Shift-Invert 幂法的拉普拉斯算子求解方法：使用 Neural Operator

目标问题

我们希望通过 **shift-invert 幂法** 来求解 **拉普拉斯算子的中间特征值**。具体来说，我们考虑以下的拉普拉斯特征值问题：

$$-\Delta u(x) = \lambda u(x), \quad (x \in \Omega),$$

其中，Δ 是 **拉普拉斯算子**，Ω 是定义域，$u(x)$ 是特征函数，λ 是特征值。

shift-invert 方法

1. **定义移位算子**：
   首先，我们考虑 **移位算子** $T_\sigma = (A - \sigma I)^{-1}$，其中：
   - $A = -\Delta$ 是拉普拉斯算子，
   - σ 是移位参数，通常选为与目标特征值接近的值。

   移位算子的特征值为：

   $$\mu_n = \frac{1}{\lambda_n - \sigma},$$

   其中 λ_n 是原算子的特征值。通过 **shift-invert 方法**，我们可以逐步收敛到**离** σ 最近的特征值及其对应的特征函数。

幂法的应用

我们利用 **幂法** 来计算 **最大特征值** 对应的特征函数。其基本步骤如下：

1. **初始状态**：
   给定一个初始猜测 u_0，通常是随机选取的初始向量。
2. **迭代更新**：
   每次迭代时，使用幂法进行更新：

   $$u_{t+1} = \frac{T_\sigma u_t}{\|T_\sigma u_t\|_{L^2}},$$

   这个操作会将 u_t 向离 σ 最近的特征函数收敛。随着迭代次数增加，u_t 会逐步接近目标特征函数。
3. **Rayleigh 商估计特征值**：
   在每次迭代后，我们可以计算 **Rayleigh 商**：

   $$\hat{\lambda} = \frac{\langle u_t, Au_t \rangle}{\langle u_t, u_t \rangle}.$$

   Rayleigh 商的值会逐步收敛到目标特征值。

Neural Operator 为什么适用

在传统的 **shift-invert 幂法** 中，我们每次迭代都需要计算 $(A-\sigma I)^{-1}$ 来得到 w，这是一种迭代运算，非常耗时，特别是对于复杂的几何形状和大规模问题。

为了提高效率，我们可以通过 **Neural Operator**（神经算子）来 **直接从** $u(x)$ 预测对应的解 $w(x)$，避免了每次反复计算算子运算的步骤。

使用 Neural Operator 的理由

1. **避免每次迭代计算** w：
   在传统方法中，每次求解 w 都需要应用 $(A-\sigma I)^{-1}$ 迭代计算。而 **Neural Operator** 通过训练神经网络，使其能够直接根据 $u(x)$ 推理出 $w(x)$，避免了反复计算算子的过程。
2. **高效推理**：
   一旦网络完成训练，我们可以 **通过一次前向传播** 来得到 $w(x)$，这大大提高了计算效率。相比传统方法的多次迭代，神经网络的推理速度非常快，适合实时应用或大规模计算。
3. **适应复杂边界条件**：
   神经网络能够通过训练自动适应不同的边界条件和几何形状，无需显式修改计算过程。这对于实际应用中，尤其是具有复杂边界的几何问题，非常有用。
4. **跨域能力**：
   神经网络经过训练后，能够适应不同的域和条件，一次训练后可以在多个不同的测试域中进行高效推理，而无需重新求解每个域的解。

Neural Operator 训练方案

1. **目标方程**：
   我们希望通过 **Neural Operator** 来训练一个网络，直接从 $u(x)$ 预测 $w(x)$，即求解方程：

$$(A-\sigma I)w(x)=u(x).$$

   通过神经网络，我们可以直接从 $u(x)$ 计算出 $w(x)$，避免了传统方法中每次都应用算子 $(A-\sigma I)^{-1}$ 的过程。
2. **神经网络架构**：
   - **输入**：$u(x)$，即当前的特征函数。
   - **输出**：$w(x)$，即通过神经网络推理得到的解。
   - 我们可以使用多层感知器（MLP）来表示 $w(x)$，并通过最小化 **PDE 残差** 来训练网络。
3. **损失函数**：
   - 我们通过最小化 **PDE 残差** 来训练网络：

$$\mathcal{L}_{\text{int}}=\mathbb{E}_{x\in\Omega}\left[\left(-\Delta w(x)-\sigma w(x)-u(x)\right)^2\right].$$

   - **边界条件损失**：对于 Dirichlet 边界条件 $w(x)=0$ 在边界上，我们添加边界损失：

$$\mathcal{L}_{\text{bc}}=\mathbb{E}_{x_b\in\partial\Omega}[w(x_b)^2].$$

   - 最终损失函数：

$$\mathcal{L}=\mathcal{L}_{\text{int}}+\omega_{\text{bc}}\mathcal{L}_{\text{bc}},$$

     其中 ω_{bc} 是边界损失的权重。
4. **外层迭代**：
   每次外层迭代时，使用训练得到的 $w(x)$ 进行归一化得到新的 $u(x)$，即：

$$u_{t+1}=\frac{w}{\|w\|_{L^2}},$$

   然后计算新的 **Rayleigh 商** 和 **残差**，来监控迭代过程的收敛情况。

停止条件

1. Rayleigh 商收敛:

在每次外层迭代中，计算 **Rayleigh 商**:

$$\hat{\lambda} = \frac{\int_\Omega \|\nabla u\|^2}{\int_\Omega u^2}.$$

当 Rayleigh 商的变化量小于预设阈值时，停止迭代:

$$\left|\hat{\lambda}(u_{t+1}) - \hat{\lambda}(u_t)\right| < \epsilon_1.$$

2. 特征函数稳定性:

当特征函数 u_t 的变化非常小，即:

$$\|u_{t+1} - u_t\|_{L^2} < \epsilon_2,$$

停止迭代。

3. 残差收敛:

计算残差:

$$\mathcal{R}_t = \left\|(A - \sigma I)u_t - \hat{\lambda}(u_t)u_t\right\|_{L^2}.$$

当残差小于预设阈值时，停止迭代:

$$\mathcal{R}_t < \epsilon_3.$$

4. 最大迭代次数:

如果达到最大迭代次数 $T_{\max}$，则强制停止。
