constant-accel

Bryan Killett

11/2/2015

Contents

```
var('x accel t c') # "x" is the midpoint distance to be travelled.
c = 299792458         #light speed, in m/s
ly = 9.4605284E15     #light year, in meters
AU = 1.4960E11        #astronomical unit, meters
assume(t>0)           #time in seconds, divided by "units" for display.
accel = 9.81*1.0      #acceleration, in m/s^2
#units=3600           #hours
#units=3600*24        #days
#units=3600*24*7      #weeks
units=3600*24*365.25 #years

#Midpoint is halfway to Mars (or wherever) http://nssdc.gsfc.nasa.gov/\
   planetary/factsheet/marsfact.html
#midpoint_distance = 4.01E11 / 2     #Mars farthest distance is 401 \
   million km = 4.01E11 meters.
#midpoint_distance = 1.00E11 / 2     #Mars close approach at 100 million \
   km = 1E11 meters.
#midpoint_distance = 0.55E11 / 2     #Mars close approach at 55 million \
   km = 0.55E11 meters.
#midpoint_distance =    30.3*AU / 2  #Neptune aphelion: https://en.\
   wikipedia.org/wiki/Neptune
#midpoint_distance =    83*AU / 2    #2012 VP113 distance from Sun in \
   2015: https://en.wikipedia.org/wiki/2012_VP113
#midpoint_distance =    86*AU / 2    #Sedna distance from Sun in 2015: \
   https://en.wikipedia.org/wiki/90377_Sedna
#midpoint_distance =    96.4*AU / 2  #Eris distance from Sun in 2014: \
   https://en.wikipedia.org/wiki/Eris_(dwarf_planet)
#midpoint_distance =  2000*AU / 2    #Inner edge of inner Oort cloud: \
   https://en.wikipedia.org/wiki/Oort_cloud#Structure_and_composition
#midpoint_distance = 20000*AU / 2    #Inner edge of outer Oort cloud: \
   https://en.wikipedia.org/wiki/Oort_cloud#Structure_and_composition
#midpoint_distance = 50000*AU / 2    #Outer edge of outer Oort cloud: \
   https://en.wikipedia.org/wiki/Oort_cloud#Structure_and_composition
#midpoint_distance =   0.0636*ly / 2 #Max speed at 1g is 0.25c,   takes \
   0.49 ship years total.
#midpoint_distance =   0.3*ly / 2    #Max speed at 1g is 0.5c,    takes \
   1.06 ship years total.
```

```
#midpoint_distance =   1.128*ly / 2  #Max speed at 1g is 0.775c,  takes \
   2.00 ship years total.
#midpoint_distance =   2.51*ly / 2   #Max speed at 1g is 0.9c,    takes \
   2.85 ship years total.
#midpoint_distance =  11.799*ly / 2  #Max speed at 1g is 0.99c,   takes \
   5.13 ship years total.
#midpoint_distance =  41.383*ly / 2  #Max speed at 1g is 0.999c,  takes \
   7.36 ship years total.
#midpoint_distance = 135.02*ly / 2   #Max speed at 1g is 0.9999c, takes \
   9.59 ship years total.
midpoint_distance =   4.25*ly / 2   #Proxima Centauri.
#midpoint_distance =   4.32*ly / 2   #Alpha Centauri.
#midpoint_distance = 100.0*ly / 2
#midpoint_distance = 30000*ly / 2    #Sag A, center of Milky Way.
#midpoint_distance = 2.5E6*ly / 2      #Andromeda galaxy. https://en.\
   wikipedia.org/wiki/Andromeda_Galaxy
#midpoint_distance = 93E9*ly / 2     #Diameter of the observable universe\
   : https://en.wikipedia.org/wiki/Observable_universe
#midpoint_distance = 1E12*ly / 2     #Ludicrously far. ~53 total ship \
   years of travel at 1g.
#midpoint_distance = 1E23*ly / 2     #Gone to plaid.  ~103 total ship \
   years of travel at 1g. See https://twitter.com/DumbSci/status\
   /656296561882697728 (Ascension was still interesting, aside from that.\
    Sorry for using big "G" rather than the correct little "g" in that \
   tweet.)

#Newtonian mechanics:
eq1 = x == 0.5*accel*t^2
soln1 = solve(eq1.subs(x=midpoint_distance),t)
accel_time = 2*soln1[0].rhs().n()
accel_time/units #At high speeds Newtonian mechanics will yield absurd \
   results.
```

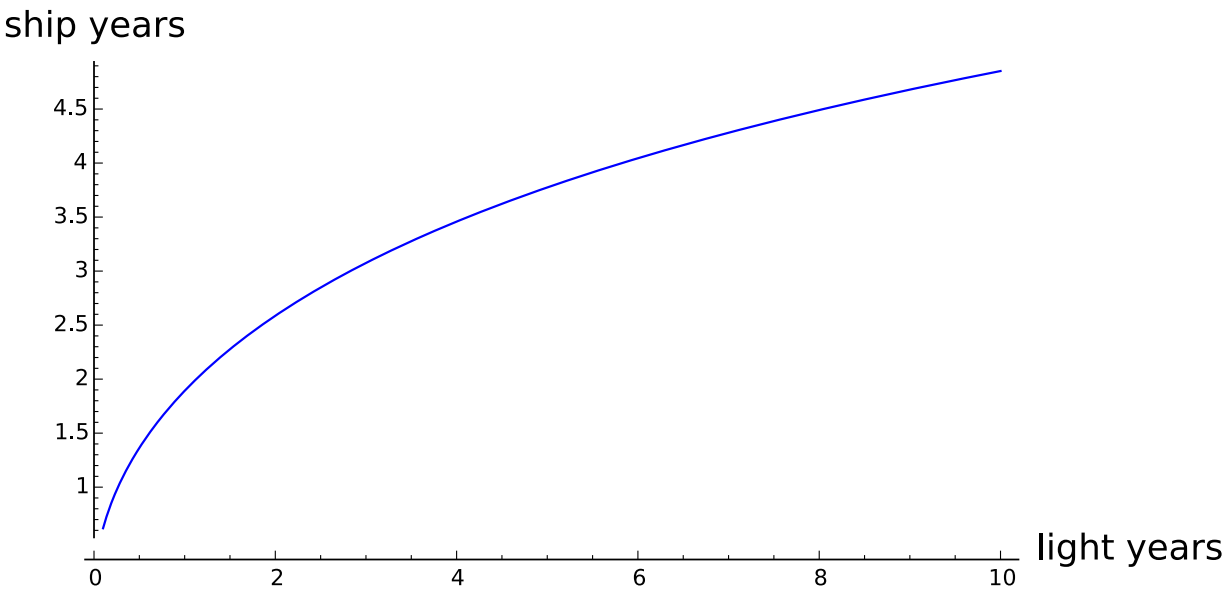
```
(x, accel, t, c)
4.05736134029145
```

```
#Maximum speed at midpoint, in m/s and fraction of "c".
accel*accel_time/2
accel*accel_time/2/c #At high speeds Newtonian mechanics will yield \
   absurd (>1) results.
```

```
6.28039075469831e8
2.09491285958178
```

```
#Relativistic travel time, as measured on the ship.
# http://math.ucr.edu/home/baez/physics/Relativity/SR/rocket.html
# http://www.webcitation.org/query?url=http://www.geocities.com/albmont/\
   relroket.htm&date=2009-10-25+12:17:16
var('accel_time_rel')
eq2 = accel_time_rel == (c/accel)*arccosh(accel*midpoint_distance/c^2 + \
   1)
soln2 = solve(eq2,accel_time_rel)
accel_time_ship = 2*soln2[0].rhs().n()
accel_time_ship/units
```

```
accel_time_rel
3.54248691970821

#Relativistic travel time, as measured on Earth.
accel_time_earth = 2*(c/accel)*sinh(accel*accel_time_ship/2/c)
accel_time_earth/units
5.87570501785570

#Maximum speed at midpoint, in m/s and fraction of "c".
c*tanh(accel*accel_time_ship/2/c)
beta = tanh(accel*accel_time_ship/2/c)
beta
2.84723419309041e8
0.949735097435443

#If engines are shut off to cruise at maximum speed, how long will it \
   take to travel cruise_distance, as measured on Earth?
#Note: total_distance can't be smaller than twice the midpoint_distance.
#total_distance = 4.25*ly
total_distance = 20*ly
#total_distance = 30000*ly
cruise_distance = total_distance - 2*midpoint_distance
cruise_distance/ly #First display cruise distance in light years.
cruise_time_earth = cruise_distance/(beta*c)
cruise_time_earth/units
15.7500000000000
16.5832174012337

#How long will it take to travel cruise_distance, as measured on ship?
cruise_time_ship = cruise_time_earth*sqrt(1-beta^2)
cruise_time_ship/units
5.19145413552563

#Total travel time as measured on the ship:
(accel_time_ship+cruise_time_ship)/units
8.73394105523384

#Total travel time as measured on Earth:
(accel_time_earth+cruise_time_earth)/units
22.4589224190894

#Plot time (in "units") measured on the ship while accelerating and \
   decelerating versus distance travelled in light years.
#(Doesn't include time spent cruising at constant velocity.)
T(x) = 2*(c/accel)*arccosh(accel*x*ly/2/c^2 + 1)/units
plot(T,(x,0.1,10),axes_labels=['light years','ship years'])
```


ship years
4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
light years