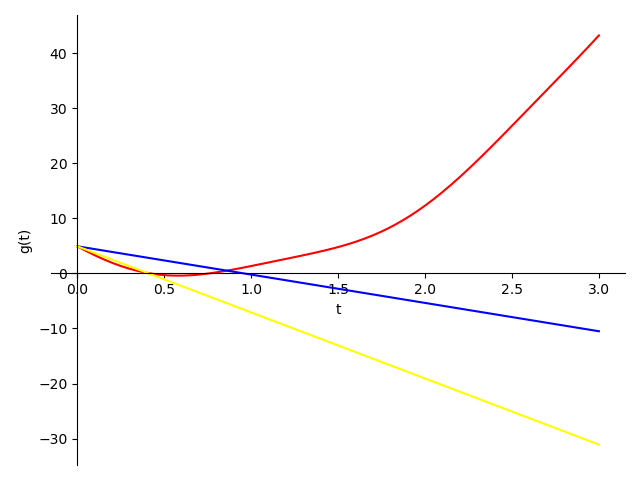
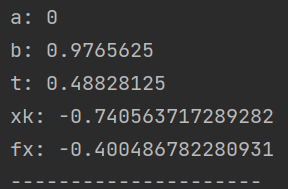
1. **Goldenstein：**

代码解释：

函数接受目标函数、搜索起始点、sigma参数、alpha参数以及起始点梯度作为输入，并返回最优步长t。在函数内部，进行迭代搜索，直到找到当前起始点最优步长t并返回。

在每一次迭代中，以当前的梯度的负方向-D作为下降方向，调用函数迭代求出步长t，计算出新的候选点，并判断该点是否满足目标函数的下降条件。如果满足条件，则更新最优解。最后得到最优解，并绘制对应求解步长t时f、g1和g2函数图像。



上左图为最后一次迭代得到的最优解点与其上一步最优步长t，右图为f、g1、g2的图线，红色为f，蓝色为g1，黄色为g2。

1. **Fletcher\_Reeves**

Fletcher\_Reeves 优化算法是一种用于求解无约束优化问题的迭代算法。该算法基于牛顿法，通过迭代寻找最优解。

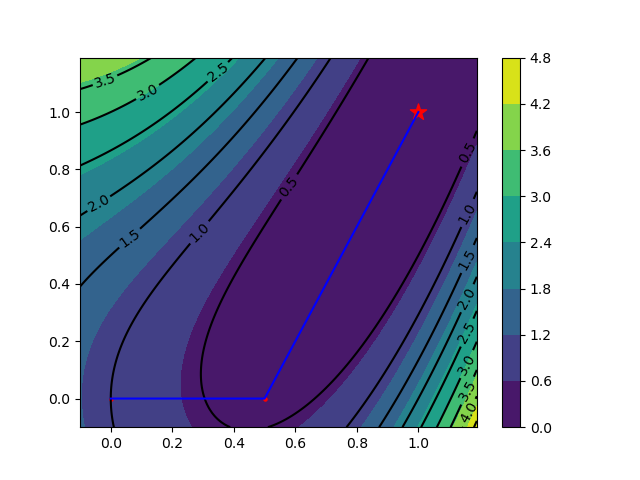
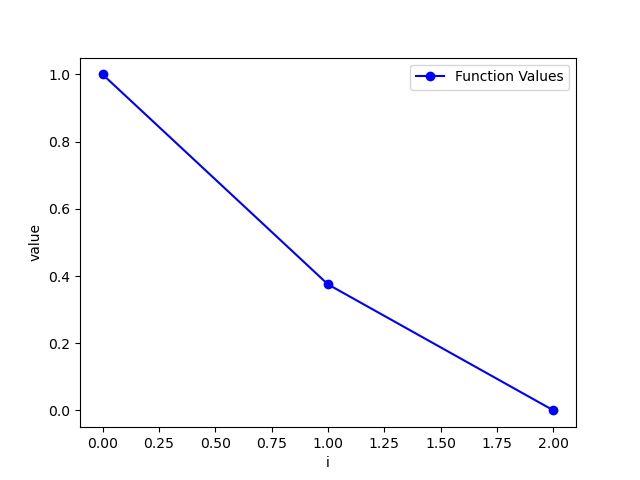
在 Fletcher\_Reeves 算法中，我们需要确定使目标函数下降的下降方向和步长，在搜索的第一步，下降方向为函数的负梯度方向,其后的每次迭代中，下降方向都为目前点的负梯度和上次下降方向的向量和(夹角为alpha = )，而每次的迭代步长由牛顿法确定。

除了计算迭代方向和步长之外，我们还需要检查迭代是否已经收敛。这可以通过计算当前迭代点与上一次迭代点之间的距离来实现。如果这个距离足够小，那么我们可以认为迭代已经收敛，算法可以停止。

总的来说，Fletcher\_Reeves 优化算法是一种基于牛顿法的迭代算法，用于求解无约束优化问题。该算法需要计算目标函数的梯度和二阶导数，并根据这些信息确定迭代方向和步长。当迭代收敛时，算法将返回最优解和最优值。

算法结果：

下图表示了初始点为(0,0)，经过两步的迭代后达到了最优解(1,1)，此时最优值为0

1. **Polak\_Ribiere**

PR算法与FR算法相似，不同处为其夹角alpha =

