cheg325 homework1 volterra coding

AUTHOR
k.wodehouse

PUBLISHED
February 14, 2025

Graphing

$$\tilde{\nabla} g(x_1, x_2) =< x_1 - x_2,\ x_2 - x_1 >$$

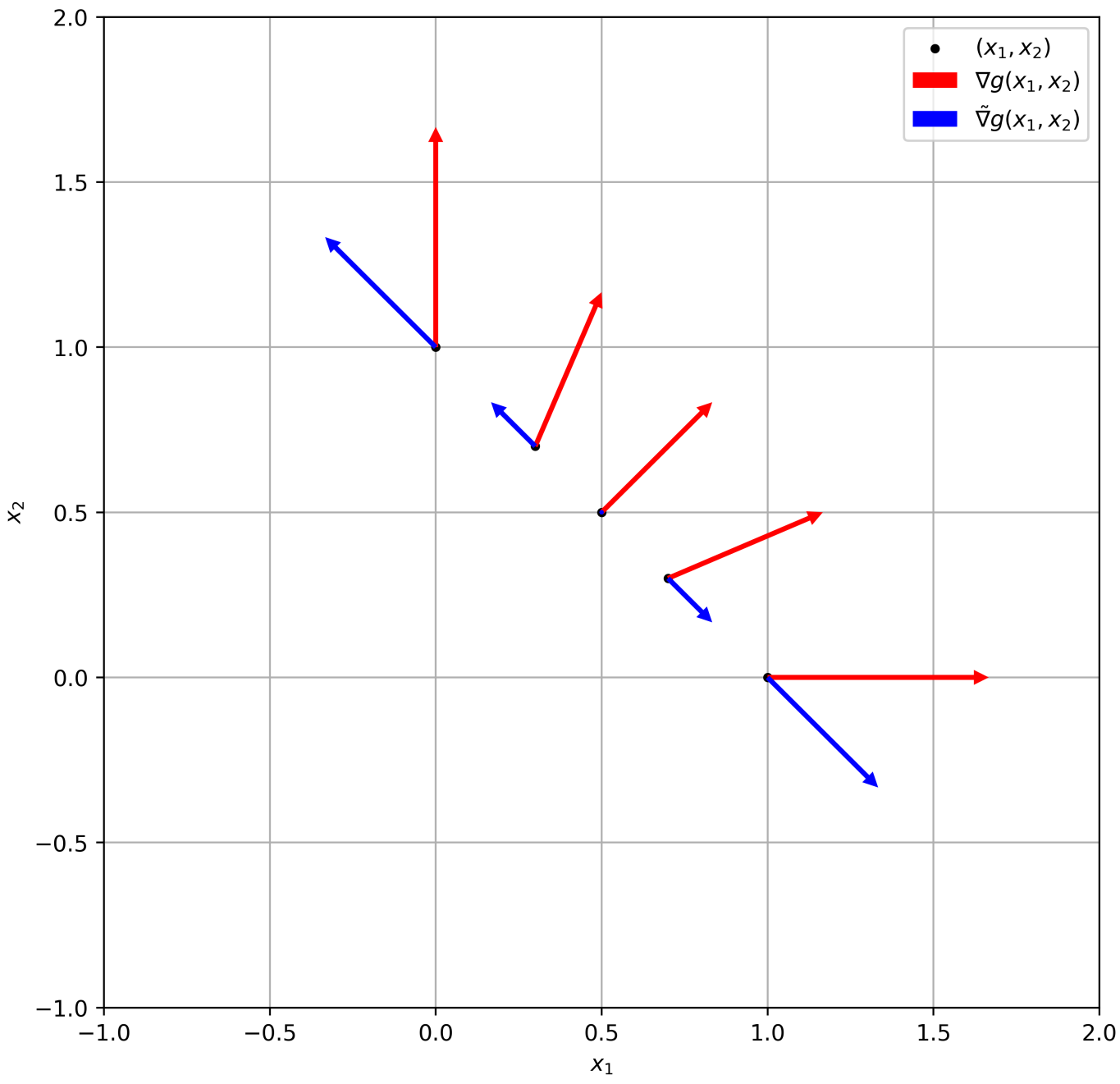
```
import numpy as np
import matplotlib.pyplot as plt

def molar_grad_g(x1, x2):
    return x1 - x2, x2 - x1

def grad_g(x1, x2):
    return 2*x1, 2*x2

x1 = np.array([1.0, 0.3, 0.5, 0.7, 0.0])
x2 = 1 - x1
xcomp, ycomp = molar_grad_g(x1, x2)
xcomp_grad, ycomp_grad = grad_g(x1, x2)


fig, ax = plt.subplots(figsize=(8,8), dpi=500, subplot_kw={'xlim':(-1,2), 'ylim':(-1,2),
ax.scatter(x1, x2, zorder=3, s=10, c='black')
ax.quiver(x1, x2, xcomp_grad, ycomp_grad, angles='xy', scale_units='xy', scale=3, width=0
ax.quiver(x1, x2, xcomp, ycomp, angles='xy', scale_units='xy', scale=3, width=0.005, zord
ax.grid(zorder=0)
ax.legend(['$(x_1, x_2)$', r'${\nabla}g(x_1, x_2)$' , r'$\tilde{\nabla}g(x_1, x_2)$']);
```



