ASSIGNMENT 4.1

We want to solve the algebraic equation $x^2 + \varepsilon x - 1 = 0$ using regular perturbation, assuming ε is small.

Step 1: Assume a Perturbation Expansion

Assume the solution x can be written as a series expansion in ε:

$x = x_0 + \varepsilon x_1 + \varepsilon^2 x_2 + \ldots$

Step 2: Substitute the Series into the Equation

Substitute the series expansion into the original equation:

$(x_0 + \varepsilon x_1 + \varepsilon^2 x_2)^2 + \varepsilon (x_0 + \varepsilon x_1 + \varepsilon^2 x_2) - 1 = 0$

Step 3: Expand the Equation

Expand the equation by multiplying out the terms:

$x_0^2 + 2\varepsilon x_0 x_1 + \varepsilon^2 (2x_0 x_2 + x_1^2) + \varepsilon x_0 + \varepsilon^2 x_1 + \varepsilon^3 x_2 - 1 = 0$

Step 4: Group Terms by Powers of ε

Group the terms by powers of ε:

ε^0 terms: $x_0^2 - 1$

ε^1 terms: $2x_0 x_1 + x_0$

ε^2 terms: $2x_0 x_2 + x_1^2 + x_1$

Step 5: Solve for x_0, x_1, and x_2

Solve for x_0, x_1, and x_2 from the equations obtained for each power of ε.

Step 5.1: Solve for x_0 (Constant Term, ε^0)

The equation for ε^0 is:

$x_0^2 - 1 = 0$

Solve for x_0:

$x_0 = \pm 1$

Step 5.2: Solve for x_1 (First-Order Term, ε^1)

The equation for ε^1 is:

$2x_0x_1 + x_0 = 0$

Solve for x_1:

$x_1 = -x_0/2$

Substitute the values of x_0:

If $x_0 = 1$, then $x_1 = -1/2$.

If $x_0 = -1$, then $x_1 = 1/2$.

Step 5.3: Solve for x_2 (Second-Order Term, ε^2)

The equation for ε^2 is:

$2x_0x_2 + x_1^2 + x_1 = 0$

Substitute the values of x_0 and x_1:

$x_2 = -(x_1^2 + x_1) / (2x_0)$

Substitute the values for x_0 and x_1:

For $x_0 = 1$ and $x_1 = -1/2$:

$x_2 = 1/8$

For $x_0 = -1$ and $x_1 = 1/2$:

$x_2 = 3/8$

Step 6: Write the Final Solution

Substitute x_0, x_1, and x_2 back into the perturbation expansion:

For $x_0 = 1$:

$x \approx 1 - \varepsilon/2 + \varepsilon^2/8$

For $x_0 = -1$:

$x \approx -1 + \varepsilon/2 + 3\varepsilon^2/8$

HINTS WITH SYNTAX

Hint 1: Define Variables

Start by defining the necessary variables using `sympy`. You need variables for `x`, ` ε`, and the coefficients `x_0`, `x_1`, and `x_2`.

Use `sp.symbols` to define these symbols.

Hint 2: Set Up the Perturbation Series

- Write the perturbation series as `x_perturbation = x0 + ε * x1 + ε **2 * x2`.
- This series represents the approximate solution we're seeking.

Hint 3: Substitute into the Original Equation

- Substitute `x_perturbation` into the original equation $x^2 + \varepsilon x - 1 = 0$.
- Use `sp.expand` to expand the equation, which will allow you to separate it into different powers of `epsilon`.

Hint 4: Collect Terms by Powers of Epsilon

- Use `sp.collect` to group the terms in the expanded equation by powers of ε.
- This step will help you identify the equations for `x_0`, `x_1`, and `x_2` separately.

Hint 5: Solve for x_0 (Zeroth-Order Term)

- Solve the equation involving the zeroth-order term (constant term) using `sp.solve`.
- The solution will give you the values of `x_0`.

Hint 6: Solve for x_1 (First-Order Term)

- The first-order term will involve `x_0` and `x_1`.
- Substitute the value(s) of `x_0` you found in the previous step into the first-order equation, and solve for `x_1`.

Hint 7: Solve for x_2 (Second-Order Term)

- The second-order term will involve `x_0`, `x_1`, and `x_2`.

- Substitute the values of `x_0` and `x_1` into the second-order equation and solve for `x_2`.

Hint 8: Combine the Solutions

- Combine the values of `x_0`, `x_1`, and `x_2` into the perturbation series.

- This will give you the final expression for `x` in terms of ε.

Hint 9: Simplify and Display the Final Solution

- Simplify the final expressions using `sp.simplify`.

