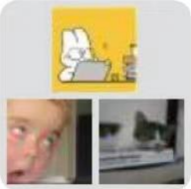
课程基本信息

- 考核：作业（40%-46%）+期末（60%）
- 作业完成时间：1周；晚交一周减当次作业20%分数；晚交超过两周，减当次作业60%分数；期末考试之后不可补交。会有少数附加作业题（每题3分，只做两道即可），小幅提高平时成绩。
- 参考答案公布：2周。
- 期末：闭卷，可携带一张A4“作弊”纸。
- 请加课程微信群。课件以及课程的一些相关信息会在微信群发布。
- 答疑时间待定。
- 课程内容“重拿轻放”原则，与教材（实际是参考书）不完全重合，超出课程内容的不要求，考核与课程平均难度保持一致。
- 学习思路：重原理重应用，轻记忆。
- 代码语言：推荐Matlab。Python也可以。为了让我们都不痛苦，避免基础语言，比如C++



群聊: 计算物理 B

该二维码7天内(9月9日前)有效，重新进入将更新

聊聊计算与物理

计算物理b
高阳

物理为什么需要计算？

- 单体问题：解析无法提供精确解
  例子：紧束缚电子的能带

- 单体问题：解析无法提供全貌
  例子：电导随磁序角度的依赖

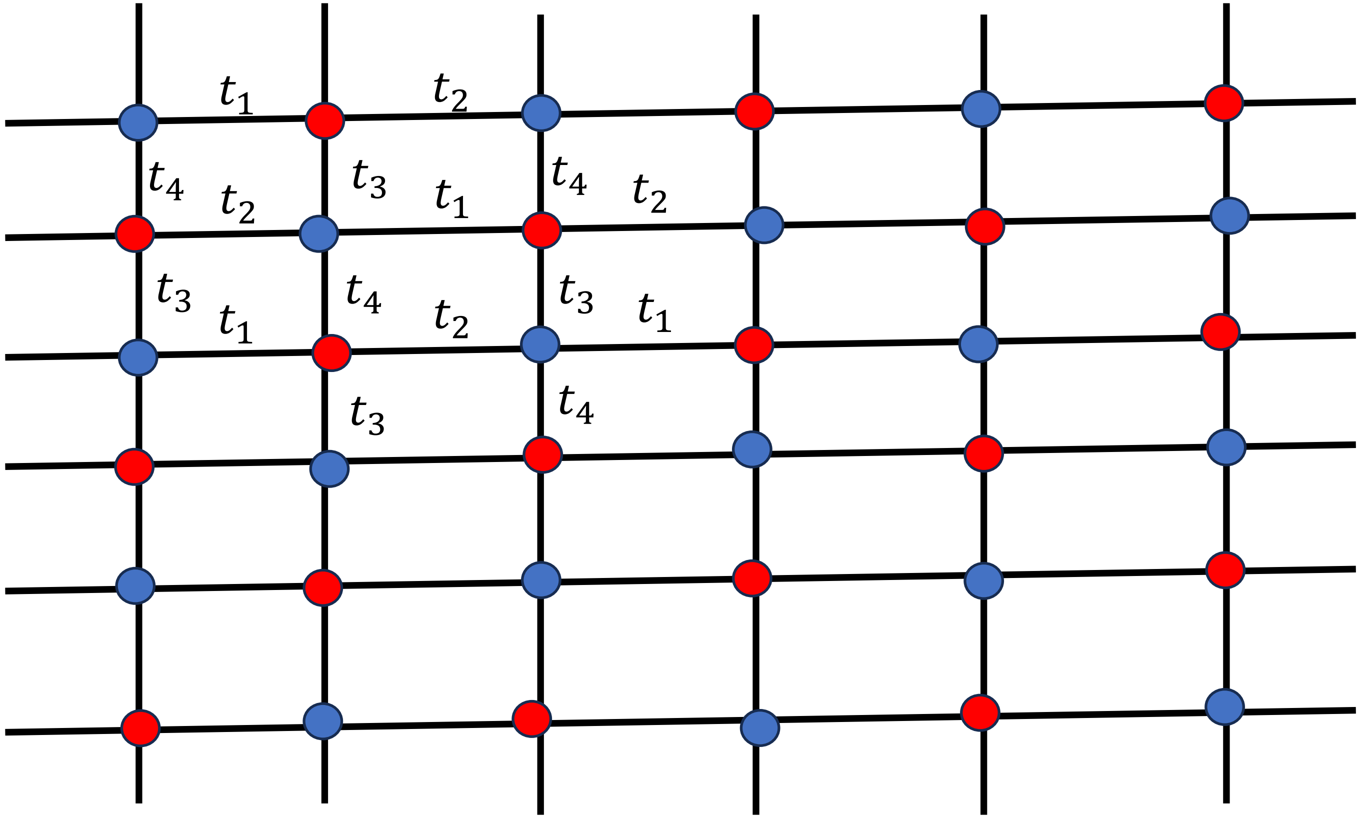
- 多体问题：体系过大，位型指数增长
  例子：二维伊辛模型的基态

计算为什么需要物理?

- 为计算方法提供图像
- 例1 退火算法搜索最小值：降温至能量最低态
- 例2 有限元方法：变分原理

计算物理示例：晶格中的电子

- 晶格：原子的周期性排列



计算物理问题示例：能量动量关系

- 哈密顿量

$$\hat{H} = \begin{pmatrix} \Delta & t_1 e^{ik_x} + t_2 e^{-ik_x} + t_3 e^{ik_y} + t_4 e^{-ik_y} \\ t_1 e^{-ik_x} + t_2 e^{ik_x} + t_3 e^{-ik_y} + t_4 e^{ik_y} & -\Delta \end{pmatrix}$$

- 能量满足

$$\hat{H}\psi(k_x, k_y) = E(k_x, k_y)\psi(k_x, k_y)$$

- 二能带可以解析，多能带需用数值对角化

计算物理问题示例：电导率

- 电导率

$$\sigma_{ij} = \int \frac{dk_x dk_y}{4\pi^2} \frac{df_0}{dE} \; v_i \, v_j$$

- 分布函数：$f_0 = \frac{1}{1+e^{\frac{E(k_x,k_y)-\mu}{k_B T}}}$

- 速度：$v_i = \frac{\partial E(k_x,k_y)}{\partial k_i}$

- 在动量空间离散化，计算能谱，计算电导率
- 改变参数可获得电导率随参量变化

计算物理问题示例：伊辛模型

- 模型
  二维点阵，每个点上有自旋，自旋可向上或向下

- 哈密顿量：$\hat{H} = \sum_{\langle ij \rangle} J S_i S_j$

- 若有N个自旋，则构型有2^N个，考虑到整体为偶函数，也有2^{N-1}个，数目增长过快

内容与思路

- 固定算法套路：蒙特卡洛算法，有限差分方法，有限元法，分子动力学法，Mathematica介绍，机器学习（待定），并行计算（待定）

- 以应用为核心，熟知基本原理

管控计算

- 算法的物理体现:
  正确性: 1. 确保算法理论上给出正确结果
  2. 从理论上找到方法判定算法正确性

- 算法本身的合理性:
  收敛性
  时间代价
  空间代价

- 程序员不想抓狂: 模块化

正确性示例1：delta函数 I

- 数学特征：$f = \delta(x), x = 0, f = \infty; x \neq 0, f = 0.$
- 应用举例：跃迁概率

$$\rho_{if} = \sum_{if} \left| T_{if} \right|^2 \delta(\varepsilon_i - \varepsilon_f - \hbar\omega)$$

