Реализация метода локально наилучшего рациональной аппроксимации степенного ряда.(1.3.6([n\n+2]))

1. Суть метода и алгоритм решения:

Дано некоторое разложение функции f в ряд Тейлора в окрестности точки $x_0 = 0$: $f = \sum_{k=0}^{\infty} c_k x^k$

Задача построить ее локальное наилучшее рациональное приближение.

Искомым приближением $[m\backslash n]_f$ будет полином $\frac{a_0+a_1x+\cdots+a_mx^m}{b_0+b_1x+\cdots+b_nx^n} = \sum_{k=0}^{\infty} d_k x^k$, где $d_i = c_i$ для $i = 0, \ldots, m+n$. В данной задаче мне необходимо реализовать случай $m = n - 2$.

Отсюда следует, что $\sum_{k=0}^{m} a_k x^k = \sum_{k=0}^{m+n} c_k x^k * \sum_{k=0}^{n} b_k x^k$. Примем $b_0 = 1$. Это не повлияет на общность.

Для нахождения оставшихся коэффициентов b_i необходимо решить СЛАУ:

$$\begin{pmatrix} c_{m-n+1} & c_{m-n+2} & \cdots & c_{m-1} & c_m \\ c_{m-n+2} & \ddots & ⋰ & ⋰ & c_{m+1} \\ \vdots & ⋰ & \ddots & ⋰ & \vdots \\ c_{m-1} & ⋰ & ⋰ & \ddots & c_{m+n-2} \\ c_m & c_{m+1} & \cdots & c_{m+n-2} & c_{m+n-1} \end{pmatrix} \begin{pmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_2 \\ b_1 \end{pmatrix} = - \begin{pmatrix} c_{m+1} \\ c_{m+2} \\ \vdots \\ c_{m+n-1} \\ c_{m+n} \end{pmatrix}$$

Если индекс i при $c_i < 0$, то $c_i = 0$. Если i при $b_i > n$, то $b_i = 0$.

Из всех этих условий следует, что:

$$\begin{aligned} a_0 &= c_0 b_0 \\ a_1 &= c_1 b_0 + c_0 b_1 \\ &\vdots \\ a_{m-1} &= c_{m-1} b_0 + c_{m-1} b_1 + \ldots + c_0 b_{m-1} \\ a_m &= c_m b_0 + c_{m-1} b_1 + \ldots + c_0 b_m \end{aligned}$$

Особенность Паде аппроксиманта заключается в том, что

$$\begin{aligned} f(0) &= R(0), \\ f'(0) &= R'(0), \\ f''(0) &= R''(0), \\ &\vdots \\ f^{(m+n)}(0) &= R^{(m+n)}(0). \end{aligned}$$

2. Программная реализация:

Данный метод реализован с помощью Wolfram Mathematica:

Входные данные: параметр n и разложение функции в ряд.

Выходные данные: найденный аппроксимант Паде.

```
padeP[f_, n_] := Module[{m = n - 2, c, cmatrix, cvect, b, a},
  c = Table[SeriesCoefficient[f[x], {x, 0, i}], {i, 0, 2 n}];
  cmatrix = Partition[cond /@ (Table[Table[i + j, {j, 1, n}], {i, m - n + 1, m}] // Flatten), n, n];
  cvect = Transpose@{- (cond /@ Table[i + 1, {i, m + 1, m + n}])};
  b = Reverse[Flatten@ ((Inverse@cmatrix).cvect) ~Join~ {1}];
  a = Table[
    t = 0;
    Do[
     If[i - j + 1 ≤ n,
      t += c[[j + 1]] * b[[i - j + 1]]]
     , {j, 0, i}];
    t
    , {i, 0, m}];
  Total@Table[a[[j + 1]] x^j, {j, 0, m}]
  ─────────────────────────────────────
  Total@Table[b[[j + 1]] x^j, {j, 0, n}]
 ]
```

3. Сравнение точности и визуализация

Стоит отметить, что в система $Wolfram\ Mathematica$ существует внутренняя функция, строящая для функции f аппроксимант Паде в окрестности точки x_0 для заданных значений m, n

Сравним для разных функций и разных наборов n на равенство полученных полиномов.

```
And @@ (Table[padeP[f1, i] == PadeApproximant[f1[x], {x, 0, {i - 2, i}}], {i, (Range@20)[[2 ;;]]}])
```

```
And @@ (Table[padeP[f2, i] == PadeApproximant[f2[x], {x, 0, {i - 2, i}}], {i, (Range@20)[[2 ;;]]}])
```

```
And @@ (Table[padeP[f3, i] == PadeApproximant[f3[x], {x, 0, {i - 2, i}}], {i, (Range@20)[[2 ;;]]}])
```

Во всех случаях выдается результат

```
True
```

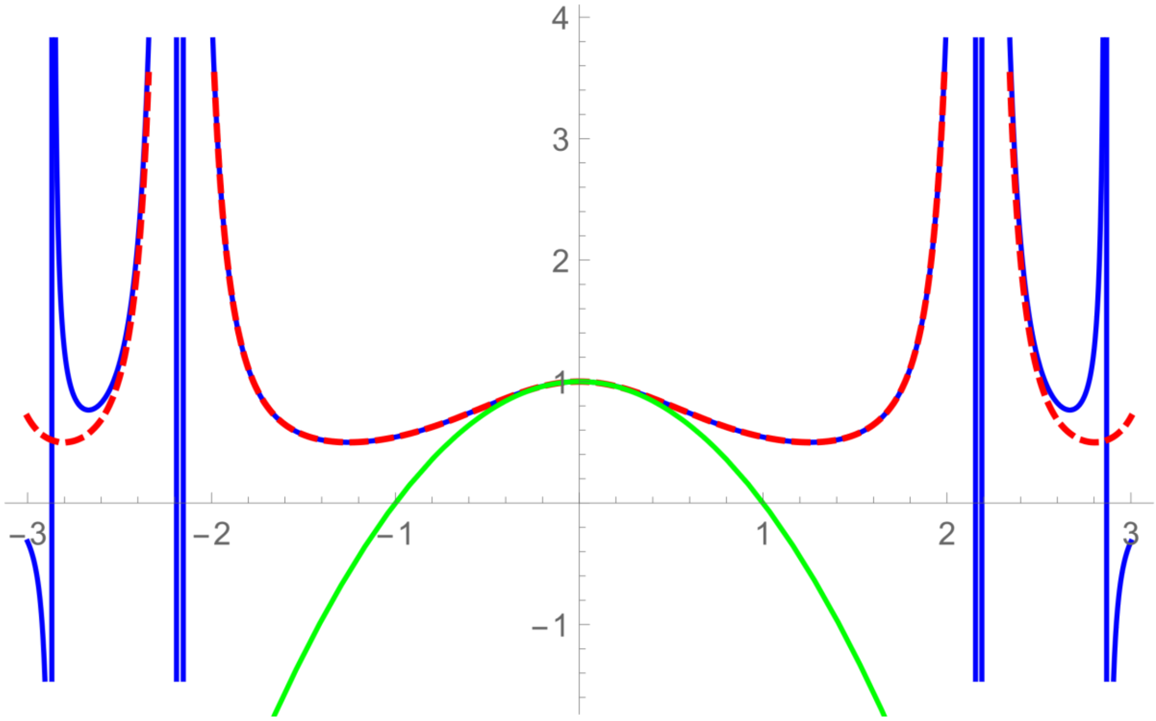
Это подтверждает правильность построения данных полиномов, при f_1, f_2, f_3 – произвольные функции.

Сравним также с помощью визуализации:

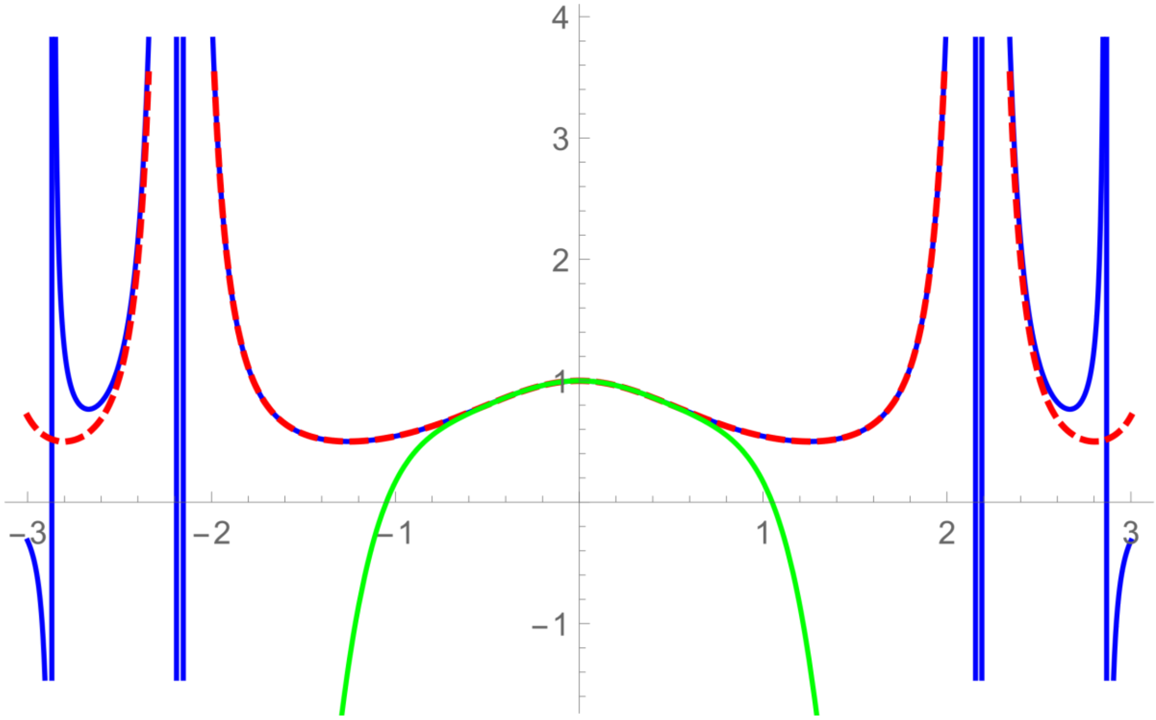
Синий график – исходная $f(x)$. Зеленый – ряд Тейлора исходной функции степени $n+m$. Красный – паде аппроксимант [n/n+2]. Будем рассматривать, как с ростом n меняются графики. Пусть $n = 2,4,\dots,10$

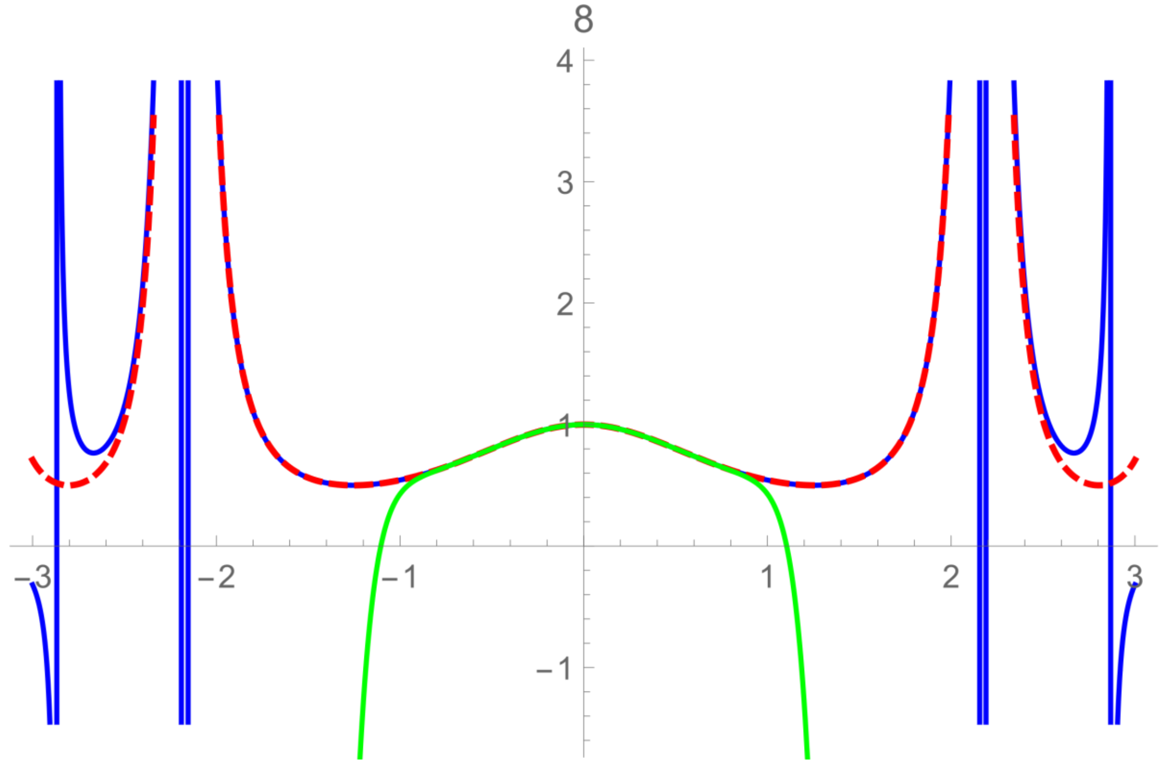
1) $f(x) = \frac{1}{1+\sin(x^2)}$

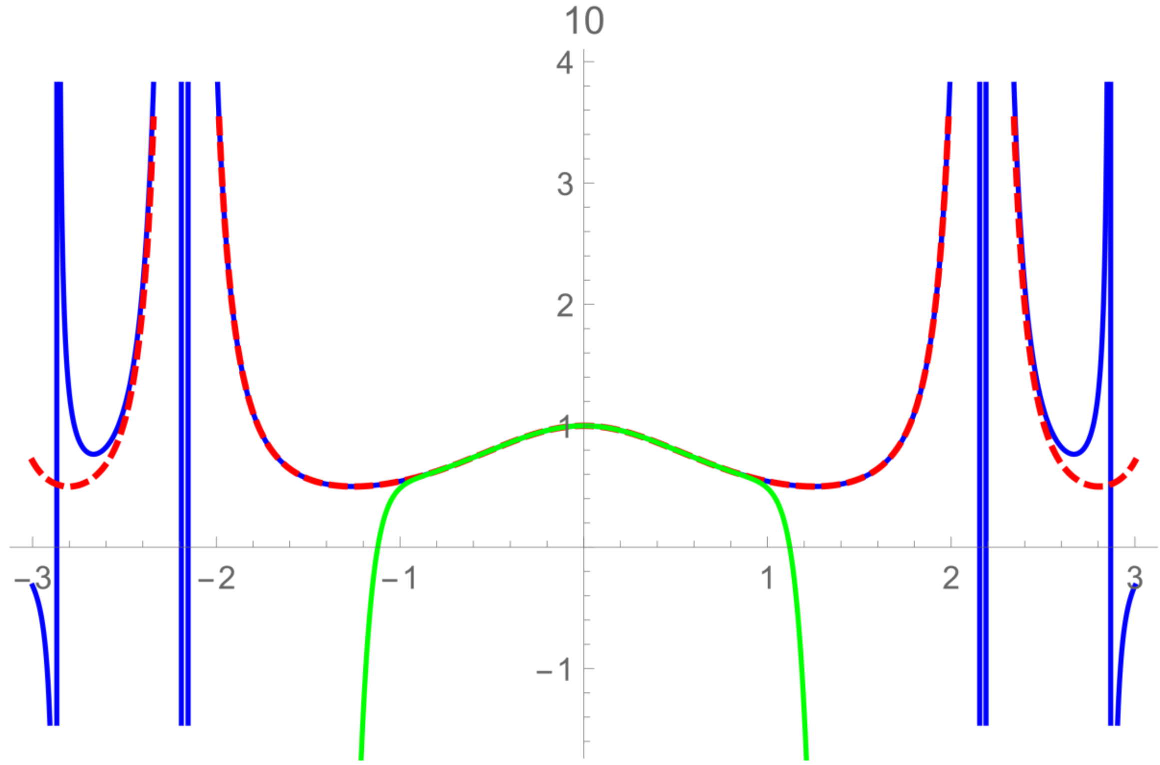
2



4



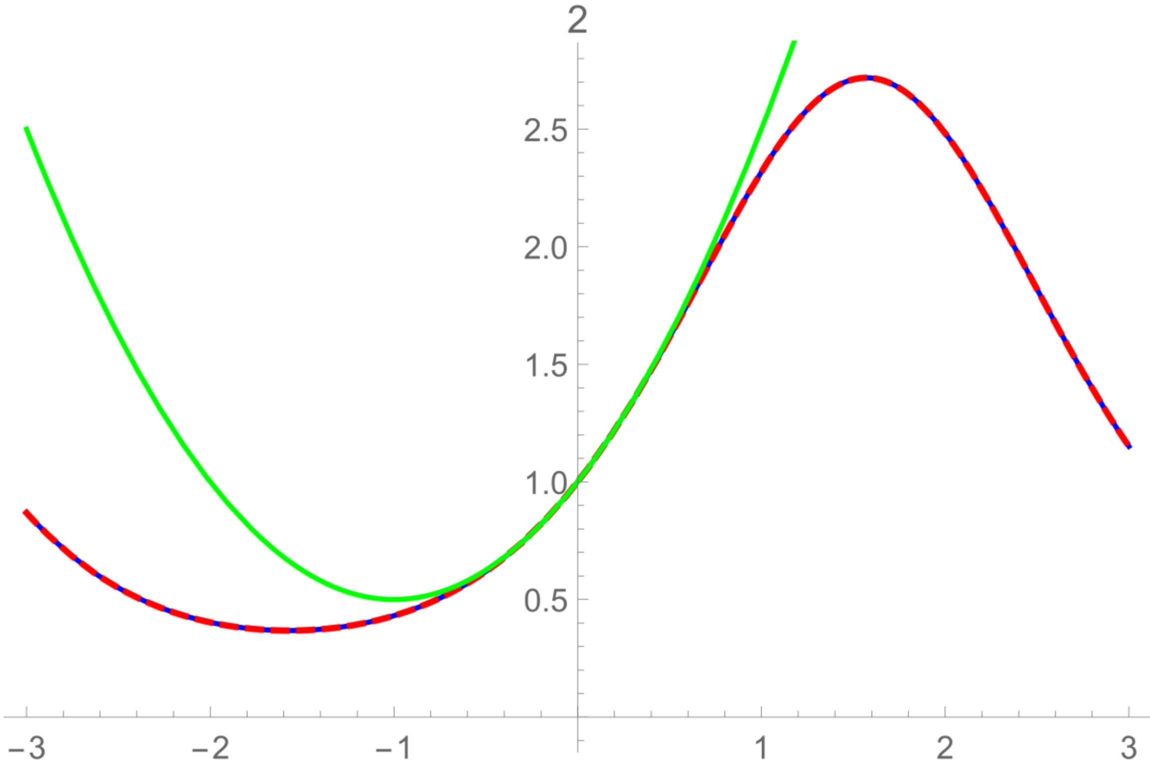


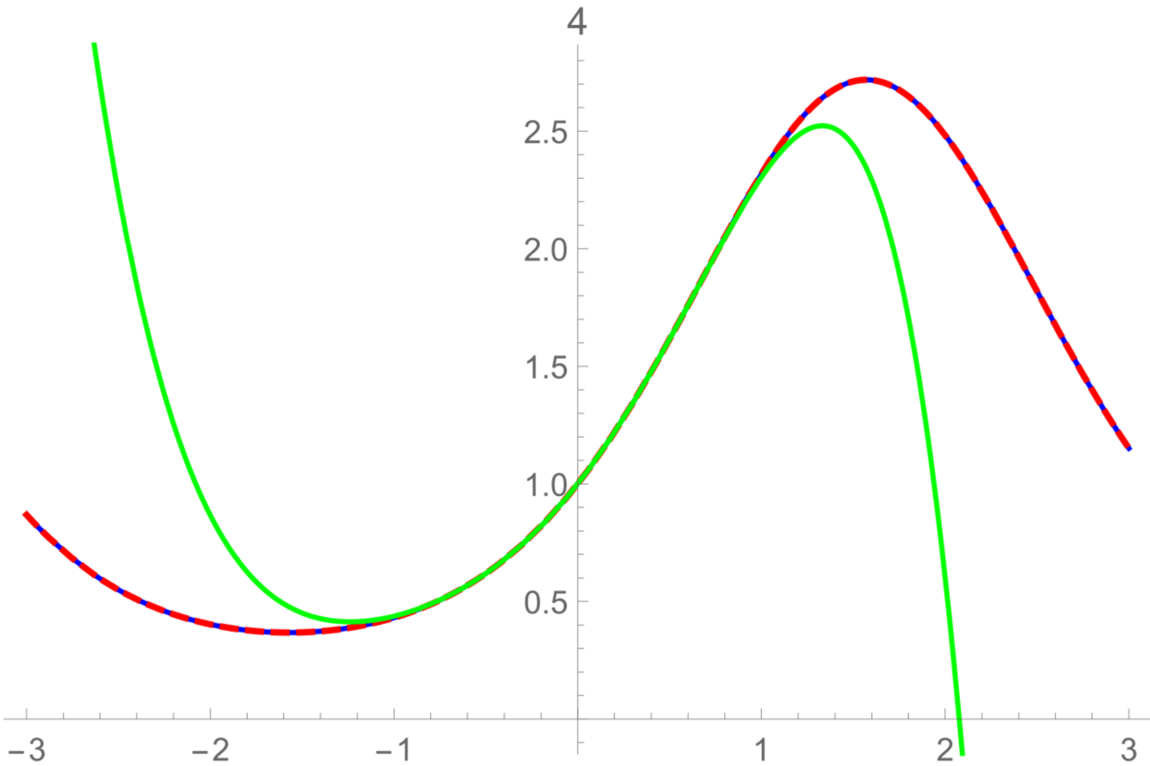


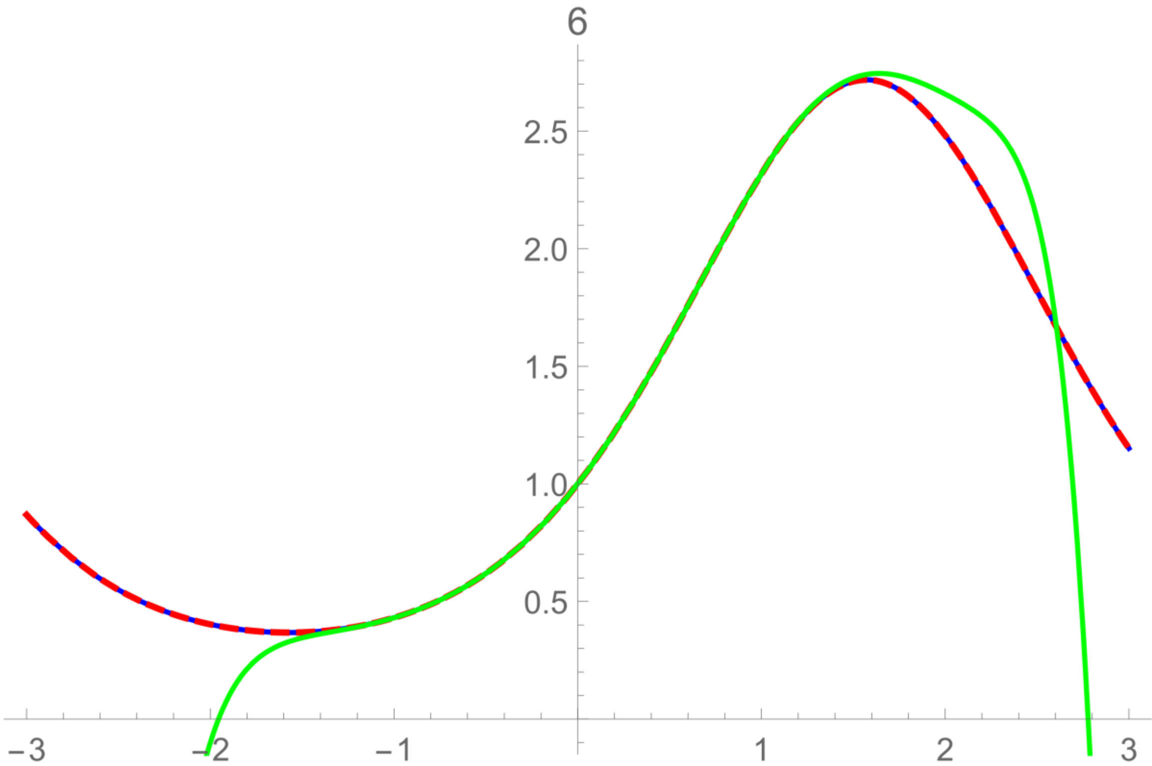
Для данного примера видно, что с ростом n качество аппроксимации локально не изменяется, достаточно $n = 2$. Однако ряд Тейлора также уступает аппроксиманту.

