1 Code

```
from pysph.sph.equation import Equation

class BlackHole2D(Equation):
    def __init__(self, dest, sources, soft=0.05, t_hit=5.0, M=1.0):
        self.soft = soft # softening length to not divide by zero
        self.t_hit = t_hit # time when the black hole crosses the origin
        self.M = M # mass of black hole
        super(BlackHole2D, self).__init__(dest, sources)

    def initialize(self, d_idx, d_au, d_av):
        d_au[d_idx] = 0.0
        d_av[d_idx] = 0.0

    # calculate the force due to the black hole
    def loop(self, d_x, d_y, d_idx, d_au, d_av, t):
        d_au[d_idx] += -self.M * d_x[d_idx] / pow((d_x[d_idx]**2 + self.soft
    **2 + (d_y[d_idx] + t - self.t_hit)**2),3.0/2.0)
        d_av[d_idx] += -self.M * (d_y[d_idx] + t - self.t_hit) / pow((d_x[
    d_idx]**2 + self.soft**2 + (d_y[d_idx] + t - self.t_hit)**2),3.0/2.0)
```

Figure 1: Class for adding the acceleration due to the primordial black hole.

```
19 # Import the eqations
20 from pysph.sph.equation import Group
21 from pysph.sph.BlackHoleEquation import BlackHole2D

40 # Domain and reference values
41 Lx = 200.0; H = 30.0; Ly = 1.5*H
42 gy = -1.0
43 Vmax = np.sqrt(abs(gy) * H)
44 c0 = 10 * Vmax; rho0 = 1.0
45 p0 = c0*c0*rho0
46 gamma = 1.0
47
48 soft = 0.05
49 t_hit = 30.0
50 Mass = 20.0
51 tf = H + t_hit # Simulation ends when the black hole reaches the bottom of the
       tank
52
53 # Reynolds number and kinematic viscosity
54 Re = 0; nu = 0.01 # Ideal fluid
55
56 # Numerical setup
57 nx = 100; dx = Lx/nx
58 ghost_extent = 5.5 * dx
59 hdx = 1.2

82 class BlackHole(Application):

171     def create_equations(self):
172         # Formulation for REF1
173         equations1 = [

194             # Main acceleration block
195             Group(equations=[

212                 # Add the black hole
213                 BlackHole2D(dest='fluid', sources=None, soft=soft, t_hit=t_hit
        , M=Mass)
214
215             ]),
216         ]
```

Figure 2: Modifications of the hydrostatic tank.