- 对于 $\epsilon_i = \hbar^2 \frac{k_x^2 + k_y^2}{2m_i}, \epsilon_f = \hbar^2 \frac{k_x^2 + k_y^2}{2m_f}$,能量守恒关联的是动量空间的一条闭合曲线，数值求和需计算delta函数
- 方法：利用delta函数的特征来近似处理

正确性示例1： delta函数 II

- 数学特征： $f = \delta(x), x = 0, f = \infty; x \neq 0, f = 0.$
- 设计1： $f \approx \frac{1}{x^2}$? 也满足基本征， 甚至是偶函数
- 不满足 $\int_{-\infty}^{+\infty} f\, dx = 1$
- 方法1： $f = -\frac{1}{\pi} Im \frac{1}{x+i\epsilon}$ ϵ 为小量。

这是物质对光吸收的响应函数里的固定因子

- 方法2： $f = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$ σ为小量
- 方法3： $g = \frac{1}{1+\exp\left(\frac{x}{\lambda}\right)}$, $f = -\frac{dg}{dx}$， λ 为小量

$x = Energy, \lambda = k_B T$时为费米分布！

正确性示例2：物理上判断正确性的常用办法 --- 对称性

- 计算反常霍尔电流

$$J_x = \sigma_{xy} E_y \quad \sigma_{xy} \propto \int dk_x dk_y dk_z \Omega_z$$

- 对称性要求

$$M_x: J_x \to -J_x, E_y \to E_y,\ M_y: J_x \to J_x, E_y \to -E_y$$

σ_{xy}需破坏镜面对称性！

同理：σ_{xz}需破坏M_x和M_z, σ_{yz}需破坏M_y和M_z

- 对于破坏M_x和M_z对称性的材料，为了计算σ_{xy}，可同时计算σ_{xz}与σ_{yz}，如果后面二者不为0，则代码有问题。

合理性示例1：收敛性

- 回到delta函数的算法

$$f=-\frac{1}{\pi}Im\frac{1}{x+i\epsilon}$$

- 经常的应用场景是 $\int g(x)\delta(x)dx \approx \int g(x)f(x)dx$
- 考虑最简单情况 $g(x)=1$ $\int_{-1}^{+1} f(x)dx$ 预期值为1
- 特例： $\epsilon=0.1,(-1,1)$分成200份：0.9365

  $\epsilon=0.05,(-1,1)$分成200份：0.9682

  $\epsilon=0.01,(-1,1)$分成200份：0.9974

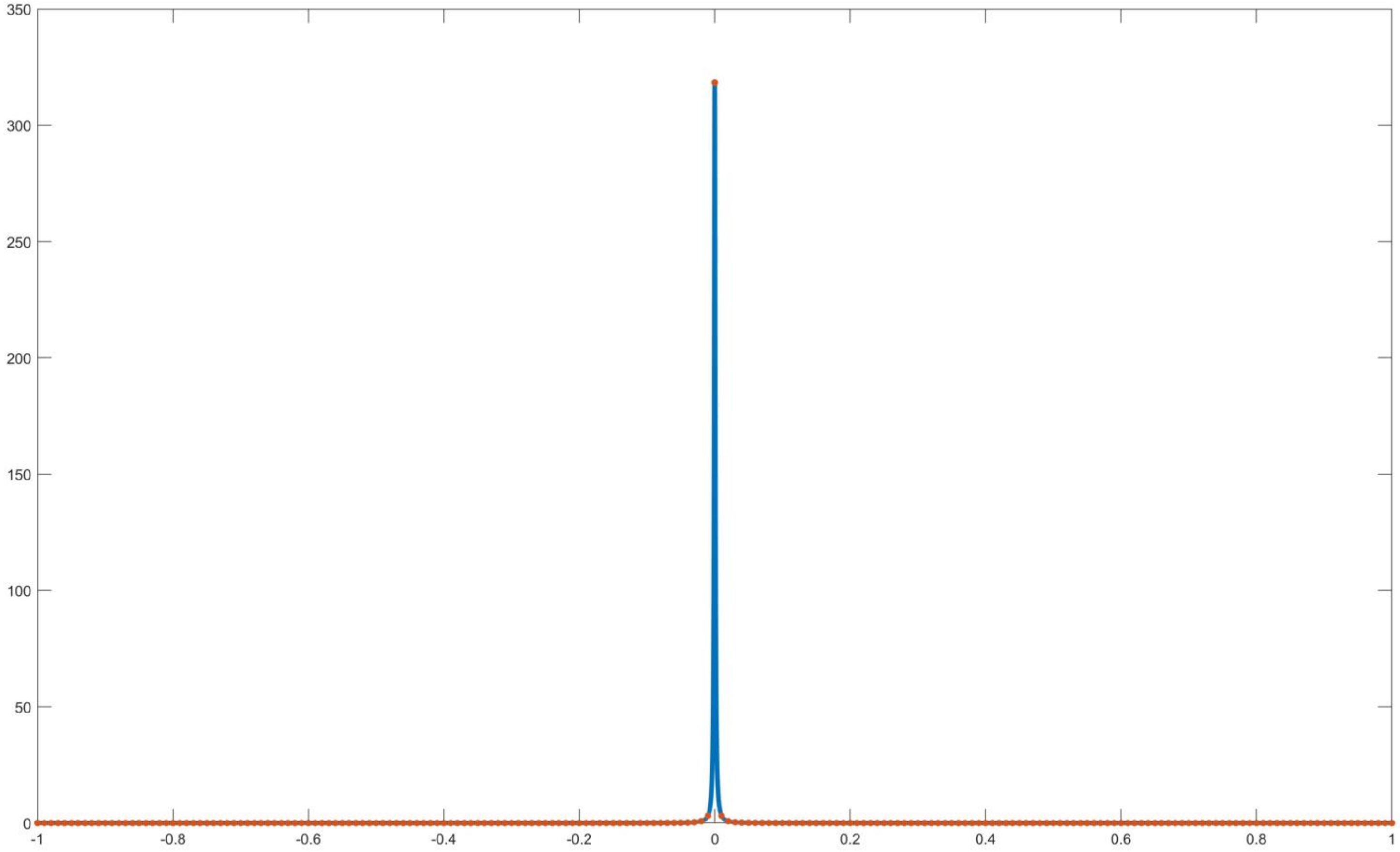
  $\epsilon=0.001,(-1,1)$分成200份：3.2865 ???

