# Examples of approximations by series based on derivatives matching ("Taylor-like" approximations)

Andrej Liptaj*

**Abstract**

Inspired by Taylor polynomials, several other approximations based on derivative-matching are proposed.

Introductory note: This text was previously published on Scribd[1].

## 1 What can be understood as Taylor-like?

The basic feature of the Taylor (Maclaurin) series is the matching of the derivatives at a given point. Thus I define:

*Taylor-like approximation of a function $f$ in the point $x_0$ of the order $N$ is a function $TL_N^{x_0}(x)$ which fulfills*

$$\frac{d^n}{dx^n} TL_N^{x_0}(x)_{|x=x_0} = \frac{d^n}{dx^n} f(x)_{|x=x_0} \tag{1}$$

*for each $0 \leq n \leq N$.*

A set of such functions naturally builds a sequence $\left(TL_N^{x_0}\right)_{N=0}^{N=\infty}$, where one hopes for the equality $f(x) = TL_\infty^{x_0}(x)$ at some non-zero interval around $x_0$. In this text, my aim is to present new Taylor-like expansions. To my knowledge there are two such expansions commonly used: Taylor polynomials and Padè approximation (rational function).

The definition (1) is general and does not constrain much the form of the function $TL$. From practical point of view a sum is very convenient. This is of course because the derivative is a linear operator. The derivative matching in case of a product $TL_N(x) = \prod_{i=0}^N g_i(x)$ or a composite function $TL_N(x) = g_0(g_1(g_2(\dots g_N(x))))$ seems to be much harder.

Also, one wants to avoid getting $N$ unknown variables to be obtained from $N$ equations (after differentiating $N-1$ times). One rather prefers the parameters to appear progressively as higher and higher derivatives are done, so that the derivative matching is feasible and easy (at each step only one parameter is fixed). One wants the coefficients that have been previously settled to remain the same for the higher-order approximation and not to re-settle all of them. I will consider in this text only sums and all my approximations will be in the form $TL_N(x) = \sum_{i=0}^N g_i(x, a_i)$, where $a_i$ are parameters.

Let me briefly summarize the classification:

- *A*: TL approximations in the form of a sum.
- *B*: TL approximations in a different form (e.g. Padè). For this category one usually needs to solve for each $TL_N(x)$ $N+1$ equations with $N+1$ unknown variables.
    - *A1*: TL approximations in the form of a sum where coefficients come to play a role progressively with increasing derivative order. This is suitable since the coefficients from the step $N-1$ can be reused (without change) in the step $N$. In the latter, one new equation and one new parameter come into the game. In practice this often requires a single coefficient to be used for each additive term.

*Institute of Physics, Bratislava, Slovak Academy of Sciences, andrej.liptaj@savba.sk
I am willing to publish any of my ideas presented through free-publishing services in a journal, if someone (an editor) judges them interesting enough. Journals in the "*Current Contents*" database are strongly preferred.

[1] *https://www.scribd.com/document/228743662/Examples-of-approximations-by-series-based-on-derivatives-matching-Taylor-like-approximations*

– *A2*: TL approximations in the form of a sum where all coefficients play a role in all derivatives of the function $TL_N(x)$. Like in case "B", one needs in each step to solve repeatedly $N+1$ equations with $N+1$ variables, unless some smart *ad hoc* approach exists.

  ∗ *A1a*: Subset of A1, where explicit formulas for the n*th* derivative are found. Usually, unfortunately, one cannot express the expansion coefficients as explicit functions of the derivative order (finding the inverse relations is hard) but has to relay on a recursive approach to get them. In case of multiplicative coefficients (triangular matrices) this is, however, an easy and quick procedure.

  ∗ *A1b*: Subset of A1 where explicit formulas for n*th* derivative are difficult to find. This might be related to the fact that derivatives have complicated structure. An arbitrary long expansion can be constructed by doing the differentiation explicitly (supposing the related equations for coefficients can be solved).

I will present examples that fall within cases *A1a*, *A1b* and *A2*. To compare different methods I decided to use in this text four test functions: $x^2$, $e^x$, $\sin(x)$ and $\ln(x+1)$.

A section-closing remark: all this work is possible only because software tools[2] are available. Doing derivatives by hand would for sure prevent me from being interested in this topic. This brings an disadvantage, the expansions presented in Sections 3.1 and 3.2 lack proofs (might be considered as hypotheses). These proofs might be feasible using the method by induction.

# 2 Remarks about possible recipes for constructing Taylor-like series

## 2.1 Building approximations with vanishing function

Let $g$ be a function obeying $g(x_0) = 0$, with $x_0$ being the point of expansion. Then one can build a series $TL_N(x) = \sum_{i=0}^{N} a_i [g(x)]^i$. For such a series it is likely that coefficients appear progressively one-by-one after each derivative. At each step a linear equation with one unknown variable is obtained and is easily solved. Its value then remains for higher derivatives a constant, without change. This fact is easily to prove: in a term $a_i [g(x)]^i$ the constant $a_i$ remains as multiplicative factor in all successive derivatives. The expression $[g(x)]^i$ transforms as follows:

$$\begin{aligned}
[g(x)]^i &\overset{\frac{d}{dx}}{\to} ig'(x)[g(x)]^{i-1}\\
&\overset{\frac{d^2}{dx^2}}{\to} ig''(x)[g(x)]^{i-1} + i(i-1)\left[g'(x)\right]^2[g(x)]^{i-2}\\
&\overset{\frac{d^3}{dx^3}}{\to} ig'''(x)[g(x)]^{i-1} + 3i(i-1)g''(x)g'(x)[g(x)]^{i-2} + i(i-1)(i-2)\left[g'(x)\right]^3[g(x)]^{i-3}
\end{aligned}$$

