horizon

January 29, 2019

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
```

```
In [19]: 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 [20]: 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 [21]: # 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())
```

```
[-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.  3.  3.  4.  4.  4.  4.  4.  4.  4.  4.  4.
  4.  4.  5.  5.  5.  5.  5.  5.  5.  5.  5.  5.  5.]
```

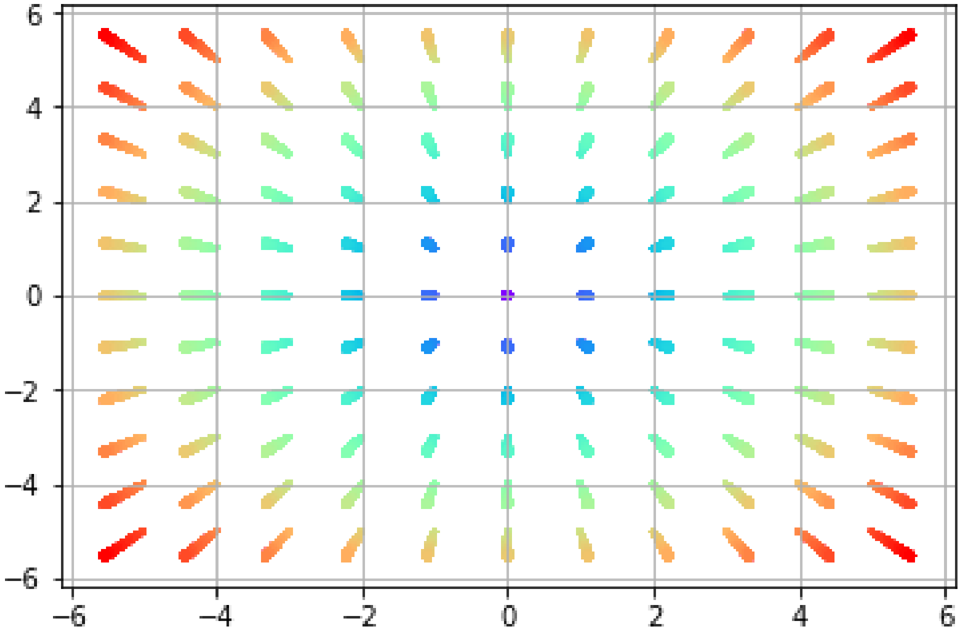
```
/anaconda3/lib/python3.5/site-packages/ipykernel_launcher.py:11: DeprecationWarning: object of
  # This is added back by InteractiveShellApp.init_path()
/anaconda3/lib/python3.5/site-packages/ipykernel_launcher.py:12: DeprecationWarning: object of
  if sys.path[0] == '':
```

```
In [22]: 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,))
             velocities = 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, H=H)
                     velocities[counter] = velocity(t, r0=np.sqrt(x0**2+y0**2), H=H)
                     counter += 1

             return xs, ys, velocities
```

```
In [23]: import matplotlib.cm
         print(matplotlib.cm.cmap_d.keys())
```

```
dict_keys(['CMRmap', 'PuOr_r', 'jet_r', 'GnBu', 'magma', 'hsv', 'YlGn_r', 'CMRmap_r', 'BrBG',
In [42]: # plot coordinates over time
         import matplotlib

         max_t = 10
         ts = np.linspace(0,max_t,max_t+1)

         t_slices = []
         for t in ts:
             xst, yst, vst = latice_arrays_at_time(t, coordinate_grid_t0)
             t_slices.append([xst, yst, vst])

         # Redshift, Normalize to extreems
         flatten = lambda l: [item for sublist in l for item in sublist]
         combinded_vs = flatten([list(s[-1]) for s in t_slices])

         cmap = matplotlib.cm.get_cmap('rainbow')
         normalize = matplotlib.colors.Normalize(vmin=min(combinded_vs), vmax=max(combinded_vs

         # plots
         for t, t_slice in zip(ts, t_slices):
             xst, yst, vst = t_slice
             cst = [cmap(normalize(v)) for v in vst]
             plt.scatter(xst,yst,c=cst, s=int((t+1)**(0.85)))

         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
```


6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6