不收敛！！

合理性示例1：画出被积函数

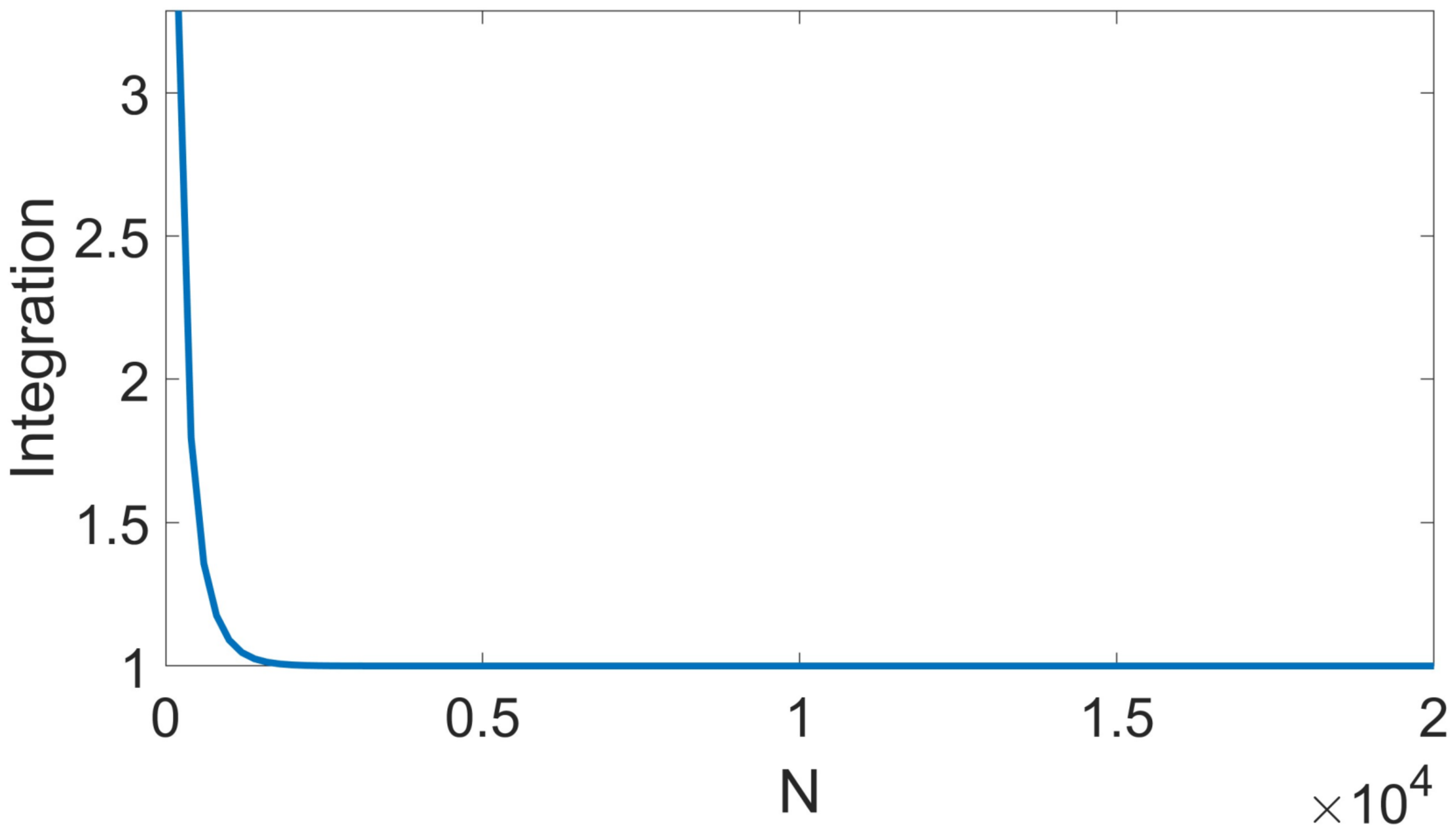
$\epsilon = 0.001, (-1,1)$分成200份

变量分划是“显微镜”，
显微程度需能匹配被积函数变化特征



合理性示例1：重新收敛

- $\epsilon = 0.001, (-1,1)$分成4000份$, 0.9994$



合理性示例2：时间成本

- Delta函数的例子：

  $\epsilon = 0.01, (-1,1)$分成200份, 0.9974

  $\epsilon = 0.001, (-1,1)$分成4000份，0.9994

  时间成本高太多了
- 可参考计算方法书里关于时间成本的分析。
- 物理中最漂亮的算法之一：第一性原理计算（DFT）

- 应找到时间成本与收敛性的平衡，这也是蒙特卡洛算法的重要意义之一

合理性示例3：空间（内存）成本

- 内存读取速率快，硬盘读取速率慢
- 常用内存空间：8G=8×10^9 Byte

    这代表什么？以32位计算机为例：整数4个字节，

    双精度浮点数8个字节，长浮点16字节。

    理论可存储：整数2×10^9,浮点数 10^9

    长浮点5×10^8
- 长浮点对应的复数矩阵大小：15811*15811
- 如果做并行，50核，每核 10^7，

    对应复矩阵：2236*2236，对于转角系统能轻松占用完

合理性示例3：错误示例

- 张量变换：

