

Группа № 1

Тема: Приближенное решение уравнений в электронных таблицах.

Приближенное решение уравнений на языке Visual Basic.

Содержание: Проект «Приближенное решение уравнения» на языке Visual Basic.

Компьютерные модели, построенные с использованием метода Монте-Карло на языке Visual Basic

Формальная модель «Определение площади круга методом Монте-Карло».

Формальная модель-задание «Бросание монеты».

Проект «Определение площади круга с использованием метода Монте-Карло» на языке Visual Basic.

Выполнить практическую работу 2.

Практическая работа № 2

«Приближенное решение квадратных уравнений на языке Visual Basic».

Цель работы – познакомиться с языками программирования, разработкой и исследованием моделей на компьютере.

На языке алгебры формальные модели записываются с помощью уравнений, точное решение которых основывается на поиске равносильных преобразований алгебраических выражений, позволяющих выразить переменную величину с помощью формулы.

Для большинства уравнений приходится использовать методы приближенного решения с заданной точностью

Основные этапы работы:

- знакомство с языками программирования;
- знакомство с методами решения математических задач в программе Excel;
- содержательная постановка задачи;
- описательная модель;
- формальная модель;
- компьютерная модель.

Написанную программу планируется использовать на уроках математики в качестве учебного пособия.

На языке алгебры формальные модели записываются с помощью уравнений, точное решение которых основывается на поиске равносильных преобразований алгебраических выражений, позволяющих выразить переменную величину с помощью формулы.

Точные решения существуют только для некоторых уравнений определенного вида (линейные, квадратные, тригонометрические и др.), поэтому для большинства уравнений приходится использовать методы приближенного решения с заданной точностью (графические или численные).

Например, нельзя найти корень уравнения $x^3 - \cos x = 0$ путем равносильных алгебраических преобразований. Однако такие уравнения можно решать приближенно графическими и численными методами

Численные методы решения нелинейных уравнений.

1.1. Постановка задачи.

Пусть имеется уравнение вида

$$f(x) = 0. \quad (1)$$

где $f(x)$ - заданная алгебраическая или трансцендентная функция.

(**Функция** называется **алгебраической**, если для получения её значения нужно выполнить арифметические операции и возведение в степень с рациональным показателем. Примеры **трансцендентных функций** - показательная, логарифмическая, тригонометрические, обратные тригонометрические.)

Решить уравнение - значит найти все его корни, то есть те значения x , которые обращают уравнение в тождество, или доказать, что корней нет. Если алгебраическое или трансцендентное уравнение достаточно сложно, то довольно редко удастся точно найти его корни. Кроме того, в некоторых случаях уравнение может содержать коэффициенты, известные лишь приблизительно, поэтому сама задача о точном нахождении корней теряет смысл. В таких случаях применяют численные (приближенные) методы решения.

Поставим задачу найти такое приближенное значение корня x_{np} , которое мало отличается от точного значения корня x^* , так что выполняется неравенство $|x^* - x_{np}| < \varepsilon$, где ε (эпсилон) – малая положительная величина – допустимая ошибка, которую мы можем заранее задать по своему усмотрению. Если корень найден с точностью ε , то принято писать $x^* = x_{np} \pm \varepsilon$.

Будем предполагать, что уравнение (1) имеет лишь изолированные корни, т.е. для каждого корня существует окрестность, не содержащая других корней этого уравнения.

1.2. Этапы приближенного решения нелинейных уравнений.

Приближенное решение уравнения состоит из двух этапов:

1. **Отделение корней**, то есть нахождение интервалов из области определения функции $f(x)$, в каждом из которых содержится только один корень уравнения (1).
2. **Уточнение корней** до заданной точности.

Отделение корней можно проводить графически и аналитически.

Для того чтобы графически отделить корни уравнения (1), необходимо построить график функции. Абсциссы точек его пересечения с осью Ox являются действительными корнями уравнения.

На практике же бывает удобнее заменить уравнение (1) равносильным ему уравнением

Пример 1. Отделить графически корень уравнения.

Решение. Для решения задачи построим график функции.

Из рисунка видно, что один из корней уравнения принадлежит отрезку , второй – отрезку . Так как рассматриваемое уравнение имеет третью степень, то должен существовать еще один корень на интервале

Аналитическое отделение корней основано на следующих теоремах.

Теорема 1. Если непрерывная функция принимает на концах

отрезка значения разных знаков, т.е. , то на этом

отрезке содержится по крайней мере один корень уравнения (1)

Теорема 2. Если непрерывная на отрезке функция принимает на

концах отрезка значения разных знаков, а производная сохраняет знак внутри отрезка , то внутри отрезка существует единственный корень уравнения $f(x) = 0$

Пример 3. Подтвердить аналитически правильность нахождения отрезка изоляции корня уравнения .

Решение. Для отрезка имеем: ; Значит, . Следовательно, корень отделён правильно.

Уточнение корней до заданной точности заключается в сужении интервала изоляции корня и выполняется одним из специальных методов. Наиболее распространенными являются метод деления отрезка пополам, метод касательных (Ньютона), метод секущих (хорд).

1.3. Уточнение корней методом деления отрезка пополам.

Метод деления отрезка пополам имеет другие названия: метод половинного деления, метод дихотомии, метод проб, метод бисекций.

Пусть корень уравнения $f(x) = 0$ отделен на отрезке ,

т.е. .

Алгоритм приближенного вычисления корня методом половинного деления.

Исходные данные:

$f(x)$ – функция;

ε – требуемая точность;

a, b – границы заданного интервала (границы поиска корня).

Результат: $x_{\text{пр}}$ – приближенный корень уравнения $f(x) = 0$.

Метод решения:

Шаг 1. Выбрать середину отрезка в качестве приближенного корня. Шаг 2.

Если , то c – искомый корень уравнения, на этом прекращаем вычисления. В противном случае перейти к шагу 3.

Шаг 3. Точный корень уравнения x^* отличается от c не более чем на половину длины отрезка, т.е. не более чем на (полученная точность). Проверяем

условие . Если условие не выполняется, т.е. полученная точность нас не устраивает (она больше, чем требуемая), то перейти к шагу 4; в противном случае прекратить вычисления, поскольку мы достигли требуемой точности, и приближенным корнем уравнения $f(x) = 0$ считать середину c отрезка .

Шаг 4. Определить интервал дальнейшего поиска корня. Из двух образовавшихся при делении отрезков переходим к той из его половин и , на концах которого функция принимает значения разных знаков.

Случай 1 (рис. 7). Корень на отрезке . , граница b сдвигается влево – заменить b на c : $b := c$.

Корень на отрезке . , граница a сдвигается вправо – заменить a на c : $a := c$.

Перейти к шагу 1.

Алгоритм деления отрезка пополам довольно медленный, но зато абсолютно застрахован от неудач. Основное достоинство метода состоит в том, что его скорость сходимости не зависит от вида функции $f(x)$. Данный метод не имеет дополнительных условий сходимости, кроме .

Так же есть методы хорд и касательных.

Графическое решение таких уравнений можно осуществить путем построения компьютерных моделей:

- построением графика функции в системе объектно-ориентированного программирования Visual Basic или Turbo Delphi
- в электронных таблицах Microsoft Excel или OpenOffice.org Calc путем построения диаграммы типа График

Найдем корень уравнения $x^3 - \cos x = 0$ приближенными методами (графическим и численным методом деления пополам числового отрезка аргумента)

Формальная модель задана уравнением, для нахождения корня уравнения разработаем компьютерную модель на языке Visual Basic.

Графический метод

```
1 Dim Graph1 As Graphics
```

```
Dim Pen1 As New Pen (Color.Black, 2)
```

```
Dim drawBrush As New SolidBrush (Color.Black)
```

```
Dim drawFont As New Font ("Arial", 10)
```

```
Dim X, Y As Single
```

```
‘Графическое решение уравнения
```

```
Private Sub Button1_Click(...)
```

```
Graph1=Me.PictureBox1.CreateGraphics()
```

```
Graph1.Clear (Color.White)
```

```
‘Печать шкал математической системы координат в компьютерной системе координат
```

```
For X=-150 To 150 Step 50
```

```
Graph1.DrawString (X/100, drawFont,_  
drawBrush, X+150, 50)
```

```
Next X
```

```

For Y=0 To 200 Step 50
Graph1.DrawString ((Y-150)/100, drawFont,_
drawBrush, 150, 200-Y)
Next Y
‘Преобразование компьютерной системы координат в математическую
систему координат
Graph1.ScaleTransform(1, -1) ‘поворот оси Y
Graph1.TranslaterTransform (150, -50) ‘Сдвиг по осям X и Y
‘Рисование осей математической системы координат
Graph1.DrawLine (Pen1, -150, 0, 300, 0) ‘ось X
Graph1.DrawLine (Pen1, 0, -150, 0, 50) ‘ось Y
For X=-150 To 150 Step 50 ‘засечки на оси Y
Graph1.DrawLine (Pen1, X, -5, X, 5)
Next X
For Y=-100 To 100 Step 50 ‘засечки на оси Y
Graph1.DrawLine (Pen1, -5, Y, 5, Y)
Next Y
‘График функции
For X=-1.5 To 1.5 Step 0.01
Y=X^3-Math.Cos(X)
Graph1.DrawEllipse (Pen1, X*100, Y*100, 1, 1)
Next X
End Sub

```

График функции пересекает ось X один раз, следовательно, уравнение имеет один корень. По графику грубо приближенно можно определить, что $x \approx 0.8$ см
Метод половинного деления

УТОЧНЕНИЕ КОРНЯ УРАВНЕНИЯ МЕТОДОМ ПОЛОВИННОГО ДЕЛЕНИЯ

При решении уравнения, как правило, заранее задается допустимая погрешность ϵ приближенного значения корня E . В процессе уточнения корней требуется найти их приближенные значения, отличающиеся от точных не более чем на ϵ .

Описанный выше способ табулирования может рассматриваться и как способ уточнения корня (хотя и крайне неэффективный). При этом можно либо постепенно уменьшать шаг табулирования, приближая его к значению ϵ , либо сделать это сразу, полагая $h = \epsilon$. В любом случае получим $b - a < \epsilon$. Тогда в качестве искомого значения корня можно выбрать середину этого отрезка, т.е. положить $E = (a + b)/2$.

Гораздо более эффективным является так называемый **метод половинного деления**.

Пусть уравнение $F(x)=0$ имеет на отрезке $[a; b]$ единственный корень, причем функция $F(x)$ на этом отрезке непрерывна. Разделим отрезок $[a; b]$ пополам точкой $c = (a + b)/2$. Если $F(c) \neq 0$ (что практически наиболее вероятно), то возможны два случая: $F(x)$ Меняет знак либо на отрезке $[a; c]$ (рис. 2.6, а), либо на отрезке $[c; b]$ (рис. 2.6, б). Выбирая в каждом случае тот из отрезков, на котором функция меняет знак, и продолжая процесс половинного деления дальше, можно дойти до сколь угодно малого отрезка, содержащего корень уравнения.

Рис. 2.6. К решению уравнения $F(x)$ методом половинного деления:

а — функция $F(x)$ меняет знак на отрезке $[a; c]$;

б — функция $F(x)$ меняет знак на отрезке $[c; b]$

Метод половинного деления вполне можно использовать как метод решения уравнения с заданной точностью. Действительно, если на каком-то этапе процесса получен отрезок $[a; b]$, содержащий корень, то, приняв приближенно $x = (a + b)/2$, получим ошибку, не превышающую значения (заметим, что речь в данном случае идет о погрешности метода). Метод половинного деления требует утомительных ручных вычислений, однако он легко реализуется с помощью программы на компьютере (блок-схему алгоритма см. на рис. 2.7). Отметим, что даже если на каком-то этапе деления отрезка пополам получится $F(c) = 0$, это не приведет к сбою алгоритма.

Блок-схема алгоритма уточнения корня уравнения $F(x)=0$ на отрезке $[a; b]$ с точностью ϵ методом половинного деления

Использование метода половинного деления в написании программы.

Написание программы и составление интерфейса

Поместить на форму:

текстовые поля TextBox1 и TextBox2 для ввода числовых значений концов отрезка A и B;

текстовое поле TextBox3 для ввода точности вычислений P;

надпись Label1 для вывода значений корня;

четыре надписи для вывода обозначений;

кнопку Button2 для запуска обработчика событий

Ввести программный код, позволяющий вычислить корень уравнения методом половинного деления с использованием цикла с постусловием, который будет выполняться, пока выполняется условие $(B-A)/2 > P$:

‘численное решение уравнения

```
Dim A, B, C, P As Single
```

```
Private Sub Button2_Click(...)
```

```
A=Val (TextBox1.Text)
```

```
B=Val (TextBox2.Text)
```

```
P=Val (TextBox3.Text)
```

```
Do
```

```
C=(A+B)/2
```

```
If (A^3-Math.Cos(A) ) * (C^3-Math.Cos(C) ) <0
```

```
Then
```

```
B=C  
Else  
A=C  
End If  
Loop While (B-A)/2>P  
Label1.Text=(A+B)/2  
End Sub
```

Из графика функции видно, что корень находится на отрезке $[0,5;1]$. Введем в текстовые поля значения концов числового отрезка, а также точность вычислений (например, 0,0001).

На надпись будет выведено значение корня: $x \approx 0,8654175$

Точность вычисления корня зависит не только от параметров используемого численного метода, но и от типа переменной. В нашем случае имеет смысл говорить о математической точности результата, которая не может превышать точность числового метода, т.е. $x \approx 0,8654$.