In[306]:=

```
(*Лабораторная работа 4
 Самута Даниил
 Группа 221703
 Вариант 8*)

(*Задание 1*)
Clear[x];
```
очистить

```
f[x_] = 12 * x^3 + 29 * x^2 - 52 * x + 11
```

Out[307]=

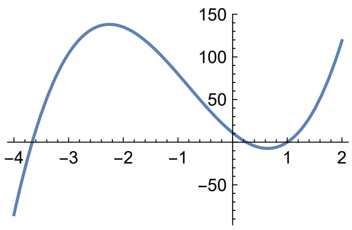
$11 - 52\,x + 29\,x^2 + 12\,x^3$

In[308]:=

```
Plot[f[x], {x, -4, 2}, AxesOrigin → {0, 0}, ImageSize → Small]
```
график функции | точка пересечения осей | размер изоб⋯ | малый

```
(*Отделяем графически корни алгебраического уравнения*)
```

Out[308]=



In[309]:=

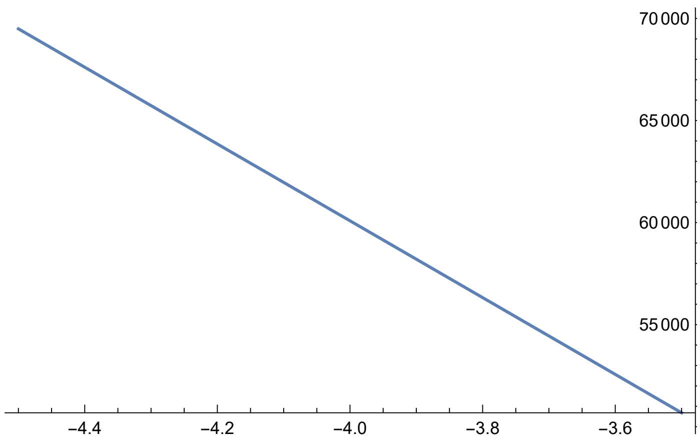
```
a = -4.5; b = -3.5; countOfItters = 1; e = 0.001; maxIters = 100;
(*Находим корень, который находится на данном отрезке*)
```

In[310]:=

In[311]:=

```
Plot[f[a] * f''[x], {x, a, b}](*f[a]*f''[x]>0, следовательно берем первую систему*)
```
график функции

Out[311]=



In[312]:=

```
x0 = b;
min = x0;
g[x_] = a - f[a]/(f[x] - f[a]) * (x - a);
sr = g[x0];
```

In[317]:=

```
While[countOfItters < maxIters,
цикл–пока
  (*Находим корень и колиество итераций с помощью метода хорд*)
    max = g[sr];
  If[(max - sr)^2/Abs[max + min - 2 * sr] < e,
  условный оператор
   Break[];];
   прервать цикл
  countOfItters++;
  min = sr;
  sr = max;
 ];
```

In[318]:=

```
countOfItters
```

Out[318]=

4

In[319]:=

In[320]:=

```
N[max] (*корень*)
численное приближени
```

Out[320]=

-3.66623

In[321]:=

```
N[(max - sr)^2/Abs[max + min - 2 * sr]] (*условие окончания итераций выполнилось, меньше e*)
численное приближение
```

Out[321]=

0.000434319

In[322]:=

In[323]:=

```
x1 = g[x0]; (*Находим уравнения двух хорд*)
f1 = f[x1];
x2 = g[x1];
f2 = f[x2];
```

In[327]:=

```
x1
```

Out[327]=

```
-3.61441
```

In[328]:=

```
f0 = f[x0]
```

Out[328]=

```
33.75
```

In[329]:=

```
f1
```

Out[329]=

```
11.1827
```

In[330]:=

```
f2
```

Out[330]=

```
3.4632
```

In[331]:=

$$A = \begin{pmatrix} x1 & 1 \\ x0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} f1 \\ f0 \end{pmatrix}$$

Out[331]=

```
{{-3.61441, 1}, {-3.5, 1}}
```

Out[332]=

```
{{11.1827}, {33.75}}
```

In[333]:=

```
solv = LinearSolve[A, B]
```

решить линейные урав

Out[333]=

```
{{197.255}, {724.141}}
```

In[334]:=

In[335]:=

```
h1[x_] = solv[[1]] * x + solv[[2]] (*1-я хорда*)
```

Out[335]=

```
{724.141 + 197.255 x}
```

In[336]:=

$$A = \begin{pmatrix} x2 & 1 \\ x0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} f2 \\ f0 \end{pmatrix}$$

Out[336]=

```
{{-3.65076, 1}, {-3.5, 1}}
```

Out[337]=

```
{{3.4632}, {33.75}}
```

In[338]:=

```
solv = LinearSolve[A, B]
```
решить линейные урав

Out[338]=

{{200.896}, {736.887}}

In[339]:=

```
h2[x_] = solv⟦1⟧ * x + solv⟦2⟧ (*2-я хорда*)
```

Out[339]=

{736.887 + 200.896 x}

In[340]:=

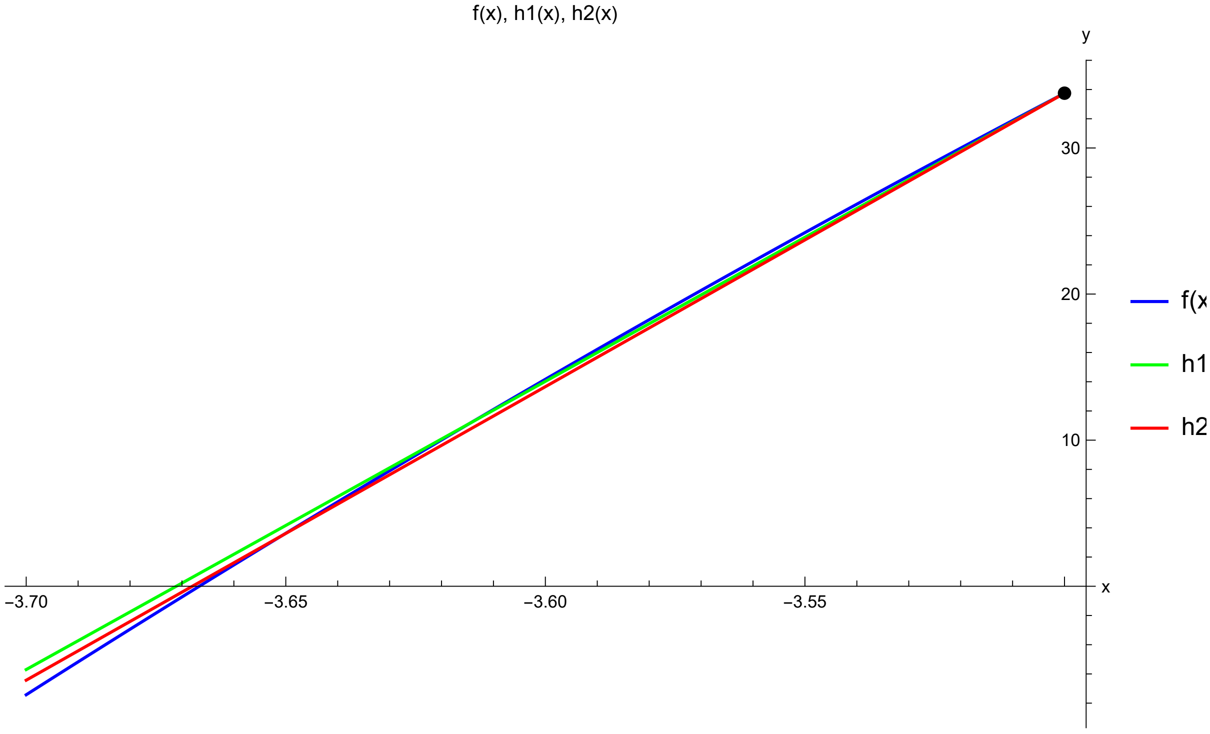
In[341]:=

```
(*График функции и хорд*)
```

In[342]:=

```
Show[Plot[f[x], {x, -3.7, -3.5}, PlotStyle → Blue, PlotLegends → {"f(x)"}],
 Plot[h1[x], {x, -3.7, -3.5}, PlotStyle → Green,
  PlotLegends → {"h1(x) - график первого приближения"}], Plot[h2[x], {x, -3.7, -3.5},
  PlotStyle → Red, PlotLegends → {"h2(x) - график второго приближеня"}],
 AxesLabel → {"x", "y"}, PlotLabel → "f(x), h1(x), h2(x)",
 ImageSize → Large, Epilog → {PointSize[Large], Point[{{x0, f0}}]}]
```
пок··· график функции · стиль графика · синий · легенды графика · график функции · стиль графика · зелёный · легенды графика · график функции · стиль графика · кр··· · легенды графика · обозначения на осях · пометка графика · размер изоб··· · круп··· · эпилог · размер то··· · крупный · точка

Out[342]=



In[343]:=

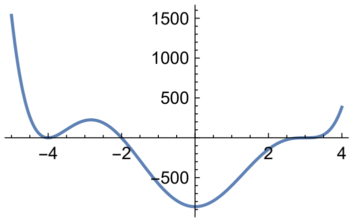
```
(*Задание 2*)
f[x_] = x^6 + x^5 - 31 * x^4 - 13 * x^3 + 306 x^2 - 864;
```

In[344]:=

```
Plot[f[x], {x, -5, 4}, AxesOrigin → {0, 0}, ImageSize → Small]
```
график функции | точка пересечения осей | размер изоб⋯ | малый

```
(*Отделяем графически корни*)
```

Out[344]=



In[345]:=

In[346]:=

```
Solve[f[x] == 0] (*Находим корни с помощью функций пакета Математика*)
```
решить уравнения

Out[346]=

{{x → -4}, {x → -4}, {x → -2}, {x → 3}, {x → 3}, {x → 3}}

In[347]:=

```
NSolve[f[x] == 0]
```
численное решение уравнений

Out[347]=

{{x → -4.}, {x → -4.}, {x → -2.}, {x → 3.}, {x → 3.}, {x → 3.}}

In[348]:=

```
Roots[f[x] == 0, x]
```
корни многочлена

Out[348]=

x == -4 || x == -4 || x == -2 || x == 3 || x == 3 || x == 3

In[349]:=

In[350]:=

```
FindRoot[f[x] == 0, {x, -3}]
```
найти корень

Out[350]=

{x → -4.}

In[351]:=

In[352]:=

```
FindRoot[f[x] == 0, {x, -1}]
```
найти корень

Out[352]=

{x → -2.}

In[353]:=

```
FindRoot[f[x] == 0, {x, 2}]
```
найти корень

Out[353]=

```
{x → 2.99999}
```

In[354]:=

```
Factor[f[x]](*Раскладываем многочлен f (x) на множители,используя функцию Factor*)
```
факторизовать ... факторизов

Out[354]=

$(-3+x)^3 (2+x) (4+x)^2$

In[355]:=

```
(*Задание 3*)
```

$f[x\_] = \sqrt{12 * x^2 + x + 3}$

Out[355]=

$\sqrt{3 + x + 12 x^2}$

In[356]:=

$g[x\_] = 2^x + 2$

Out[356]=

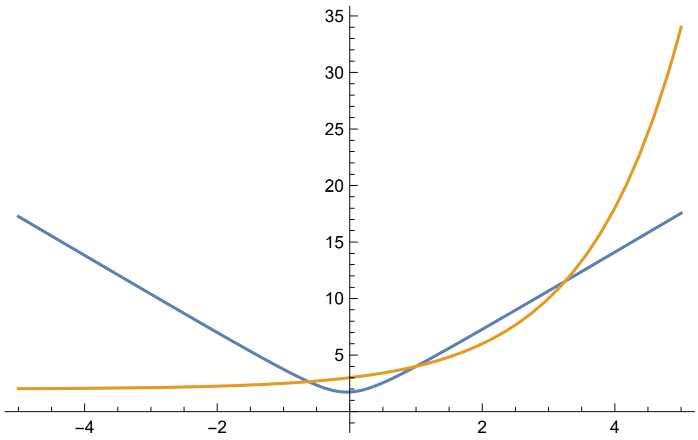
$2 + 2^x$

In[357]:=

```
Plot[{f[x], g[x]}, {x, -5, 5}]
```
график функции

```
(*Отделяем графически корни трансцендентного уравнения с помощью
функции Plot*)
```
график функции

Out[357]=



In[358]:=

```
(*Метод Ньютона*)
```

$p[x\_] = \sqrt{3 + x + 12 x^2} - 2^x - 2$

Out[358]=

$-2 - 2^x + \sqrt{3 + x + 12 x^2}$

In[359]:=

```
p1[x_] = p'[x]
```

Out[359]=

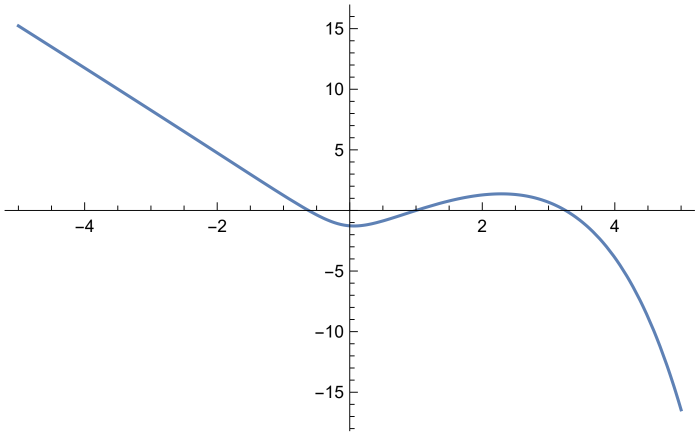
$$\frac{1+24\,x}{2\,\sqrt{3+x+12\,x^2}}-2^x\,\mathrm{Log}[2]$$

In[360]:=

```
Plot[p[x], {x, -5, 5}]
```

график функции

Out[360]=



In[361]:=

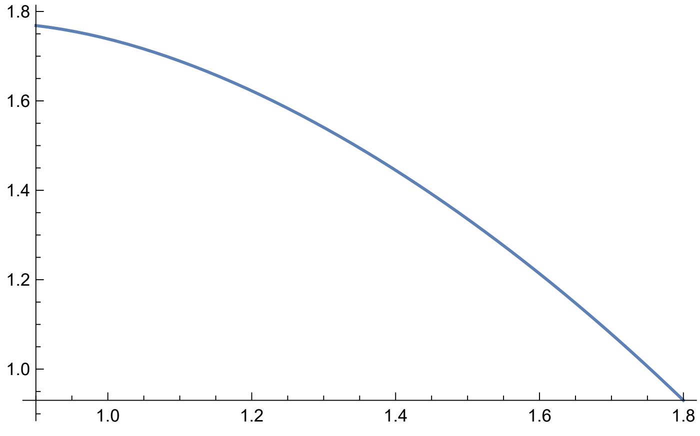
```
a = 0.9; b = 1.8;(*Проверяем условие f'(x)
и f''(x) сохраняют знак на отрезке[a,b].*)
```

In[362]:=

```
Plot[p1[x], {x, a, b}]
```

график функции

Out[362]=



In[363]:=

```
d[x_] = p''[x]
```

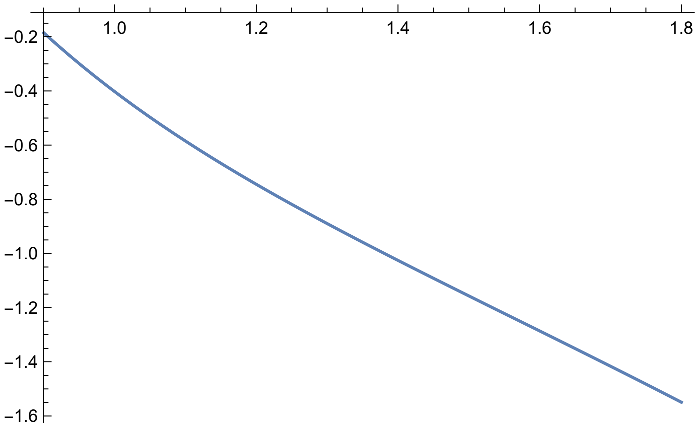
Out[363]=

$$-\frac{(1+24\,x)^2}{4\,\left(3+x+12\,x^2\right)^{3/2}}+\frac{12}{\sqrt{3+x+12\,x^2}}-2^x\,\mathrm{Log}[2]^2$$

In[364]:=

```
Plot[d[x], {x, a, b}]
```
график функции

Out[364]=



In[365]:=

```
N[p[0.9] * d[0.9]]
```
численное приближение

```
(*В качестве начального приближения x следует брать тот конец отрезка [a,b],
 для которого выполняется условие p[x]*p''[x]>0)
```

In[365]:=

```
countOfItters = 0; e = 0.001; maxIters = 100;
(*Находи корень уравнеия и количество итераций с помощью метода Ньютона*)
```

In[366]:=

```
While[countOfItters < maxIters,
  countOfItters++;
  x1 = x0 - p[x0]/p'[x0];
  If[Abs[x1 - x0] < e,
   Break[];];
  x0 = x1;
 ];
```
цикл-пока; абсолютное значение; прервать цикл

In[367]:=

```
countOfItters(*количество итераций*)
```

Out[367]=

3

In[368]:=

```
x1(*корень*)
```

Out[368]=

-0.622041

In[369]:=

In[370]:=

```
FindRoot[p[x] == 0, {x, 0.5}] (*Находим корень с помощью функции пакета Математика*)
найти корень
```

Out[370]=

```
{x → 1.}
```

In[371]:=

```
(*Метод секущих*)
```

In[372]:=

```
countOfItters = 0; e = 0.001; maxIters = 100;
(*Находим корень уравнения и количество итераций с помощью метода секущих,
начальное приближение берем из метода Ньютона*)
```

In[373]:=

```
x0 = 0.9;
x1 = 0.95
```

Out[374]=

```
0.95
```

In[375]:=

```
While[countOfItters < maxIters,
цикл-пока
  countOfItters++;
  x2 = x1 - (x1 - x0)/(p[x1] - p[x0]) * p[x1];
  If[Abs[x1 - x0] < e,
  ···  абсолютное значение
   Break[];];
   прервать цикл
  x0 = x1;
  x1 = x2;
 ];
```

In[376]:=

```
countOfItters (*Количество итераций*)
```

Out[376]=

```
3
```

In[377]:=

```
x2 (*Корень*)
```

Out[377]=

```
1.
```

In[378]:=

```
(*Задание 4*)
(*1 корень*)

a = 0.9
b = 1.8
```

Out[378]=

```
0.9
```

Out[379]=

```
1.8
```

In[380]:=

```
1.
```

Out[380]=

```
1.
```

In[381]:=

```
M = FindMaximum[p1[x], {x, a, b}]
```

найти максимум

```
(*Находим максимальное значение производной на отрезке [0.9, 1.8]*)
```

Out[381]=

```
{1.77524, {x → 0.828435}}
```

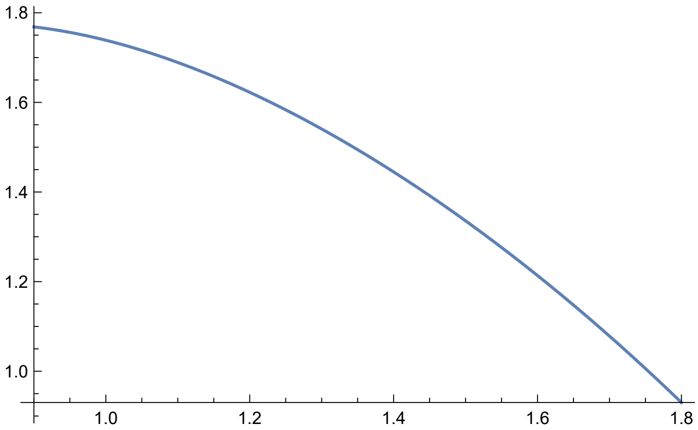
In[382]:=

```
Plot[p1[x], {x, a, b}](*Смотрим,
```

график функции

```
какой знак у производной на данном отрезке, чтобы определить знак λ*)
```

Out[382]=



In[383]:=

```
2 / M[[1]] (*|λ|<2/M – условие для λ*)
```

Out[383]=

```
1.12661
```

In[384]:=

```
λ = 1(*Берем λ меньше по модулю этого
  числа и по знаку совпадающее со знаком производной*)
```

Out[384]=

```
1
```

In[385]:=

```
φ[x_] = x - λ * p[x] (*Переходим к уравнению пригодному для итераций*)
```

Out[385]=

$2 + 2^x + x - \sqrt{3 + x + 12 x^2}$

In[386]:=

```
φ1[x_] = D[φ[x], x];(*Находим производную и смотрим, чтобы значение
```

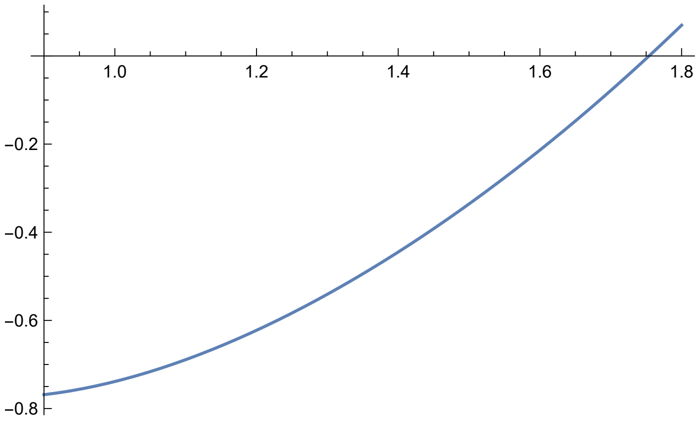
дифференцировать

```
 производной на данном отрезке в каждой точке по модулю было меньше единицы*)
```

In[387]:=

```
Plot[φ1[x], {x, a, b}]
```
график функции

Out[387]=



In[388]:=

In[389]:=

```
countOfItters = 0; e = 0.001; maxIters = 100 ; iterations = 0;(*Находим 1-
 ый корень уравнения и количество итераций с помощью метода простой итерации*)
```

In[390]:=

```
x0 = 1.35;(*x0 (начальное приближение) – любое число из отрезка [a, b]*)
```

In[391]:=

```
For[iterations, iterations < maxIters, iterations++,
  x1 = φ[x0];
  If[Abs[x1 - x0] < e,
   Break[]];
  x0 = x1;];
```
цикл ДЛЯ; абсолютное значение; прервать цикл

In[392]:=

```
x1 (*первый корень*)
```

Out[392]=

1.00037

In[393]:=

```
iterations(*потребовавшееся число итераций*)
```

Out[393]=

21

In[395]:=

$p[x\_] = \sqrt{3 + x + 12 x^2} - 2^x - 2$ (*2-ой корень*)

Out[395]=

$-2 - 2^x + \sqrt{3 + x + 12 x^2}$

In[396]:=

```
p1[x_] = p'[x]
```

Out[396]=

$$\frac{1 + 24 x}{2 \sqrt{3 + x + 12 x^2}} - 2^x \text{Log}[2]$$

In[397]:=

```
a = -1
b = 0
```

Out[397]=

-1

Out[398]=

0

In[399]:=

```
M = First@FindMaximum[{Abs[p[x]], a ≤ x ≤ b}, x]
```

первый найти максимум абсолютное значение

(*Находим максимальное значение производной на отрезке [-1, 0]*)

Out[399]=

1.26795

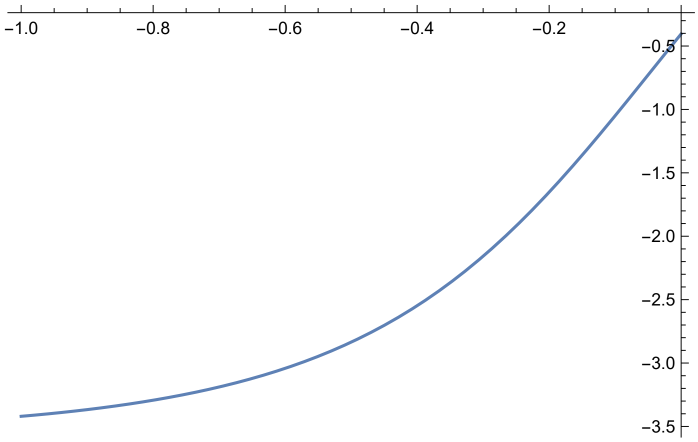
In[400]:=

```
Plot[p1[x], {x, a, b}] (*Смотрим какой знак у производной на данном отрезке,
```

график функции

чтобы определить знак λ*)

Out[400]=



In[401]:=

2 / M (*|λ|<2/M - условие для λ*)

Out[401]=

1.57735

In[402]:=

λ = -0.5 (*Берем λ меньше по модулю этого числа и по знаку совпадающее со знаком производной*)

Out[402]=

-0.5

In[403]:=

```
φ[x_] = x - λ * p[x]
(*Переходим к уравнению пригодному для итераций*)
```

Out[403]=

$$x + 0.5 \left(-2 - 2^{x} + \sqrt{3 + x + 12 x^{2}}\right)$$

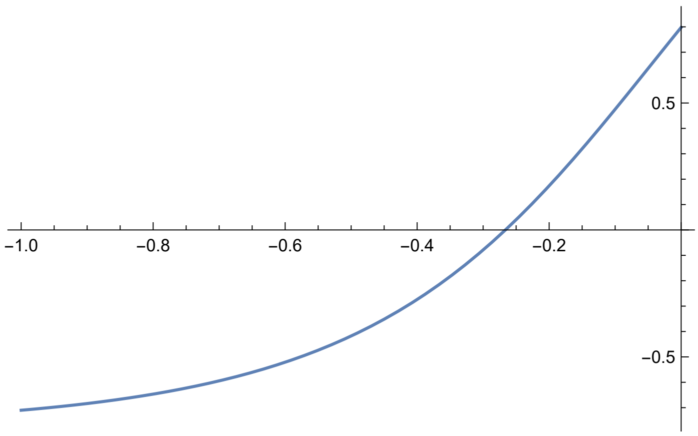
In[404]:=

```
φ1[x_] = D[φ[x], x];
         дифференциировать
(*Находим производную и смотрим, чтобы значение производной
 на данном отрезке в каждой точке по модулю было меньше единицы*)
```

In[405]:=

```
Plot[φ1[x], {x, a, b}]
график функции
```

Out[405]=



In[406]:=

```
countOfItters = 0; e = 0.001; maxIters = 100 ; iterations = 0;
(*Находим 2-
 ой корень уравнения и количество итераций с помощью метода простой итерации*)
```

In[407]:=

```
x0 = -0.5;(*x₀(начальное приближение) – любое число из отрезка [a, b]*)
```

In[408]:=

```
For[iterations, iterations < maxIters, iterations++,
цикл ДЛЯ
  x1 = φ[x0];
  If[Abs[x1 - x0] < e,
  … абсолютное значение
   Break[]];
   прервать цикл
  x0 = x1;];
```

In[409]:=

```
x1 (*второй корень*)
```

Out[409]=

-0.621808

In[410]:=

```
iterations(*потребовавшееся число итераций*)
```

Out[410]=

9

In[411]:=

```
(*Находи 3 корень*)
```

In[412]:=

```
a = 3
b = 4
```

Out[412]=

3

Out[413]=

4

In[414]:=

```
M = First@FindMaximum[{Abs[p[x]], a ≤ x ≤ b}, x]
(*Находим максимальное значение производной на отрезке [3, 4]*)
```