  $o_{ijkl} = \sum_{abcd} S_{ia}S_{jb}S_{kc}S_{ld}\ o_{abcd}$

- 对于三维张量：

  o_{ijkl}的大小：$3^4 = 81$ ；S的大小：9

- 如果采取直乘与缩并（mathematica内置程序）：

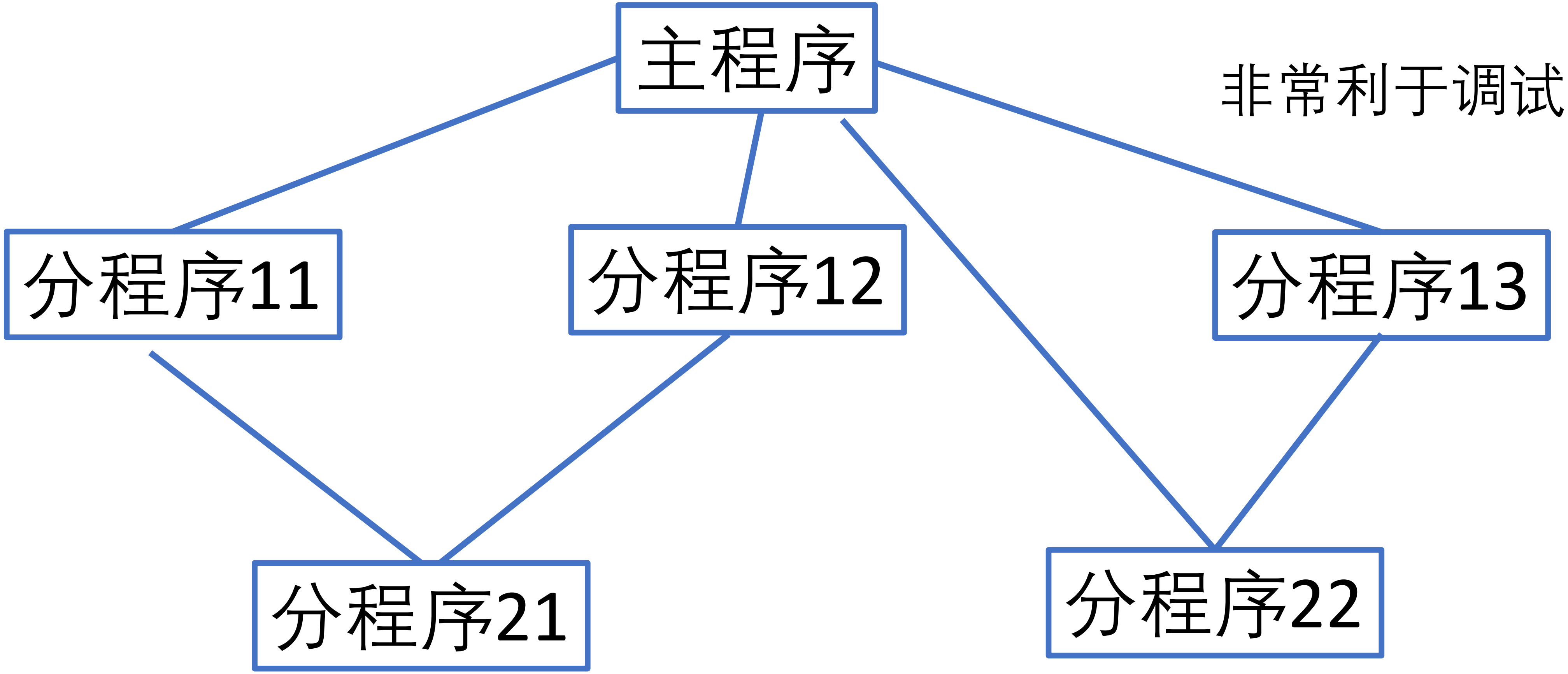
  直乘这一步$S_{ia}S_{jb}S_{kc}S_{ld}\ o_{abcd}$会产生一个12阶张量，

  大小为 $3^{12} = 531441$，会极大的增加内存消耗。

同时，高阶矩阵的寻址和求和亦十分低效。

- 正确做法：写求和代码或者用现有求和代码，保持低阶矩阵运算。

轻松性示例：模块化



非常利于调试与扩充

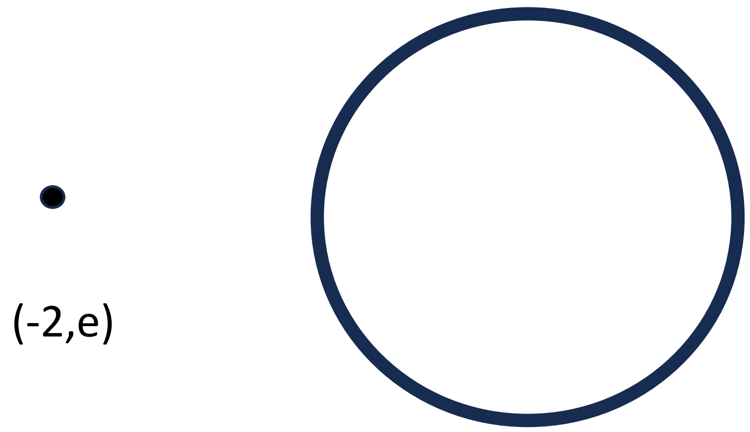
。。。。。。

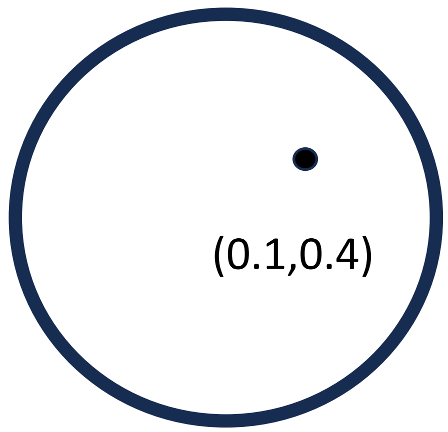
算法设计/学习原理

- 理解问题基本需求
- 基于数学或物理理论给出方案
- 列出算法步骤
- 编写程序
- 提出测试思路，并测试优化程序
- 结合算法的原理检查是否有无法处理的特例，提出解决方案并测试优化

我也创算法：点在内与点在外

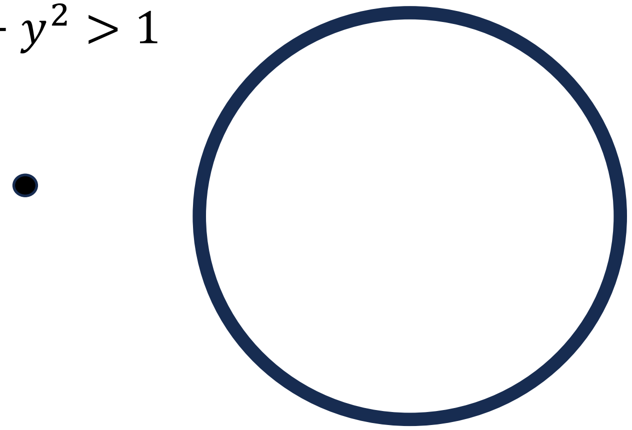
基本问题：考虑二维平面上的简单闭合曲线（同构于单位圆），给定某个点之后，判断点与线的相对关系（曲线内还是外）。

