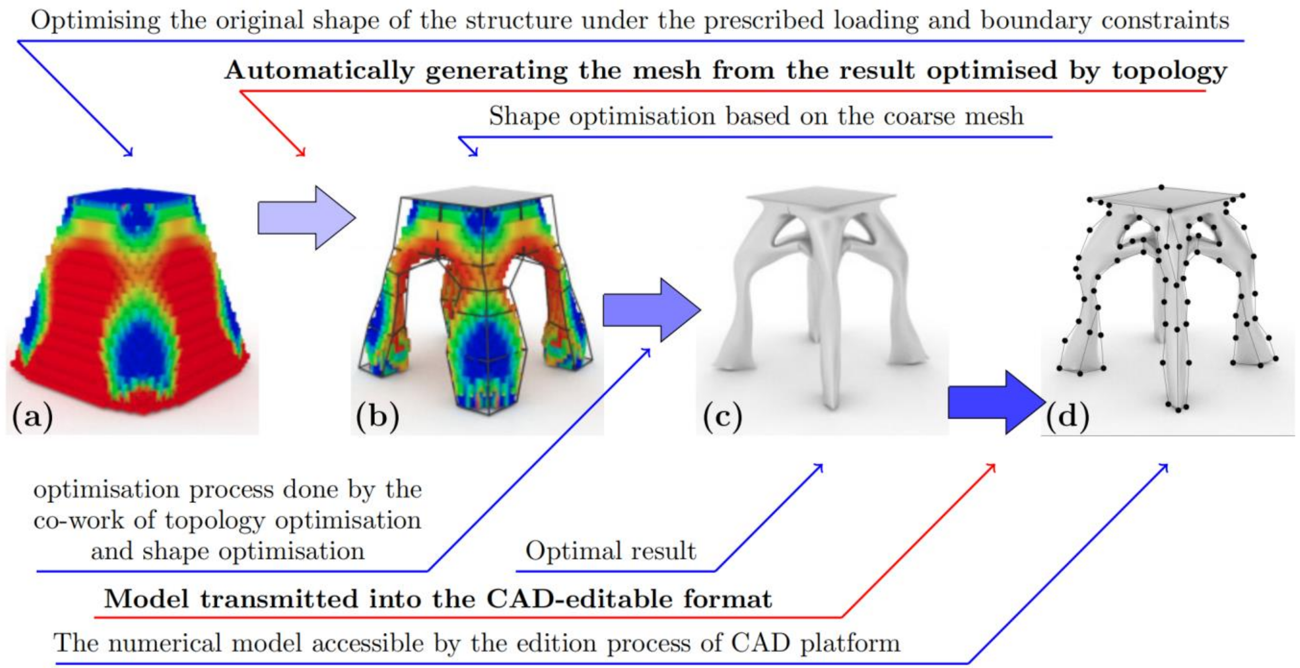
数值优化

- 背景
- 梯度优化理论
- 案例分析

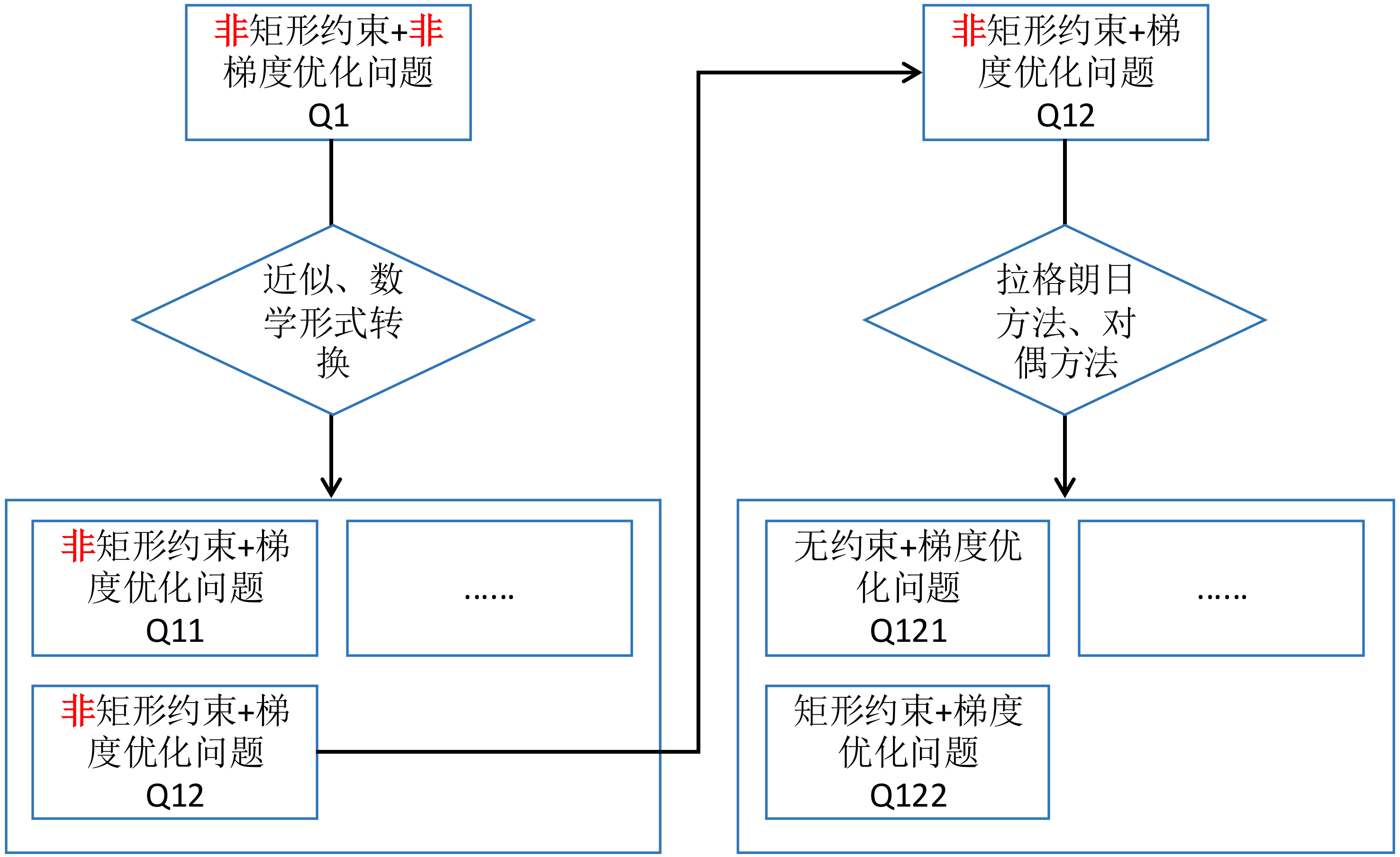
数值优化 - 背景

- Everything can be optimised
  - 几何形态能被优化
    - 曲线曲面拟合
    - 曲线曲面光顺
  - 几何数据能被优化
    - 距离求值
    - 投影
    - 求交
  - 设计方案能被优化
    - 拓扑优化
    - generative design（创成式设计）



数值优化 - 背景

- 优化分类
  - 按方法：
    - 梯度优化，极其常用，但是在全局求解上有局限性
    - 非梯度优化（花招很多，收敛性有问题），感兴趣可以看“贝叶斯优化”
  - 按约束：
    - 无约束及矩形区域约束（数值上稳定），“简单约束”
    - 非矩形约束（方法复杂）
- 最终
  - 都会回到求解简单约束下的梯度问题
  - 那全局性怎么保证？工程方法：初值采样，择优录取



数值优化 - 梯度优化理论

- 范式
  - $\boldsymbol{x}$ = optimisation variable，优化变量
  - $f(.)$ = objective function，目标函数，限定为R^{nx1}到R^1的映射
  - n = optimisation problem dimension，优化问题维度
  - c = equality constraints，等式约束，均可以转化为不等式约束
  - g = inequality constraints，不等式约束
- 优化结果
  - $\boldsymbol{x}^*$=argmin $f(\boldsymbol{x})$ or argmax $f(\boldsymbol{x})$
  - $f^* = f(\boldsymbol{x}^*)$

$$
\begin{aligned}
&\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ or } \max_{\boldsymbol{x}} f(\boldsymbol{x}) \\
s.t.\quad & \boldsymbol{x} \in \mathbb{R}^n, \\
& f \in \mathbb{R}, \\
& c_i(\boldsymbol{x}) = 0, \quad i \in [1, \ldots, n_c], \\
& g_j(\boldsymbol{x}) < 0, \quad j \in [1, \ldots, n_g].
\end{aligned}
$$

