**选项一：多重积分的数值计算（参考p90）**

**题目描述**

使用 MATLAB 实现多重积分的数值计算，完成以下任务：

1. 利用 MATLAB 自带的数值积分函数 integral2 或 integral3，求解以下积分问题：
2. 对于以下三重积分：
   * 使用 MATLAB 的 integral3 函数计算结果。
   * 编写一个自定义的数值积分函数，通过嵌套循环的方式实现多重积分，验证 MATLAB 自带函数的结果。
3. 可视化多重积分的区域：
   * 绘制积分 和 的积分区域。
   * 使用 surf 或 mesh 绘制相关函数的图形，直观展示积分的范围和函数的形状。

**实验要求**

1. **理论部分**：简要推导和解释两道积分的积分顺序及积分区域。
2. **代码部分**：
   * 提交清晰、注释良好的 MATLAB 代码。
   * 对比 MATLAB 内置数值积分结果和自定义方法的误差。
3. **分析总结**：
   * 描述实验过程和遇到的挑战。
   * 给出每个积分的最终结果和误差分析。
   * 提供可视化结果，并说明函数形状和积分区域对结果的影响。

**扩展任务（选做）**

1. 将积分 的计算范围调整为：
   * 对比调整后结果和原积分的变化。
2. 使用**蒙特卡罗方法**对 进行估算，分析该方法与传统数值积分方法的效率和精度。

**选项二：随机游走求解多变量函数的最小值（参考p121）**

**问题描述**

使用随机游走算法，寻找以下多变量函数的全局最小值：

1. 函数定义域为：

*x*∈[−5, 5],  *y*∈[−5, 5]

1. 要求：
   * 使用随机游走算法搜索全局最优解。
   * 设置一个初始点，并通过随机扰动不断更新当前解。
   * 记录每一步的解和函数值，直到满足收敛条件（例如，函数值的变化小于阈值或达到最大迭代次数）。
2. 实现以下功能：
   * 可视化搜索路径和函数曲面。
   * 输出最优解和对应的函数值。

**问题提示**

1. **函数实现**  
   定义目标函数 f(*x*, *y*)：

f = @(x, y) sin(x) .\* cos(y) + (x-1).^2 / 4 + (y-2).^2 / 6; //MATLAB代码

1. **随机游走算法流程**
   * 初始化当前解 (*x*, *y*) 和步长 Δ。
   * 在每一步随机选择一个方向，更新解 (x, y)：
   * 如果新的解更优，则接受更新，否则保持原解。
2. **收敛条件**
   * 迭代次数达到上限。
   * 函数值的变化小于设定阈值。
3. **可视化**
   * 绘制函数的三维曲面。
   * 将随机游走的路径叠加在曲面上。

**扩展要求**

1. 增加可调整参数，例如初始步长、收敛阈值和最大迭代次数。
2. 比较不同初始点对最优解的影响。
3. 在函数复杂度较高时，分析随机游走算法的性能与局限性。
4. 完成代码后，分析搜索路径的收敛行为，并总结随机游走算法的优缺点。

**选项三：遗传算法求解多配送中心选址与路径优化问题（参考p138）**

**题目描述**

某物流公司需要在一个二维平面区域内选择多个配送中心的位置，并为配送中心分配需求点，优化以下两个目标：

1. **配送中心选址：** 确定 *k* 个配送中心的位置，最小化所有需求点到其最近配送中心的加权距离总和。
2. **路径优化：** 在每个配送中心的服务区域内，规划配送路径，使总配送路径长度最小。

这是一个**多目标优化问题**，要求设计并实现一个基于遗传算法的解决方案。

**已知条件**

1. 平面区域为 [0, 100] [0, 100]。
2. 各需求点的坐标和需求量如下：

| **需求点编号** | **坐标 (x, y)** | **需求量 (weight)** |
| --- | --- | --- |
| 1 | (20, 30) | 5 |
| 2 | (50, 50) | 10 |
| 3 | (70, 80) | 8 |
| 4 | (10, 90) | 6 |
| 5 | (90, 20) | 7 |
| 6 | (30, 40) | 4 |
| 7 | (60, 10) | 3 |
| 8 | (85, 85) | 9 |
| 9 | (5, 60) | 2 |
| 10 | (45, 70) | 6 |