From the rule on product differentiation it follows, that the lowest power in $g$ that any term of the form $P\left(g', g'', ..., g^{(N)}\right) g^i$ can reach after a single differentiation is $g^{i-1}$ ($P$ is polynomial). So the expression $[g(x)]^i$ can become non-zero only after $i$ differentiations (supposing the derivatives of $g$ behaving well, i.e. being finite). Thus, after $i$ differentiations one ends up with a finite expression multiplied by the constant $a_i$. Taking into account other terms of a lower degree together with their constants $a_0 \cdots a_{i-1}$, one sets $a_i$ such as to provide the desired value for the derivative in question. I give later some examples that follow this principle. Since this approach is very general, many different TL series can be derived in this way.

## 2.2 Building approximations with $x^i$ as a function argument

This seems to be even more efficient way of constructing TL series. The approximation takes the form

$$TL_N(x) = \sum_{i=0}^{N} a_i g(x^i)$$

[2] I would like to thank *(wx)Maxima* creators.

and is done around $x_0 = 0$. When differentiating a term from the sum one arrives to

$$\begin{aligned} g(x^i) &\overset{\frac{d}{dx}}{\rightarrow} ix^{i-1}g'(x^i) \\ &\overset{\frac{d^2}{dx^2}}{\rightarrow} i(i-1)x^{i-2}g'(x^i) + i^2x^{2i-2}g''(x^i) \\ &\overset{\frac{d^3}{dx^3}}{\rightarrow} i(i-1)(i-2)x^{i-3}g'(x^i) + 3i^2(i-1)x^{2i-3}g''(x^i) + i^3x^{3i-3}g'''(x^i) \end{aligned}$$

Thanks to the chain rule, each derivative and each term contains power of $x$ which is multiplied by the appropriate derivative of the function $g$. Since the power can be lowered at maximum by 1 at each higher derivative, the differentiated term $g(x^i)$ can become non-zero only after $i$ derivatives, or later. This guarantees that the coefficients appear progressively. The approximation is more "efficient" than the one from the previous section, because the powers of $x$ appear any time $g$ undergoes a differentiation making vanish the given term for $x_0 = 0$. And this is true for many terms. The number of terms entering the derivative matching is usually quite smaller than in the previous case and therefore the derivative matching much quicker. In spite that, in cases I studied (most common function), I did not observe the series coefficients to be over-constrained by the derivatives, i.e. the series not flexible enough to match the derivatives. I give an example of this type of series later.

### 2.3 Building approximations with repetitively integrable function

Imagine a function $g$ which can be repeatedly integrated within elementary functions. One can then construct a TL series by integration. Each term represents a higher integration and is set to zero at $x_0$ by choosing an appropriate integration constant. This hides a danger: there are high chances that within each term a polynomial resulting from the constant integration is built.

Let $g$ be non-zero at $x_0$ (so that $a_0 g(x_0)$ matches the value of the function one wants to approximate). Then one has:

$$TL_0(x) = a_0 g(x) \, .$$

Next I define the primitive function of $g$, noted $g_{I1}$, that fulfills

$$g_{I1}(x_0) = 0.$$

The series becomes

$$TL_1(x) = a_0 g(x) + a_1 g_{I1}(x)$$

The constant $a_1$ is chosen in the way the expression, after one derivative, matches the derivative of the approximated function. The next step is analogical. I note $g_{I2}$ the primitive function of $g_{I1}$ that vanishes at $x = x_0$. One arrives to

$$TL_2(x) = a_0 g(x) + a_1 g_{I1}(x) + a_2 g_{I2}(x) \, .$$

Obviously the value of the expression is given by $a_0$, the derivative by $a_0$ and $a_1$ and the second derivative by $a_0$, $a_1$ and $a_2$. The generalization is straightforward:

$$TL_N(x) = \sum_{i=0}^{i=N} a_i g_{Ii}(x) \, .$$

Although the approach works well from the theoretical point of view, I failed to find an example without pile-upping polynomials. With polynomials one easily adjusts "any derivatives anywhere" and so I have to admit this approach might lack elegance.

The nice feature of this approach comes from the fact, that one does not need to differentiate each term when doing derivative matching. Its enough to $N$ times differentiate the first one, by construction the others follow it in next steps. The matrix of the linear problem looks like $(N + 1 = 4)$

$$\begin{pmatrix} f \\ f' \\ f'' \\ f''' \end{pmatrix} = \begin{pmatrix} d_0 & 0 & 0 & 0 \\ d_1 & d_0 & 0 & 0 \\ d_2 & d_1 & d_0 & 0 \\ d_3 & d_2 & d_1 & d_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

with the obvious notation $d_i = \frac{d^i}{dx^i} g(x)_{|x=x_0}$.

