What it takes to solve PDEs?

方法框架

1.求解一个特定的方程：

I.FDM

II.FEM

III.FVM/FVEM

IV.Mesh-free（PINN, RFM，...）

2.求解一类特定的方程/一个特定的算子

I.Deep Operator Network (DeepONet)

II.Fourier Neural Operator (FNO)

III.Physics-Informed Neural Operators

3.通用方程求解器

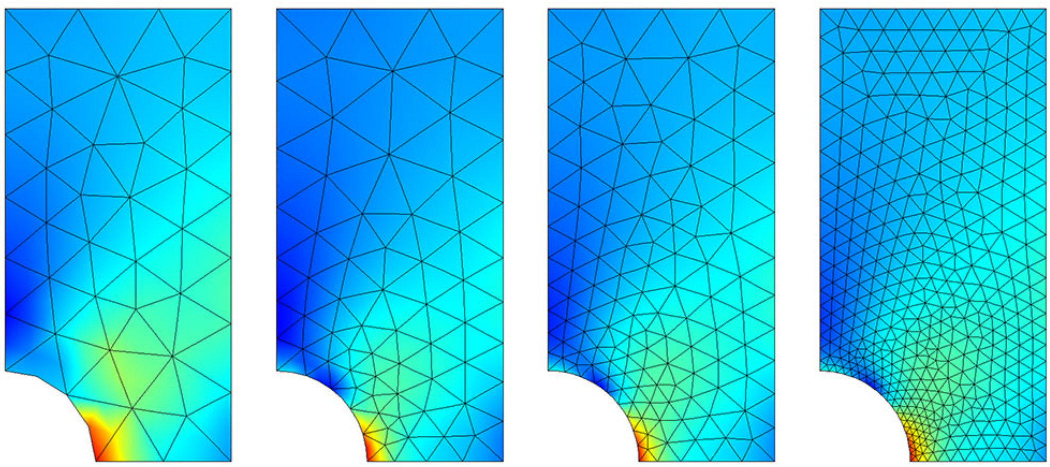
有限元方法

FDM、FEM、FVM的求解过程是非常类似的，因此用有限元方法（FEM）来代替。

1. 有限元方法求解PDE时的输入

有限元方法的输入涉及到几何信息、物理参数、边界条件、初始条件：

- 几何方面：
    - **几何模型**：定义求解域的几何形状。对于二维和三维问题，域的离散化需要通过网格划分来近似描述连续域。
    - **网格信息**：网格的节点坐标、单元的拓扑结构、单元类型（如三角形、四边形、六面体等）。
      几何模型说白了就是求解域，这里的网格和DPOT的gridset.py文件里的网格不一样，DPOT里明显是“差分”，即离散的等间距的取点，而不是画网格，而有限元方法要求把一个给定形状的几何求解域划分成具体的一块一块的网格（比如画成井字棋一样的网格），下面是一个网格的例子



https://cn.comsol.com/multiphysics/mesh-refinement?parent=physics-pdes-numerical-042-32

- 方程方面：
  - **方程形式**：所求解的PDE形式，即具体的数学公式（如 $\Delta u = -f, \rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} + \mathbf{f}$）
  - **边界条件**：定义在边界上的条件，如**迪里克雷边界条件**（给定边界上的函数值）、**诺依曼边界条件**（给定边界上的导数值或通量）、**罗宾边界条件。**
  - **初始条件**：如果是随时间演化的PDE，如热传导或流体力学问题，则需要定义初始时刻的状态，如初始温度场、初始速度场等。
  - **材料参数**：包括物理属性，如材料的弹性模量、泊松比、导热系数、密度等。不同的PDE问题（如热传导、流体力学、弹性力学等）需要不同的物理参数。

    具体的一个例子：

    方程形式：

$$\begin{cases} \partial_t u(x,t) - \alpha^2 \partial_x^2 u(x,t) = 0, & x \in [x_0, x_1], t \in [0,T], \\ u\left(x_0, t\right) = g_1(t), & t \in [0,T], \\ u\left(x_1, t\right) = g_2(t), & t \in [0,T], \\ u(x,0) = h(x), & x \in [x_0, x_1], \end{cases}$$

    边界条件：

$$\begin{aligned} u\left(x_0, t\right) = g_1(t), \quad t \in [0,T], \\ u\left(x_1, t\right) = g_2(t), \quad t \in [0,T], \end{aligned}$$

    初值条件：$u(x,0) = h(x)$

    材料参数：α

2. 有限元方法求解PDE时的输出

- **解的离散值**：在网格节点(x,y)计算出的场量值。例如，位移场、温度场、压力场等。

有限元求解偏微分方程（PDE）的数学步骤

问题描述：二维泊松方程

我们考虑在二维区域 Ω上求解泊松方程：

$$-\nabla\cdot(\kappa(x,y)\nabla u(x,y)) = f(x,y), \quad (x,y)\in\Omega$$

其中，$\kappa(x,y)$ 是空间上的系数函数（例如导热系数），$f(x,y)$是外部源项或右手边函数，$u(x,y)$是待求解的未知函数。边界条件如下：

$$u(x,y) = u_D(x,y), \quad \partial\Omega_D$$

$$\kappa(x,y)\frac{\partial u}{\partial n} = g(x,y), \quad \partial\Omega_N$$

输入

1. **几何区域**Ω的定义
   - 定义求解区域的几何形状，可以是矩形、圆形等简单几何，也可以通过网格划分描述复杂形状。
2. **材料参数** $\kappa(x, y)$
   - 可以是常数，也可以是随位置变化的函数。
3. **源项** $f((x, y))$
   - 右手边的源项函数，描述外部的驱动力或热源等。
4. **边界条件**
   - **迪里克雷边界条件** $u_D(x, y)$，定义在 $\partial\Omega_D$ 上的已知函数。
   - **诺依曼边界条件** $g(x, y)$，定义在 $\partial\Omega_N$上的通量。

1. 弱形式

首先，将PDE的**强形式**转化为**弱形式**：

- 选择一个测试函数 $v \in H_0^1(\Omega)$（通常是试探空间中的函数），并将方程乘以测试函数 v，然后对区域 Ω 进行积分：

  $\int_\Omega -\nabla \cdot (\kappa(x, y)\nabla u(x, y))v(x, y)\, d\Omega = \int_\Omega f(x, y)v(x, y)\, d\Omega$
- 使用分部积分，将二阶导数转化为一阶导数：

$$u_h(x, y) = \sum_{j=1}^{N} u_j \phi_j(x, y)$$

其中，$\phi_j(x, y)$ 是第 j 个节点对应的基函数，u_j 是该节点的解值。

测试函数的选择：同样用有限维的基函数来近似测试函数 $v(x, y)$：

$$v_h(x, y) = \phi_i(x, y)$$

其中 ϕ_i 是与第 i 个节点对应的基函数。

3. 组装有限元方程

将离散化的解和测试函数代入弱形式，得到离散化后的代数方程组。对于每个测试函数 (\phi_i)，方程为：

$$\sum_{j=1}^{N} \int_\Omega \kappa(x, y)\nabla\phi_j(x, y) \cdot \nabla\phi_i(x, y)\, d\Omega\, u_j = \int_\Omega f(x, y)\phi_i(x, y)\, d\Omega + \int_{\partial\Omega_N} g(x, y)\phi_i(x, y)\, d\Gamma$$

写成矩阵形式：

$$\mathbf{Ku} = \mathbf{F}$$

其中，

- $\mathbf{K}$ 是**刚度矩阵**，其元素为 $\mathbf{K}_{ij} = \int_\Omega \kappa(x, y) \nabla \phi_j(x, y) \cdot \nabla \phi_i(x, y) \, d\Omega$。
- $\mathbf{u}$ 是待求解的**未知数向量**，包含每个网格节点上的解 u_j。
- $\mathbf{F}$ 是**右端项向量**，其元素为 $\mathbf{F}_i = \int_\Omega f(x, y) \phi_i(x, y) \, d\Omega + \int_{\partial\Omega_N} g(x, y) \phi_i(x, y) \, d\Gamma$。

4. 应用边界条件

- **迪里克雷边界条件**：在迪里克雷边界 $\partial\Omega_D$ 上的节点，直接将已知的 $u_D(x, y)$ 代入，并调整刚度矩阵和右端项向量以反映这些固定值。

5. 数值求解

通过求解线性方程组 $\mathbf{K}\mathbf{u} = \mathbf{F}$，可以得到节点上的解 u_j，即离散化解 $u_h(x, y)$。

6. 后处理

在解出 $u_h(x, y)$ 之后，可能需要计算导数（如应力、温度梯度等），并通过插值方法得到区域内部的解值。解的可视化通常通过等值线图、变形图等方式呈现。

输出

1. **节点上的解** u_j：求解后的离散解 $u_h(x, y)$ 在每个节点上的值，通常用于近似整个区域内的场解。
2. **梯度和导数场**：通过对 $u_h(x, y)$ 的导数计算出梯度场（如应力、应变、温度梯度等）。
3. **误差估计**：误差分析，如通过与解析解的比较或者后验误差估计方法。
4. **可视化结果**：解的可视化，例如温度场、位移场、应力分布等。

工业软件（如ANSYS）求解PDE时的输入

- **几何建模**：通过GUI或者导入CAD模型定义几何域，软件自动生成网格。
- **材料参数**：包括物理属性，如材料的弹性模量、泊松比、导热系数、密度等。不同的PDE问题（如热传导、流体力学、弹性力学等）需要不同的物理参数。
- **方程和求解器设置**：选择合适的物理场模块，如结构力学模块、热传导模块、流体模块等，同时设置PDE的求解方法（静态分析、动态分析、线性或非线性分析）。
- **边界条件和初始条件**：通过界面设置或导入实验数据，定义迪里克雷、诺依曼边界条件，设置初始条件。
- **网格划分**：网格的节点坐标、单元的拓扑结构、单元类型（如三角形、四边形、六面体等）。