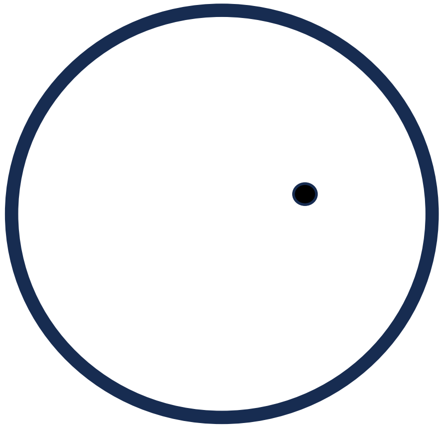




我也创算法：动机

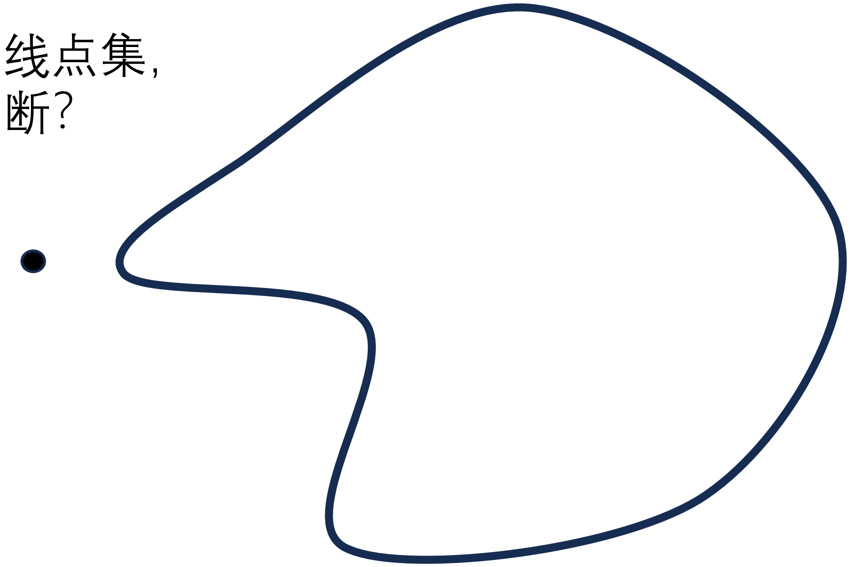
$x^2 + y^2 > 1$





$x^2 + y^2 < 1$

假如知道曲线点集，
这个如何判断？



物理实例：画等能线 $\epsilon = \cos k_x + \cos k_y$

$$\hat{H}(k)\,\psi_k = \epsilon_k\,\psi_k$$

判断最值点在等能线里还是外

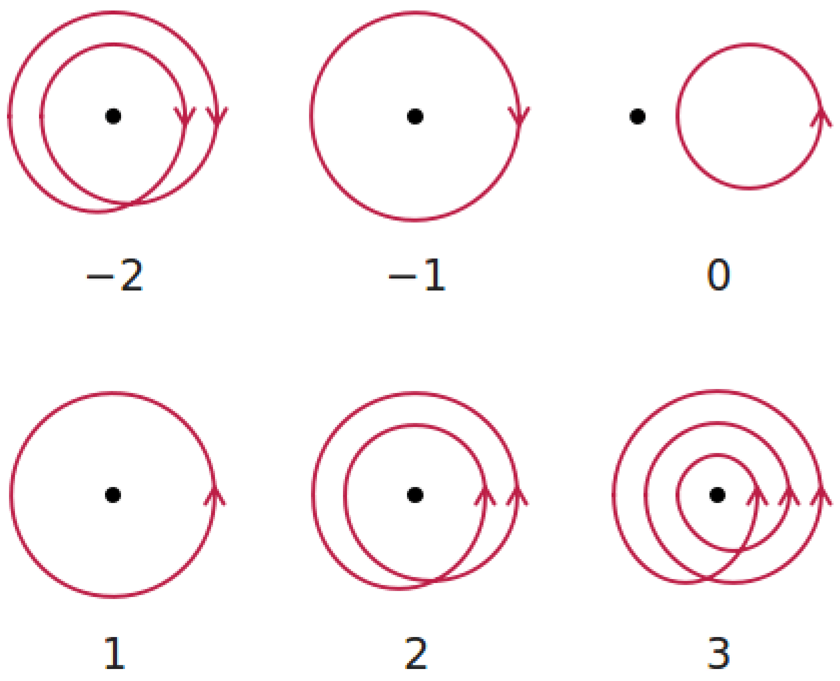
最一般的情形：可给出曲线上的一套点集

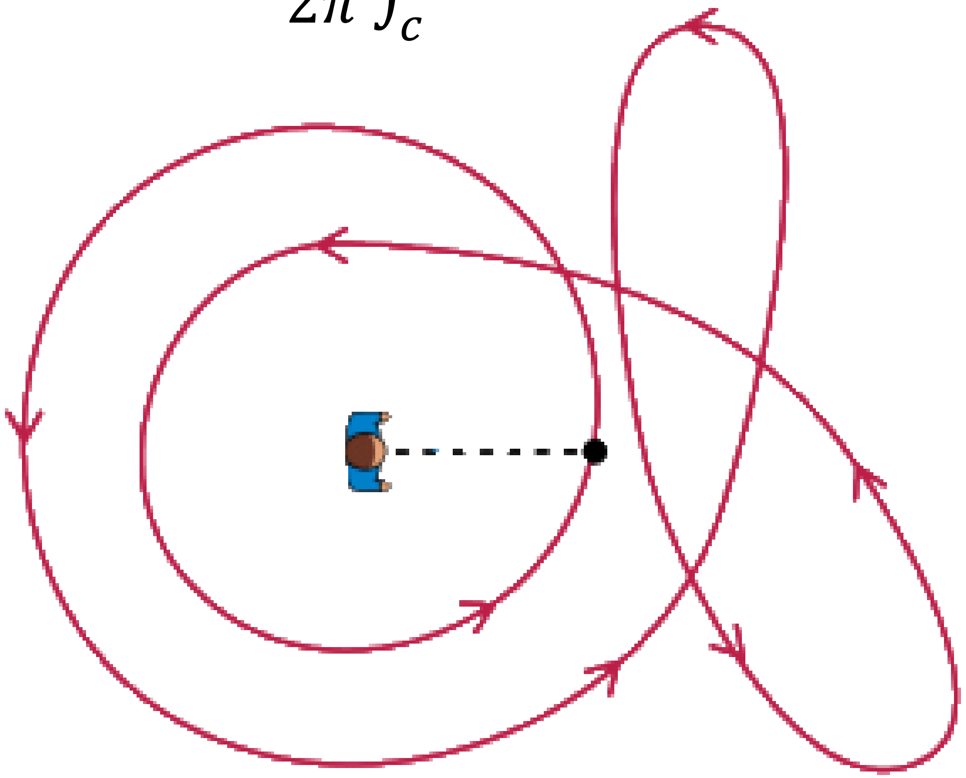
$(x_i, y_i),\quad i = 1,2,\dots,N$

判断另一点在曲线内或外

我也创算法：基于绕数的算法构思

$$f: S^1 \to S^1 \qquad n = \frac{1}{2\pi}\int_c d\theta$$



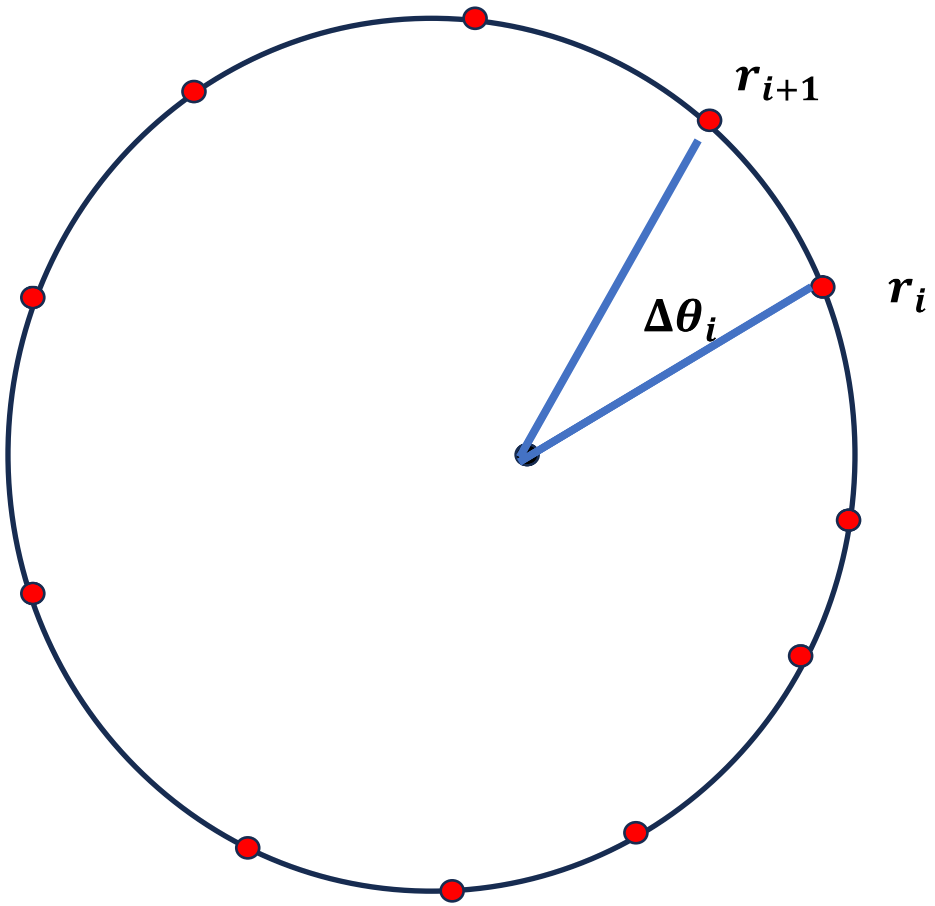


我也创算法：构思细节

$$n = \frac{1}{2\pi}\int_c d\theta$$

$$\sin\Delta\theta_i = \frac{(\boldsymbol{r_i} \times \boldsymbol{r_{i+1}})_z}{|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$$

当点足够近的时候，$\Delta\theta_i \approx \sin\Delta\theta_i$



我也创算法：算法步骤
（写作形式是考核要求）

- 生成单位圆上的点集序列：$\phi_i = \frac{2\pi}{N} * i, i = 0,1,\ldots,N$
  $x_i = \cos\phi_i$，$y_i = \sin\phi_i$ （注意：此方法保证了点集顺序排列）

- 计算相邻点与被测点所夹的角度 $\Delta\theta_i = \frac{(\boldsymbol{r_i} \times \boldsymbol{r_{i+1}})_z}{|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$

- 计算绕数 $n = \frac{1}{2\pi}\sum_{i=0}^{N-1} \Delta\theta_i$

- 判断内外 $n = 0$ 点在外，$n = \pm 1$ 点在内。

我也创算法：细节考量 1

- 对相等的判定

  $n=0 \quad n=\pm 1$

  计算机由于存储方法的设定，很多时候无法准确判断绝对的相等

