БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет НиДО

Специальность ИиТП

Контрольная работа № 1

по дисциплине «Методы численного анализа»

Выполнил студент: Дегтярев А.А.

группа 393551

Зачетная книжка № 902021-26

Минск 2016

**Вариант 6**

**6.** На депозитный счет банка изначально положена сумма A = 6000 рублей. Годовой банковский процент составляет p = 8.5 %. Требуется определить средства на счете после t = 3 лет.

Задачу решить аналитически и с помощью Matlab

**Решение**

Если предположить что начисление по процентам происходят каждый год, тогда решение сводится к следующей формуле

**Matlab**

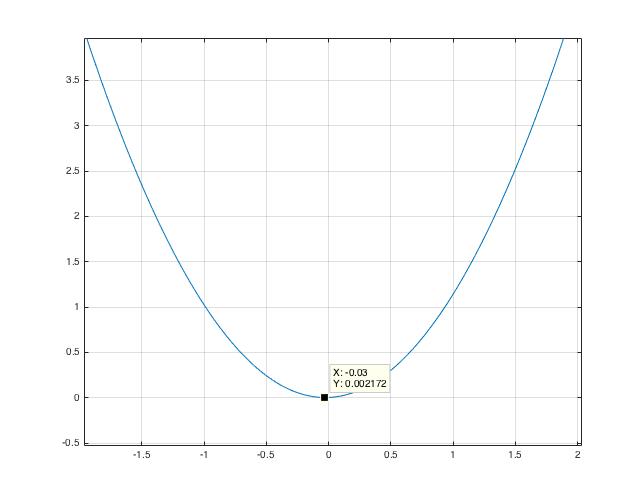
|  |
| --- |
| function KR1\_6  a = 6000  p = 0.085  t = 3  disp(a\*(1+p)^3)  end  Output: 7.6637e+03 |

**16.** Отделить графически и найти методом итераций действительные корни уравнения ах2+bx+c=0 c пятью верными знаками. Задачу решить аналитически и в системе Matlab

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | d |
| 1,08 | 0,06 | 0,003 | 1,952 |

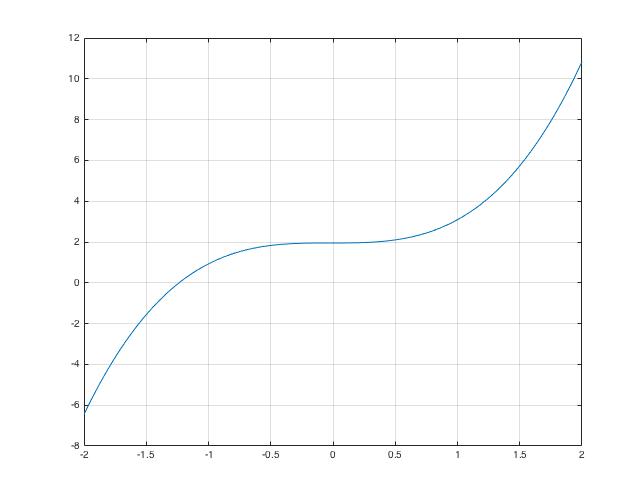
**Решение**

Построив приблизительный график функции убедился в том, что действительных корней уравнения нет:  
  
f(x)>0



**Matlab**

|  |
| --- |
| function KR1\_16  syms x  a = 1.08  b = 0.06  c = 0.003  d = 1.952  f = a\*x^2+b\*x+c  x = -2:0.01:2  figure  plot(x,eval(f))  return  end |

Предположительно, формулировка задачи неверная и подразумевалось уравнение вида ax^3+bx^2+cx+d=0, тогда, графически можно определить наличие одного действительного корня уравнения:  
 

Начальное приближение -1.5<x<-1  
  
Довольно быстрый способ нахождения x – Метод ньютона  
Достигнем точности 0.00001 за 3 итераций:

Итерационная формула:

|  |  |
| --- | --- |
| n | x |
| 0 | -1.50000 |
| 1 | -1.28033 |
| 2 | -1.23766 |
| 3 | -1.23613 |
| 4 | -1.23613 |

**Matlab**

|  |
| --- |
| a = '1.08'  b = '0.06'  c = '0.003'  d = '1.952'  f0 = a\*x^3+b\*x^2+c\*x+d  disp(NewtonMethod(f0,-1.5,-1,0,0.000001))    function res = NewtonMethod(f0,a,b,iter,e)  syms x  fa0 = subs(f0,x,a);  fa1 = subs(diff(f0),x,a);  newa = a - fa0 / fa1  iter = iter+1  if abs(newa-a) < e  res = [a,iter]  return  else  res = NewtonMethod(f0,newa,b,iter,e)  return  end  end |

**26.** Вычислить по формуле Симпсона приближенное значение определенного интеграла с шагом .



Задачу решить аналитически и в системе Matlab.

