**21374389 牛鹏军 210711**

**理论分析**

|  |
| --- |
| c676966c21fbf6895281a7a42b031026_ |

正常参照教材算法7.2黄金分割法将伪代码实现即可，预计封装成两个函数，其一为f(x)，其二为黄金分割一维极值搜索算法。

**算法设计**

|  |
| --- |
| 906897d4db91746265f8edea0e773d5 |

**编程实现**

|  |
| --- |
| #include <stdio.h>  #include <math.h>  double f(double x) {  // 定义函数f(x)  return x\*x\*x-2\*x\*x-3\*x+4;  }  double golden\_section\_search(double a0, double b0, double e) {  double b=b0;  double a=a0;  const double phi = (sqrt(5)-1) / 2; // 黄金分割比例  double x1 = phi\*a+(1-phi)\*b; // 计算内点1  double x2 = a+b-x1; // 计算内点2  int i=0;  while (fabs(b - a) > e) {  i++;  if(f(x1)<f(x2)){  b=x2;  x2=x1;  x1=a+b-x2;  }else{  a=x1;  x1=x2;  x2=a+b-x2;  }  }  printf("迭代 %d 次\n",i);  return (a + b) / 2; // 返回极值点的估计值  }  int main() {  double a, b, e;  printf("输入区间[a, b]的端点a和b（空格分隔）：");  scanf("%lf %lf", &a, &b);  printf("输入容差值e：");  scanf("%lf", &e);  double x = golden\_section\_search(a, b, e);  printf("极值点的估计值为：%lf\n", x);  return 0;  } |

**测试分析**

当选取区间1到3，并取容差值为0.01时，得到极小值点估计值为1.868444

|  |
| --- |
|  |

用matlab绘制出该函数的图像做验证：

|  |
| --- |
| % 定义函数 f(x)  f = @(x) x.^3 - 2\*x.^2 - 3\*x - 4;  % 使用 fminsearch 函数寻找极小值点  x\_min = fminsearch(f, 0);  % 生成 x 的取值范围  x = linspace(-3, 4, 100);  % 计算对应的 y 值  y = f(x);  % 绘制函数图形  plot(x, y, 'b', 'LineWidth', 2);  hold on;  grid on;  % 绘制极小值点  plot(x\_min, f(x\_min), 'ro', 'MarkerSize', 10, 'LineWidth', 2);  % 添加标题和轴标签  title('Graph of f(x) = x^3 - 2x^2 - 3x - 4 with Minimum Point');  xlabel('x');  ylabel('f(x)'); |
| untitled |

**结论**

采用黄金分割法得到的结果比较精准，迭代次数上也具有一定的优越性能。