The solution can be written using an analogical matrix

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} w_0 & 0 & 0 & 0 \\ w_1 & w_0 & 0 & 0 \\ w_2 & w_1 & w_0 & 0 \\ w_3 & w_2 & w_1 & w_0 \end{pmatrix} \begin{pmatrix} f \\ f' \\ f'' \\ f''' \end{pmatrix}.$$

The functions $w_i = w_i(d_0, \ldots, d_i)$ are universal and can be determined once and for all. The first 6 functions stand

$$\begin{aligned} w_0 &= \frac{1}{d_0},\ w_1 = -\frac{d_1}{d_0^2},\ w_2 = -\frac{d_0 d_2 - d_1^2}{d_0^3},\ w_3 = -\frac{d_0^2 d_3 - 2d_0 d_1 d_2 + d_1^3}{d_0^4}, \\ w_4 &= -\frac{d_0^3 d_4 - 2d_0^2 d_1 d_3 - d_0^2 d_2^2 + 3d_0 d_1^2 d_2 - d_1^4}{d_0^5}, \\ w_5 &= -\frac{d_0^4 d_5 - 2d_0^3 d_1 d_4 + (3d_0^2 d_1^2 - 2d_0^3 d_2) d3 + 3d_0^2 d_1 d_2^2 - 4d_0 d_1^3 d_2 + d_1^5}{d_0^6}. \end{aligned}$$

The matrix actually falls into the category named "(triangular) Toeplitz matrix" and an appropriate algorithm can be used to invert it (see [1]).

# 3 Examples of Taylor-like series

The only examples that fit the definition *A1a* from Section 1 I was able to find are (single) function-weighted polynomials, not following any of the recipes from Section 2. I present them in the two first subsections. Next, in Sections 3.3 and 3.4, I give two examples of the type *A1b* which follow the instructions from Sections 2.1 and 2.2. The fifth example falls into the category *A*2. Here the matrix of the linear problem is not triangular but is easy to build. The last example demonstrates the integral approach from Section 2.3.

## 3.1 Example one

$$TL_N(x) = \sum_{i=0}^{N} a_i x^i \exp(x)$$

The derivatives at $x_0 = 0$ are

| derivative order | derivative value |
|---|---|
| 0 | $a_0$ |
| 1 | $a_0 + a_1$ |
| 2 | $a_0 + 2a_1 + 2a_2$ |
| 3 | $a_0 + 3a_1 + 6a_2 + 6a_3$ |
| 4 | $a_0 + 4a_1 + 12a_2 + 24a_3 + 24a_4$ |
| 5 | $a_0 + 5a_1 + 20a_2 + 60a_3 + 120a_4 + 120a_5$ |
| 6 | $a_0 + 6a_1 + 30a_2 + 120a_3 + 360a_4 + 720a_5 + 720a_6$ |

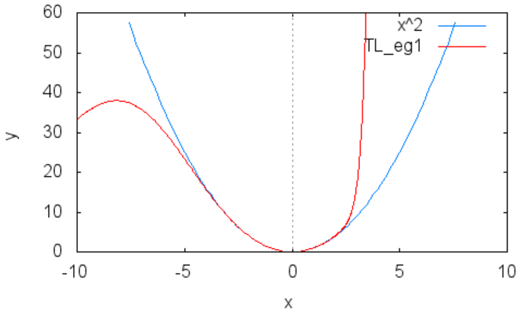
The pattern is ($N > n$):

$$\frac{d^n}{dx^n}\left[\sum_{i=0}^{N} a_i x^i \exp(x)\right]_{|x=0} = \sum_{i=0}^{n} \frac{n!}{(n-i)!} a_i.$$

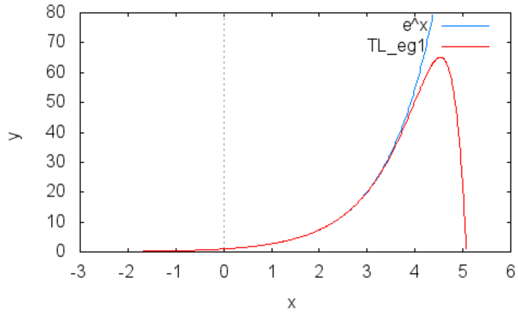
The recursive formula for coefficients stands

$$\begin{aligned} a_0 &= f^{(0)}, \\ a_n &= \frac{1}{n!}\left\{ f^{(n)} - \sum_{i=0}^{n-1} \frac{n!}{(n-i)!} a_i \right\}. \end{aligned}$$

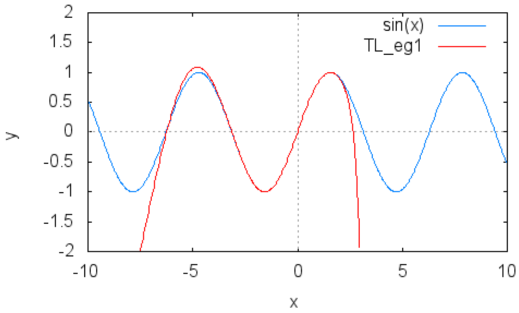
In Figure 1 I depicted the approximations of the four test functions by this series with 10 derivatives (+1 value) matched.



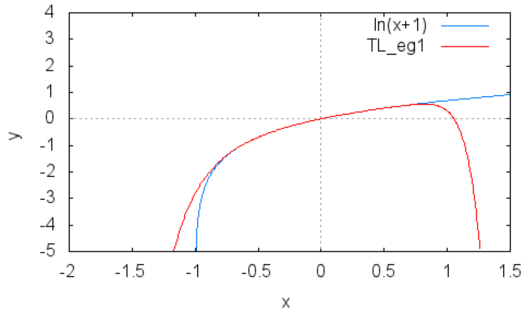

(a) $f(x) = x^2$




(b) $f(x) = e^x$




(c) $f(x) = \sin(x)$




(d) $f(x) = \ln(x+1)$

Figure 1: Approximations of four test functions with $TL_{10}(x) = \exp(x) \sum_{i=0}^{10} a_i x^i$ (example one).