数值优化 - 梯度优化理论

- 重要问题
  - （1）“我们要优化什么？”数学建模，将问题进行公式化，转化成为梯度优化标准范式，定义**优化变量**与**目标函数**与**约束**
  - （2）“我们沿什么方向寻找结果？”优化变量的**更新规则**，确定规则是可以保证问题收敛，并合理进行**初始值采样**（initial guess）
  - （3）“我们要找到什么时候？”梯度优化全部依赖于迭代方法，根据精度要求定义好**收敛条件**
- 说明
  - （1）是问题特化，对症下药
  - （2）有规律可循
  - （3）有常规设定

数值优化 - 梯度优化理论

- “我们要优化什么？”
  - 拉格朗日算子：将约束优化问题转化为无约束问题
  - 构造拉格朗日函数

$$L(\boldsymbol{x},\lambda)=f(\boldsymbol{x})+\lambda g(\boldsymbol{x})$$

  - 原问题可以通过解下列方程组

$$\begin{cases}\nabla_{\boldsymbol{x}}L(\boldsymbol{x},\lambda)=0\\ g(\boldsymbol{x})=0\end{cases}$$

  - $\nabla_{\boldsymbol{x}}L=0$ 意味着L对于所有$\boldsymbol{x}$的分量都是偏导为0，在目标函数$f(.)$为凸函数的时候就是最优解。上述方程组还可以理解为最优解要么在g内的无约束问题的解，要不然就一定在边界上。可以阅读，解释得很通俗：https://zhuanlan.zhihu.com/p/440297403
  - 那么问题来了？万一函数不是凸函数？

$$\begin{aligned}&\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ or } \max_{\boldsymbol{x}} f(\boldsymbol{x})\\ s.t.\quad &\boldsymbol{x}\in\mathbb{R}^n,\\ &f\in\mathbb{R},\\ &c_i(\boldsymbol{x})=0,\quad i\in[1,\ldots,n_c],\\ &g_j(\boldsymbol{x})<0,\quad j\in[1,\ldots,n_g].\end{aligned}$$

数值优化 - 梯度优化理论

- 插一个队“我们要找到什么时候？”
  - 首先明确，梯度优化方法都是迭代方法，指导迭代的最重要信息，就是我们怎么定义收敛条件
  - 抛砖引玉：假设迭代步k和k+1间出现了以下情况，则可以理解为收敛

$$\left|\frac{f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)}{f(\boldsymbol{x}_k)}\right| \leq \epsilon$$

  - 这个微量ϵ，“epsilon”，是一个微量，也是迭代精度
  - 更严格收敛条件，可以要求连续多次迭代步下降且相对误差满足一个微量，这里不做赘述，但要明确两点
    - 收敛条件均是迭代步之间相差小于某种精度
    - 严格的收敛条件必然会在迭代精度提升的同时导致迭代时间变长，“鱼和熊掌”

数值优化 - 梯度优化理论

- 再回到“我们沿什么方向寻找结果？”
  - 我们从收敛条件倒推迭代原则，假设我们有两个相邻的迭代步k与k+1，以及它们的函数值f_k与f_{k+1}，我们对f_{k+1}进行泰勒展开

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &= f(\boldsymbol{x}_k) + \nabla f^\top(\boldsymbol{x}_k) \cdot (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \\ &\quad + \frac{1}{2}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)^\top \cdot \nabla^2 f(\boldsymbol{x}_k) \cdot (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \end{aligned}$$

  - 然后定义$D = f_{k+1} - f_k$，所以D应该收敛到0，即最小的非负数

$$D = \nabla f^\top(\boldsymbol{x}_k) \cdot (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{1}{2}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)^\top \cdot \nabla^2 f(\boldsymbol{x}_k) \cdot (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k)$$

