**《数值计算方法》 实验报告**



**非线性方程组的数值解法**

**学 号 3022244012**

**姓 名 覃千尖**

**学 院 智算学部**

**专 业 计科1班**

**年 级 2022**

**任课教师 林迪**

**2024年 11 月 16 日**

目录

[一． 实验内容及要求 2](#_Toc400058929)

[二． 实验结果分析 2](#_Toc19120946)

[三、 优劣比较 4](#_Toc1390676019)

[四、 实验总结 5](#_Toc2027321032)

# 实验内容及要求

1.编写不动点迭代、斯特芬森加速迭代和牛顿迭代的通用程序。

(1) 设计一种不动点迭代格式，求解函数f(x) = x^2 − 3x + 2 − e ^x和g(x) =

x ^3 + x^ 2 + 10x − 20的根，要求该迭代格式收敛。然后再使用斯特芬森

加速迭代，计算到|xk − xk−1 | < 10^−8为止。

(2) 用牛顿迭代，同样计算到|xk − xk−1 | < 10^−8。输出迭代初值、迭代次数

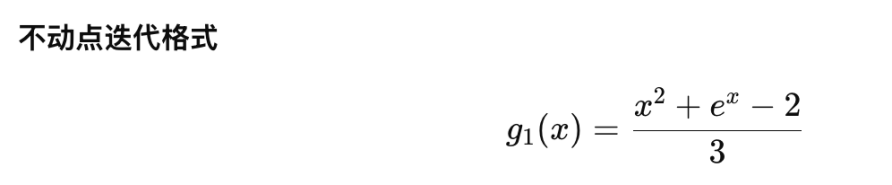
及各次迭代值，比较方法优劣。

# 实验结果分析

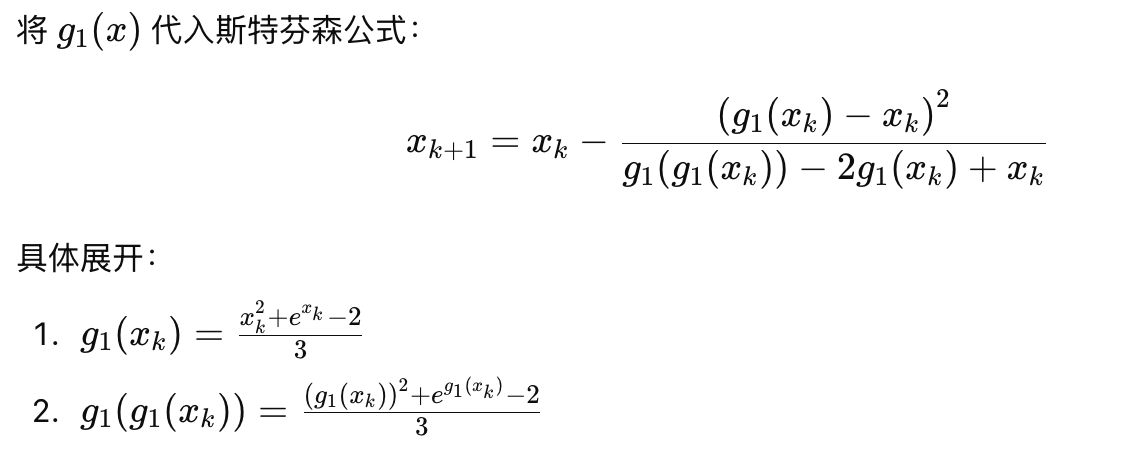
1. 函数f(x) = x^2 − 3x + 2 − e ^x

1 . 不动点迭代：

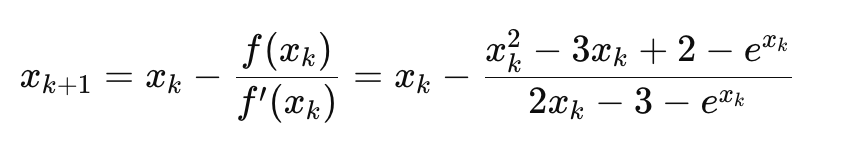
1. 选定迭代公式：



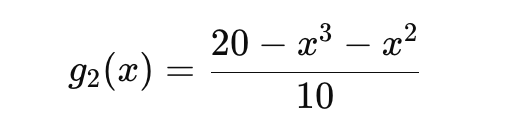
1. 迭代次数：14
2. 方程的根：x = 0.2575302868550145
3. 斯特芬森加速迭代
4. 选定迭代公式：



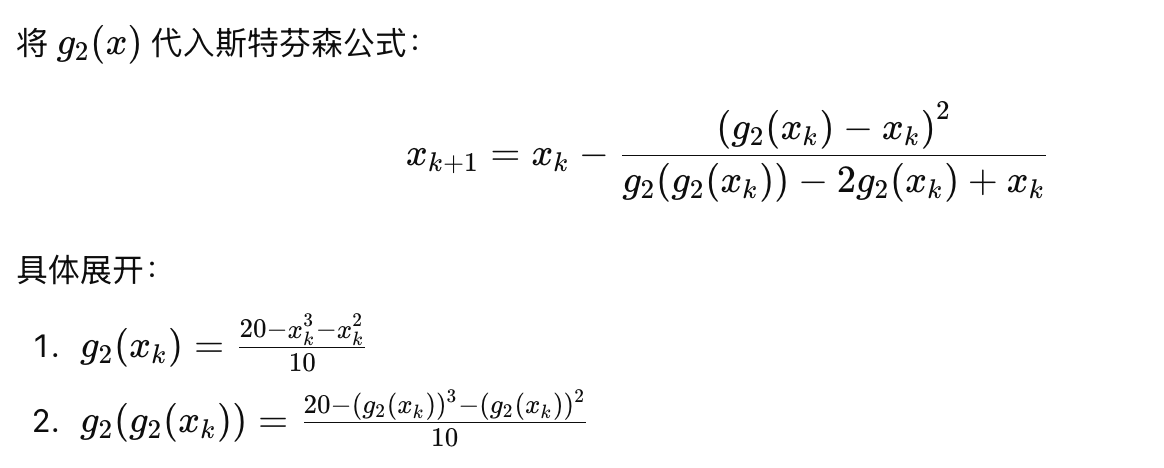
1. 迭代次数：4
2. 方程的根： x = 0.25753028543986073
3. 牛顿迭代
4. 迭代公式：



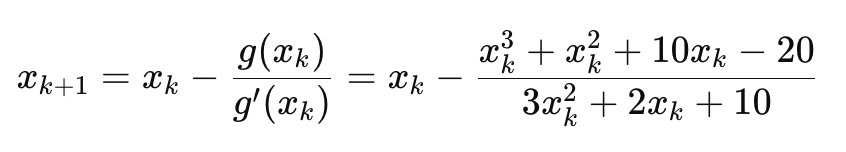
1. 迭代次数：4
2. 方程的根： x = 0.2575302854398608
3. 函数g(x) = x ^3 + x^ 2 + 10x − 20
4. 不动点迭代：
5. 迭代公式：



1. 迭代次数： 255
2. 方程的根：x = 1.4680721943427395
3. 斯特芬森迭代加速：
4. 迭代公式：



1. 迭代次数：4
2. 方程的根：x = 1.4680721990944698
3. 牛顿迭代：
4. 迭代公式：



1. 迭代次数：4
2. 方程的根：x = 1.4680721990944698

# 优劣比较

### 1. ****固定点迭代法（Fixed-point Iteration）****

* **收敛性**：固定点迭代法的收敛速度较慢。从你提供的结果可以看到，固定点迭代在进行大量迭代后逐渐收敛，但需要非常多的迭代才能达到较为稳定的解。
* **计算量**：需要进行大量的迭代，效率相对较低。根据你的输出，方法需要进行多达100次迭代以上才能接近解，且每次迭代计算的时间较长。
* **稳定性**：固定点迭代法依赖于迭代函数的收敛性，如果选择的迭代函数不合适，可能会导致不收敛或收敛慢。因此，对于不满足收敛条件的函数，固定点法可能表现较差。

### 2. ****斯特芬森加速法（Steffensen's Acceleration）****

* **收敛性**：斯特芬森加速法是对固定点迭代法的加速版本，它通过对迭代结果进行加速处理，通常能够比固定点法更快收敛。在你的结果中，经过几次加速迭代后，解在较少的迭代次数内就达到了稳定值（例如3次加速后便已经接近最终解）。
* **计算量**：尽管每次迭代仍然需要计算函数值，但加速可以显著减少迭代次数，减少计算时间，相较于固定点法更高效。
* **稳定性**：斯特芬森加速法通过加速固定点法的收敛，具有较好的稳定性和较快的收敛速度，但同样也受到初始猜测和加速条件的影响。如果起始值选择不当，加速效果可能不显著。

### 3. ****牛顿法（Newton's Method）****

* **收敛性**：牛顿法的收敛速度非常快，特别是在接近解时表现出二次收敛特性，即解每次迭代后会迅速精确到更高的精度。在你提供的结果中，牛顿法在经过少数几次迭代后就达到了非常精确的解，甚至在3到4次迭代后就收敛到了最终值。
* **计算量**：牛顿法每次迭代需要计算函数的导数，这可能会导致额外的计算量。但由于其快速收敛，所需的迭代次数远远少于固定点迭代法，因此总体计算量通常较小。
* **稳定性**：牛顿法要求函数在解附近具有良好的导数，并且起始猜测应当接近真实解。如果初始值选择不当，牛顿法可能会发散或进入不合适的解。因此，牛顿法需要谨慎选择初值。

# 实验总结



**适用场景**：

* **固定点迭代法**适合简单的迭代方法，能处理简单的方程，但效率较低。
* **斯特芬森加速法**适用于需要改进收敛性的场景，能够在固定点法的基础上加速收敛，效果较为理想。
* **牛顿法**则是最为高效且快速的迭代方法，适合要求精度和速度高的场景，但对初值和导数要求较高。