0) * dp**3
#Peso = massa * g
#Fe = densidadefluido * volume * g # força de empuxo

#k = (Peso - Fe) / massa
#alfa = (densidadefluido * Ac * cd) / (2.0 * massa)
```

```
################ Parâmetros: k e alfa (NOVO!!) 1.0007
denig = 1.0003

razao_dens1 = denig # razão das densidades do fluido e do corpo 1.
k1 = (1.0 - razao_dens1) * g # k1 deve ser < 0!!
alfa1 = 3.0 * razao_dens1 * cd / dp1 / 4.0 # ATENÇÃO!! Essa expressão é válida apenas para um corpo esférico.

razao_dens2 = denig # razão das densidades do fluido e do corpo 2.
k2 = (1.0 - razao_dens2) * g # k2 deve ser < 0
alfa2 = 3.0 * razao_dens2 * cd / dp2 / 4.0

razao_dens3 = denig # razão das densidades do fluido e do corpo 3.
k3 = (1.0 - razao_dens3) * g # k3 deve ser < 0
alfa3 = 3.0 * razao_dens3 * cd / dp3 / 4.0
################

v0y1 = 2.0; v0y2 = v0y1; v0y3 = v0y1

arg1 = atan(math.sqrt(alfa1 / abs(k1)) * v0y1)
arg2 = atan(math.sqrt(alfa2 / abs(k2)) * v0y2)
arg3 = atan(math.sqrt(alfa3 / abs(k3)) * v0y3)

y0 = cubo.pos.y; y1 = y0 - raio

# print y0

t = 0.0; t0 = 0.0; dt = 0.03

delta1 = - 1.0; delta2 = delta1; delta3 = delta1

time.sleep(4.0)

while True:

    rate(150)

    temp1 = cubo.pos.y; temp2 = cilindro.pos.y; temp3 = bola.pos.y

    t += dt # isso serve para atualizar o tempo t

    if (cubo.pos.y < recip.pos.y - recip.height/2.0 + cola.pos.y or delta1 < 0.0):
        cubo.pos.y = y0 - (1.0 / alfa1) * log(cos(arg1 - math.sqrt(abs(k1) * alfa1) * (t - t0)) / cos(arg1))
        delta1 = cubo.pos.y - temp1
    else:
        cubo.pos.y = recip.pos.y - recip.height/2.0 + cola.pos.y

    if (cilindro.pos.y < recip.pos.y - recip.height / 2.0 + cola.pos.y - cilindro.axis.y/2.0 or delta2 < 0.0):
        cilindro.pos.y = y1 - (1.0 / alfa2) * log(cos(arg2 - math.sqrt(abs(k2) * alfa2) * (t - t0)) / cos(arg2))
        delta2 = cilindro.pos.y - temp2
    else:
        cilindro.pos.y = recip.pos.y - recip.height / 2.0 + cola.pos.y - cilindro.axis.y/2.0

    if (bola.pos.y < recip.pos.y - recip.height / 2.0 + cola.pos.y  or delta3 < 0.0):
        bola.pos.y = y0 - (1.0 / alfa3) * log(cos(arg3 - math.sqrt(abs(k3) * alfa3) * (t - t0)) / cos(arg3))
        delta3 = bola.pos.y - temp3
    else:
        bola.pos.y = recip.pos.y - recip.height / 2.0 + cola.pos.y
```

ALGORITMO DA ANIMAÇÃO 3

```
# coding: iso-8859-1

from visual import *
from visual.text import *
import math
import time

janela = display(title = 'Fisica dos Fluidos')
janela.fullscreen = True # tela cheia
janela.forward = (-0.5, 0.0, -0.5)
janela.range = 10 # altera o zoom da janela
janela.ambient = 0.4 # altera do luminosidade do ambiente