  此时应改为不等式

  $x=x_0 \quad \Rightarrow \qquad |x-x_0|<\epsilon\,, \qquad \epsilon$为一个小量，比如0.001

  对于我们的算法，可改为

  $|n|<0.001, \quad |n+1|<0.001,\ |n-1|<0.001$

我也创算法：细节考量 2

- 如何提高角度计算的精度？

$$\sin\Delta\theta_i = \frac{(\boldsymbol{r_i} \times \boldsymbol{r_{i+1}})_z}{|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|} = \lambda$$

当点足够近的时候，$\Delta\theta_i \approx \sin\Delta\theta_i$

- 可以用逆函数 $\Delta\theta_i = \sin^{-1}\lambda$
- 逆函数计算代价相对于加减乘除等基本运算会慢很多
- 利用泰勒展开：$\sin x = x - \frac{x^3}{6} = \lambda$
- 利用微扰法获得更准确的角度 $\Delta\theta = \lambda + \delta\lambda$

$$\delta\lambda - \frac{1}{6}\left(\lambda^3 + 3\lambda^2\delta\lambda + 3\lambda\delta\lambda^2 + \delta\lambda^3\right) = 0$$

$$\Rightarrow \quad \delta\lambda = \frac{\lambda^3}{6-3\lambda^2} \approx \frac{\lambda^3}{6}$$

我也创算法：算法更改

- 生成单位圆上的点集序列：$\phi_i = \frac{2\pi}{N} * i, i = 0,1, \dots, N$

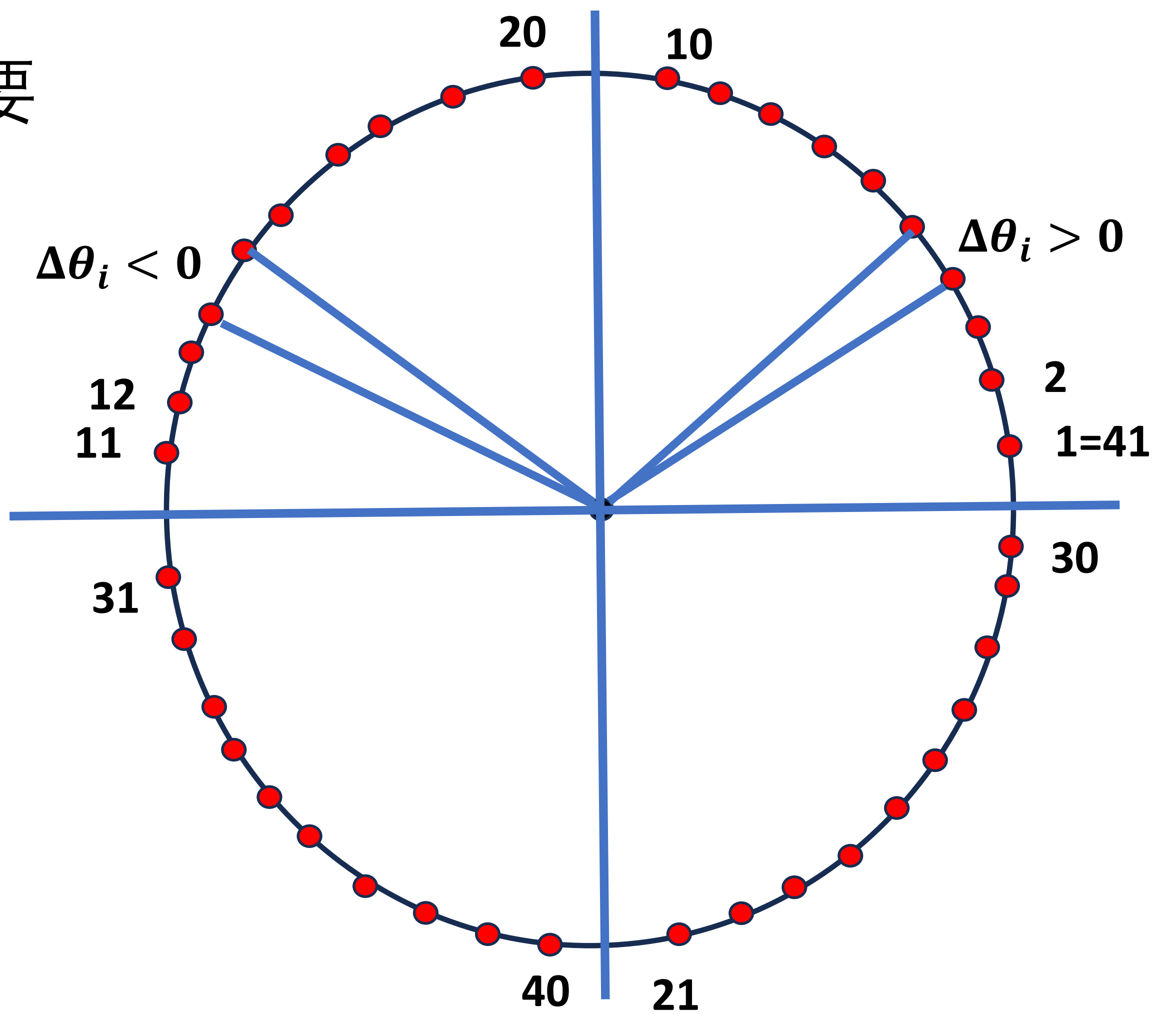
  $x_i = \cos\phi_i$，$y_i = \sin\phi_i$
- 计算相邻点与被测点所夹的角度 $\lambda = \frac{(\boldsymbol{r_i} \times \boldsymbol{r_{i+1}})_z}{|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$ $\Delta\theta_i = \lambda + \frac{\lambda^3}{6}$
- 计算绕数 $n = \frac{1}{2\pi} \sum_{i=0}^{N-1} \Delta\theta_i$
- 判断内外 $n = 0$ 点在外，$n = \pm 1$ 点在内。