数值优化 - 梯度优化理论

- 在D的定义下，只要能够满足D一直下降到0就行，所以我们来设计一个迭代方法
- $-\nabla f(\boldsymbol{x}_k)$是$f$在$\boldsymbol{x}_k$处下降最快的方向
- 这里α_k是一个常数，所以迭代是沿着导数反方向进行的，那么这个α_k应该怎么设计才能使D下降呢？带入α_k：

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k),$$
$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = -\alpha_k \nabla f(\boldsymbol{x}_k).$$

$$D = -\alpha_k \|\nabla f(\boldsymbol{x}_k)\|^2 + \frac{1}{2}\alpha_k^2 \nabla f^\top(\boldsymbol{x}_k) \nabla^2 f(\boldsymbol{x}_k) \nabla f(\boldsymbol{x}_k)$$

- 然后D对α_k求导为0：$\alpha_k^* = \dfrac{\|\nabla f(\boldsymbol{x}_k)\|^2}{\nabla f^\top(\boldsymbol{x}_k) \nabla^2 f(\boldsymbol{x}_k) \nabla f(\boldsymbol{x}_k)}$

- 经典牛顿法的迭代规则为：$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \nabla^2 f(\boldsymbol{x}_k)^{-1} \nabla f(\boldsymbol{x}_k)$
- 下降方向为：$\boldsymbol{d}_k = -\nabla^2 f(\boldsymbol{x}_k)^{-1} \nabla f(\boldsymbol{x}_k)$
- 步长为：$\alpha_k = 1$

数值优化 - 梯度优化理论

- 所以要定义一个迭代方法只需要关注两点
  - 迭代方向 $\boldsymbol{d}_k$
  - 迭代步长 α_k

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$$

- 迭代方向的选择
  - 梯度下降法：直接用目标函数梯度作为$\boldsymbol{d}_k$
  - Quasi-Newton法：采用一阶导数/二阶导数
  - BFGS法：采用一阶导数/二阶导数近似
- 迭代步长选择
  - Line search：线性搜索

数值优化 - 案例分析

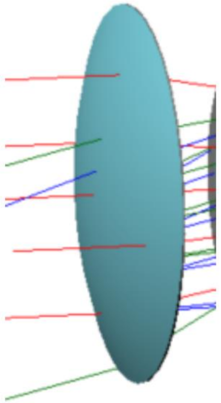
- 案例背景
  - 场景：光学软件厂商，光线与光学透镜求交算法
  - 需求：效率达每秒十万级别光线求交完成，精度达10E-9
- 透镜方程均具有解析隐式表达