####### FLUIDO - RECIPIENTE - TIRA - BASE!!!
lado = 8.0 # lado da caixa
verm = 50; verd = 100; azul = 255.0 # os valores dos parâmetros "verd", "verm" e "azul" correspondem respectivamente aos do código RGB de cores (consultar tabela na internet)
norma = 255.0
cola = frame()
cola.pos = (-3.0, -1.0, 0.0)

fluido = box(frame = cola, pos = (0.0, 0.0, 0.0), length = lado, height = lado/2, width = lado/3, color = (verm/norma,verd/norma,azul/norma),
          opacity = 0.5) # o fluido
recip = box(frame = cola, pos = (0.0, fluido.height/2 + lado/15.99, 0.0), length = lado, height = lado/8, width = fluido.width, color =
         color.white, opacity = 0.1) # o recipiente
compx = 0.05
tira = box (pos = (fluido.length/2.0 + cola.pos.x + compx / 2.0, fluido.pos.y + cola.pos.y, cola.pos.z), length = compx, height = 1.5 * compx,
         width = fluido.width, color = color.white) # tira que marca a metade da altura do fluido
superficie = box (pos = (2.0 + cola.pos.x,-(fluido.height + 0.305)/2 + cola.pos.y, cola.pos.z), length = 1.9 * lado, height = 0.3, width = 5.0,
            color = color.yellow) # a base
################

alty = 0.3
text(pos = (0.0, recip.pos.y + recip.height + cola.pos.y + alty, 0.0), axis = (0.1,0,0), length = 0.5, height = 0.4, width = 0.6, string =
    'ESCOAMENTO DE LIQUIDO', color = color.green, depth = 0.2, justify = 'center')

text(pos = (fluido.pos.x - fluido.length/2 + cola.pos.x + 1.5, fluido.pos.y + fluido.height/2 + cola.pos.y - 0.5, fluido.pos.z + fluido.width/2 + cola.pos.z),
    axis = (0.1,0,0), length = 0.5, height = 0.4, width = 0.6, string = 'AGUA', color = color.orange, depth = 0.2, justify = 'center')

####### ORIFÍCIOS!!!
r_orif = 0.15

distmet1 = 1.8 # distância do orifício 1 (a partir da meia altura do fluido): pode ser positivo (distância para cima) ou negativo (distância para baixo)
cilindro1 = cylinder (pos = (fluido.length/2.0 + cola.pos.x, fluido.pos.y + cola.pos.y + distmet1, -fluido.width/3.0 + cola.pos.z), axis =
              (r_orif/2.0, 0.0, 0.0), radius = r_orif, color = (205/norma,92/norma,92/norma), material=materials.rough)
distmet2 = 1.5 # distância do orifício 2 (a partir da meia altura do fluido): pode ser positivo (distância para cima) ou negativo (distância para baixo)
cilindro2 = cylinder (pos = (fluido.length/2.0 + cola.pos.x, fluido.pos.y + cola.pos.y + distmet2, fluido.width/3.0 + cola.pos.z), axis =
              (r_orif/2.0, 0.0, 0.0), radius = r_orif, color = (205/norma,92/norma,92/norma), material=materials.rough)
distmet3 = -0.3 # distância do orifício 3 (a partir da meia altura do fluido): pode ser positivo (distância para cima) ou negativo (distância para baixo)
cilindro3 = cylinder (pos = (fluido.length/2.0 + cola.pos.x, fluido.pos.y + cola.pos.y + distmet3, cola.pos.z), axis = (r_orif/2.0, 0.0, 0.0),
              radius = r_orif, color = (205/norma,92/norma,92/norma), material=materials.rough)
distmet4 = 0.3 # distância do orifício 3 (a partir da meia altura do fluido): pode ser positivo (distância para cima) ou negativo (distância para baixo)
cilindro4 = cylinder (pos = (fluido.length/2.0 + cola.pos.x, fluido.pos.y + cola.pos.y + distmet4, cola.pos.z), axis = (r_orif/2.0, 0.0, 0.0),
              radius = r_orif, color = (205/norma,92/norma,92/norma), material=materials.rough)
```

