Orbits Optimisation Exercise

Description

Problem

- Imagine we want to launch a probe to land on a distant object
  - e.g. manned mission to Mars

- Need to reach a certain point at a certain time with a certain velocity

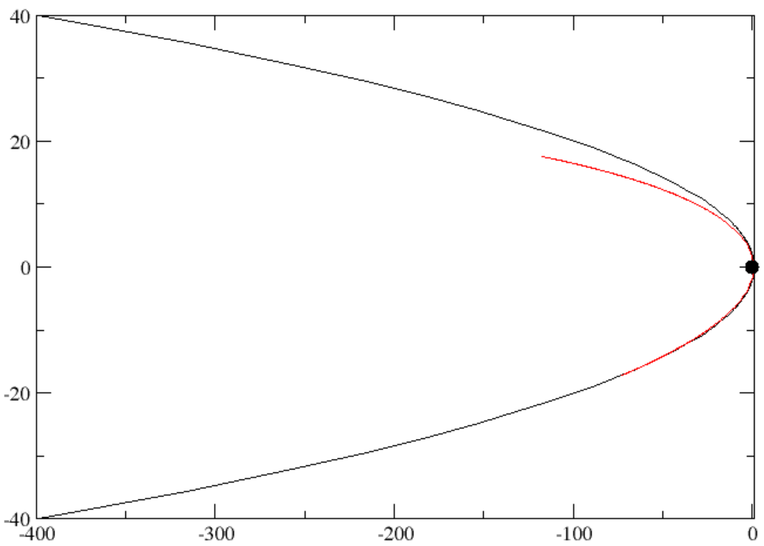
- In this exercise
  - given a fixed starting point, try to find the correct launch velocity that, after a fixed time, arrives at target point with correct velocity

Optimisation Problem

- Define an **objective function**
  - equals zero for correct solution
  - small (but positive) for a close solution
  - large (and positive) for a bad solution
- Run many simulations with different launch velocities
  - search for settings that minimise the objective function

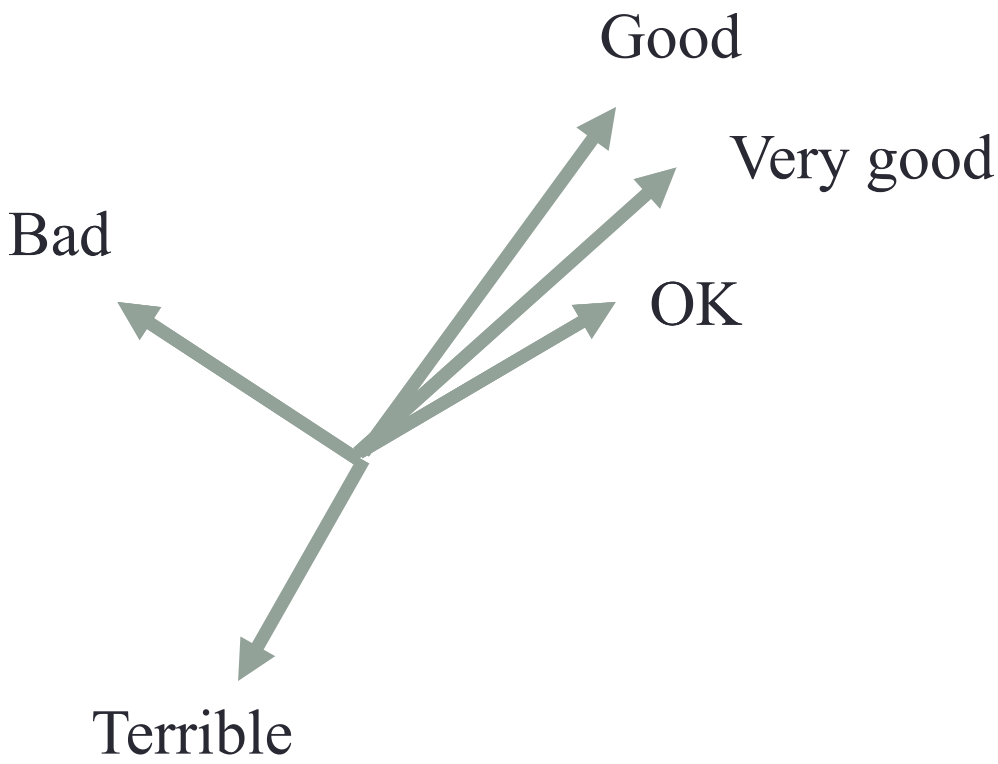
- Definition of objective function: objf
  - actual position and velocity are (x,y) and (vx, vy)
  - target values are (xf, yf) and (vxf, vyf)
  - $objf^2 = ((x-xf)/xf)^2 + ((y-yf)/yf)^2 + ((vx-vxf)/vxf)^2 + ((vy-vyf)/vyf)^2$

Initial setup




```
Orbit simulation

mass of object 1: 1000.000000
mass of object 2: 0.100000
end time for the integration: 30.000000
integration time step: 0.000100

Initial guess for velocity vector is (5.000000, 0.600000)

Starting time integration (300000 steps)...
... finished

Final  position = (-117.489232, 17.561779), velocity = (-3.934065, 0.228667)
Target position = (-119.165926, 21.929982), velocity = (-4.046287, 0.369329)

Error in final values = 0.430924
```

Picking the velocity



Approaches

- Search points on a 2D grid that includes many velocities
  - pick the best
- Steepest descent
  - change the velocity and accept if it reduces the objective function
  - reduce the size of the change as we approach solution
- Feed into a black-box optimiser
  - “please minimise f(vx, vy)”
- etc. etc. ...