4. 工业软件的输出

- **场量解的可视化**：例如应力场、位移场、温度场、速度场。
- **数值结果导出**：可导出关键节点或单元的数值结果，如CSV、TXT格式的数值表。
- **误差分析与后处理**：ANSYS提供误差估计、灵敏度分析、优化分析等功能。

Mesh-free方法

Mesh-free方法之PINN方法

pinn的输入为$(\mathbf{x},t)=(x,y,z,t)$:

输出为$u(\mathbf{x},t)=u(x,y,z,t)$。

Random Feature Method

输入：基函数数量M，配点数量（样本点数量Q），方程形式（用在Loss function的构造里）

Algorithm 1 The random feature method.

Input: Number of basis functions M; number of collocation points Q; rule for generating collocation points; + 补充：加上方程形式（在2. 10公式里）
Output: The approximate solution u_M;
1: Construct M random feature functions $\{\phi_m\}$ and the PoU $\{\psi_n\}$;
2: Sample points $C = C_I \cup C_B$ according to some predetermined rule;
3: Evaluate equations at C_I and boundary conditions at C_B;
4: Construct the loss function (2.10) (M is not necessarily equal to N);
5: Solve the optimization problem;
6: Return u_M;

输出

数值解$u_M(\mathbf{x},t)=u_M(x,y,z,t)$

求解步骤

- 使用随机特征函数和单位分解法（PoU）求解PDE的算法

1. 定义向量解$u_M(x)$:

$$u_M(\boldsymbol{x})=\sum_{i=1}^{M}u_m\phi_m(\boldsymbol{x})=\left(\sum_{m=1}^{M}u_m^1\phi_m^1(x),\cdots,\sum_{m=1}^{M}u_m^{K_I}\phi_m^{K_I}(x)\right)^T$$

2. 进行区域划分（Partition of unity）得到$\{x_n\}_{n=1}^{M_p}\subset\Omega$，对$x_n$归一化:

$$\tilde{x}=\frac{1}{r_n}(x-x_n),\quad n=1,\cdots,M_p$$

3. 构造局部随机特征函数

$$\phi_{nj}(x)=\sigma(k_{nj}\cdot\tilde{x}+b_{nj}),\quad j=1,\cdots,J_n$$

4. 局部解通过PoU函数 $\psi_n(x)$ 组合，最终近似解为:

$$u_M(x)=\sum_{n=1}^{M_p}\psi_n(x)\sum_{j=1}^{J_n}u_{nj}\phi_{nj}(x)$$

5. 为了捕捉解的大尺度特征，在PoU局部基函数的基础上加入全局随机特征函数:

$$u_M(x)=u_g(x)+\sum_{n=1}^{M_p}\psi_n(x)\sum_{j=1}^{J_n}u_{nj}\phi_{nj}(x)$$

6. 损失函数求解：

$$\text{Loss} = \sum_{x_i \in C_I} \sum_{k=1}^{K_I} \lambda^k \| L_k u_M(x_i) - f^k(x_i) \|_{l_2}^2 + \sum_{x_j \in C_B} \sum_{\ell=1}^{K_B} \lambda^\ell \| B_\ell u_M(x_j) - g^\ell(x_j) \|_{l_2}^2$$

其中u_M的表达式为：

$$u_M(x) = \left(\sum_{n=1}^{M_p} \psi_n(x) \sum_{j=1}^{J_n} u_{nj}^1 \phi_{nj}^1(x), \cdots, \sum_{n=1}^{M_p} \psi_n(x) \sum_{j=1}^{J_n} u_{nj}^{K_I} \phi_{nj}^{K_I}(x) \right)^T$$

使用最小二乘法求解上述问题，得到了u_M解的数值。

微分方程大模型方法

设数据的tensor维度为$(x, y, t, Channel)$四个维度，$Channel$代表一个数据集对应的方程中有多少物理变量。

输入：

$u[:, :, 0:T, :]$

即前T帧的时间序列

输出：

$u[:, :, T+1, :]$

即第$T+1$帧

一个具体的例子：（Unisolver在做downstream task时怎么搞的）

•**compressible Navier-Stokes equation:** predict the future 11 timesteps of vorticity, pressure and density given the initial 10 timesteps of observations.

•**The shallow-water equations****: The task is to predict the future 91 timesteps of water depth given the first 10 timesteps of observations.

•**The 2D Diffusion-Reaction Equation:** The task is to predict the future 91 timesteps of u and v given the initial 10 timesteps of observations.

•**PDEArena-NS1/2:** Given the initial 10 timesteps of observations, the task on the fixed-force dataset is to predict the future 4 timesteps of velocity, while on the varying-force dataset, the number of timesteps to predict is 46.

•**CFDBench**: The task is to predict the future 10 timesteps of velocity given the initial 10 timesteps of observations.