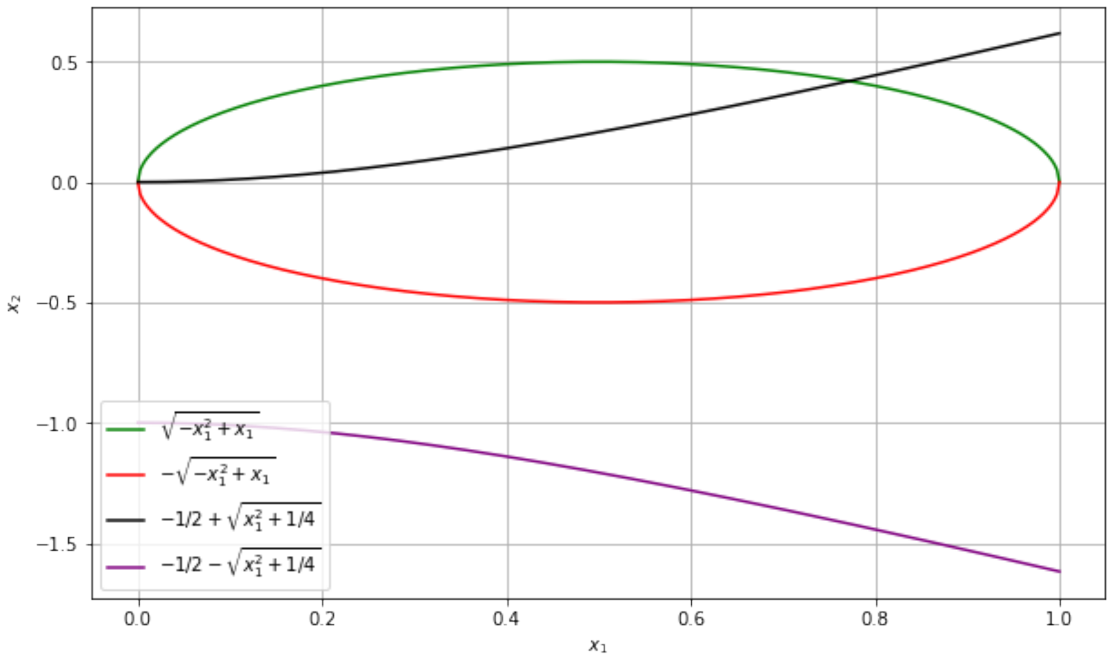

[370]:

```
# p1 fixed point theorem
def rhs(x):
    """ right hand side, takes in a vector of dimension 2 x 1.
    returns 2 x 1 output G(x).
    """
    assert x.shape[0] == 2 and x.shape[1] == 1
    x1, x2 = x[0], x[1]
    G1 = np.sqrt(x2**2+x2)
    G2 = np.sqrt(-x1**2+x1)
    Gx = np.array([G1, G2])
    Gx = Gx.reshape(len(x), -1)
    assert Gx.shape[0] == 2 and Gx.shape[1] == 1
    return Gx

def fixed_point_iteration(G, x0=None, tol=1e-5):
    """ fixed point iteration. G is right hand side, with
    random starting point (by default) drawn from [0, 1]x[-1/2, 1/2],
    convergence criteria is inf norm. """
    if x0 is None:
        x0 = np.random.uniform([0, -1/2], [1, 1/2], size=(1, 2))
    x0 = np.array(x0)
    # reshape
    x0 = x0.reshape(-1, 1)
    n = x0.shape[0]
    # history
    all_x = []
    all_x.append(x0)
    x_prev = 0
    x_curr = x0
    while np.linalg.norm(x_curr - x_prev, ord=float('inf')) > tol:
        x_prev = x_curr
        x_curr = G(x_curr)
        all_x.append(x_curr)
    len_history = len(all_x)
    history = np.array(all_x).reshape(-1, len_history)
    last_x = history[:, -1]
    return last_x, history
```

[371]:

```
last, _ = fixed_point_iteration(rhs) # fixed point 1
print(last)
```

```
[0.7726402  0.41964714]
```

[ ]:

[ ]:

[ ]:

[ ]: