Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа по

Вычислительной математике №3

Вариант «Метод простых итераций»

Работу выполнила:

Касьяненко В.М.

Группа:

P3220

Санкт-Петербург,

2024

Описание численного метода.

Метод простых итераций – это численный метод для нахождения корней системы нелинейных уравнений, основанный на принципе последовательных итераций, в которых текущее приближение корней системы используется для вычисления следующего приближения.

Допустим, у нас есть система нелинейных уравнений:

Мы можем переписать её в виде:

Где является некоторой функцией, преобразовывающей систему в эквивалентную форму. Затем мы начинаем с некоторого начального приближения ​ и итеративно обновляем их значения по следующему правилу:

Процесс повторяется до тех пор, пока изменения ​ станут достаточно малы или до достижения максимального числа итераций.

Блок-схема

Изображение выглядит как зарисовка, диаграмма, рисунок, Технический чертеж

Автоматически созданное описание

Код численного метода

def solve\_by\_fixed\_point\_iterations(system\_id, number\_of\_unknowns, initial\_approximations):  
 max\_iterations = 100  
 tolerance = 1e-5  
 step\_length = 0.0001  
  
 def fixed\_point\_iteration(func, initial\_guesses, max\_iter, tol, step\_length):  
 iterations = 0  
 x\_prev = initial\_guesses  
 while iterations < max\_iter:  
 x\_next = func(x\_prev, step\_length)  
 if all(abs(x\_next[i] - x\_prev[i]) < tol for i in range(number\_of\_unknowns)):  
 break  
 x\_prev = x\_next  
 iterations += 1  
 return x\_next, iterations  
  
 functions = get\_functions(system\_id)  
  
 def system(x, step\_length):  
 return [x[i] - step\_length \* functions[i](x) for i in range(number\_of\_unknowns)]  
  
 solution, iterations = fixed\_point\_iteration(system, initial\_approximations, max\_iterations,   
tolerance, step\_length)  
 return solution

Примеры работы программы

1)

Входные данные:  
1

2

1.0

2.0  
Выходные данные:

0.9916078527646449

1.9900660838909958

2)

Входные данные:  
3

3

0.5

-1.0

2.0  
Выходные данные:

0.4538072181466193

-1.0304576235085257

1.9538281206568267

3)

Входные данные:

3

3

0.0

0.0

0.0

Выходные данные:

0.0009950238612479096

-0.001990174530636173

0.0029851821351471173

4)

Входные данные:

1

1

0.0

Выходные данные:

0.0

5)

Входные данные:

2

2

1.0

2.0

Выходные данные:

1.445721981124815

2.1435695839180386

Выводы

В результате выполнения лабораторной работы был реализован метод простых итераций для решения систем нелинейных уравнений, основанный на идее последовательного уточнения начального приближения до достижения требуемой точности. Метод простых итераций легко реализуем и понятен, а также дает возможность решения большого класса нелинейных систем уравнений. Метод простых итераций преобразует исходную систему уравнений в эквивалентную систему, где каждое уравнение содержит одну из неизвестных в явном виде, а остальные неизвестные выражены через уже найденные значения на предыдущей итерации. Это позволяет свести решение к поиску неподвижной точки функции итерации.

Сравнение с методом Ньютона и методом бисекции. Метод простых итераций является простым в реализации. Он не требует вычисления производных функций и может быть применен к широкому спектру задач. Однако его сходимость может быть медленной, что может привести к необходимости выполнять большое количество итераций для достижения требуемой точности. Метод Ньютона, с другой стороны, обеспечивает быструю сходимость, особенно вблизи решения. Однако для его применения требуется вычисление производных функций, что может быть сложно или даже невозможно в некоторых случаях. Кроме того, метод Ньютона может быть неустойчивым при выборе неправильного начального приближения или для некоторых видов систем. Метод бисекции является простым и гарантирует сходимость к решению на отрезке, где функция меняет знак. Он не требует вычисления производных и легко реализуется. Однако метод бисекции медленно сходится и применим только к уравнениям с одной переменной.

Анализ применимости метода: метод простых итераций хорошо подходит для простых систем нелинейных уравнений, особенно если есть хорошее начальное приближение и известно, что решение находится вблизи этого приближения. В случае многомерных систем уравнений (с большим числом неизвестных) метод простых итераций может быть менее эффективным из-за необходимости итерировать по всем переменным сразу. Также метод простых итераций склонен к нахождению только локальных решений системы уравнений. Если необходимо найти глобальные решения, может потребоваться использовать другие методы, такие как метод Ньютона.

Общая алгоритмическая сложность метода простых итераций может быть выражена как , где – количество неизвестных, а – максимальное количество итераций.

Метод простых итераций может иметь ограниченную точность из-за приближенных вычислений. Это может привести к накоплению погрешности на каждой итерации, что в итоге может ухудшить точность результата. Число итераций, необходимых для достижения заданной точности, также важно для анализа численной ошибки. Большое число итераций может быть затратным с точки зрения вычислительных ресурсов, в то время как недостаточное количество итераций может привести к неточному результату. Метод простых итераций также может быть чувствителен к начальным приближениям. Небольшие изменения в начальных условиях могут привести к значительным изменениям в итоговом решении.

Таким образом, метод простых итераций является хорошим инструментом для численного решения систем нелинейных уравнений, но требует внимательной настройки и анализа для достижения точных и стабильных результатов.