**Matlab**

|  |
| --- |
| KR1\_26((x^3+4)^(1/2),2,12,1)  function res = KR1\_26(f0,a,b,h)  syms x  xn = a  i = 0  summ = double(subs(f0,x,a)) + double(subs(f0,x,b))  while(xn<(b-1))  xn = xn + h  i = i + 1  partialSumm = double(subs(f0,x,xn));  if(0 == mod(i,2))  summ = summ + 2 \* partialSumm;  else  summ = summ + 4 \* partialSumm;  end  end  res = summ \* h / 3  end  Output: 198.9008 |

**36.** Найти решение дифференциального уравнения у'=f(x, у) с заданным начальным условием у (х0)=у0:

у'=3х2+6у у(0)=3;

а) в виде пяти отличных от нуля членов разложения в степенной ряд;

б) методом Рунге-Кутта с шагом h=0,1 на отрезке [0; 0,5] с точностью 10-5.

Сравнить полученные результаты.

Решить задачу аналитически и в системе Matlab.

**Решение**

Начнем с метода Рунге-Кутта где

Коффициенты :

Ниже приведена таблица итераций

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | k1 | k2 | k3 | k4 | x | y |
| 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 1 | 1.8030 | 2.3477 | 2.5110 | 1.926 | 0.1 | 5.2472 |
| 2 | 3.1603 | 4.1151 | 4.4016 | 3.4394 | 0.2 | 9.1860 |
| 3 | 5.5386 | 7.2100 | 7.7114 | 6.0223 | 0.3 | 16.0866 |
| 4 | 9.7000 | 12.6227 | 13.4996 | 10.5370 | 0.4 | 28.1669 |
| 5 | 16.9751 | 22.0834 | 23.6159 | 18.4251 | 0.5 | 49.3000 |

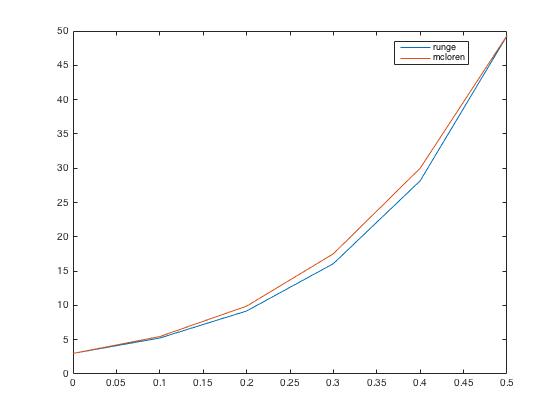
Решим задачу разложением в степенной ряд

Разложим в ряд Маклорена, первые 5 членов искомой будут иметь вид:

Таблица значений

|  |  |  |
| --- | --- | --- |
| i | x | y |
| 0 | 0 | 3 |
| 1 | 0.1 | 5.4653 |
| 2 | 0.2 | 9.8936 |
| 3 | 0.3 | 17.5273 |
| 4 | 0.4 | 30.0016 |
| 5 | 0.5 | 49.3438 |

Сравним результат:



**Matlab**

|  |
| --- |
| %  y1 = 3\*x^2+6\*y  lb = 0.0  ub = 0.5  h = 0.1  %  r = RungeMethod(y1,lb,ub,h)    y2 = 3 + 18\*x + 54\*x^2 + 109\*x^3 + 0.5\*327\*x^4  resy2=[3,double(subs(y2,x,0.1)),  double(subs(y2,x,0.2)),  double(subs(y2,x,0.3)),  double(subs(y2,x,0.4)),  double(subs(y2,x,0.5))]    plot(r.x,r.y,r.x,resy2)  return  function res = RungeMethod(f0,a,b,h)  syms x y  xk = a  yk = 3  resx = [xk]  resy = [yk]  while(xk<b)  xk = xk+h  k1 = h\*double(subs(subs(f0,x,xk),y,yk))  k2 = h\*double(subs(subs(f0,x,xk+h\*0.5),y,yk+k1/2))  k3 = h\*double(subs(subs(f0,x,xk+h\*0.5),y,yk+k2/2))  k4 = h\*double(subs(subs(f0,x,xk+h),y,yk+h\*k3))  yk = yk + (k1+2\*k2+2\*k3+k4)/6  resx = [resx,xk]  resy = [resy,yk]  end  res = struct('x',resx,'y',resy)  plot(resx,resy)  return  end |

**46.** Данные о сроке службы электроламп, соответствующие нормальному ходу технологического процесса, имеют вид:

|  |  |  |  |
| --- | --- | --- | --- |
| Срок службы ламп, час | 900-1100 | 1100-1300 | 1300-1500 |
| Количество ламп | 1000 | 6000 | 3000 |
| Контрольная выборка | 60 | 80 | 60 |

На основании контрольной выборки, объем которой приведен ниже, сделать заключение об устойчивости технологического процесса, об увеличении дисперсии или необходимости наладки оборудования.

Для решения поставленной задачи необходимо посчитать среднее выборочное значение срока службы ламп и сравнить его с теоретическим. Если выборочное среднее *m(x)* больше или совпадает с теоретическим *m(v),* то делается вывод об устойчивости технологического процесса.

**Решение**

Как видно m(x)-m(v) = 1200-1240 = -40, следовательно, **технологический процесс неустойчив.**

**56.** Методом наименьших квадратов найти эмпирическую формулу у=ах+b зависимости х и у, заданной таблицей и сделать прогноз на три шага вперед.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | 1 | 2 | 3 | 4 | 5 |
| y | 3.9 | 4.9 | 3.4 | 1.4 | 1.9 |

Как известно по методу наименьших квадратов, коэффициенты определяются системой:

Эмпирическая формула: