

# Chapter 1

## Extrapolation to zero

In this short chapter we will describe shortly the concept of *extrapolation to zero* and how we can apply it.

### 1.1 Motivation

Let  $T : ]0, \varepsilon[ \rightarrow \mathbb{R}$  be function and assume that we have

$$T(h) = T_0 + ah^n + O(h^{n+1}). \quad (1.1)$$

for  $h \rightarrow 0$ . We are interested in computing  $T_0 = \lim_{h \rightarrow 0} T(h)$  up to some desired accuracy. In order to do that we might have to compute  $T(h)$  for very small  $h$ . That might not be feasible since  $T(h)$  might be very expensive to compute for small  $h$  or even impossible due to numerical instabilities. Hence we would like to somehow accelerate the convergence of  $T$  to 0. A nice way to do that is the *Richardson extrapolation scheme* which goes as follows: Let  $0 < r < 1$ . Plug  $rh$  into (1.1). Then we get

$$T(rh) = T_0 + ar^n h^n + O(h^{n+1}). \quad (1.2)$$

Now multiply (1.1) by  $r^n$ , subtract it from (1.2) and divide the result by  $1 - r^n$ . Then we get:

$$R(h) = T_0 + O(h^{n+1})$$

where

$$R(h) := \frac{T(rh) - r^n T(h)}{1 - r^n}.$$

Note that  $R(h)$  has  $O(h^{n+1})$  convergence to  $T_0$  while  $T(h)$  has  $O(h^n)$ , i.e.  $R(h)$  converges asymptotically faster. But what did we actually do? We took the linear polynomial in  $t^n$  which goes through  $(rh, T(rh))$  and  $(h, T(h))$  and let  $R(h)$  be its value at 0, i.e. we interpolated the points and then evaluated the interpolation polynomial outside the interval; hence the term *extrapolation*. This should serve as a motivation for the sequel.

### 1.2 The extrapolation table

We always think of  $T$  as arising from some numerical scheme e.g. the trapezoidal rule and then  $T_0$  is the integral of some function. Thus we do not require that  $T$  is necessarily defined for all values near 0, but only on a discrete set  $\mathbb{H}$  which has 0 as an accumulation point. In what follows, we will thus refer to  $T$  as a *method* for computing  $T_0$ .

**Definition 1.1.** Let  $T$  be a method for computing  $T_0$ . We say that  $T$  has an asymptotic expansion in  $h^p$  up to order  $pm$  if there exist constants  $\tau_p, \tau_{2p}, \dots, \tau_{mp} \in \mathbb{R}$  such that

$$T(h) = T_0 + \tau_p h^p + \tau_{2p} h^{2p} + \dots + \tau_{mp} h^{mp} + O(h^{(m+1)p}) \quad (1.3)$$

for  $h \rightarrow 0$ ,  $h \in \mathbb{H}$ .

Let  $(x_1, y_1), \dots, (x_k, y_k)$  be a collection of points such that  $x_1, \dots, x_k$  are distinct. Then there exists a unique polynomial  $P$  of degree  $k - 1$  which interpolates the points, i.e.  $P(x_i) = y_i$  for all  $i$ . We say that  $P$  is the *interpolation polynomial* for the points. Let  $p > 0$  be an integer and points  $(x_1^p, y_1), \dots, (x_n^p, y_n)$  such that  $x_i^p$  are distinct, be given. Let  $P$  be the interpolation polynomial for the points. We then call  $P(h^p)$  the *interpolation polynomial in  $p$*  for these points.

Let  $T$  be a method with asymptotic expansion in  $p$  up to  $pm$ . The extrapolation process works as follows: We compute  $T(h)$  for some points  $h_1, h_2, \dots, h_k$  where  $k \leq m$ . Then we compute the interpolation polynomial  $P$  in  $h^p$  which goes through  $(h_1, T(h_1)), \dots, (h_k, T(h_k))$ . We then hope that  $P(0)$  gives a good approximation  $T_0$ .

In order to compute  $P(0)$  we use the *Neville scheme*. Let  $P_{ij}(h^p) := P(h^p; h_{i-j+1}^p, h_i^p)$  be the interpolation polynomial in  $h^p$  which interpolates  $(h_{i-j+1}^p, T(h_{i-j+1})), \dots, (h_i^p, T(h_i^p))$  and set  $T_{ij} := P_{ij}(0)$ . Then according to the Neville scheme we can compute  $T_{ij}$ ,  $j \leq i$ , in the following recursive way:

1.  $T_{i1} := T(h_i)$  for  $i = 1, \dots, k$ .
2.  $T_{ij} := T_{i,j-1} + \frac{T_{i,j-1} - T_{i-1,j-1}}{r^p - 1} = \frac{r^p T_{i,j-1} - T_{i-1,j-1}}{r^p - 1}$  for  $1 < j \leq i$  where  $r := h_{i-j+1}/h_i$ .

If we align  $T_{ij}$  to a triangular table, we call that the *extrapolation table*.

### 1.3 Convergence

If we have a numerical method or scheme that has an asymptotic expansion as (1.3), then the error decays polynomially as  $h \rightarrow 0$ . It is known (see e.g. theorem 9.22 in [1]) that  $T_{ij}$  has polynomial decay of higher degree, as  $h \rightarrow 0$ , than  $T$ . Let  $\varepsilon_k := |T_{kk} - T|$ . We want to analyse how  $\varepsilon_k$  behaves as  $k \rightarrow +\infty$ , i.e. how  $\varepsilon_k$  behaves when we increase the number of extrapolation steps. Let  $N_n$  be some measure of the effort needed to compute  $T_{kk}$ . In what follows we will test numerically the qualitative hypothesis that the error converges exponentially with the computational effort i.e.

$$\varepsilon_k \sim A \exp(-cN_k^q) \quad (1.4)$$

for constants  $A, c, q$ . Note that if  $\varepsilon_k = A \exp(-cN_k^q)$  then  $\ln \varepsilon_k = b - cN_k^q$  so in order to test the hypothesis we will do the following: Assume that we have the error  $\varepsilon_k$  for  $k = 1, \dots, n$ . Then we will compute

$$(b^*, c^*, q^*) := \operatorname{argmin}_{(b,c,q)} \left\{ \sum_{k=1}^n |\ln \varepsilon_k - (b - cN_k^q)|^2 \right\} \quad (1.5)$$

and see whether the points  $(N_k, \ln \varepsilon_k)$  fit well to the graph of  $t \mapsto b - ct^q$ . Here  $b = \ln A$ .

We will also test the hypothesis that the error converges exponentially with the number of extrapolation steps, i.e. whether

$$\varepsilon_k \sim A \exp(-ck^q) \quad (1.6)$$

for constants  $A, c, q$ .

We define the following relative residuals:

$$\rho_{\ln} := \frac{\sum_{k=1}^n |\ln \varepsilon_k - (b^* - c^* N_k^{q^*})|^2}{\sum_{k=1}^n |\ln \varepsilon_k|^2}$$

and

$$\rho_{\text{lin}} := \frac{\sum_{k=1}^n |\varepsilon_k - A^* \exp(-c^* N_k^{q^*})|^2}{\sum_{k=1}^n |\varepsilon_k|^2}$$

where  $A^* := \exp(b^*)$ . We will use these residuals as one measurement of the goodness of fit.

In addition to that we will do a simple *cross validation* by fitting the model to subsets of the data and see whether the parameters vary a lot. If they vary a lot, we conclude that the fitting is unstable. If they are almost the same we will be more confident in that the model is actually appropriate. The cross validation strategy we will use goes as follows: Suppose that we have done a curve fitting on  $(x_k, y_k)$  for  $k = 0, 1, \dots, n$ . Let  $r = \max(10, [n/2])$ . Then we will do the curve fitting for  $(x_k, y_k), \dots, (x_{k+r-1}, y_{k+r-1})$  for  $k = 3, \dots, n-r-1$  and compute the relative variance of the parameters. Let  $a_k$ ,  $k = 1, \dots, m$  be numbers. Then we define their mean value by  $\bar{a} := \frac{1}{m} \sum_{k=1}^m a_k$ , and the relative variance by

$$\frac{1}{m\bar{a}^2} \sum_{k=1}^m (a_k - \bar{a})^2.$$

In order to visualize the stability of the fit, we will plot all the curves obtained in the cross validation, on top of each other. We call this plots *stack plots*.

## 1.4 Code

The following Python function computes the extrapolation table for some scheme which has an asymptotic expansion in  $h^p$ .

```
#sc (Scheme): The scheme to extrapolate
#prob: The problem to apply the scheme to. We assume that sch is an
#       implementation of Scheme which can be applied to prob.
#seq (Sequence): The sequence to use in the extrapolation
#hp (bool): Indicates whether to use high precision arithmetic (true)
#           or standard double precision (false).
#returns: The extrapolation table as an np.array of np.arrays.
def extrapolate(sc, prob, seq, hp):
    n = len(seq)
    X = np.array([np.array([mpf(0) if hp else 0] * (i+1)) for i in range(n)])
    #X[i][j] = T_ij
```

```
for i in range(n):
    X[i][0] = sc.apply(prob, seq[i])
    for j in range(1, i + 1):
        #r = h_{i-j} / h_i = seq[i] / seq[i-j]
        #rp = r^p.
        #Must cast the elements of seq to hp numbers if in hp mode.
        rp = ((mpf('1') * seq[i]) / (mpf('1') * seq[i-j])) if hp else seq[i] / seq[i-j]) ** sc.p
        X[i][j] = (rp * X[i][j-1] - X[i-1][j-1]) / (rp - 1)

return X
```

# Chapter 2

## Romberg quadrature

### 2.1 The algorithm

Let  $f : [a, b] \rightarrow \mathbb{R}$  be a function and  $I := \int_a^b f(x)dx$ . The *trapezoidal rule* is a method for approaching  $I$  which works as follows: Let  $a = t_0 < t_1 < \dots < t_n = b$  be a subdivision of  $[a, b]$ . On each of the intervals  $[t_{i-1}, t_i]$  we approximate  $\int_{t_{i-1}}^{t_i} f(x)dx$  by the area of a trapezoid with vertices  $(t_{i-1}, 0)$ ,  $(t_{i-1}, f(t_{i-1}))$ ,  $(t_i, f(t_i))$ ,  $(t_i, 0)$  i.e. by  $\frac{1}{2}(t_i - t_{i-1})(f(t_{i-1}) + f(t_i))$ . Hence we approximate  $I$  by

$$I = \sum_{i=1}^n \int_{t_{i-1}}^{t_i} f(x)dx \approx \sum_{i=1}^n \frac{1}{2}(t_i - t_{i-1})(f(t_{i-1}) + f(t_i)).$$

If  $t_i - t_{i-1} = \frac{1}{n}(b - a) =: h$  for each  $i$  then the above estimate becomes

$$I \approx h \left( \frac{1}{2}(f(a) + f(b)) + \sum_{i=1}^{n-1} f(a + ih) \right) \quad (2.1)$$

We define  $T_f(h)$  as the right hand side in (2.1) for all  $h \in \{(b - a)/n \mid n \in \mathbb{Z}_+ \dots\} =: \mathbb{H}$ .

Let  $F : [0, n] \rightarrow \mathbb{R}$  be a  $2k + 1$  times continuously differentiable function,  $n$  a positive integer. Then by Euler's summation formula (see formula 298 in [2]) we have

$$\sum_{i=0}^n F(i) = \int_0^n F(x)dx + \frac{1}{2}(F(0) + F(n)) + \sum_{i=1}^k \frac{B_{2i}}{(2i)!}(F^{(2i-1)}(n) - F^{(2i-1)}(0)) + R_k \quad (2.2)$$

where  $R_k = \int_0^n P_{2k+1}(x)F^{(2k+1)}(x)dx$ ,  $B_m$  are the *Bernoulli numbers* and  $P_m$  the *Bernoulli polynomials*. If we let  $F(x) := f(a + xh)$  then we get the following asymptotic expansion for the trapezoidal rule:

**Theorem 2.1.** *Let  $f : [a, b] \rightarrow \mathbb{R}$  be  $2k + 1$  times continuously differentiable and  $h := (b - a)/n$ . Then*

$$T_f(h) = I + \sum_{i=1}^k \frac{B_{2i}}{(2i)!}(f^{(2i-1)}(b) - f^{(2i-1)}(a))h^{2i} + h^{2k+1}R_k(h) \quad (2.3)$$

where

$$R_k(h) = \int_a^b P_{k+1} \left( n \frac{x-a}{b-a} \right) f^{(2k+1)}(x)dx. \quad (2.4)$$

The following code is a trivial implementation of the trapezoidal rule. The *TrapezoidalRule* class in an implementation of the abstract class *Scheme* which represents a numerical scheme or method, which has asymptotic expansion in  $h^p$ . The Scheme class has a method named *apply* which takes in a problem to which the scheme is applied to. The argument  $m$  in the apply-method is the number of subintervals that should be used.

```
class TrapezoidalRule(Scheme):
    def __init__(self):
        super(TrapezoidalRule, self).__init__(2)

    def apply(self, inte, m):
        (a,b) = inte.interval
        h = (b - a) / m
        I = 0.5 * (inte.f(a) + inte.f(b))
        for i in range(1, m):
            I += inte.f(a + i * h)

        return I * h
```

Assume that we have computed the value of  $T_f(h)$  for  $h = h_1, \dots, h_k$  and we want to extrapolate to zero, i.e. we want to compute the value at zero of the interpolation polynomial in  $h^2$  for  $(h_i^2, T_f(h_i))$ ,  $i = 1, \dots, k$ . Denote by  $T_{ij}$  the value at zero of the polynomial of degree  $j - 1$  in  $h^2$  which goes through  $(h_{i-j+1}^2, T(h_{i-j+1}))$ ,  $\dots$ ,  $(h_i^2, T(h_i))$ . The Neville scheme gives us the following algorithm for computing  $T_{ij}$ ,  $1 \leq j \leq i \leq k$ , recursively:

1.  $T_{i1} := T_f(h_i)$  for  $i = 1, \dots, k$ .
2.  $T_{ij} := T_{i,j-1} + \frac{T_{i,j-1} - T_{i-1,j-1}}{\left(\frac{h_{i-j+1}}{h_i}\right)^2 - 1}$  for  $2 \leq j \leq i$ .

## 2.2 Numerical experiments

In this section we are going to apply Romberg quadrature to various functions and also try different sequences. We will analyse how different sequences perform in the sense that we want to measure how many function evaluations we need to attain a prescribed precision.

We will try various functions and the following sequences:

- The harmonic sequence:  $a_n = n$ ,  $n > 0$ .
- The Romberg sequence:  $a_n = 2^{n-1}$ ,  $n \geq 1$ .
- The Bulirsch sequence:  $a_1 = 1$ ,  $a_2 = 2$ ,  $a_3 = 3$  and  $a_{n+2} = 2 \cdot a_n$  for  $n \geq 2$ . Its first elements are

$$1, 2, 3, 4, 6, 8, 12, 16, 24, 32, \dots$$

Suppose that we are approximating the integral  $I := \int_a^b f(x)dx$  using Romberg quadrature. We will use the stepsizes  $h_k := (b - a)/a_k$  for the extrapolation. Let  $T_{ij}$ ,  $i \geq 0$  and  $j \leq i$  be the extrapolation table we get and  $\varepsilon_k := |T_{kk} - I|$  be the error on the diagonals. Let  $N_k$  be the number of function evaluations needed to compute  $T_{kk}$ . We will use  $N_k$  as the measurement of computational effort as mentioned in section 1.3 and we will try to fit the exponential convergence model introduced there. We will also plot the logarithm of the error against the number of extrapolation steps. Note that  $N_k = \sum_{i=1}^k (a_i + 1)$  where

$(a_i)$  is our sequence, so in case of the Harmonic sequence, we have  $N_n = n(n+3)/2 \approx n^2/2$  for  $n$  large. Hence if  $\varepsilon_n \sim A \exp(-cN_n^q)$  then

$$\varepsilon_n \sim A \exp(-c/2^q n^{2q})$$

for  $n$  large. Thus if the error converges exponentially with the number of function evaluations, it will also converge exponentially with the number of extrapolation steps, and the exponent in the latter fitting will be twice the parameter from the former.

If our sequence is the Romberg sequence then  $N_k = \sum_{i=1}^k (2^{i-1} + 1) = 2^k + k - 1 \approx 2^k$  for  $k$  large, so if  $\varepsilon_k \sim A \exp(-cN_k^q)$  then

$$\varepsilon_k \sim A \exp(-c2^{kq})$$

for  $k$  large, which is not exponential convergence. Hence possibilities of having exponential convergence in the number of steps and in the number of evaluations are mutually exclusive for the Romberg sequence.

For the model fitting we will plot the logarithm of the error against the number of function evaluations and the number of extrapolation steps. Then we will try to fit the points on curve of a function of the form  $t \mapsto b - ct^q$  and we will report the mean and relative variance of  $A := e^b$ ,  $c$  and  $q$ . We will also provide the plot of the base 10 logarithm of the error against the number of function evaluations.

We conduct the experiments in Python 3 and use the high precision arithmetic library mpmath for all the computations. The precision will be set to 500 significant digits so will not have to worry about numerical instabilities.

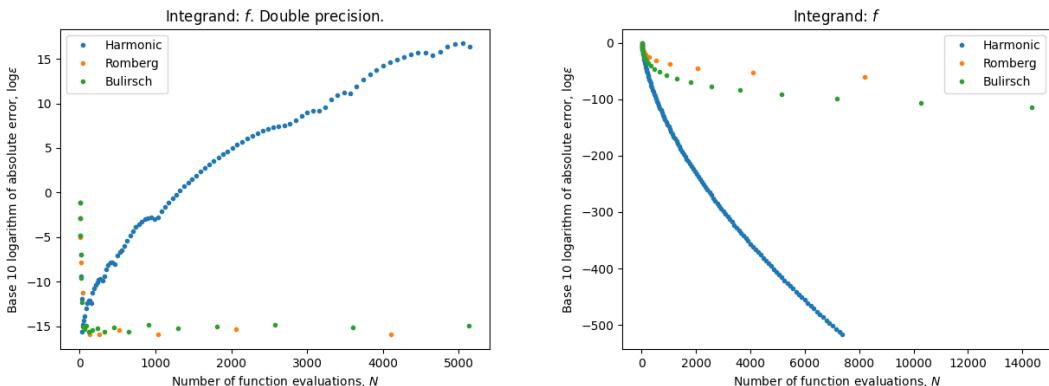
Now we will consider the results of the experiments.

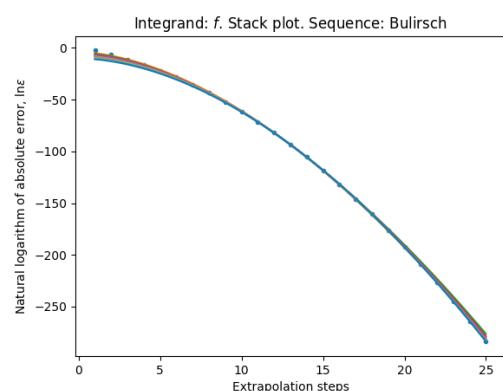
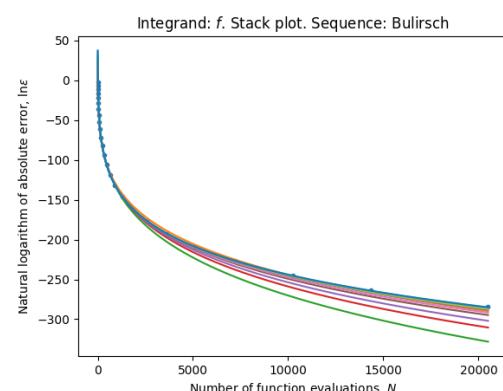
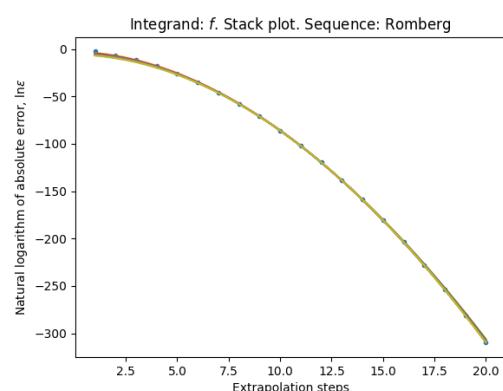
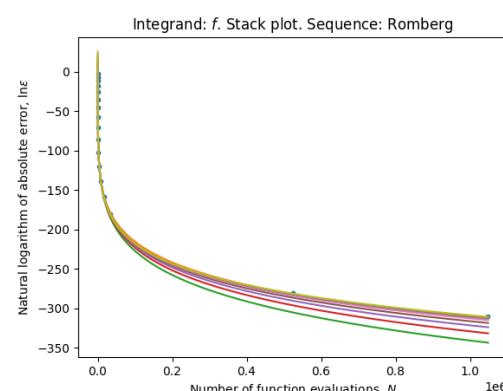
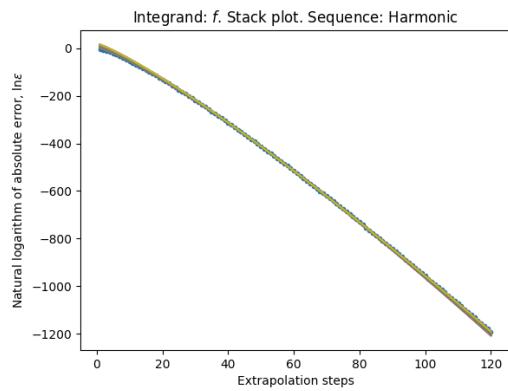
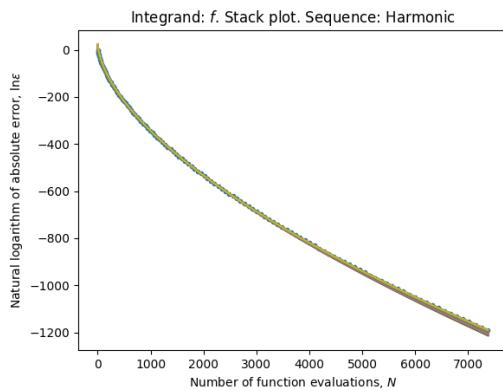
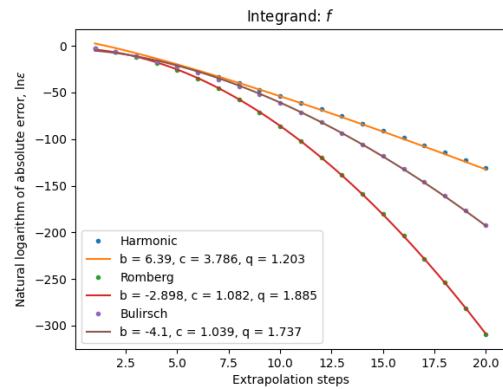
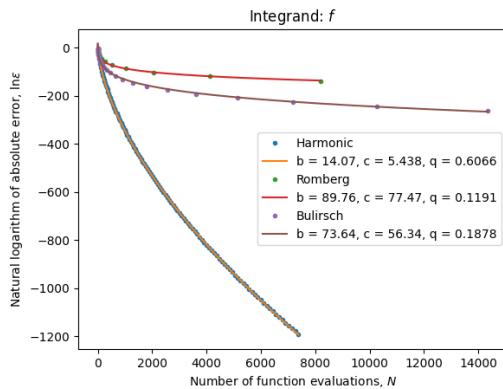
### 2.2.1 Cosine squared

The first function we are going to try is

$$f : [0, 1] \rightarrow \mathbb{R}, \quad f(x) := \cos^2(x)$$

which is entire.





Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\text{lin}}$	$\rho_{\ln}$
Harmonic	$8.644 \cdot 10^{12}$	7.493	5.843	0.004665	0.5999	0.00017	$1.605 \cdot 10^7$	$6.464 \cdot 10^{-6}$
Romberg	$1.071 \cdot 10^{73}$	6	95.86	0.06674	0.1125	0.01945	$1.14 \cdot 10^7$	0.0003937
Bulirsch	$3.78 \cdot 10^{76}$	7.998	87.35	0.1063	0.164	0.03428	$2.502 \cdot 10^{10}$	0.0007942
Harmonic	$8.372 \cdot 10^7$	6.367	4.002	0.003408	1.193	0.0001069	$2.727 \cdot 10^4$	$3.325 \cdot 10^{-6}$
Romberg	0.0127	0.6464	1.015	0.002443	1.904	$7.669 \cdot 10^{-5}$	0.5932	$1.117 \cdot 10^{-5}$
Bulirsch	0.002302	2.227	0.9216	0.01377	1.771	0.0004207	0.8609	$3.924 \cdot 10^{-5}$

We see that the harmonic sequence performs best, then Bulirsch and then Romberg. In standard double precision arithmetic, we get down to machine level precision using Romberg or Bulirsch, but we are like 2 digits from there, using the harmonic sequence.

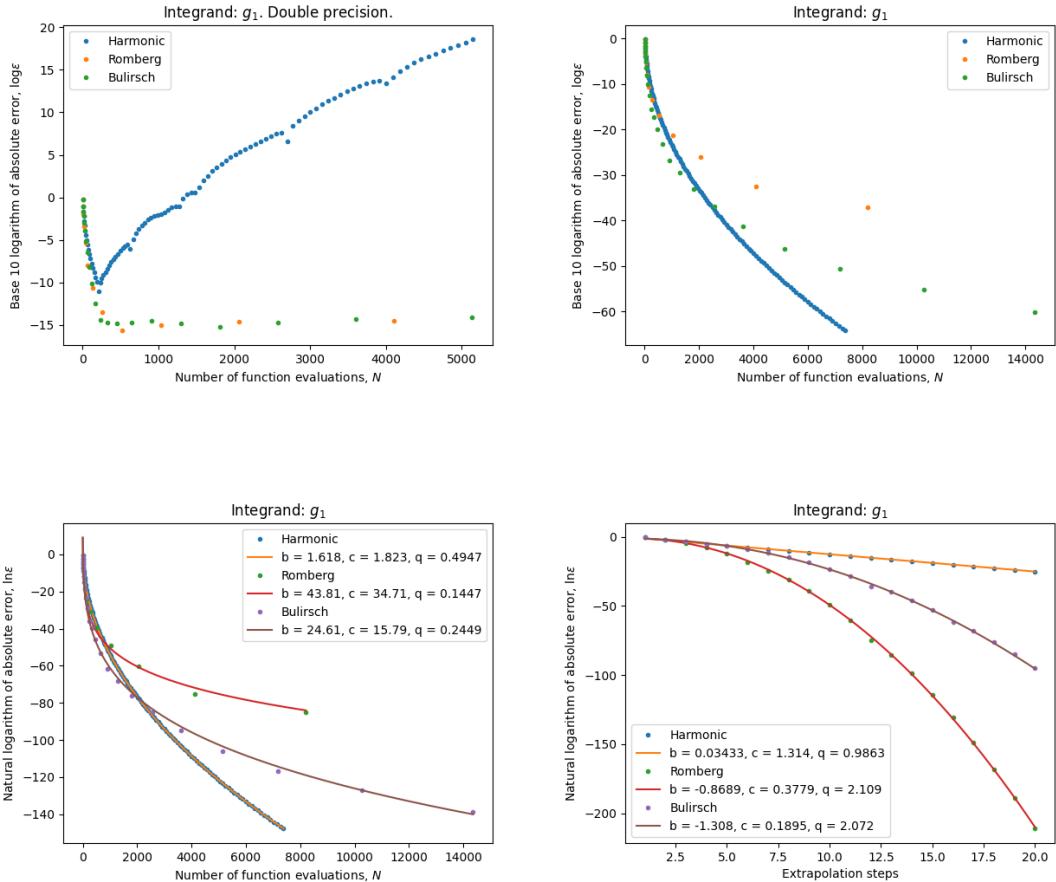
For the Romberg and Bulirsch sequence we clearly have exponential convergence in the number of steps but not in the number of evaluations.

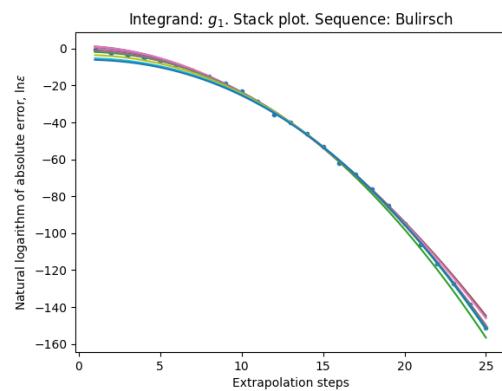
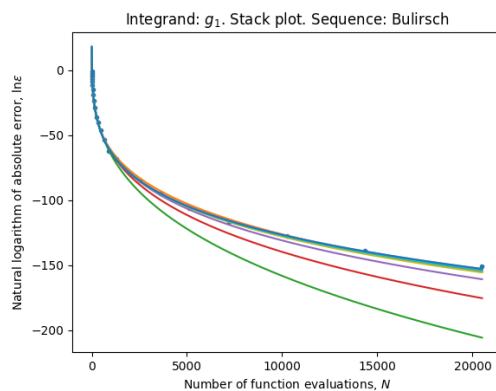
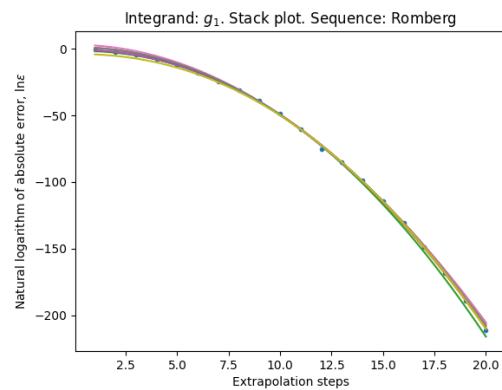
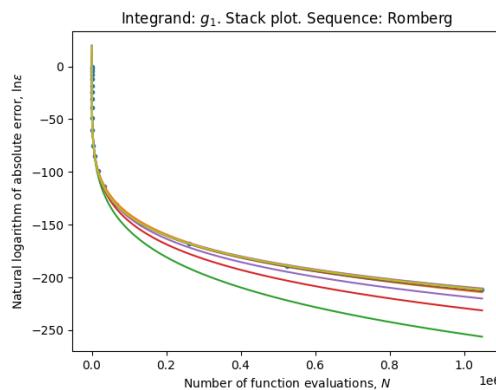
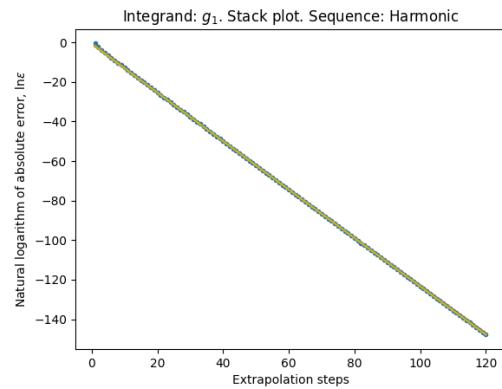
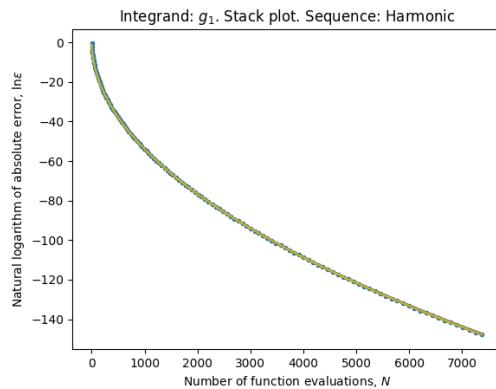
For the harmonic sequence we get unreasonably large values of  $A$  and we can see from the stack plots that the fitting is not so stable.