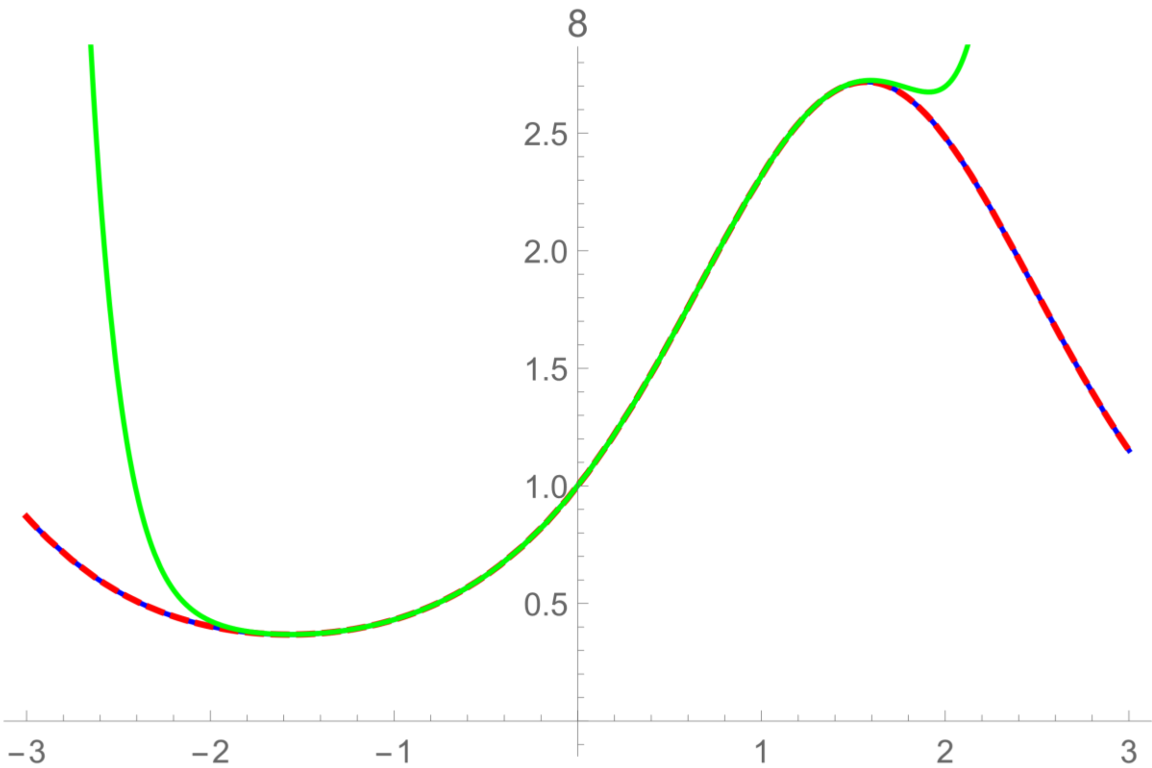
Рассмотрим более наглядный пример

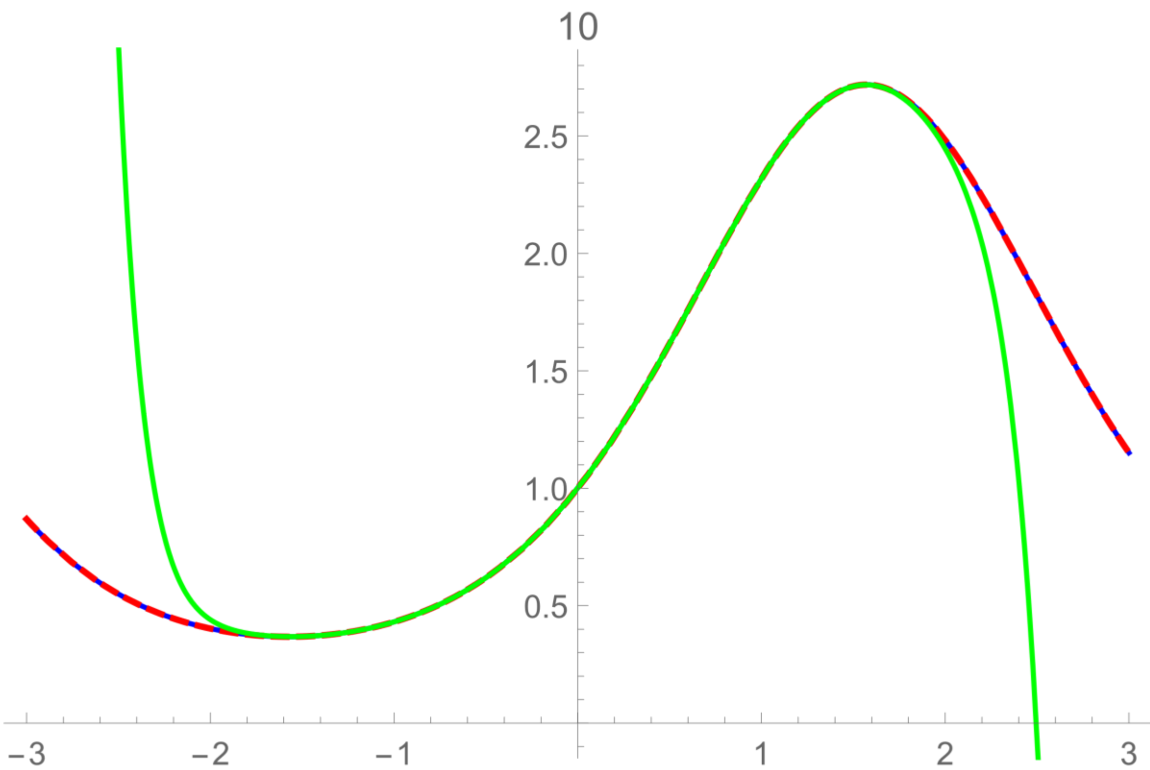
2) $f(x) = e^{\sin(x)}$





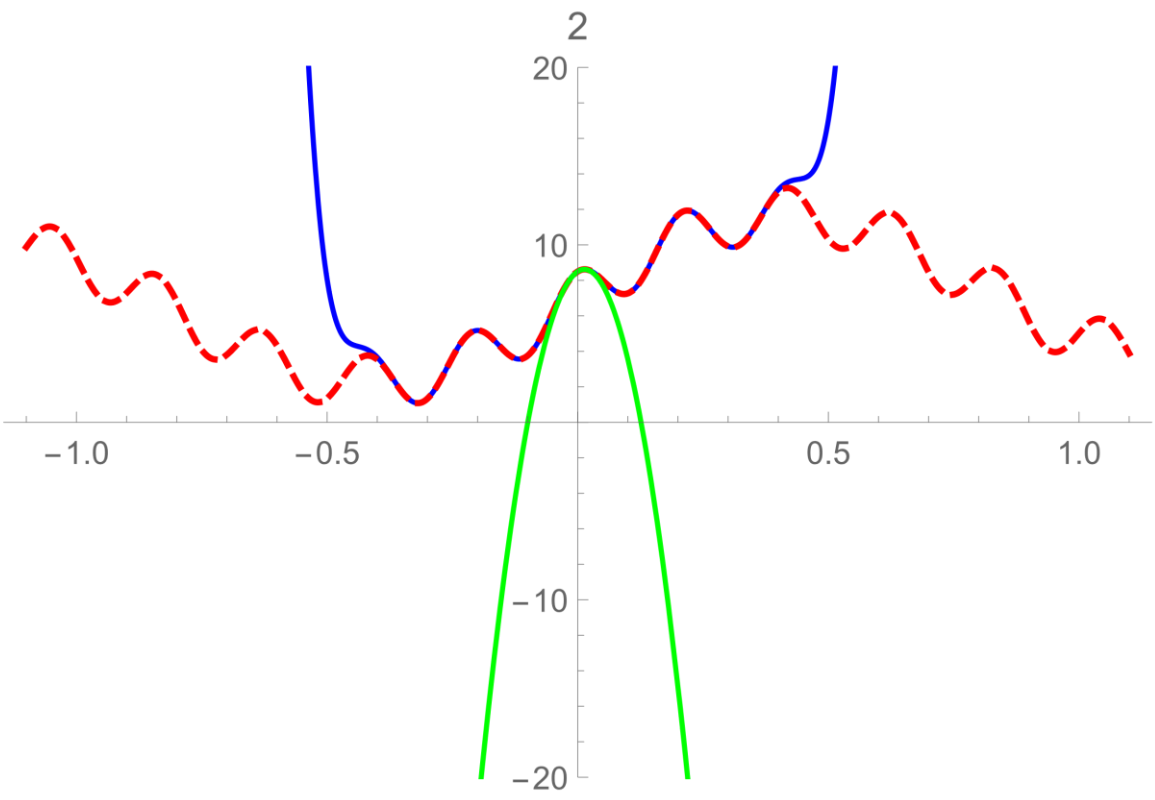

6
2.5
2.0
1.5
1.0
0.5
−3
−2
−1
1
2
3

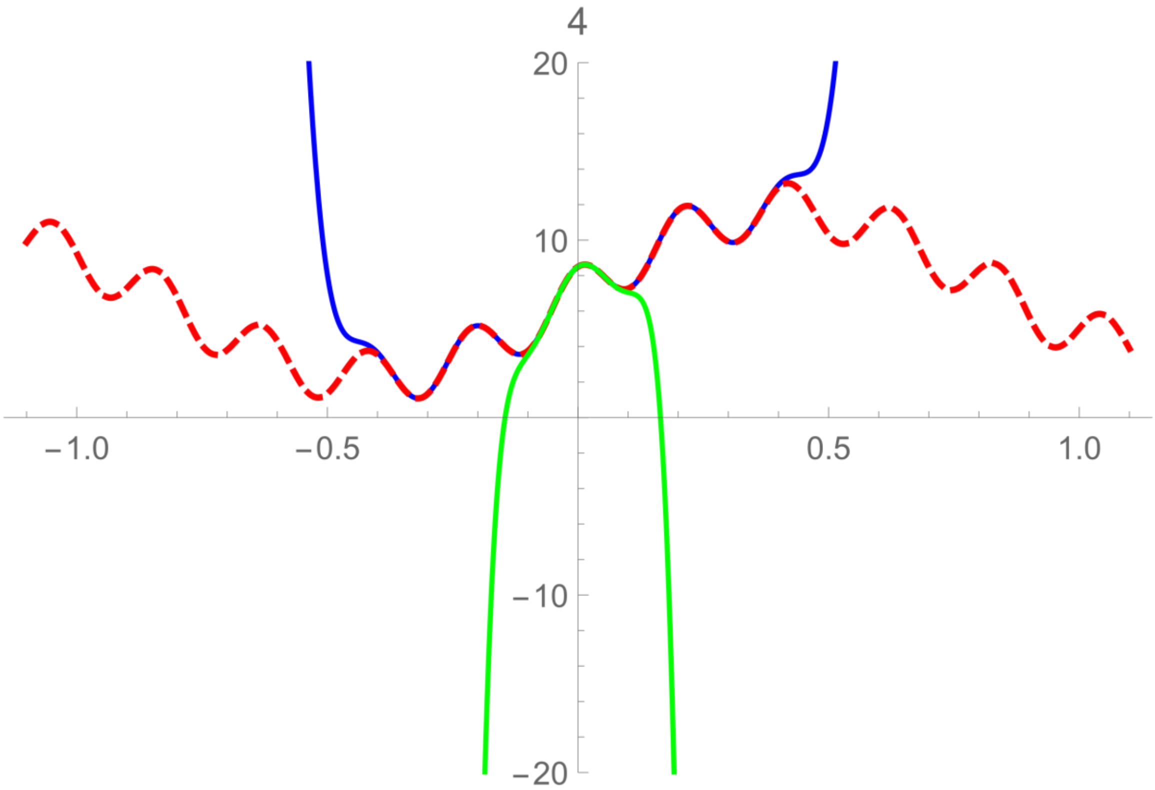

8
2.5
2.0
1.5
1.0
0.5
−3
−2
−1
1
2
3

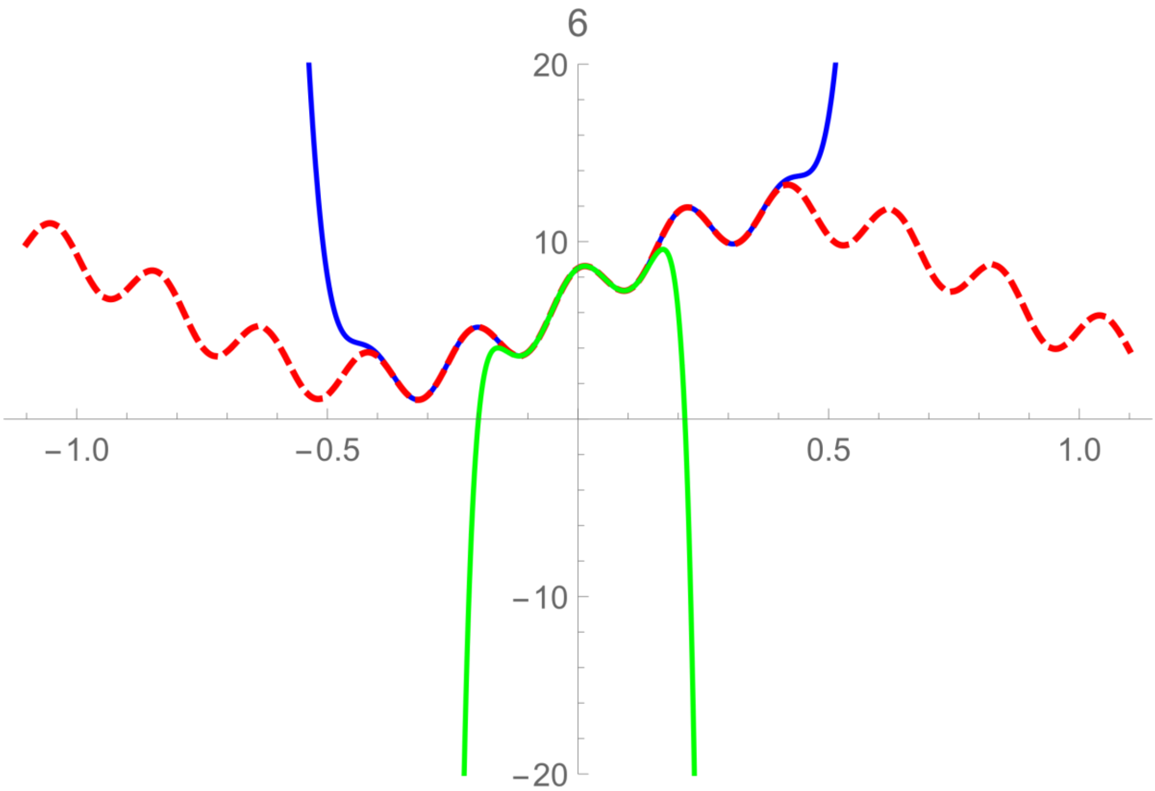


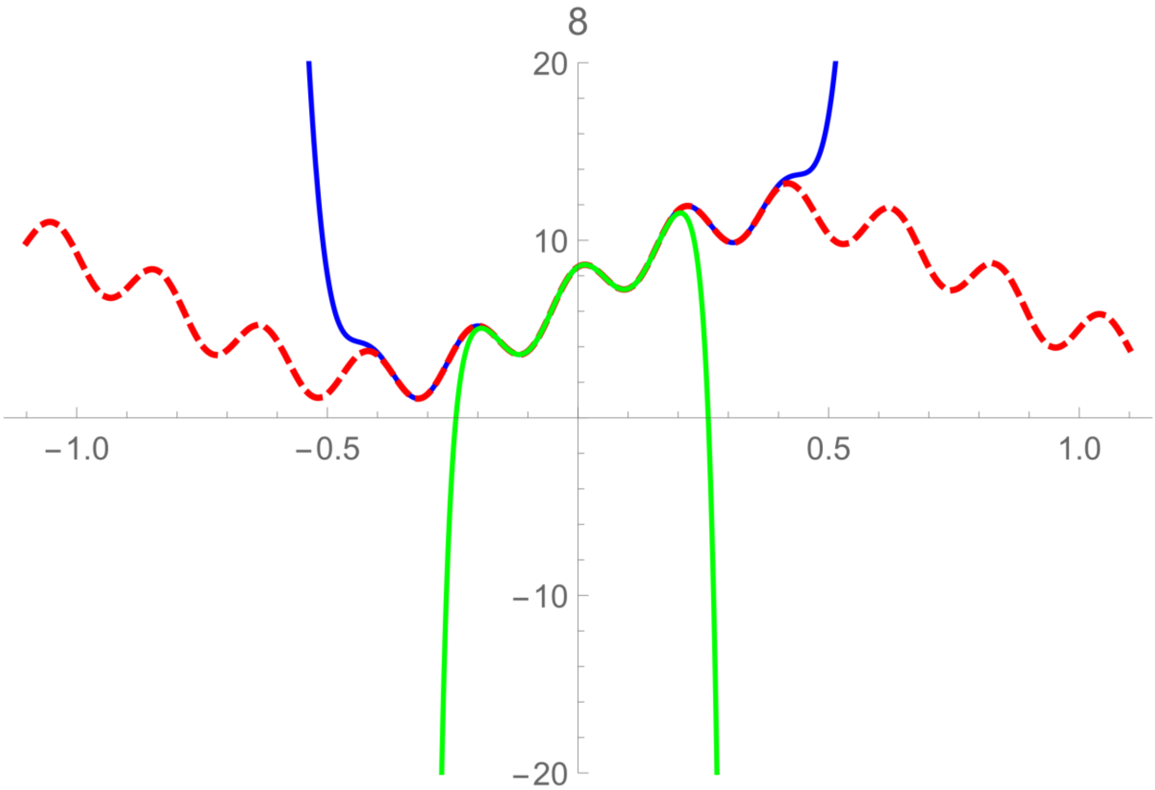
На данном примере видно, что с повышением степени ряда Тейлора, он все равно уступает аппроксиманту.

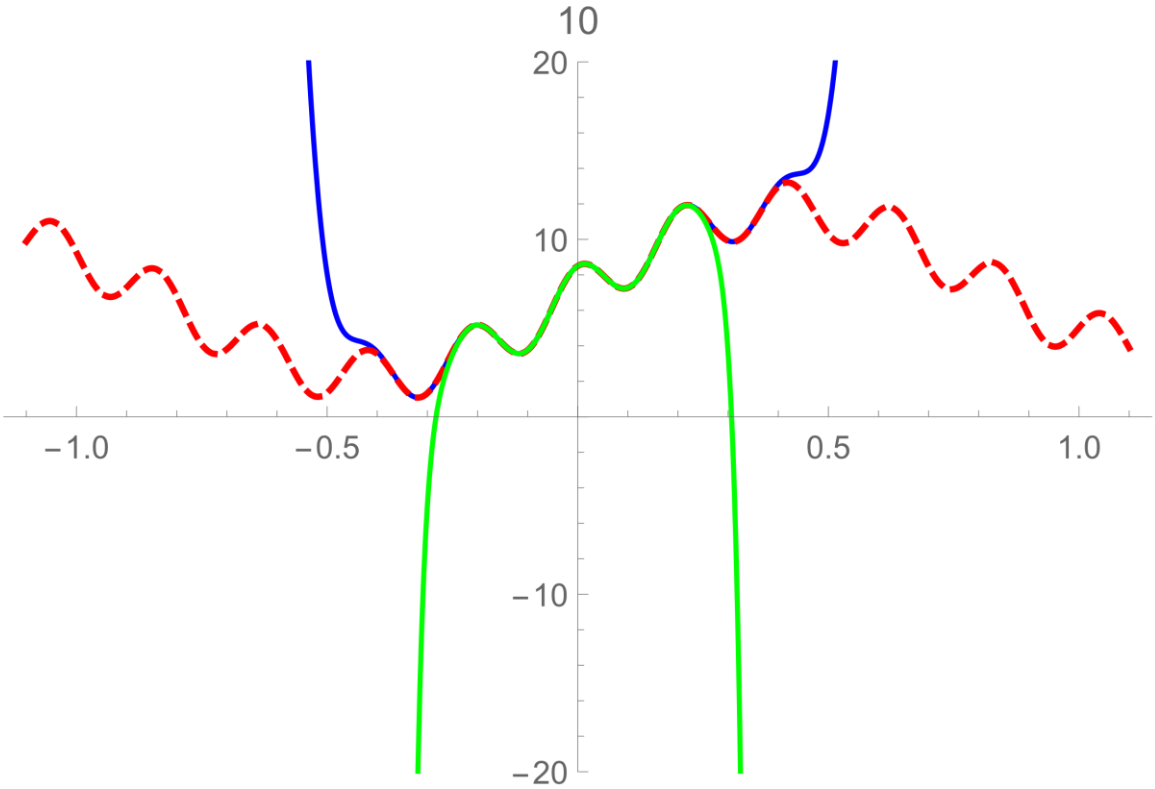
3) $f(x) = 1.5\cos(30x) + 4\sin(4x) + \sin(2x) + 7 - \frac{x^2}{12}$




4
20
10
−10
−20
−1.0
−0.5
0.5
1.0


6
20
10
−10
−20
−1.0
−0.5
0.5
1.0





По данным графикам видно, что паде аппроксимант намного лучше справляется с поставленной задачей, чем ряд Тейлора при том же n

Однако в данном случае если взять, например, $n = 20$, можно наблюдать, что ряд Тейлора побеждает, но он смог превзойти результат паде аппроксиманта лишь при довольном большом значении n, что подтвержает, что паде аппрокимант работает все равно лучше, нежели степенной ряд, поскольку паде аппроксимант достиг довольно точного локального приближения при гораздо меньших n