

Why the James Webb Space Telescope won't see the first galaxies

[Tom Fuchs](#)

tomfuchs@gmail.com

I explain why the Webb telescope won't see the first galaxies. Among the furthest galaxies it sees will be ones that look much older than expected. I make the case that, rather than assume that galaxies can mature faster than we thought, we should adopt a model of the universe in which space itself doesn't expand. The new model is based on a new, experimentally confirmed metric for Schwarzschild geometry. Unlike the Schwarzschild metric, the new metric obeys the equivalence principle, thereby it can explain our observations when space itself doesn't expand. In the new model: space expands relatively, so that the expansion depends on the observer; an object thrown upward can accelerate up; black holes aren't predicted, so there's no black hole information paradox; and there's no flatness problem, horizon problem, or need for dark energy. I give code to numerically integrate the relativistic motion of an object thrown upward.

1 An object thrown upward can accelerate up

A formal statement of the Einstein [equivalence principle](#) (EP) is

In any and every locally Lorentz (inertial) frame, the laws of special relativity must hold.

See the equations of special relativity at [The Relativistic Rocket](#), for a rocket having a constant proper acceleration $a > 0$. For example:

$$v = \frac{at}{\sqrt{1 + (at/c)^2}} \quad (1)$$

This equation returns the velocity of a rocket after the time t , as measured in the local inertial frame (LIF) in which the rocket blasted off. The speed of light is c .

[Equations for a falling body](#) gives the velocity of a free-falling object that was dropped in a uniform gravitational field (ignoring air resistance) as

$$v = at \quad (2)$$

where a is the acceleration of gravity and t is the elapsed time.

The EP shows that The Relativistic Rocket's (1) supplants (2):

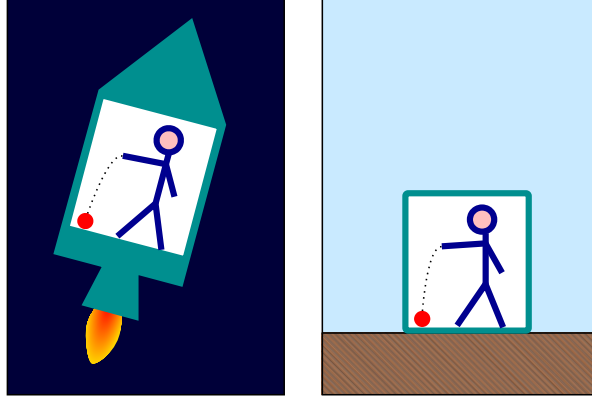


Figure 1: Ball falling to the floor in an accelerating rocket (left) and on Earth (right). By [Pbroks13 / Markus Poessel \(Mapos\), CC BY-SA 3.0](#), via Wikimedia Commons.

According to the EP, the laws of special relativity hold in both scenarios in Fig. 1. The Relativistic Rocket equations describe the ball's motion relative to the rocket. Then those equations describe the ball's motion relative to the room on Earth as well, where a is the acceleration of gravity. The time t in (1) is measured in the ball's LIF, in which the rocket (or room) blasted off when the ball was dropped.

Experiment #1: A ball is thrown upward from the ground to a maximum height $H = 10$ km in a hypothetical $1 g$ uniform gravitational field.

Let's model this experiment in the ground's frame, using the following Relativistic Rocket equations:

$$T = \frac{c}{a} \operatorname{acosh}(ad/c^2 + 1) \quad (3)$$

$$\gamma = \cosh \frac{aT}{c} \quad (4)$$

$$d = \frac{c^2}{a} \left(\cosh \frac{aT}{c} - 1 \right) = \frac{c^2}{a} (\gamma - 1) \quad (5)$$

The thrower is like the person in the rocket in Fig. 1. The elapsed time in their frame (i.e. how much they age) is T . To plot the ball's rise and fall we plot h , the ball's height in the ground's frame, versus T . The distance d is measured in the ball's frame; in the ground's/rocket's frame it's length contracted by the [gamma factor](#) γ , so $h = d/\gamma$. At the ball's apex its velocity $v = 0$, so that $\gamma = 1$ and $h = d$ there.