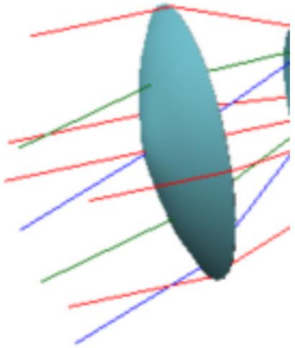
$$S(x,y,z)=0$$

- 比如柱面

$$z=\frac{cR^2}{1+\sqrt{1-c^2R^2}}$$

$$S(x,y,z)=z-\frac{c(x^2+y^2)}{1+\sqrt{1-c^2(x^2+y^2)}}=0$$





数值优化 - 案例分析

- 解题思路：
  - 效率与要求都要，就要充分利用透镜面解析表达的特性
  - 光线可以用解析表达，所以考虑直接利用解析形式进行求解
- 射线方程

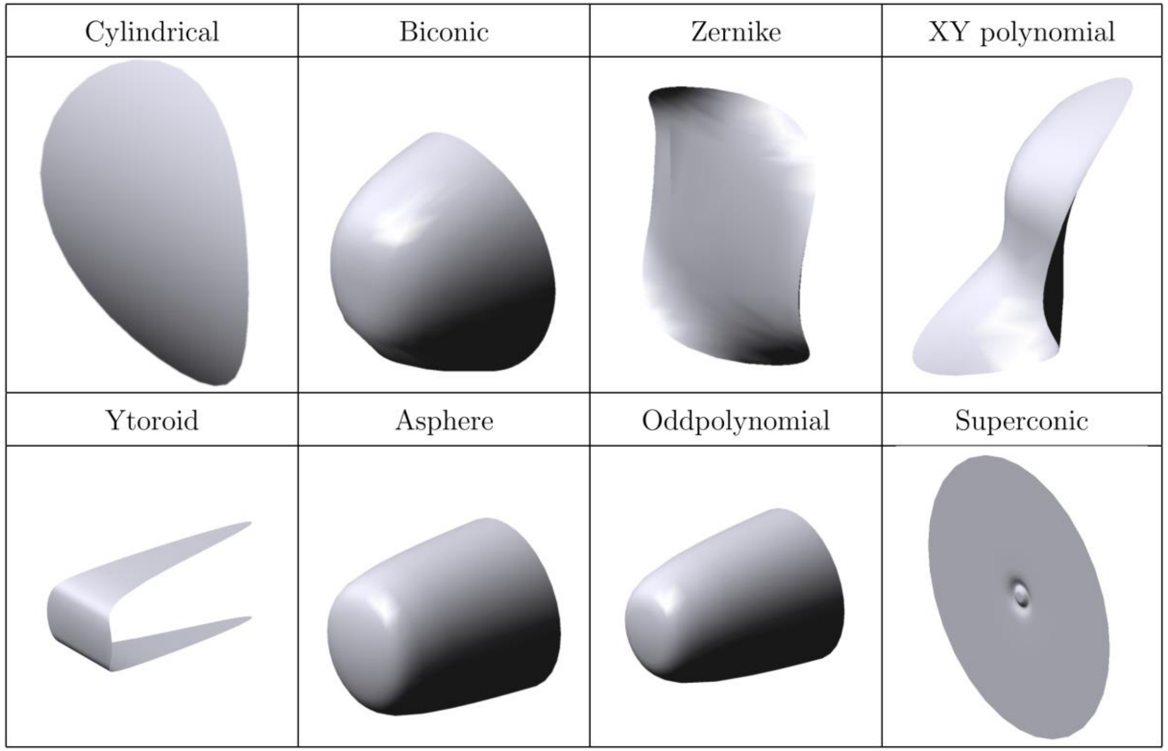
$$\boldsymbol{p}(t) = \boldsymbol{p}_0 + t \cdot \boldsymbol{p}_1$$

- 交点必同时存在于S和$\boldsymbol{p}$上，所以把射线方程代入曲面方程

$$S(t) = S(x(t), y(t), z(t)) = 0$$

- 上述方程求解可以转换为优化问题，完成数学建模

$$\min_t F(t), \quad F(t) = \frac{1}{2}(S(x(t), y(t), z(t)))^2$$



数值优化 - 案例分析

- 更新策略：会用到$F(t)$的一阶导数和二阶导数，链式法则推导

$$\frac{\partial S}{\partial t} = \frac{\partial S}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial S}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial S}{\partial z}\frac{\partial z}{\partial t}$$

$$\frac{\partial F}{\partial t} = S \cdot \frac{\partial S}{\partial t}$$

$$\frac{\partial^2 S}{\partial t^2} = \frac{\partial^2 S}{\partial x^2}\frac{\partial x}{\partial t}\frac{\partial x}{\partial t} + \frac{\partial^2 S}{\partial y^2}\frac{\partial y}{\partial t}\frac{\partial y}{\partial t} + \frac{\partial^2 S}{\partial z^2}\frac{\partial z}{\partial t}\frac{\partial z}{\partial t} + 2(\frac{\partial^2 S}{\partial xy}\frac{\partial x}{\partial t}\frac{\partial y}{\partial t} + \frac{\partial^2 S}{\partial xz}\frac{\partial x}{\partial t}\frac{\partial z}{\partial t} + \frac{\partial^2 S}{\partial yz}\frac{\partial y}{\partial t}\frac{\partial z}{\partial t})$$

$$\frac{\partial^2 F}{\partial t^2} = (\frac{\partial S}{\partial t})^2 + S \cdot \frac{\partial^2 S}{\partial t^2}$$

- 收敛条件：常规即可
- 初始值：离散网格快速粗筛，GPU加速