- Display the solutions for `x` to see the final perturbative approximation in terms of ε.

Hint 10: Construct and Simplify the Approximate Solutions

- What to Do: Combine the solutions x_0, x_1, and x_2 into the perturbation series. Use sp.simplify() to simplify the expression.
- Why: This yields the final approximate solutions for x in terms of ϵ.

Hint 11: Display the Results

- What to Do: Use print() to display the results for x_0, x_1, x_2, and the final approximate solutions.

Bonus Hint: Explore `sympy` Functions

- Explore other useful `sympy` functions like `sp.factor`, `sp.expand`, and `sp.solve` to manipulate and solve the equations.

SYNTAX	WORK
sp.expand()	To expand an expression, you can use the sp.expand() function.
sp.collect()	To collect terms by powers of a particular variable, use the sp.collect() function.
sp.coeff()	The sp.coeff() function in SymPy is used to extract the coefficient of a specified term in an expression.

Step 1: Outer Solution (Leading-Order Approximation)

Consider the original equation: $\varepsilon^2 * x^2 + x - 1 = 0$

For the outer solution, we set epsilon = 0 (leading-order approximation):

$0 * x^2 + x - 1 = 0$

This simplifies to: $x - 1 = 0$

Therefore, the outer solution is: $x_{outer} = 1$

This solution is valid away from regions where the small $\varepsilon^2 * x^2$ term dominates.

Step 2: Inner Solution (Boundary Layer Analysis)

In regions where the small $\varepsilon^2 * x^2$ term is important,

we introduce a boundary layer variable:

Let $x = \varepsilon * xi$, where xi is the new variable.

Substitute into the original equation: $\varepsilon^2 * (\varepsilon * xi)^2 + \varepsilon * xi - 1 = 0$

Simplifying, we get: $\varepsilon^4 * xi^2 + \varepsilon * xi - 1 = 0$

Dividing the equation by ε, we obtain: $\varepsilon^3 * xi^2 + xi - 1/\varepsilon = 0$

As $- \varepsilon > 0$, the dominant balance is given by: $xi - 1/\varepsilon = 0$

Therefore, the inner solution is: $xi_{inner} = 1/\varepsilon$

Converting back to x, we get: $x_{inner} = 1$

Step 3: Matching the Solutions

We check if the inner and outer solutions match in an overlapping region:

As $xi \to \infty$ (which means $x \to 0$), the inner solution approaches: $x_{inner} = 1$

The outer solution was also $x_{outer} = 1$, so the solutions match.

No further modification is necessary as the solutions are consistent across regions.

Step 4: Uniform Approximation (If Needed)

In some singular perturbation problems, we might combine the inner and outer solutions.

For this specific example, both inner and outer solutions are $x = 1$, so the

uniform solution is simply: $x_{uniform} = 1$

This solution is valid across the entire domain.

SYNTAX AND PSEUDO CODE

Step 1: Define Variables

Define the small parameter epsilon and the variable x.

Example: epsilon, x = symbols('epsilon x')

Step 2: Define the Original Equation

Write down the equation to be analyzed.

Example: equation = epsilon^2 * x^2 + x - 1

Step 3: Outer Solution (Leading-Order Approximation)

Set epsilon to 0 in the equation and solve for x.

Example: outer_equation = equation.subs(epsilon, 0)

outer_solution = solve(outer_equation, x)

Step 4: Inner Solution (Boundary Layer Analysis)

Introduce a new variable xi for the boundary layer.

Example: xi = symbols('xi')

x_inner = epsilon * xi # Boundary layer scaling

Substitute x = epsilon * xi into the original equation.

Example: inner_equation = equation.subs(x, x_inner)

Simplify the inner equation and divide by epsilon to obtain leading order.

Example: inner_equation_simplified = simplify(inner_equation / epsilon)

Solve the inner equation for xi.

Example: inner_solution = solve(inner_equation_simplified, xi)

Step 5: Matching the Inner and Outer Solutions

Analyze the inner solution in the limit as xi -> infinity and compare with the outer solution.

Example: matching_inner_to_outer = [sol.subs(xi, 1/x).simplify() for sol in inner_solution]

Check if the solutions match or if further adjustment is needed.

Step 6: (Optional) Uniform Approximation

Combine the inner and outer solutions if necessary to create a uniform solution.

Example: uniform_solution = x_outer + inner_correction_term

Step 7: Output the Solutions

Print or return the outer solution, inner solution, and uniform approximation if applicable.

Example: print("Outer Solution:", outer_solution)

print("Inner Solution:", inner_solution)

print("Uniform Solution:", uniform_solution if defined)