我也创算法： 细节考量 3

- 点的顺序排列非常重要

10 到11， 20到21，
30到31， 40 到41会有跳变

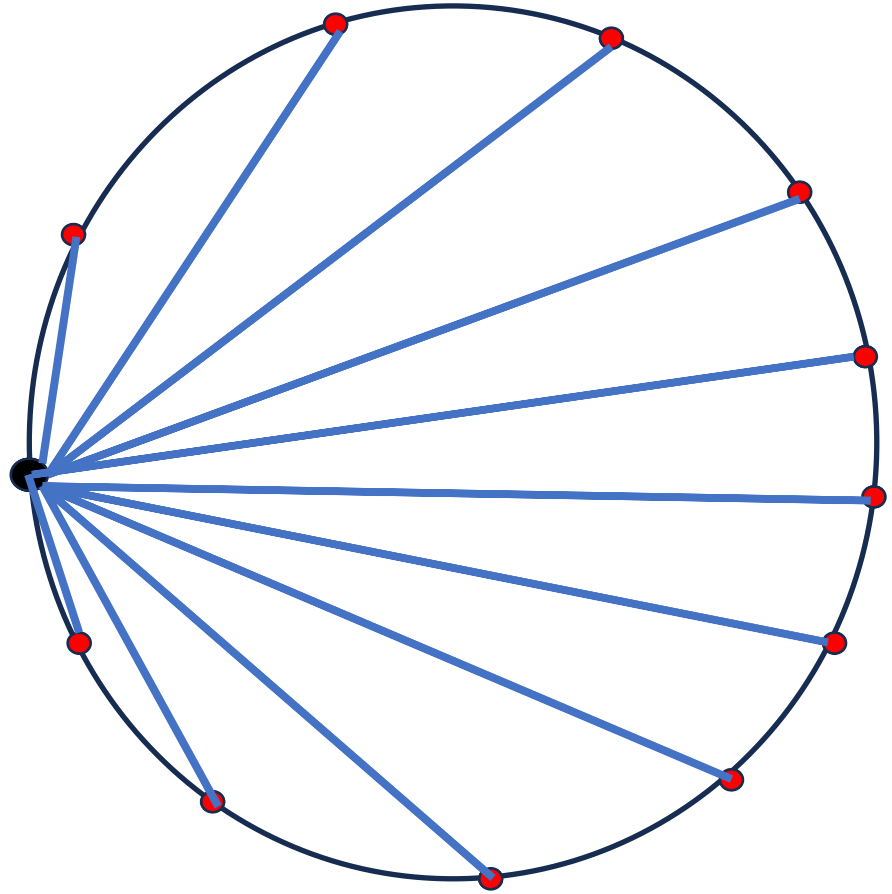
对应跳变值为： $\frac{\pi}{2}, -\pi, -\pi, \frac{\pi}{2}$
正负角度抵消，
跳变值留下，
对应角度积分为 π， 绕数不再是整数