Do a one-time calculation of the elapsed time for the ball to fall from its apex to the ground, using (3):

$$T_a = \frac{c}{a} \operatorname{acosh}(aH/c^2 + 1) \quad (6)$$

Use $c = 299,792,458$ m/s, $a = 9.80665$ m/s², and $H = 10,000$ m. The ground accelerates like a rocket to reach the ball after the time T_a . Since an object's fall reverses its rise, T_a is also the time for the ball to rise from the ground to its apex. So you need many values of T that range from 0 to $2T_a$.

Calculate γ for each value of T , using (4) with an offset to the time:

$$\gamma = \cosh \frac{a(T_a - T)}{c} \quad (7)$$

Calculate d for each value of T , using (5) with an offset to the distance, and the γ from (7):

$$d = H - \frac{c^2}{a} (\gamma - 1) \quad (8)$$

Now we can plot h (in km) versus T :

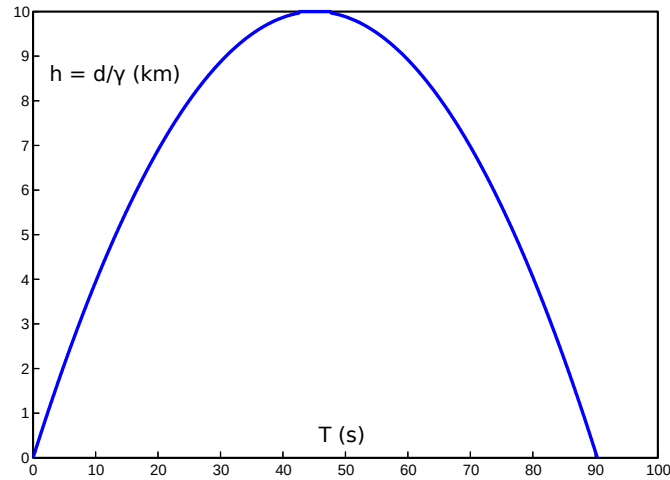


Figure 2: Ball thrown upward from the ground to a maximum height $H = 10$ km in a $1 g$ uniform gravitational field.

Compare Fig. 2 to a plot drawn by a [projectile motion calculator](#) that uses Newton's equations for a uniform gravitational field. They look alike, because Newton's equations approximate the Relativistic Rocket equations when both the acceleration of gravity a and the maximum height H are sufficiently small.

Experiment #2: A ball is thrown upward from the ground to a maximum height $H = 10$ light years in a hypothetical $1 g$ uniform gravitational field.

Use $c = 1$ ly/yr, $a = 1.03$ ly/yr² $\approx g$, and $H = 10$ ly.

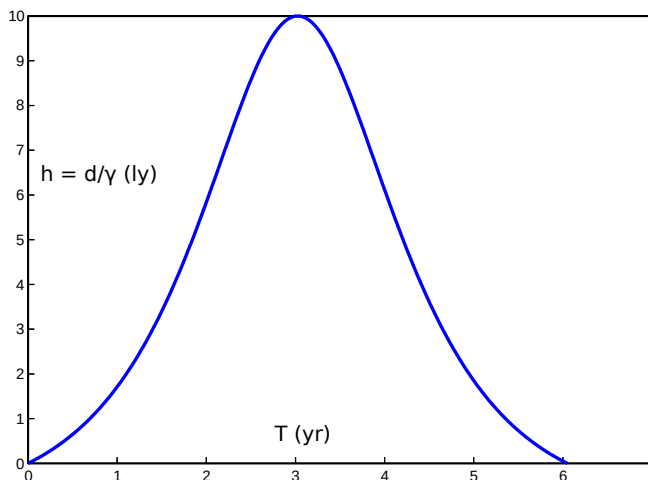


Figure 3: Ball thrown upward from the ground to a maximum height $H = 10$ ly in a $1 g$ uniform gravitational field.

See that Fig. 3 looks close to the plot at The Relativistic Rocket, in the section “The Equivalence Principle and a Stone Thrown Upwards”. (The site uses $a = 1 \text{ ly/yr}^2$ there.) A projectile motion calculator that uses Newton’s equations would invalidly plot a parabola.