### 3.2 Example two

$$TL_N(x) = \sum_{i=0}^{N} a_i x^i \left[\sin(x) + \cos(x)\right]$$

The derivatives at $x_0 = 0$ are

| derivative order | derivative value |
|---|---|
| 0 | $+a_0$ |
| 1 | $+a_0 + a_1$ |
| 2 | $-a_0 + 2a_1 + 2a_2$ |
| 3 | $-a_0 - 3a_1 + 6a_2 + 6a_3$ |
| 4 | $+a_0 - 4a_1 - 12a_2 + 24a_3 + 24a_4$ |
| 5 | $+a_0 + 5a_1 - 20a_2 - 60a_3 + 120a_4 + 120a_5$ |
| 6 | $-a_0 + 6a_1 + 30a_2 - 120a_3 - 360a_4 + 720a_5 + 720a_6$ |

The pattern is $(N > n)$:

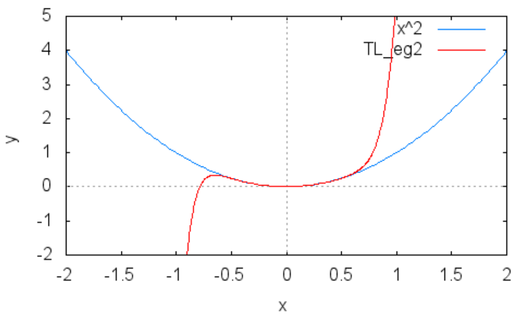
$$\frac{d^n}{dx^n}\left\{\sum_{i=0}^{N} a_i x^i \left[\sin(x) + \cos(x)\right]\right\}_{|x=0} = \sum_{i=0}^{n} c_i \frac{n!}{(n-i)!} a_i$$

with $c_i$ double alternating unity, for example[3] $c_i = \sqrt{2} \sin\left[\frac{\pi}{4} + (n - i - 1)\frac{\pi}{2}\right]$. The recursive formula for the coefficients is

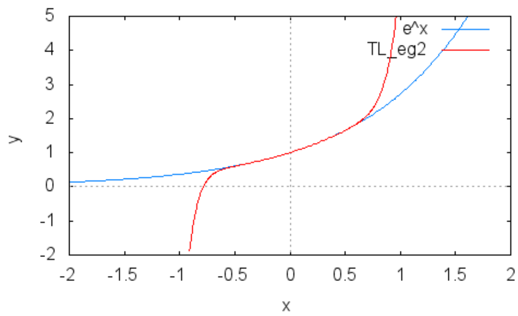
$$\begin{aligned} a_0 &= f^{(0)}, \\ a_n &= \frac{1}{n!}\left\{f^{(n)} - \sum_{i=0}^{n-1} c_i \frac{n!}{(n-i)!} a_i\right\}. \end{aligned}$$

The analogy with exponential weighted example before is hardly surprising. In Figure 2 the approximations of the four test functions by this series with 10 derivatives (+1 value) matched are shown.

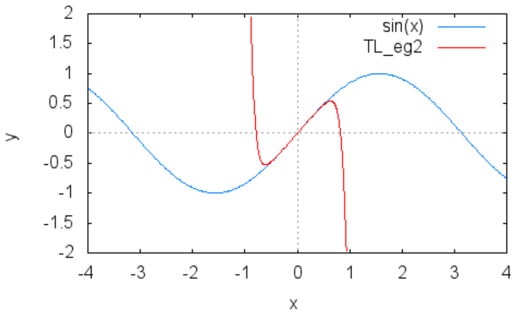
[3] I was unable to invent a more simple expression and even in this case I inspired myself by the web page http://mathhelpforum.com/calculus/180051-infinite-series-double-alternating-signs.html.



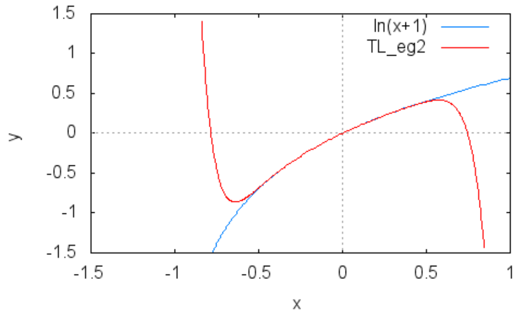

(a) $f(x) = x^2$




(b) $f(x) = e^x$




(c) $f(x) = \sin(x)$




(d) $f(x) = \ln(x+1)$

Figure 2: Approximations of four test functions with $TL_{10}(x) = [\sin(x) + \cos(x)] \sum_{i=0}^{10} a_i x^i$ (example two).

### 3.3 Example three

$$TL_N(x) = \sum_{i=0}^{N} a_i \left( \frac{x}{\sqrt{x^2+1}} \right)^i$$

I have chosen this example because the function inside the power is a "nice" well-behaved function, without singularities, bounded and bijective.