and now

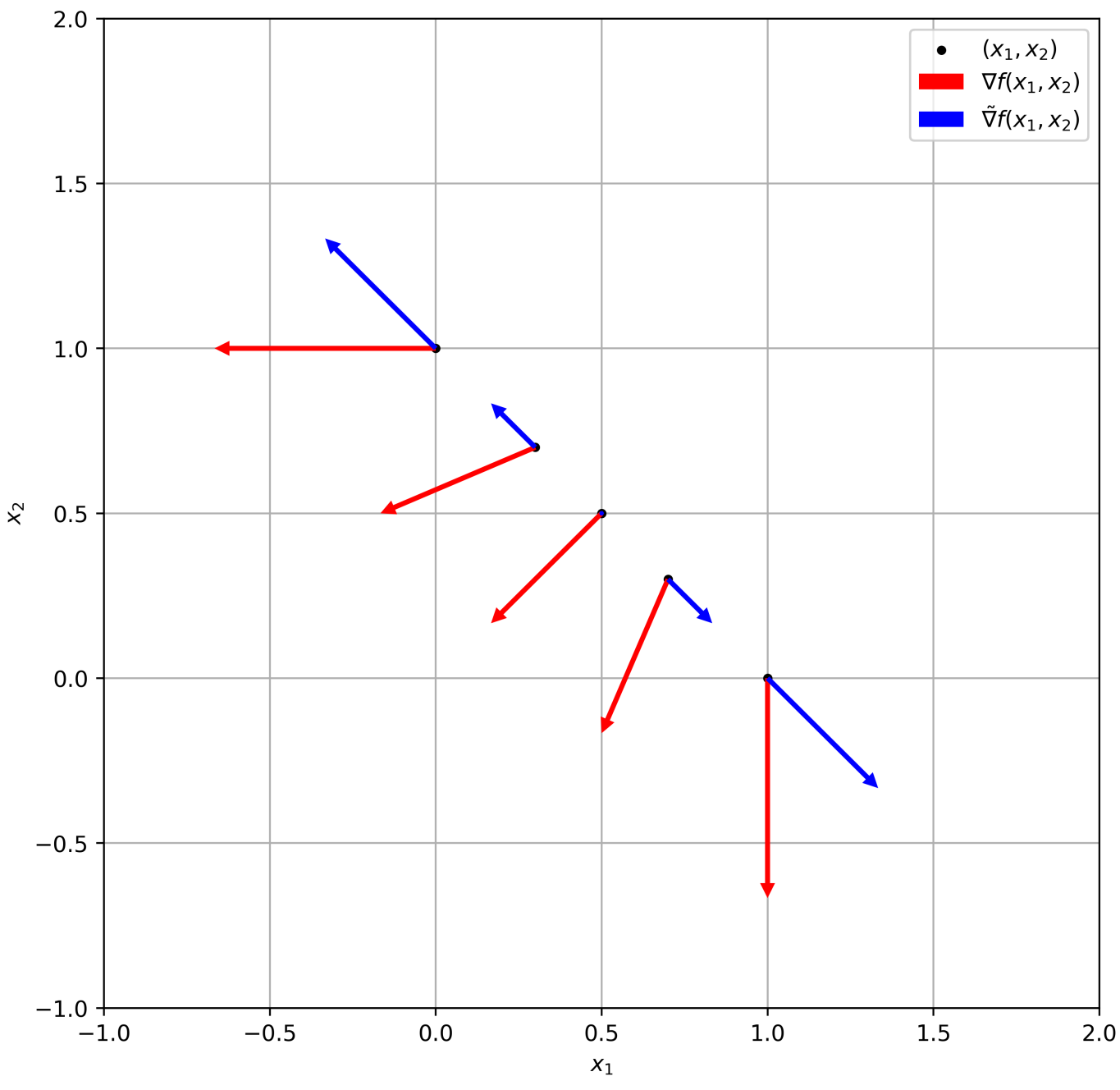
$$\tilde{\nabla} f(x_1, x_2) =< x_1 - x_2,\ x_2 - x_1 >$$

```
def molar_grad_f(x1, x2):
    return x1 - x2, x2 - x1

def grad_f(x1, x2):
    return 2*(x1 - 1), 2*(x2 - 1)

x1 = np.array([1.0, 0.3, 0.5, 0.7, 0.0])
x2 = 1 - x1
xcomp, ycomp = molar_grad_f(x1, x2)
xcomp_grad, ycomp_grad = grad_f(x1, x2)

fig, ax = plt.subplots(figsize=(8,8), dpi=500, subplot_kw={'xlim':(-1,2), 'ylim':(-1,2),
ax.scatter(x1, x2, zorder=3, s=10, c='black')
ax.quiver(x1, x2, xcomp_grad, ycomp_grad, angles='xy', scale_units='xy', scale=3, width=0
ax.quiver(x1, x2, xcomp, ycomp, angles='xy', scale_units='xy', scale=3, width=0.005, zord
ax.grid(zorder=0)
ax.legend(['$(x_1, x_2)$', r'${\nabla} f(x_1, x_2)$', r'$\tilde{\nabla} f(x_1, x_2)$']) ;
```




```
# this is filler text -- ignore.
```