Fig. 3 shows that the ball accelerates up initially, and when it falls it decelerates as it nears the ground.

2 Space expands and contracts relatively

The ball accelerates up in Fig. 3 due to the reverse of length contraction (hereafter “length expansion”) of the space between the thrower and the ball, that occurs in the ground’s frame. Throw a ball upward, while imagining that a string dangles down from it. The string moves relative to you, so it’s length contracted in your frame, however slightly. The string expands toward its proper length while passing you, so some of the ball’s movement away from you is due to the length expansion of the string. As this length expansion still happens when you remove the string from the picture, it’s simpler seen as an expansion of space that occurs between you and a free-falling object that’s rising away from you. It’s a relative expansion that causes the object to accelerate up when the expansion is large enough. The reverse happens between you and a free-falling object that’s falling toward you. In that case the space between you and the object relatively contracts due to length contraction. This causes the ball to decelerate down in Fig. 3.

The Relativistic Rocket gives an equation to predict when an object will accelerate up or decelerate down (in which case its plot of h vs. T will be bell-shaped, however slightly):

$$a > (\sqrt{2} - 1) c^2/H \quad (9)$$

and says “This means that if you want to perform such an experiment, either $[a]$ or H will have to be very large.” Note that the minimum a is inversely proportional to H .

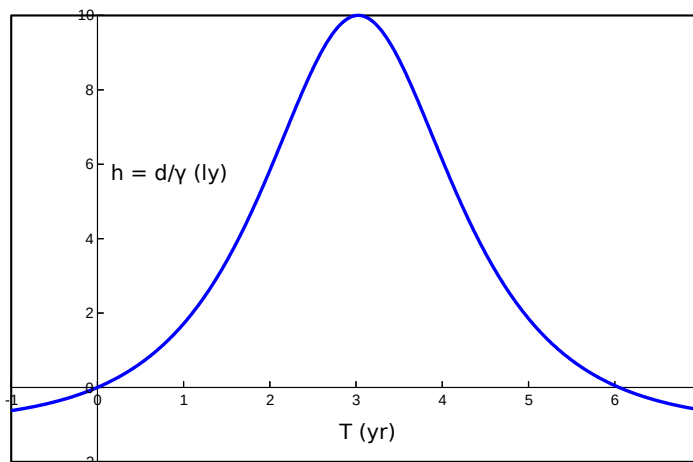


Figure 4: This is Fig. 3 with extended tails.

The relative expansion and contraction of space also affects light. Shine light upward. Then the maximum height H in (9) is infinite, so the light accelerates away. (A ball thrown upward at close to c is a proxy for the light that recedes faster.) The right tail of Fig. 4 shows that light shone downward decelerates away. Measure the height of a building with a sufficiently precise [laser rangefinder](#), from both the top and bottom of the building. This measures the time taken by a laser pulse to reflect off a target and return to its sender. Whether the height is measured from the top or bottom, the same set of one-way pulses are used; only their order differs. The height measures taller from the top, because clocks run faster there due to gravitational time dilation, hence more time elapses between the start and end of the same set of one-way pulses. Fig. 4 shows what happens to the light. When the height is measured from the bottom, the light accelerates up and decelerates down, its speed averaging $> c$. When the height is measured from the top, the light decelerates down and accelerates up, its speed averaging $< c$. The proper height of the building is between the two heights reported by the rangefinder.

3 Numerical integration for a non-uniform gravitational field

The Relativistic Rocket equations apply only to uniform gravitational fields. [Here is code](#) to numerically integrate the relativistic motion of a ball thrown upward to a given maximum radial coordinate in a non-uniform gravitational field. Click the Run button to get the output:

```
The ball apexes at 1.0000e+01 ly after 3.0230e+00 yr
Its velocity v = 0.3559% of c at the apex
```

This result approximates the expected result for experiment #2, as depicted by Fig. 3. The program stops when the ball apexes. You can change the inputs at the top of the program, as explained therein. The closer the reported velocity v at the apex is to the expected zero, the

better the approximation. To get closer to $v = 0$ increase the `segmentCount`, the number of steps in the numerical integration. At a high enough `segmentCount` the program times out when run online. You can install the [Go language](#) to run the program on your own machine. The program outputs data points for T and h when `outputCount > 0`.