我也创算法：细节考量 4

- 算法对于正好在球上的点不适用

会错误判定为点在圆外



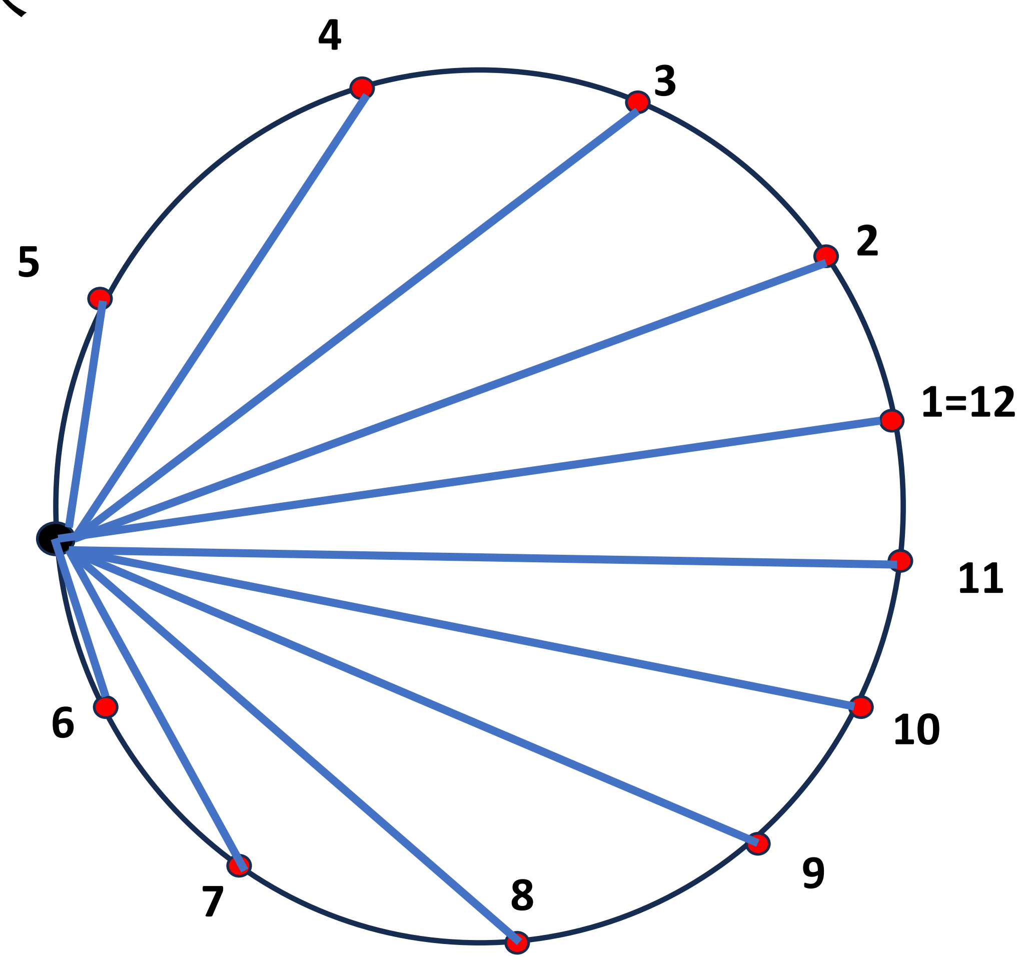
我也创算法：如何辨别？

- 辨别思路：5到6的转角会过大
- 计算完整三角函数：

$$\sin\Delta\theta_i = \frac{(\boldsymbol{r}_i \times \boldsymbol{r}_{i+1})_z}{|\boldsymbol{r}_i| * |\boldsymbol{r}_{i+1}|}$$

$$\cos\Delta\theta_i = \frac{\boldsymbol{r}_i^2 + \boldsymbol{r}_{i+1}^2 - (\boldsymbol{r}_i + \boldsymbol{r}_{i+1})^2}{2|\boldsymbol{r}_i| * |\boldsymbol{r}_{i+1}|}$$

如果$\cos\Delta\theta_i$ 接近-1，则在圆上



我也创算法：算法更改

- 生成单位圆上的点集序列：$\phi_i = \frac{2\pi}{N} * i, i = 0,1,\ldots,N$

  $x_i = \cos\phi_i$，$y_i = \sin\phi_i$

- 计算相邻点与被测点所夹的角度 $\lambda = \frac{(\boldsymbol{r_i} \times \boldsymbol{r_{i+1}})_z}{|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$ $\Delta\theta_i = \lambda + \frac{\lambda^3}{6}$

  $\mu = \frac{\boldsymbol{r_i^2} + \boldsymbol{r_{i+1}^2} - (\boldsymbol{r_i} + \boldsymbol{r_{i+1}})^2}{2|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$

- 如果 $\mu < -1 + 0.1$，点在圆上。

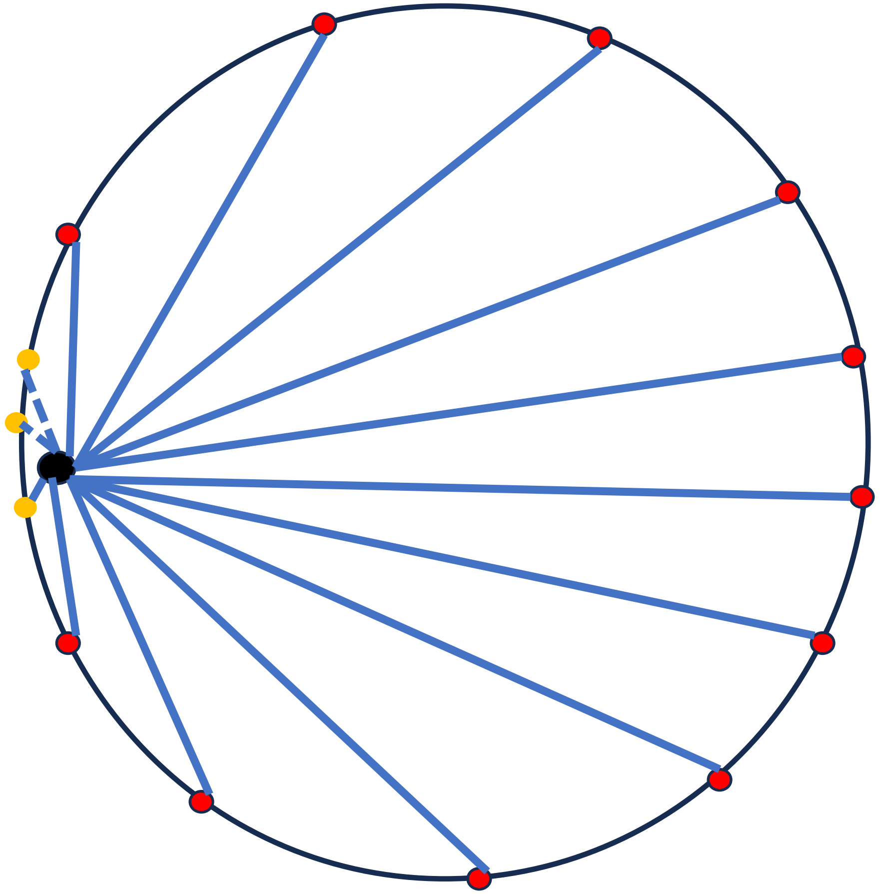
- 计算绕数 $n = \frac{1}{2\pi} \sum_{i=0}^{N-1} \Delta\theta_i$

- 判断内外 $n = 0$ 点在外，$n = \pm 1$ 点在内。

我也创算法：细节考量 5

- 对于点在圆内或者圆外，但足够接近圆，上述算法也会误判。

解决思路，加密！
如果点数加密，结果不变，则保持原判定。



我也创算法：算法更改

- 生成单位圆上的点集序列：$\phi_i = \frac{2\pi}{N} * i, i = 0,1,\dots,N$

  $x_i = \cos\phi_i$，$y_i = \sin\phi_i$
- 计算相邻点与被测点所夹的角度　　$\lambda = \frac{(\boldsymbol{r_i} \times \boldsymbol{r_{i+1}})_z}{|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$　$\Delta\theta_i = \lambda + \frac{\lambda^3}{6}$

  $\mu = \frac{\boldsymbol{r_i^2} + \boldsymbol{r_{i+1}^2} - (\boldsymbol{r_i} + \boldsymbol{r_{i+1}})^2}{2|\boldsymbol{r_i}| * |\boldsymbol{r_{i+1}}|}$
- 如果 $\mu < -1 + 0.1$，在$\boldsymbol{r_i}$和$\boldsymbol{r_{i+1}}$中间加入若干点，并重复上述步骤进行判定，如果仍有$\mu < -1 + 0.1$则点在圆上。
- 计算绕数　　$n = \frac{1}{2\pi}\sum_{i=0}^{N-1}\Delta\theta_i$
- 判断内外　$n = 0$ 点在外，$n = \pm 1$ 点在内。

作业

1. 请将点在圆内和圆外的算法中的角度计算精度提高到五阶，给出具体角度计算方式，并写出新的算法步骤。

2. （附加）请利用电磁学中的高斯定理，给出判断一个点在单位球内、球外、球上的算法，写出算法步骤，并简述原理。（鼓励大家自己写代码验算算法是否正确）（又附：此问题源自电子的能带结构。在三维布里渊区里确定好电子等能面与极值点之后如何判定等能面所包裹是哪个能量极值点。）