**Solve Elliptic PDE Function Documentation**

**Description**

The **solveEllipticPDE** function numerically solves the two-dimensional elliptic partial differential equation (PDE), specifically Poisson's equation, using a finite difference method. It discretizes the spatial domain and computes the solution over the given domain.

**Input Arguments**

* **L**: Length of the square spatial domain (2D).
* **Nx**: Number of spatial grid points along the x-axis.
* **Ny**: Number of spatial grid points along the y-axis.
* **f**: Function handle representing the source term in Poisson's equation −∇2𝑢=𝑓(𝑥,𝑦)−∇2*u*=*f*(*x*,*y*).
* **u\_left**: Boundary condition at the left boundary.
* **u\_right**: Boundary condition at the right boundary.
* **u\_top**: Boundary condition at the top boundary.
* **u\_bottom**: Boundary condition at the bottom boundary.

**Output Argument**

* **u**: Matrix representing the computed solution of the PDE. Each element of the matrix corresponds to the solution value at a spatial grid point.

**Method Explanation**

1. **Initialization**: Initialize parameters such as spatial discretization steps, spatial grid points, and construct the grid in the spatial domain.
2. **Construct Coefficient Matrix**: Call the **constructEllipticMatrix** function to construct the coefficient matrix 𝐴*A* representing the discretized form of Poisson's equation.
3. **Compute Source Term**: Evaluate the source term f(𝑥,𝑦)*f*(*x*,*y*) at each spatial grid point.
4. **Solve Linear System**: Solve the linear system 𝐴𝑢=𝐹*Au*=*F*, where 𝐴*A* is the coefficient matrix, 𝑢*u* is the solution vector, and 𝐹*F* is the vector containing the evaluated source term.
5. **Reshape Solution**: Reshape the solution vector into a matrix representing the solution over the spatial domain.
6. **Apply Boundary Conditions**: Apply the boundary conditions to the solution matrix.
7. **Visualization**: Visualize the computed solution using a surface plot.