первый найти максимум абсолютное значение

Out[414]=

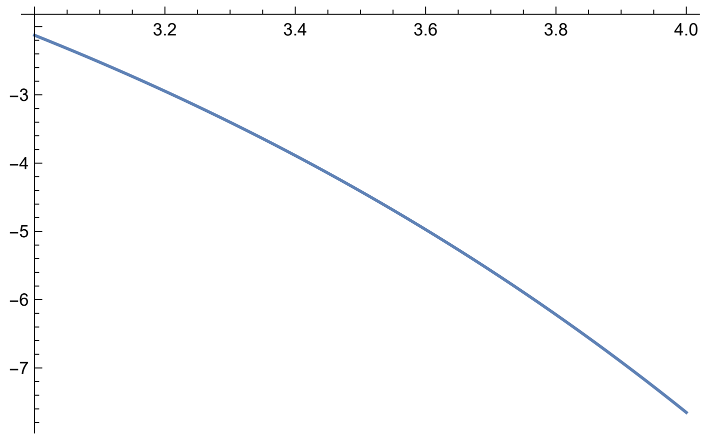
3.89326

In[415]:=

```
Plot[p1[x], {x, a, b}]
(*Смотрим, какой знак у производной на данном отрезке, чтобы определить знак λ*)
```

график функции

Out[415]=



In[416]:=

```
2 / M (*|λ|<2/M - условие для λ*)
```

Out[416]=

0.513708

In[417]:=

```
λ = -0.25
(*Берем λ меньше по модулю этого числа
 и по знаку совпадающее со знаком производной*)
```

Out[417]=

−0.25

In[418]:=

```
φ[x_] = x - λ * p[x]
(*Переходим к уравнению пригодному для итераций*)
```

Out[418]=

$x + 0.25\left(-2 - 2^{x} + \sqrt{3 + x + 12\,x^{2}}\right)$

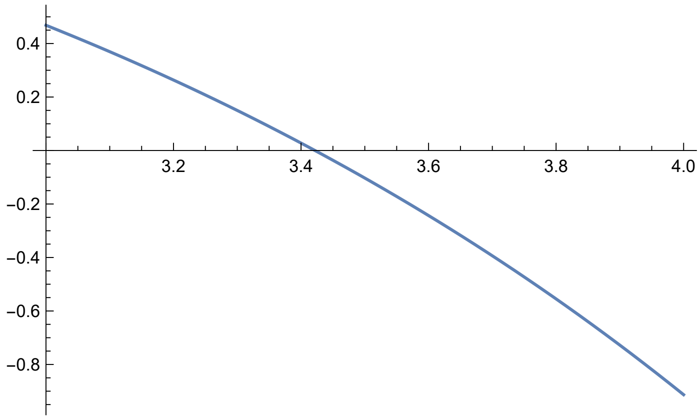
In[419]:=

```
φ1[x_] = D[φ[x], x];
         дифференцировать
(*Находим производную и смотрим, чтобы значение производной
 на данном отрезке в каждой точке по модулю было меньше единицы*)
```

In[420]:=

```
Plot[φ1[x], {x, a, b}]
график функции
```

Out[420]=



In[421]:=

```
countOfItters = 0; e = 0.001; maxIters = 100 ; iterations = 0;
(*Находим 3-
 ий корень уравнения и количество итераций с помощью метода простой итерации*)
```

In[422]:=

```
x0 = 3.5;(*x0(начальное приближение) - любое число из отрезка [a, b]*)
```

In[423]:=

```
For[iterations, iterations < maxIters, iterations++,
цикл ДЛЯ
  x1 = φ[x0];
  If[Abs[x1 - x0] < e,
  … абсолютное значение
   Break[]];
   прервать цикл
  x0 = x1;];
```

In[424]:=

```
x1 (*третий корень*)
```

Out[424]=

```
3.25604
```

In[425]:=

```
iterations(*потребовавшееся число итераций*)
```

Out[425]=

```
3
```

In[426]:=

```
(*5 задание*)
(*Решаем уравнение из задания 3 с помощью функций пакета Математика*)
Solve[p[x] == 0, Reals]
```
решить уравнения | множество действительных чисел

Out[426]=

$\{\{x \to 1\}, \{x \to -0.622\ldots\}, \{x \to 3.26\ldots\}\}$

In[427]:=

```
NSolve[p[x] == 0, Reals]
```
численное решение··· | множество дейст

Out[427]=

$\{\{x \to -0.622041\}, \{x \to 1.\}, \{x \to 3.25594\}\}$

In[428]:=

```
FindRoot[p[x] == 0, {x, -1}]
```
найти корень

Out[428]=

$\{x \to -0.622041\}$

In[429]:=

```
FindRoot[p[x] == 0, {x, 0.5}]
```
найти корень

Out[429]=

$\{x \to 1.\}$

In[430]:=

```
FindRoot[p[x] == 0, {x, 3}]
```
найти корень

Out[430]=

$\{x \to 3.25594\}$

In[431]:=

```
(*Задание 6*)

f[x_, y_] = Sinh[2 * y - 3] - 3 * x + 1
```
гиперболический синус

Out[431]=

$1 - 3x - \mathrm{Sinh}[3 - 2y]$

In[432]:=

$g[x\_, y\_] = (x^2 + y^2)^2 - 32 * (y^2 - x^2)$

Out[432]=

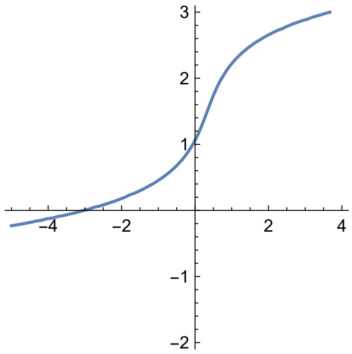
$-32(-x^2 + y^2) + (x^2 + y^2)^2$

In[433]:=

In[434]:=

```
g1 = ContourPlot[f[x, y] == 0, {x, -5, 4}, {y, -2, 3}, Axes → True,
     контурный график                                  оси    истина
   Frame → False, ImageSize → Small]
   рамка   ложь    размер изоб… малый
```

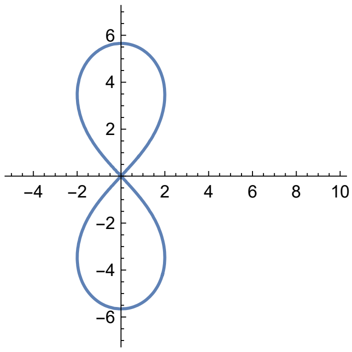
Out[434]=



In[435]:=

```
g2 = ContourPlot[g[x, y] == 0, {x, -5, 10}, {y, -7, 7}, Axes → True,
     контурный график                                   оси    истина
   Frame → False, ImageSize → Small]
   рамка   ложь    размер изоб… малый
```

Out[435]=



In[436]:=

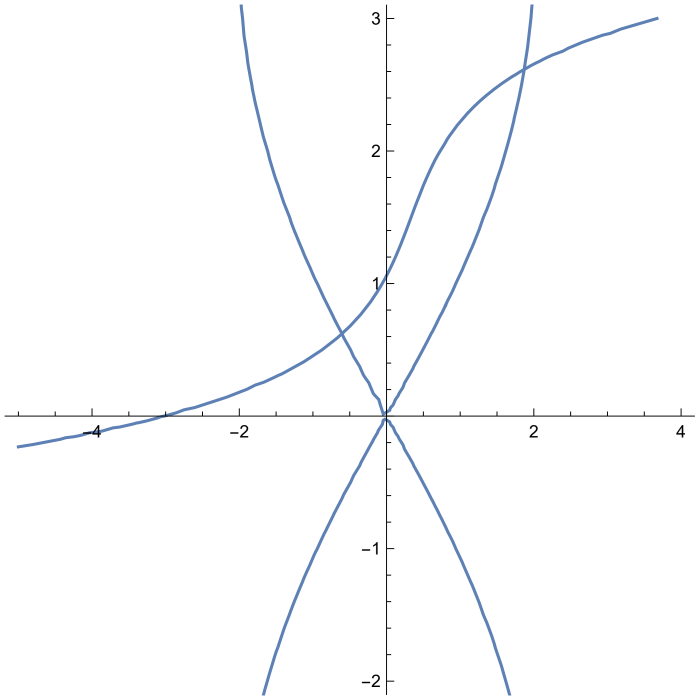
```
(*изображаем на одном чертеже кривые f (x,y)=0 и g(x,y)=0*)
```

In[437]:=

```
Show[g1, g2, ImageSize → Medium]
```

показать   размер изоб⋯ средний

Out[437]=



In[438]:=

```
(*Решаем данную систему с помощью функции пакета Математика*)
```

In[439]:=

```
FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, 2}, {y, 2.5}]
```

найти корень

Out[439]=

```
{x → 1.87109, y → 2.61681}
```

In[440]:=

```
FindRoot[{f[x, y] == 0, g[x, y] == 0}, {x, -1}, {y, 0.5}]
```

найти корень

Out[440]=

```
{x → -0.605969, y → 0.620382}
```

In[441]:=