```
1 numbers=left  def fun(A, b, theta=.1, m=50): n = A.shape[1] x = np.zeros(n) for k in range(m): x = x - theta * (A
  @ x - b) return x
```

Algorithm 1: Unknown Python code

1. Please describe what each line of the code does (please do not write into the pseudocode).
2. Which algorithm is implemented, and what is its purpose? Which role does `theta` play here?

Solution:

1. Description:

   $(1P)$ 1 Function declaration with input: $A, b, \theta := 0,35, m = 50$

   $(1P)$ 2 Set $n :=$ number of rows of A

   $(1P)$ 3 Set $x := (0, \dots, 0)^T \in \mathbb{R}^n$

   $(1P)$ 4 /

   $(1P)$ 5 Start for-loop from $k = 0, \dots, m-1$

   $(1P)$ 6 Update x by $x - \theta(Ax - b)$

   $(1P)$ 7 Output x after m iterations

2. The Richardson Iteration is implemented. $(1P)$
   It is used to solve a linear system $Ax = b$ iteratively. $(1P)$
   θ is preconditioner to assure that $\rho(\underbrace{\theta(I - A)}_{\text{iteration matrix}}) < 1$. $(2P)$