```
####### ESCOAMENTO DO FLUIDO (FILETE)!!!
r_fil = 0.05
filete1 = curve(pos = cilindro1.pos, radius = r_fil, color = color.cyan, material = materials.plastic)
filete2 = curve(pos = cilindro2.pos, radius = r_fil, color = color.cyan, material = materials.plastic)
filete3 = curve(pos = cilindro3.pos, radius = r_fil, color = color.cyan, material = materials.plastic)
filete4 = curve(pos = cilindro4.pos, radius = r_fil, color = color.cyan, material = materials.plastic)
################

g = 9.8
t = 0.0; dt = 6.e-4

dif1 = cilindro1.pos.y - (fluido.pos.y - fluido.height/2.0 + cola.pos.y)
x01 = cilindro1.pos.x + cilindro1.axis.x/2.0; y01 = cilindro1.pos.y; xt1 = x01; yt1 = y01
v0x1 = math.sqrt(2.0 * g * (fluido.height - dif1)); v0y1 = 0.0

dif2 = cilindro2.pos.y - (fluido.pos.y - fluido.height/2.0 + cola.pos.y)
x02 = cilindro2.pos.x + cilindro2.axis.x/2.0; y02 = cilindro2.pos.y; xt2 = x02; yt2 = y02
v0x2 = math.sqrt(2.0 * g * (fluido.height - dif2)); v0y2 = 0.0

dif3 = cilindro3.pos.y - (fluido.pos.y - fluido.height/2.0 + cola.pos.y)
x03 = cilindro3.pos.x + cilindro3.axis.x/2.0; y03 = cilindro3.pos.y; xt3 = x03; yt3 = y03
v0x3 = math.sqrt(2.0 * g * (fluido.height - dif3)); v0y3 = 0.0

dif4 = cilindro4.pos.y - (fluido.pos.y - fluido.height/2.0 + cola.pos.y)
x04 = cilindro4.pos.x + cilindro4.axis.x/2.0; y04 = cilindro4.pos.y; xt4 = x04; yt4 = y04
v0x4 = math.sqrt(2.0 * g * (fluido.height - dif4)); v0y4 = 0.0

ca = 0; cb = ca; cc = ca ; cd = ca

time.sleep(5.0)

while True:
    rate(350)
    if (yt1 >= - fluido.height/2.0 + cola.pos.y):
        filete1.append(pos = (xt1, yt1, cilindro1.pos.z))
        xt1 = x01 + v0x1 * t
        yt1 = y01 - v0y1 * t - g * t * t / 2.0
    elif (ca == 0):
        ellipsoid(pos = (xt1, yt1, cilindro1.pos.z), length = 1.0, height = 0.1, width = 0.5, color = color.cyan, material = materials.wood)
        ca = 1


    if (yt2 >= - fluido.height/2.0 + cola.pos.y):
        filete2.append(pos = (xt2, yt2, cilindro2.pos.z))
        xt2 = x02 + v0x2 * t
        yt2 = y02 - v0y2 * t - g * t * t / 2.0
    elif (cb == 0):
        ellipsoid(pos = (xt2, yt2, cilindro2.pos.z), length = 1.0, height = 0.1, width = 0.5, color = color.cyan, material = materials.wood)
        cb = 1

    if (yt3 >= - fluido.height/2.0 + cola.pos.y):
        filete3.append(pos = (xt3, yt3, cilindro3.pos.z))
        xt3 = x03 + v0x3 * t
        yt3 = y03 - v0y3 * t - g * t * t / 2.0
    elif (cc == 0):
        ellipsoid(pos = (xt3, yt3, cilindro3.pos.z), length = 1.0, height = 0.1, width = 0.5, color = color.cyan, material = materials.wood)
        cc = 1

    if (yt4 >= - fluido.height/2.0 + cola.pos.y):
        filete4.append(pos = (xt4, yt4, cilindro4.pos.z))
        xt4 = x04 + v0x4 * t
        yt4 = y04 - v0y4 * t - g * t * t / 2.0
    elif (cd == 0):
        ellipsoid(pos = (xt4, yt4, cilindro4.pos.z), length = 1.0, height = 0.1, width = 0.5, color = color.cyan, material = materials.wood)
        cd = 1

    if (ca + cb + cc + cd >= 6): break
```

```
t += dt
```