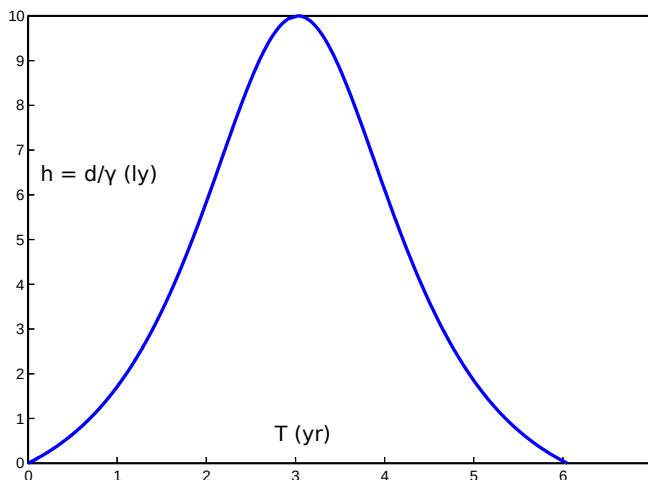


Figure 5: This is Fig. 3, except that the ball's rise to its apex uses the data output by the numerical integration program, for a non-uniform gravitational field where the acceleration of gravity stays close to $1 g$.

[Here is code](#) for a helper program to get inputs for the numerical integration program. Click the Run button to get the output:

```
Use these inputs for the numerical integration program:
  M = 1.0300e+12 ly
  rMin = 1e+06 ly
  rDiff = 6.1500e+01 ly
Acceleration of gravity at
  rMin = 1.0300e+00 ly/yr^2
  rMax = 1.0299e+00 ly/yr^2
The ball apexes at 1.0000e+01 ly after 3.0252e+00 yr
```

This program helps especially for getting inputs for a gravitational field that's close to uniform. The output above is for experiment #2, using [geometric units](#). The time to reach the apex is predicted by The Relativistic Rocket's (6). See that the acceleration of gravity at `rMin` is about the same as at `rMax` ($= rMin + rDiff$). Increase the input for the `uniformityLevel` to make the gravitational field more uniform. Decrease the `uniformityLevel`, even to a negative value, for a less uniform field. But then don't expect the results to approximate those from the numerical integration program.

Let's use the helper program to get inputs for the numerical integration program for experiment #1. Change the inputs at the top to `a = 9.80665`, `H = 10000`, `MLabel = "kg"`, `dLabel = "m"`, `tLabel = "s"`, `G = 6.67430e-11`, and `c = 299792458`. Run the program. When `uniformityLevel = 5`, the output includes

Use these inputs for the numerical integration program:

M = 1.4693e+27 kg
 rMin = 1e+08 m
 rDiff = 1.0000e+04 m

The ball apexes at 1.0000e+04 m after 4.5160e+01 s

Input those values into the numerical integration program, along with the values above for dLabel, tLabel, G, and c. Run the program. When segmentCount = 1e+7, the output is

The ball apexes at 1.0000e+04 m in 4.5154e+01 s
 Its velocity v = 0.0000% of c at the apex

When the following values for Earth are input:

M = 5.9724e+24 kg
 rMin = 6.3781e+06 m

the output is

The ball apexes at 1.0000e+04 m in 4.5227e+01 s
 Its velocity v = 0.0000% of c at the apex

Here are finer points of the numerical integration program:

The rDiff doesn't mean the same as the maximum height H that's input into the helper program. The height there is the distance d covered by the ball in its frame. The helper program finds rDiff for a uniform gravitational field by using the Relativistic Rocket equation

$$t = \sqrt{(d/c)^2 + 2d/a} \quad (10)$$

The inputted H is plugged in for d . This gives the elapsed time in the ball's frame. Then rDiff is found by plugging that value for t into

$$d = \frac{1}{2}at^2 \quad (11)$$

from [Equations for a falling body](#). In (11), d is the same as the difference between two radial coordinates, or rDiff. Radial coordinate distances are length contracted in the ball's frame until the ball apexes, so rDiff > H .

