```python
# newton - Newton-Raphson solver
#
# For APC 524 Homework 3
# CWR, 18 Oct 2010

import numpy as N
import functions as F

class Newton(object):
    def __init__(self, f, tol=1.e-6, maxiter=20, dx=1.e-6):
        """Return a new object to find roots of f(x) = 0 using Newton's method.
        tol:     tolerance for iteration (iterate until |f(x)| < tol)
        maxiter: maximum number of iterations to perform
        dx:      step size for computing approximate Jacobian"""
        self._f = f
        self._tol = tol
        self._maxiter = maxiter
        self._dx = dx

    def solve(self, x0):
        """Return a root of f(x) = 0, using Newton's method, starting from
        initial guess x0"""
        x = x0
        for i in xrange(self._maxiter):
            fx = self._f(x)
            if N.linalg.norm(fx) < self._tol:
                return x
            x = self.step(x, fx)
        return x

    def step(self, x, fx=None):
        """Take a single step of a Newton method, starting from x
        If the argument fx is provided, assumes fx = f(x)"""
        if fx is None:
            fx = self._f(x)
        Df_x = F.ApproximateJacobian(self._f, x, self._dx)
        h = N.linalg.solve(N.matrix(Df_x), N.matrix(fx))
        return x + h

```