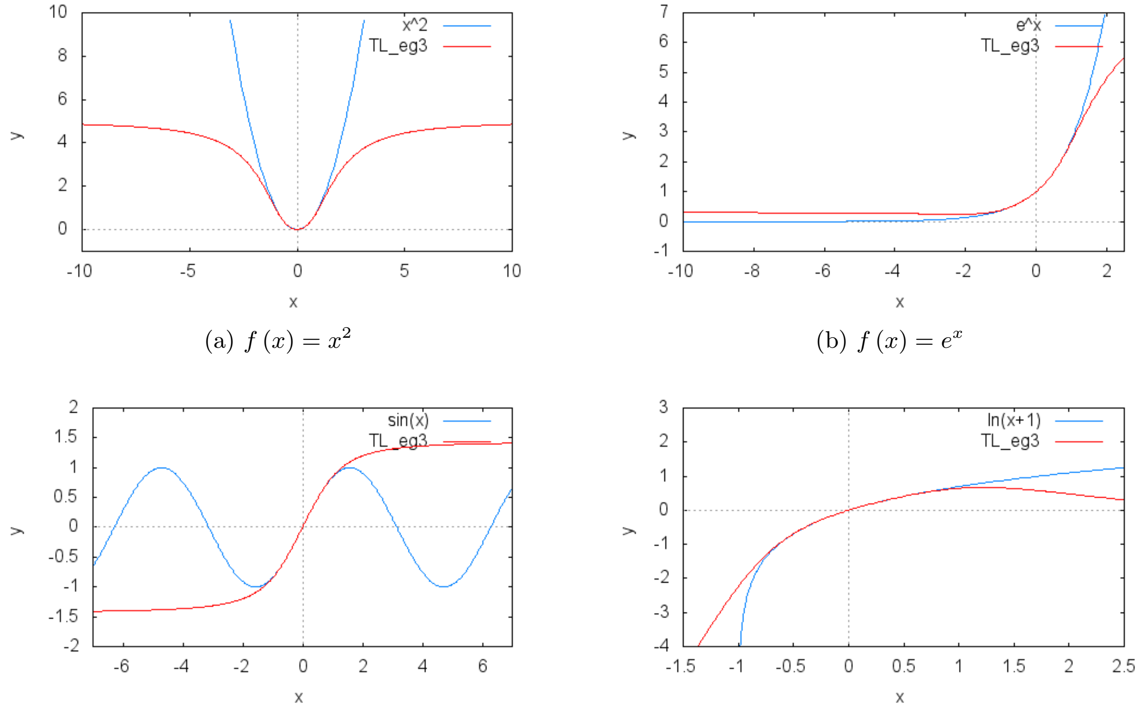
The differentiation at $x_0 = 0$ yields

| derivative order | derivative value |
|---|---|
| 0 | $a_0$ |
| 1 | $a_1$ |
| 2 | $2a_2$ |
| 3 | $6a_3 - 3a_1$ |
| 4 | $24a_4 - 24a_2$ |
| 5 | $120a_5 - 180a_3 + 45a_1$ |
| 6 | $720a_6 - 1440a_4 + 720a_2$ |
| 7 | $5040a_7 - 12600a_5 + 9450a_3 - 1575a_1$ |
| 8 | $40320a_8 - 120960a_6 + 120960a_4 - 40320a_2$ |
| 9 | $362880a_9 - 1270080a_7 + 1587600a_5 - 793800a_3 + 99225a_1$ |
| 10 | $3628800a_{10} - 14515200a_8 + 21772800a_6 - 14515200a_4 + 3628800a_2$ |

The derivatives of this function become complex and I failed to find a general pattern, although some sub-patterns can be observed[4]. Without explicit $i$-dependent formulas, one can still make use of this approximation: the coefficients appear progressively one-by-one and different terms are obtained by actually performing the differentiation explicitly. The result is represented by a less-than triangular matrix, the number of coefficients in each line grows less than the line number. The approximations of the four test functions are presented in Figure 3.

Other functions suitable for the approach from Section 2.1 (or Section 2.2) might be $\sin(x)$, $\arctan(x)$, $\exp(x-1)$ or $\operatorname{arsinh}(x)$ and may others.

[4]First term in each line increases as factorial, the second term with the line number $i$ is the first term$\times\frac{i-2}{2}$.




(a) $f(x) = x^2$

(b) $f(x) = e^x$

(c) $f(x) = \sin(x)$

(d) $f(x) = \ln(x+1)$

Figure 3: Approximations of four test functions with $TL_{10}(x) = \sum_{i=0}^{10} a_i \left(\frac{x}{\sqrt{x^2+1}}\right)^i$ (example three).

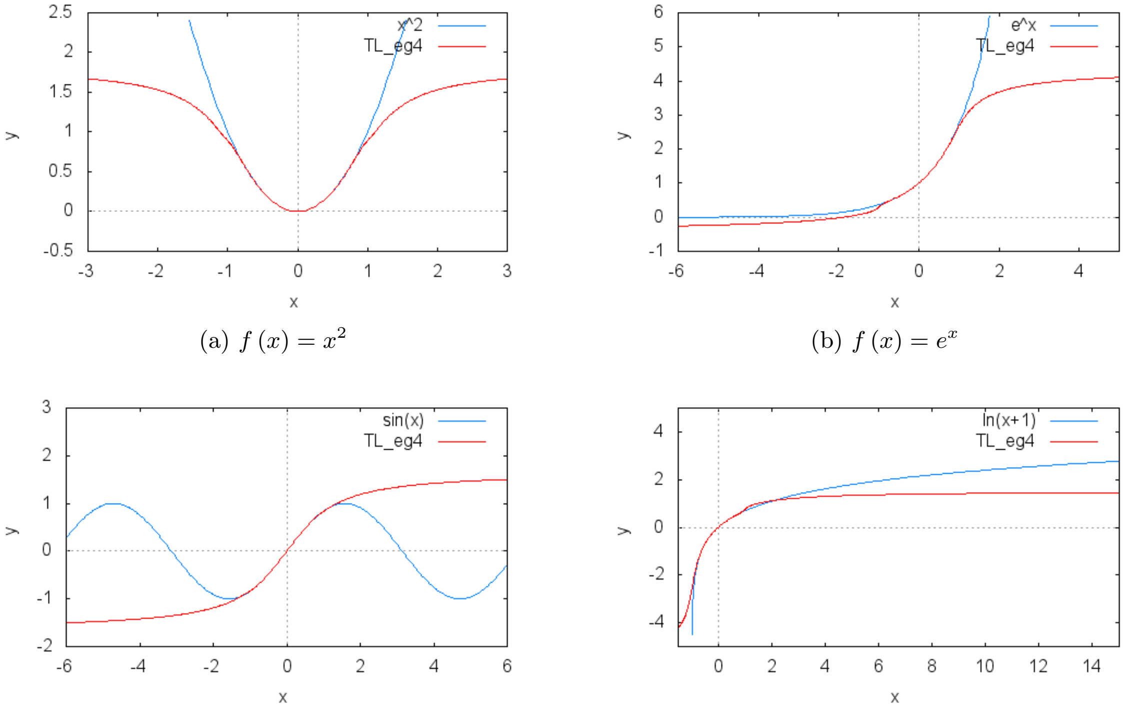
### 3.4 Example four

$$TL_N(x) = \sum_{i=0}^{N} a_i \arctan\left(x^i\right)$$

The derivatives at $x_0 = 0$ are

| derivative order | derivative value |
|---|---|
| 0 | $a_0$ |
| 1 | $a_1$ |
| 2 | $2a_2$ |
| 3 | $6a_3 - 3a_1$ |
| 4 | $24a_4$ |
| 5 | $120a_5 + 24a_1$ |
| 6 | $720a_6 - 240a_2$ |
| 7 | $5040a_7 - 720a_1$ |
| 8 | $40320a_8$ |
| 9 | $362880a_9 - 120960a_3 + 40320a_1$ |
| 10 | $3628800a_{10} + 725760a_2$ |

Number of coefficients in each line is small, the derivative matching can be done very efficiently. Like in the previous case, I did no manage to find neither direct nor recursive $i$-dependent expression for derivatives. Many other candidate functions can be used for this type of approximation. Examples of approximation using arctan are show in Figure 4.




(a) $f(x) = x^2$

(b) $f(x) = e^x$

(c) $f(x) = \sin(x)$

(d) $f(x) = \ln(x+1)$

Figure 4: Approximations of four test functions with $TL_{10}(x) = \sum_{i=0}^{10} a_i \arctan\left(x^i\right)$ (example four).

### 3.5 Example five

$$TL_N(x) = \sum_{i=0}^{N} \frac{a_i}{x^i}$$

with expansion done around $x_0 = 1$. The derivatives are can be expressed using direct $n$-dependent formulas

$$[TL_N(x)]^{(n)} = \sum_{i=0}^{N} (-1)^n \frac{(i+n-1)!}{(i-1)!} a_i.$$

One however gets a full matrix that needs to be inverted. To study the goodness of this approximation, I shifted all test function to $x_0 = 1$ and did the matrix inversion using computer. The results are displayed in Figure 5.

### 3.6 Example six

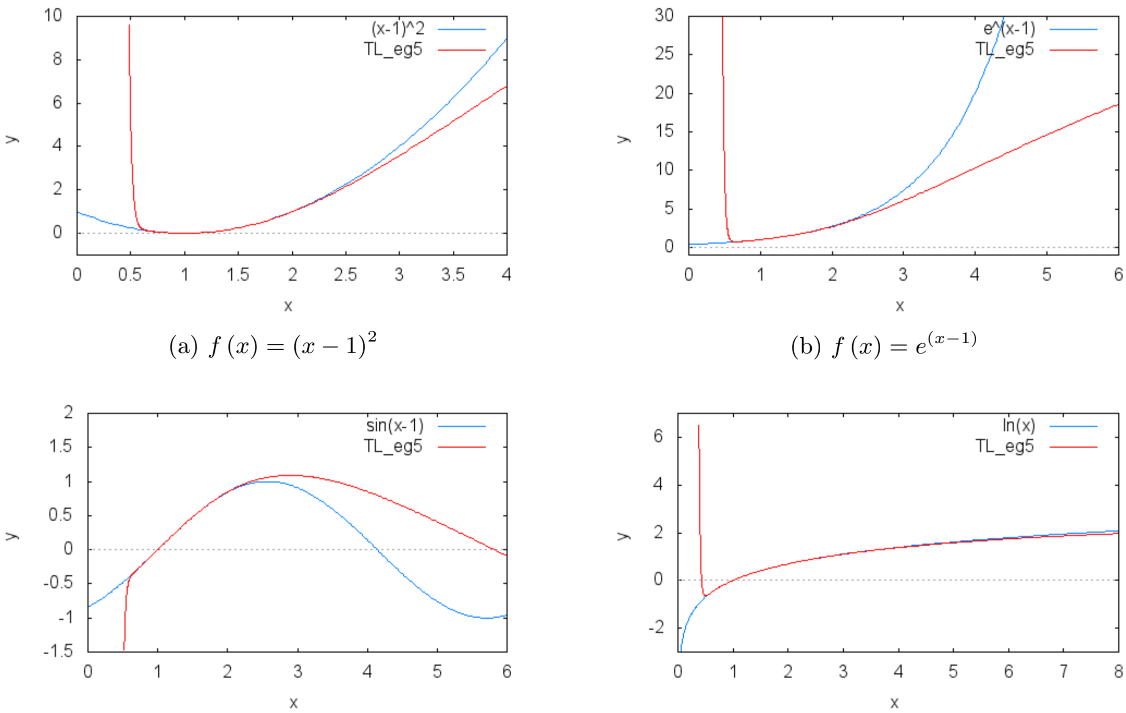
I made quite some effort to find approximations based on higher integrals, where the integration-constant polynomial would not appear. Searching without success, I prefer not to discuss easy examples with polynomials appearing, such as repetitive integrals of $e^x$: $a_0\left(e^x\right)$, $a_0\left(e^x\right) + a_1\left(e^x - 1\right)$, $a_0\left(e^x\right) + a_1\left(e^x - 1\right) + a_2\left(e^x - x - 1\right)$, $a_0\left(e^x\right) + a_1\left(e^x - 1\right) + a_2\left(e^x - x - 1\right) + a_3\left(e^x - \frac{1}{2}x^2 - x - 1\right)$, etc.. Instead I prefer to change a little bit the point of view, but still stick to the idea presented in Section 2.3.

Let me assume that the order of the approximation $N$ is given in advance. I present here an example where a function $h$ has $N-1$ derivatives vanishing at $x_0 = 0$, and only following derivatives are not zero. Then I will interpret the derivative $h^{(N)}$ as the function $g$ from Section 2.3 and lower order derivatives of $h$ as integrals of $g$.

The example is

$$h = [\sin(x)]^{2n+1} + [\sin(x)]^{2n}, \; 2n = N.$$

This function has $N-1$ derivatives vanishing at the origin, the $N$th is different from zero. The derivatives




(a) $f(x) = (x-1)^2$

(b) $f(x) = e^{(x-1)}$

(c) $f(x) = \sin(x-1)$

(d) $f(x) = \ln(x)$

Figure 5: Approximations of four test functions with $TL_{10}(x) = \sum_{i=0}^{10} \frac{a_i}{x^i}$ (example five).

behave as follows[5]:

$$\sum_{i=0}^{M} a_i [\sin(x)]^{M-i} [\cos(x)]^i \overset{\frac{d}{dx}}{\to} \sum_{i=0}^{M} b_i [\sin(x)]^{M-i} [\cos(x)]^i ,$$

where

$$b_i = (1 - \delta_{i,0})(M - i + 1) a_{i-1} - (1 - \delta_{i,M})(i+1) a_{i+1}.$$

To respect the usual approximation degree, I chose $N = 10$ ($n = 5$):

$$h = [\sin(x)]^{11} + [\sin(x)]^{10} .$$

[5]This relation can be represented by a $M+1$ dimensional matrix $Q$ in the coefficient space. Its elements are (indices start with "0"):

- *if* $(j = i + 1)$ *then* $Q_{ij} = -j$
- *else if* $(i = j + 1)$ *then* $Q_{ij} = -Q_{M-i,M-j}$
- *else* $Q_{ij} = 0$.

In general this matrix cannot be inverted. It is because some functions [e.g. $\sin^2(x)$] do not have the primitive function in the form of a sum of products of trigonometric powers. However, for $M$ odd, the matrix can be explicitly inverted $W = Q^{-1}$ and the primitive function calculated. The coefficients are:

- *if* $(i = j)$ *then* $W_{ij} = 0$
- *else if* $(i > j)$ *then* $W_{ij} = -W_{M-i,M-j}$
- *else:*
  - *if* $(i$ *is odd*$)$ *then* $W_{ij} = 0$
  - *else:*
    - *if* $(j$ *is even*$)$ *then* $W_{ij} = 0$
    - *else* $W_{ij} = \frac{A}{B}$, *where* $A = \frac{(j-1)!!}{i!!}$ *and* $B = \frac{(M-i)!!}{(M-j-1)!!}$.

Interestingly, even if $M$ is even, one can integrate $\sin^a(x)\cos^b(x)$ if both, $a$ and $b$ are odd. For this case one however needs to use the matrix $\widetilde{W} = \frac{1}{2}W$ and proceed analogically (here $W$ is a matrix built according to the given algorithm).

The function $g$ then takes the form:

$$g = \sum_{i=0}^{11} \alpha_i \left[\sin(x)\right]^{11-i} \left[\cos(x)\right]^i + \sum_{i=0}^{10} \beta_i \left[\sin(x)\right]^{10-i} \left[\cos(x)\right]^i$$

with $\alpha_i$ and $\beta_i$:

| i | $\alpha_i$ | $\beta_i$ |
|---|---|---|
| 0 | -79549811 | -46063360 |
| 1 | 0 | 0 |
| 2 | 3003336710 | 1514960640 |
| 3 | 0 | 0 |
| 4 | -11603988000 | -4904524800 |
| 5 | 0 | 0 |
| 6 | 9514169280 | 3149798400 |
| 7 | 0 | 0 |
| 8 | -1696464000 | -381024000 |
| 9 | 0 | 0 |
| 10 | 39916800 | 3628800 |
| 11 | 0 | - |

The integrals of $g$ can be obtained through

$$g_{Ii} = h^{(10-i)}$$

and formally the series takes the form

$$TL_{10}(x) = \sum_{i=0}^{10} a_i g_{Ii}(x) .$$

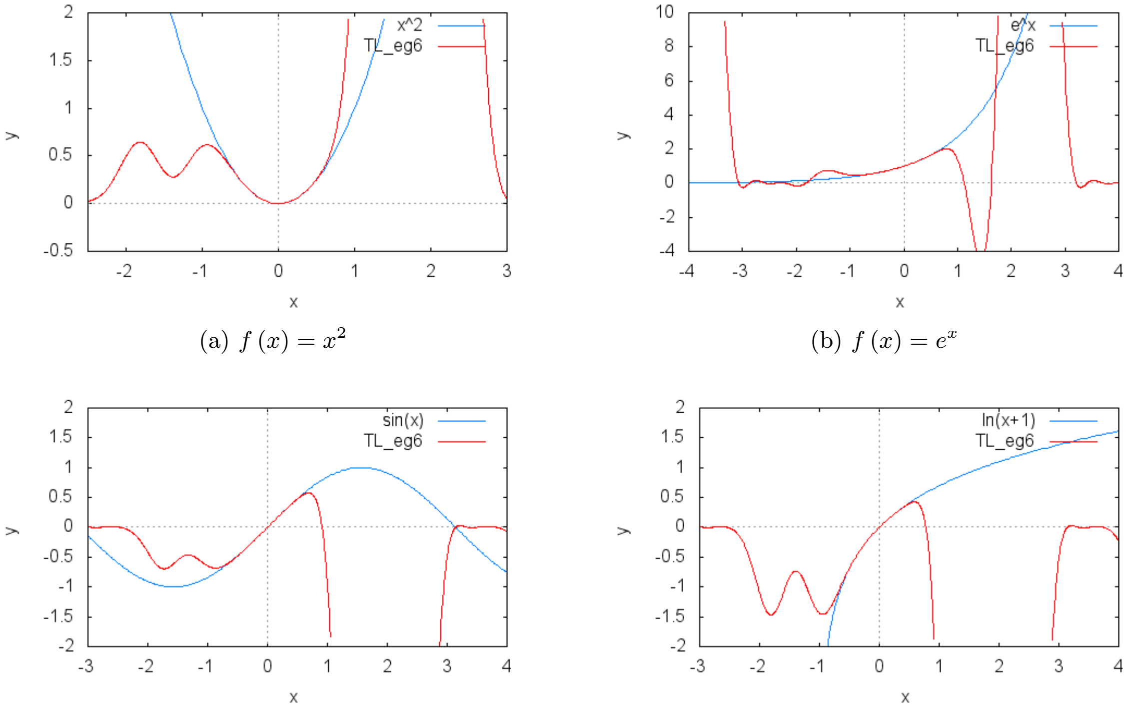
Each function $g_{Ii}$ might have up to 11+12=23 additive terms. However all of them have the same structure and only differ in coefficients. They can be therefore effectively summed up, so that the whole expression for $TL_{10}(x)$ has at most 23 terms. This can be usually further shortened by trying to factorize an appropriate powers of sin or cos[6].

The value and the first ten derivatives of the function $g$ at $x_0 = 0$ are

| derivative order | derivative value |
|---|---|
| 0 | 3628800 |
| 1 | 39916800 |
| 2 | -798336000 |
| 3 | -11416204800 |
| 4 | 116237721600 |
| 5 | 2117705990400 |
| 6 | -14280634368000 |
| 7 | -326823111014400 |
| 8 | 1610191341158400 |
| 9 | 45823744187020800 |
| 10 | -173115586215936000 |

The approximations are constructed in accordance with what was presented in Section 2.3, they are shown in Figure (6).

[6]Let me remark that even though the expression can by still considered long, it might be quite suited for computer calculations. When approximating the function at some point $x_1$, one needs to do only two "long" calculations of $\sin(x_1)$ and $\cos(x_1)$. The rest to get to the value of $Tl_N(x_1)$ consists in doing a rather small number of multiplications (higher powers of sin and cos) and additions.




(a) $f(x) = x^2$

(b) $f(x) = e^x$

(c) $f(x) = \sin(x)$

(d) $f(x) = \ln(x+1)$

Figure 6: Approximations of four test functions with $TL_{10}(x) = \sum_{i=0}^{10} a_i g_{Ii}(x)$ (example six).

## 4 Discussion, conclusion

In this text I presented some recipes and examples of Taylor-like approximations. A waste space for additional work remains. One might want to:

- become more explicit and rigorous (find and prove coefficients as function of the derivative order),
- study properties of expansions, their convergence (its speed, radius),
- propose other, maybe more elegant expansions.

When I decided to dig into the idea of Taylor-like approximations, I hoped to find nice and simple expansions with rapid convergence. This task was only partially achieved. Judging by constructed examples, the most rapid convergence seems to be found in the case of sum of inverses for $x > 1$ (example five) and exponential-weighted polynomials (example one). However, even these two approximations can hardly compete with Taylor polynomials. In addition, in many situations it is difficult to find a general expression for derivatives as function of the derivative order.

In cases of bounded functions used for expansion $(\arctan(x), \frac{x}{\sqrt{x^2+1}})$, one might hypothesize about better convergence properties, with the function value not “so easy” to go to infinity.

Even though I can hardly think of an immediate application, one might still hope that one of the presented examples might be suited for some particular tasks and problems one wants to solve in mathematics or physics.

## References

[1] Fu-Rong Lin, Wai-Ki Ching, Michael K. Ng, Fast inversion of triangular Toeplitz matrices, Theoretical Computer Science, Volume 315, Issues 2–3, 6 May 2004, Pages 511-523, ISSN 0304-3975, http://dx.doi.org/10.1016/j.tcs.2004.01.005.