The code is similar to that in [How the twins each age less than the other](#) for a twin paradox experiment. The ball is the stationary twin. The thrower is the traveling twin, as if accelerating in a rocket. Their velocity relative to each other, in either's frame, is calculated by

$$v_i = \sqrt{2GM \left(\frac{1}{r} - \frac{1}{r+d} \right)} \quad (12)$$

from [Equations for a falling body](#), described therein as the "Instantaneous velocity v_i of a falling object that has travelled distance d on a planet with mass M and radius r ." We can use this velocity as well for the ball that's rising, since an object's fall reverses its rise. Except that this

velocity isn't relativistic/valid because it can be $\geq c$. So v_i is converted to a relativistic velocity using

$$v_{\text{new}} = \frac{v_{\text{old}}}{\sqrt{1 + (v_{\text{old}}/c)^2}} \quad (13)$$

This equation derives from the following reasoning: Drop an object from a radial coordinate. For the first small segment of the object's fall, in which the acceleration of gravity a is constant, (2) approximates (12), and The Relativistic Rocket's (1) approximates the new velocity v_i . So, substituting the two terms at in (1) with the velocity returned by (12) gives the new velocity v_i . This is the same as dividing v_{old} by the gamma factor, since

$$\gamma = \sqrt{1 + (at/c)^2} \quad (14)$$

from The Relativistic Rocket. This conversion effectively changes general relativity's equation for escape velocity (which is (12) when d is infinite) such that the value it returns is always $< c$, thus black holes aren't supported. See also [A solution to the black hole information paradox](#), which gives a new metric for Schwarzschild geometry. More on this below.

The message "Reduce the segmentCount" means that division by zero would otherwise occur, due to limited precision.

4 Space itself expanding is superfluous

I posit that instead of space itself expanding, space expands only relatively due to length expansion, still stretching light. This eliminates the [flatness problem](#), which depends on the assumption that space itself is expanding. The Relativistic Rocket says "For distances greater than about a thousand million light years, the formulae given here are inadequate because the universe is expanding." When space expands only relatively then those equations for flat spacetime apply to arbitrarily large distances in principle. Parallel light rays can remain parallel indefinitely.

The [gamma factor](#) equation shows that all objects have a gamma factor γ as a function of their velocity, and regardless of their other properties. You might need a LIF to measure or predict velocity or γ , but not to apply them. As you measure, a radially moving free-falling object is length contracted by its current factor of γ along that axis of motion, and so is its proper distance from you. Imagine that a string of unconnected beads, that are all momentarily at rest relative to the object, dangles from the object all the way to you. See that the string is length contracted. (Using beads in this way makes the tidal force ignorable.) For example, the Andromeda galaxy is somewhat length contracted as we measure, and its proper distance from us is contracted by the same gamma factor. The space between you and the object length expands or further contracts as the object decelerates or accelerates radially under gravity.

The Relativistic Rocket shows that a rocket can blast off from Earth, accelerate and decelerate at $1g$ to arrive at the center of our galaxy at relative rest, 30,000 light years away from Earth,

while its crew ages just ~ 20 years. At the midpoint of the trip the rocket is less than $(20 \text{ yr} / 2) \times c = 10$ light years away from the center as measured by the crew, due to length contraction. The whole Milky Way is contracted by the same gamma factor, lest it be physically deformed. Let's model the second half of the trip. Let a ball be free-floating at the midpoint. A plot of the ball's height over time in the rocket's frame is the left half of the plot given by eqs. (6)-(8).

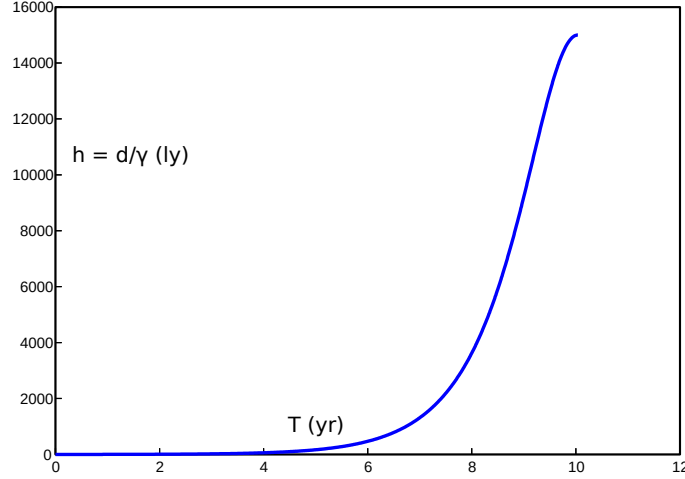


Figure 6: Ball passed by a rocket at the midpoint of a trip from Earth to the center of our galaxy, arriving at relative rest. The rocket decelerates at $1 g$.

Fig. 6 shows that the ball accelerates up initially. This is due to length expansion of the space between the rocket and the ball, that occurs in the rocket's frame. The whole Milky Way length expands by the same change in the gamma factor. When the rocket arrives at the center of the galaxy at relative rest, the galaxy is no longer length contracted. If the rocket kept its engines running and accelerated back to the ball, then the plot would be the full bell-shaped plot given by eqs. (6)-(8). As if the crew had thrown the ball upward to a maximum height $H = 15,000$ ly, and then it fell back to the rocket/"ground", all in a time $T \approx 20$ years.

Since a whole galaxy length contracts when a rocket accelerates across it, why isn't a whole black hole contracted to zero length when an observer reaches its event horizon at the speed of light? That's an illogical discrepancy. When the whole galaxy contracts, so must any black holes in it, lest the galaxy physically deform. The gamma factor equation tells us that we're free to think of velocity in terms of γ . Knowing an object's velocity, if only locally, we can know to what extent the whole object is contracted. If the numerical integration program didn't do the conversion described above using (13), which effectively changes general relativity's equation for escape velocity such that black holes aren't supported, then it wouldn't predict that an object can accelerate up. Therefore, its results for a gravitational field that's negligibly non-uniform (i.e. where the ground accelerates in a LIF) wouldn't always approximate the results from eqs. (6)-(8). This means that the Relativistic Rocket equations can't be derived from the Schwarzschild metric, thus the metric violates the EP, and means that a theory of gravity that obeys the EP can't predict black holes.

The universe needn't have a center or an edge when space itself doesn't expand or contract. All of the objects in any region of space can coalesce under gravity when space itself doesn't expand or contract. We can "run the film backward" to see that the reverse is true as well. As there's no size limit for the region, it can be the whole universe, with no center or edge. For an informative description of an expanding flat universe, see the book *Relativity Visualized*, by [Lewis Carroll Epstein](#), in the chapter "The Big Bang".

5 An object thrown upward in a highly non-uniform gravitational field

Let's use the numerical integration program to plot a ball thrown upward to a great height in a highly non-uniform gravitational field. Change the inputs at the top to $M = 1e+14$, $r_{\text{Min}} = 1e+10$, $r_{\text{Diff}} = 1e+12$, $d_{\text{Label}} = \text{"ly"}$, $t_{\text{Label}} = \text{"yr"}$, $G = 1$, and $c = 1$. Run the program. When $\text{segmentCount} = 1e+7$, the output is

```
The ball apexes at 1.0328e+11 ly after 2.9044e+10 yr
Its velocity v = 0.4428% of c at the apex
```

Increasing the segmentCount to $2e+10$ improves the results to

```
The ball apexes at 1.0328e+11 ly after 2.9077e+10 yr
Its velocity v = 0.0099% of c at the apex
```

which gives the plot:

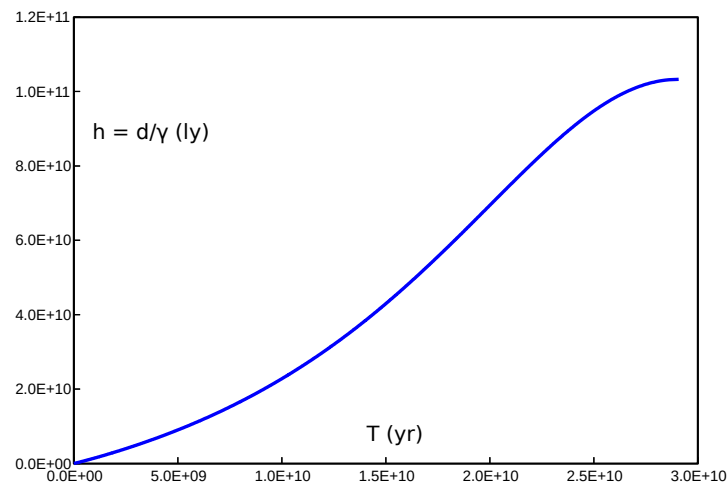


Figure 7: Ball thrown upward to a great height in a highly non-uniform gravitational field.

The ball apexes at ~ 103 billion light years after ~ 29 billion years. The acceleration of gravity ranges from ~ 1 one-millionth of g at r_{Min} (the ground, if only the "surface" of a sphere that contains the mass M) to ~ 1 ten-billionth of g at the ball's apex. So see that the acceleration of gravity needn't be large to cause an object to accelerate up, as predicted locally by (9).

6 Why the Webb telescope won't see the first galaxies

Eqs. (6)-(8) and the numerical integration program show that

- An object can in principle be thrown upward to any distance and fall back to the ground, all in an arbitrarily small time T . Then an object or light can in principle rise to any distance, or fall from any distance, in an arbitrarily small time T . This eliminates the [horizon problem](#). A region of space that's any distance from us can have been in causal contact with a region of space that's any distance from us in the opposite direction, since any age of our length-expanding universe > 0 as we measure. The [observable universe](#) can now be arbitrarily large.
- Free-falling objects that rise at sufficiently close to c accelerate away. This obviates the need for [dark energy](#). The evidence for dark energy is high-redshift supernovae that seem to be accelerating away. Since space itself doesn't expand in the new model, a high redshift indicates a velocity close to c .

It follows that the [first galaxies](#) can in principle be much further away or much more redshifted than the current model predicts. For this reason, and because we've already seen galaxies that look much older than expected, such as [ALESS 073.1](#), I expect that the Webb telescope won't be able to image the first galaxies.

Appendix A - Code for the programs

Below is the [Go language](#) code for the numerical integration program that's referenced in section 3, in case the link to the code is broken. You can run the code at the [Go Playground](#) after fixing the formatting.

```
package main

import (
    "fmt"
    "math"
)

const (
    // The mass of the massive body
    M = 1.0300e+12

    // The ground's radial coordinate (r-coordinate)
    rMin = 1e+06

    // The ball is thrown upward from rMin and apexes at rMin + rDiff
    rDiff = 6.1500e+01

    // The number of steps in the numerical integration below. More
    steps gives greater accuracy
    segmentCount = 1e+7
)
```

```

// The distance label used for output
dLabel = "ly"

// The time label used for output
tLabel = "yr"

// The gravitational constant
G = 1

// The speed of light
c = 1

// The number of data points to output
outputCount = 0

// End of user input

// The maximum r-coordinate
rMax = rMin + rDiff

// The r-coordinate distance that is considered in each step
rSegment = rDiff / segmentCount
)

func main() {
// Test whether "+Inf" would be output, due to limited precision
if rMin + (float64(segmentCount - 1) * rSegment) == rMax {
    fmt.Println("Reduce the segmentCount")
    return
}

outputEvery := 0
if outputCount > 0 {
// Output data every outputEvery-th iteration
// Adjust for: The origin and last data points are always
output
    outputCount2 := math.Max(1, float64(outputCount) - 2)
    outputEvery = int(math.Ceil(segmentCount / outputCount2))
    fmt.Printf("T, h = d/\gamma\r\n") // Print data header
    fmt.Printf("%0.4e, %0.4e\r\n", 0.0, 0.0) // Print origin
data point
}

// The twins in this twin paradox experiment are the ball and the
thrower
d := 0.0 // The distance the twins move relative to each other in
the ball's frame
T := 0.0 // The thrower's aging
v := 0.0 // Their velocity relative to each other, in either's
frame

for i := 0; i < segmentCount; i++ {
    r := rMin + (float64(i) * rSegment)

