|  |  |  |  |
| --- | --- | --- | --- |
| **The Physics of Baumgartner Skydiving** | | | |
| **Formulas and Constants** | | | |
| **Definitions** | | | |
| : The mass of Baumgartner and his equipment.   |  |  | | --- | --- | |  | kg | |  | m2/  m2 | |  | m/s2 | |  | s |   : The projected area of the Baumgartner.  : The drag coefficient of Baumgartner.  **𝑔**: The acceleration due to gravity.  Figure 1: Baumgartner and his equipment  : The density of the troposphere around Baumgartner.  : The velocity of Baumgartner.  : The time pass after Baumgartner began falling.  : The distance between ground and Baumgartner.  : The temperature of the troposphere around Baumgartner.  : The pressure of the troposphere around Baumgartner.  : A very short time. | | | |
| **Calculation of Constant** | | | |
| We hypothesized that is constant. We found out is about kilograms and set m/s2. Reaching the terminal velocity signified that the resistance equals gravity, so we got .  According to the official simulation before he skydived, Baumgartner reached the terminal velocity (km/hrm/s) at a height of about meters. We substituted into the function of density and got . Then found m2 by using .  According to the practical situation, Baumgartner reached the terminal velocity (km/hrm/s) at a height which is similar to the simulation when he was **nosedived**. Through the same way, we found another  m2.  The previous didn’t include the nosedived situation, so we take the average of these two as the  m2 of Baumgartner before he opened the parachute.  After Baumgartner opened his parachute, his velocity fluctuated between about and m/s. We thought it is probably because of the instability from stratosphere. We take the average(m/s) as the velocity after he opened. Through the same way, we got the of him and his parachute. | | | |
| **Simulation by Computer Program** | | | |
| **Flow Path** | | | |
| * **“”** here is the **program usage.** It means **to change the value of LHS into the value of RHS.**   **Yes**  **Yes**  **No**  **Yes**  **No**  **No**  Original State  Result                        **Yes**  **No** | | | |
|  | | | |
| **Program (made by phthon)**  v(m/s)  h(m)  t(s)  38969  0  540  1363  Figure 3: h-t graph | | | |
| QR code: more about the assignment  362.3  t(s)  540  0  264  Figure 2: v-t graph  These two figures show the physics relations of height, time, and velocity. The more details of the programs can be found in the website.  The simulation by computer program told us Baumgartner spent seconds on falling. It is close to the practical situation. We thought the error probably from unsettle troposphere, of Baumgartner and gravity.  The underneath is the sourse code. | | | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | import math  import os  h = 38969 # Initial height  v = 0 # Initial speed  g = -9.8 # The gravity field is considered fixed  m = 110 # The mass of Baumgartner and his equipment  t = 0 # Initial time  a = 0.7322 # The product of the coefficient of confrontation and the cross-sectional area of the windward  def T(h): # Temperature versus height function  if h < 11000:  return 288.14 - 0.00649 \* h  elif h > 25000:  return 141.89 + 0.00299 \* h  else:  return 216.64  def p(h): # Air pressure versus height function  if h < 11000:  return 101.29 \* (T(h) / 288.08) \*\* 5.256  elif h > 25000:  return 2.488 \* (T(h) / 216.6) \*\* (-11.388)  else:  return 22.65 \* math.e \*\* (1.73 - 0.000157 \* h)  def r(h): # Density versus height function  return p(h) / (0.2869 \* T(h)) | 35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | def f(v, a, h): # Resistance function  return (1 / 2) \* r(h) \* v \* v \* a  dt = 0.001  print('Height is', h)  while True:  if h < 2440: # After opening the parachute  v = v + (g + f(v, 140.6816, h) / m) \* dt  else: # Before opening the parachute  v = v + (g + f(v, a, h) / m) \* dt  h = h + v \* dt  t = t + dt  if h > 2440:  print('\_' \* int(39 - (h / 1000)) + "B" + ' ' \* int((h / 1000)) + '| V:', int(v), 't:', str(int(t)) + ' '\*20, end='\r')  else:  print('\_' \* int(37 - (h / 1000)) + "(>B" + ' ' \* int((h / 1000)) + '| V:', int(v), 't:', str(int(t)) + ' '\*20, end='\r')  if h < 1363: # Landing  break  print('\_' \* int(37 - (h / 1000)) + "(>B" + ' ' \* int((h / 1000)-1) + '| V:', int(v), 't:', str(int(t)) + ' '\*20)  print('Landing!!!!')  print('landing in '+str(int(t))+'(sec)') # Landing time  print('actually is', t, 'sec.')  os.system('pause') |
| **References** | | | |
| Natalie Wolchover. (2012). The Physics of the First-Ever Supersonic Skydive. Retrieved from <https://www.livescience.com/23710-physics-supersonic-skydive.html> (November 3, 2019)  Drag coefficient. (November 14, 2019). <https://en.wikipedia.org/wiki/Drag_coefficient> (November 3, 2019)  <https://www.zhihu.com/question/59208808>  <https://www.youtube.com/watch?v=raifrxbHxV0>  <https://www.youtube.com/watch?v=vvbN-cWe0A0> | | | |