horizon

January 29, 2019

```
In [2]: import numpy as np
        import matplotlib.pyplot as plt

In [69]: H = 10**(-2)
         def radius(t, r0=1, H=H):
             return np.exp(H*t + np.log(r0))

         def velocity(t, r0=1, H=H):
             return H*radius(t,r0=r0,H=H)

         def acceleration(t, r0=1, H=H):
             return H**2*radius(t,r0=r0,H=H)

In [70]: def polar_coors(t, r0=1, theta=0, H=H):
             return (radius(t, r0=r0, H=H), theta)

         def cart_coors(t, x0=1, y0=0, H=H):

             # convert cartesian to polar
             r0 = np.sqrt(x0**2 + y0**2)
             theta = np.arctan2(y0,x0)

             # find new positions
             r = radius(t, r0=r0, H=H)
             theta = theta

             # convert back to cartesian
             x = r*np.cos(theta)
             y = r*np.sin(theta)

             return (x,y)


In [71]: # generate latice of points
         x_step = 1
         y_step = 1

         x_range = [-5,5]
         y_range = [-5,5]

         nx = (x_range[1] - x_range[0] + 1)/x_step
         ny = (y_range[1] - y_range[0] + 1)/y_step

         xs = np.linspace(x_range[0], x_range[1], nx)
         ys = np.linspace(y_range[0], y_range[1], ny)

         xx, yy = np.meshgrid(xs, ys, indexing='ij')
         coordinate_grid_t0 = np.array([xx, yy])

         print(coordinate_grid_t0[0,:,:].flatten())
```

```
11
[-5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -5. -4. -4. -4. -4. -4. -4. -4.
 -4. -4. -4. -4. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -3. -2. -2. -2.
 -2. -2. -2. -2. -2. -2. -2. -2. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1.
 -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  1.  1.  1.  1.  1.  1.
  1.  1.  1.  1.  1.  2.  2.  2.  2.  2.  2.  2.  2.  2.  2.  2.  3.  3.
  3.  3.  3.  3.  3.  3.  3.  4.  4.  4.  4.  4.  4.  4.  4.  4.  4.
  4.  4.  5.  5.  5.  5.  5.  5.  5.  5.  5.  5.  5.]
```

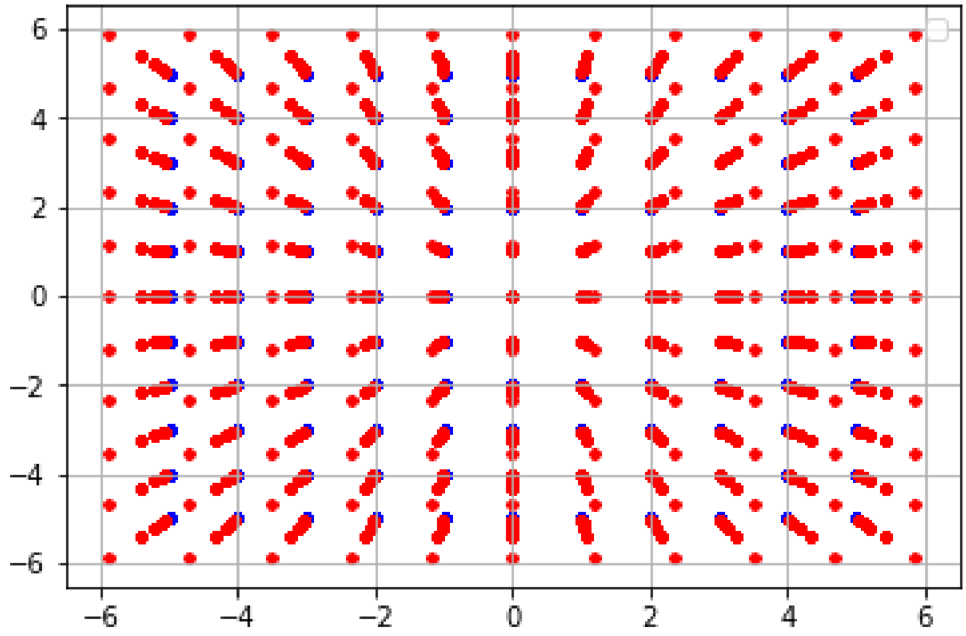
```
/anaconda3/lib/python3.5/site-packages/ipykernel_launcher.py:11: DeprecationWarning: object of
  # This is added back by InteractiveShellApp.init_path()
/anaconda3/lib/python3.5/site-packages/ipykernel_launcher.py:12: DeprecationWarning: object of
  if sys.path[0] == '':
```

```
In [72]: def latice_arrays_at_time(t, coordinate_grid_t0, H=H):
             n = coordinate_grid_t0.shape[1]**2*coordinate_grid_t0.shape[2]**2
             xs = np.empty(shape=(n,))
             ys = np.empty(shape=(n,))
             counter = 0
             for x0 in coordinate_grid_t0[0,:,:].flatten():
                 for y0 in coordinate_grid_t0[1,:,:].flatten():
                     # apply function
                     xs[counter],ys[counter] = cart_coors(t, x0=x0, y0=y0)
                     counter += 1
             return xs, ys
```

```
In [77]: # plot coordinates over tme

         xs0, ys0 = latice_arrays_at_time(0, coordinate_grid_t0)
         xs1, ys1 = latice_arrays_at_time(1, coordinate_grid_t0)
         xs2, ys2 = latice_arrays_at_time(2, coordinate_grid_t0)
         xs4, ys4 = latice_arrays_at_time(4, coordinate_grid_t0)
         xs8, ys8 = latice_arrays_at_time(8, coordinate_grid_t0)
         xs16, ys16 = latice_arrays_at_time(16, coordinate_grid_t0)
```

```
plt.scatter(xs0,ys0, c='b', s=10)
plt.scatter(xs1,ys1, c='r', s=9)
plt.scatter(xs2,ys2, c='r', s=8)
plt.scatter(xs4,ys4, c='r', s=7)
plt.scatter(xs8,ys8, c='r', s=6)
plt.scatter(xs16,ys16, c='r', s=5)

#plt.plot(xs,ys, c='r', label='f')

plt.legend()
plt.grid()
plt.show()
```

```
/anaconda3/lib/python3.5/site-packages/ipykernel_launcher.py:3: RuntimeWarning: divide by zero
  This is separate from the ipykernel package so we can avoid doing imports until
No handles with labels found to put in legend.
```


6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6