```

```

// From
https://en.wikipedia.org/wiki/Equations_for_a_falling_body
vOld := math.Sqrt((2 * G * M) * ((1 / r) - (1 / rMax)))

// From the Relativistic Rocket site
gamma := math.Sqrt(1 + math.Pow(vOld / c, 2))

// Convert vOld to a relativistic velocity, as explained in
the paper
v = vOld / gamma

// rSegment is divided by gamma because r-coordinate
distances length expand in the ball's frame as it rises
// For example, see the muon experiment at
https://en.wikipedia.org/wiki/Experimental_testing_of_time_dilation
// r-coordinate distances length contract in the muon's
frame as it falls
dSegment := rSegment / gamma

d += dSegment

// dSegment / v is the time taken in the ball's frame to
cover dSegment
// dSegment / gamma is the distance the ball covers in the
ground's frame
T += dSegment / (v * gamma)

if outputCount > 0 {
    if i % outputEvery == 0 || i == (segmentCount - 1) {
        fmt.Printf("%0.4e, %0.4e\r\n", T, d / gamma)
    }
}

}

fmt.Printf("The ball apexes at %0.4e %s after %0.4e %s\r\n", d,
dLabel, T, tLabel)
fmt.Printf("Its velocity v = %0.4f%% of c at the apex\r\n", (v /
c) * 100)
}

```

Below is the Go language code for the helper program that's referenced in section 3.

```

package main

import (
    "fmt"
    "math"
)

const (
    // The acceleration of gravity at rMin
    a = 1.03

    // The ball's maximum height
    H = 10

```

```

// The mass label used for output
MLabel = "ly"

// The distance label used for output
dLabel = "ly"

// The time label used for output
tLabel = "yr"

// The gravitational constant
G = 1

// The speed of light
c = 1

// Increase this number to make the gravitational field more
uniform, at the expense of larger numbers output for "Use these inputs"
// Decrease this number (even to a negative value) to make the
gravitational field less uniform
uniformityLevel = 5

// End of user input

aLabel = dLabel + "/" + tLabel + "^2"
)

func main() {
// These equations are from the Relativistic Rocket site
// The ball's aging
tExpected := math.Sqrt(math.Pow(H / c, 2) + 2 * H / a)
// The thrower's aging
TExpected := (c / a) * math.Acosh(a * H / math.Pow(c, 2) + 1)

// Calculate the difference d between rMin and rMax, given t
// This equation is from
https://en.wikipedia.org/wiki/Equations\_for\_a\_falling\_body
rDiff := a * math.Pow(tExpected, 2) / 2

// Let the acceleration of gravity at rMin be the acceleration a
// Calculate an rMin where the acceleration of gravity at rMax
won't differ much
rMin := math.Pow(10, math.Floor(math.Log10(rDiff)) +
uniformityLevel)

M := math.Pow(rMin, 2) * a / G

rMax := rMin + rDiff

fmt.Printf("Use these inputs for the numerical integration
program:\n")
fmt.Printf("\tM = %0.4e %s\n", M, MLabel)
fmt.Printf("\trMin = %0.0e %s\n", rMin, dLabel)
fmt.Printf("\trDiff = %0.4e %s\n", rDiff, dLabel)
fmt.Printf("Acceleration of gravity at\n")

```

```
        fmt.Printf("\trMin = %0.4e %s\n", a, aLabel)
        fmt.Printf("\trMax = %0.4e %s\n", G * M / math.Pow(rMax, 2),
aLabel)
        fmt.Printf("The ball apexes at %0.4e %s after %0.4e %s\n",
float64(H), dLabel, TExpected, tLabel)
    }
```

Text is available under the [Creative Commons Attribution-ShareAlike License 4.0](https://creativecommons.org/licenses/by-sa/4.0/).