1. 配送中心数量 *k* = 2。

**优化目标**

1. **选址目标：**  
   最小化总加权距离：

其中，(*xj*, *yj*) 是第 *j* 个配送中心的位置，(*xi*, *yi*) 是需求点的位置。

1. **路径目标：**  
   每个配送中心的服务区域内，规划路径，使总配送路径长度最小。路径目标为：

其中， 表示配送中心 *j* 的最短路径规划问题。

**实现要求**

1. **基因编码：**  
   编码包括两部分：
   * **配送中心位置编码：** 每个配送中心的 (*x*, *y*) 坐标。
   * **需求点分配编码：** 每个需求点被分配到的配送中心编号。
2. **适应度函数：**  
   将 *f*1 和 *f*2 合并为一个加权适应度函数：

其中， 和 为权重系数。

1. **操作步骤：**
   * **初始种群生成：** 随机初始化配送中心坐标和需求点分配方案。
   * **选择操作：**使用轮盘赌选择适应度较高的个体。
   * **交叉操作：**对配送中心位置和需求点分配分别进行交叉。
   * **变异操作：**对配送中心位置坐标或需求点分配进行随机扰动。
   * **路径规划子问题：** 在每次迭代中，利用贪心算法或动态规划求解 。
2. **终止条件：**  
   当适应度值收敛或达到最大迭代次数时停止。

**输出结果**

1. **配送中心的位置：** 最优的 *k* 个配送中心坐标。
2. **需求点的分配：** 每个需求点分配到的配送中心。
3. **路径规划：** 每个配送中心服务区域内的最优配送路径及总路径长度。
4. **优化过程：** 显示适应度值的迭代变化曲线。

**扩展任务**

* **动态需求点场景：** 需求点需求量和位置随时间变化。
* **多目标优化：** 独立优化 *f*1 和 *f*2，输出 Pareto 前沿解。

**提示**

1. 使用 MATLAB 的 **遗传算法工具箱** 或自行实现。
2. 为路径规划子问题，可以嵌套调用 MATLAB 的 **tsp\_ga** 或自行设计 TSP 求解方法。
3. 对结果进行可视化，展示平面区域中配送中心和需求点的分布以及路径规划图。

**选项四：图像压缩与重建应用（参考p160）**

**题目描述**

利用奇异值分解 (SVD) 方法对灰度图像进行压缩和重建。通过选取不同的奇异值个数，研究图像压缩效果与重建质量之间的关系。

**任务要求**

1. **图像加载与预处理：**
   * 导入一幅灰度图像，将其转换为矩阵形式。
2. **奇异值分解 (SVD)：**
   * 使用 svd 函数对图像矩阵进行分解，得到奇异值、左奇异矩阵和右奇异矩阵。
3. **图像压缩：**
   * 按照用户指定的奇异值个数 *k*，截断奇异值，重构压缩后的图像。
   * 计算压缩率：

其中 *m* 和 *n* 分别为图像矩阵的行列数。

1. **图像重建与评价：**
   * 重构图像并计算重建误差（均方误差 MSE）。

其中 *A*为原始图像，*B* 为重构图像。

* + 显示不同 *k* 值对应的压缩图像及其质量分析。

1. **用户交互：**
   * 让用户输入多个 *k* 值，比较这些值对压缩率和重建效果的影响。

**输出结果**

1. 压缩后的图像显示。
2. 奇异值个数 *k*、压缩率、重建误差（MSE）的对比表格或图表。
3. 综合评价压缩和重建的性能。

**输入数据**

测试图像可以使用 MATLAB 自带图像，例如 cameraman.tif。

**选项五：对图像进行前景和背景分割（参考p197）**

**题目描述**

利用图像处理技术和简单的分类方法，完成对一幅图像的前景和背景分割。实现过程中可以结合灰度阈值、边缘检测、区域生长等方法对图像的像素进行分类。具体要求如下：

1. **图像输入：** 从文件读取或通过 MATLAB 提供的图像，如 cameraman.tif。
2. **图像预处理：**
   * 转换为灰度图像（如果为彩色图）。
   * 应用滤波器以去除噪声（如高斯滤波或均值滤波）。
3. **分割方法：**
   * 使用简单的全局阈值（如 Otsu 阈值法）实现初步分割。
   * 尝试结合区域生长或形态学处理优化分割结果。
4. **输出：**
   * 原始图像。
   * 分割后的前景和背景显示图。

**任务要求**

1. **实现**一种方法对前景和背景进行分割，并分析分割结果的准确性。
2. **扩展任务：** 增加前景提取的准确性，如在分割后优化前景的形状或去除背景的噪声。
3. **报告：** 总结方法的实现、分割的效果以及可能的改进方向。