### 2.2.2 Function with poles

Now we will consider the following function:

$$g_a : [-1, 1] \rightarrow \mathbb{R}, \quad g_a(x) := \frac{1}{a^2 + x^2}, \quad a > 0$$

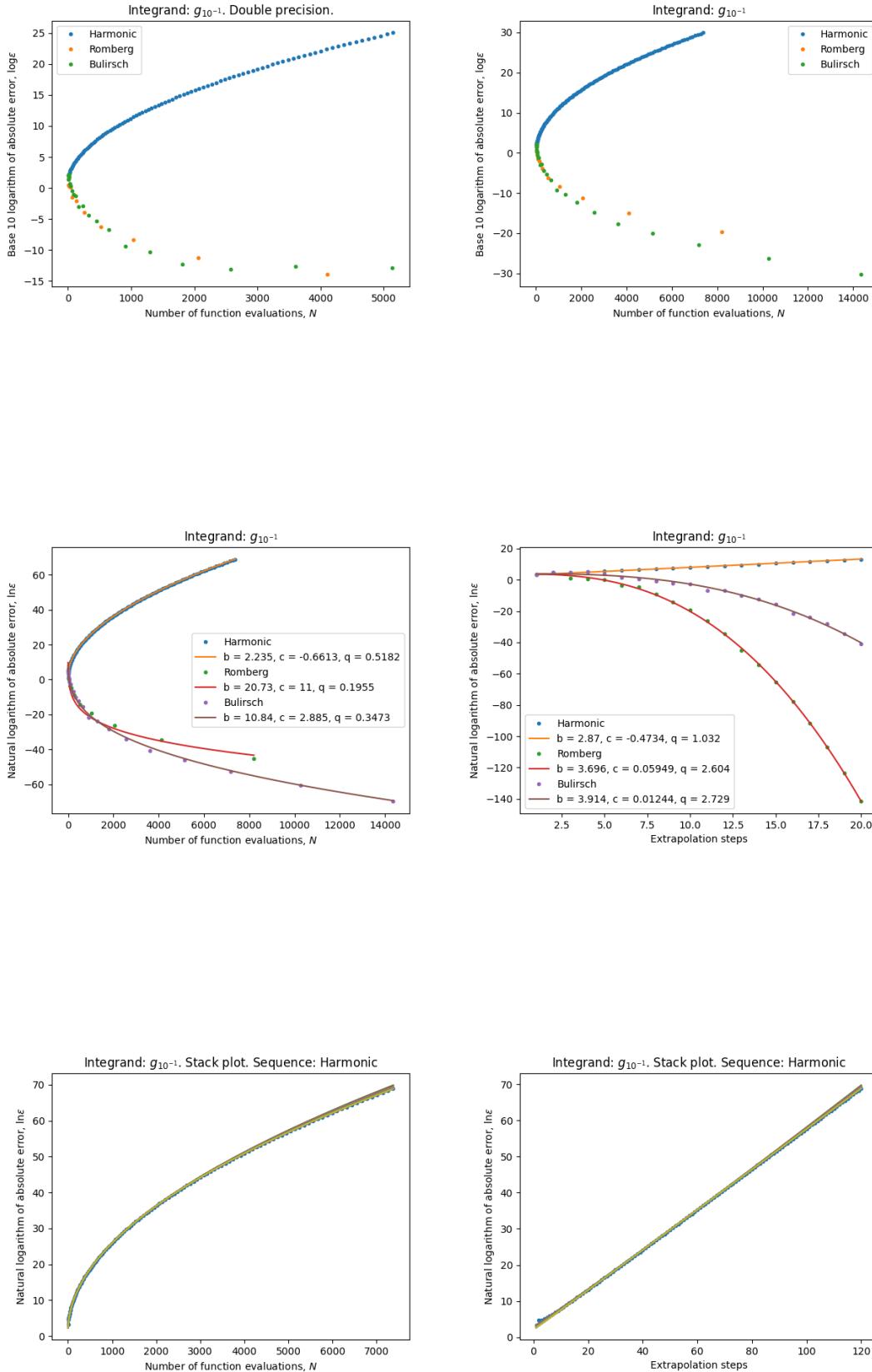


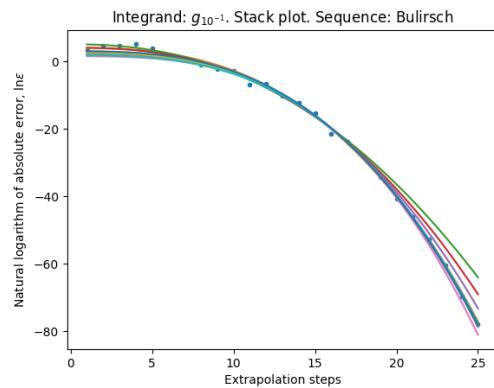
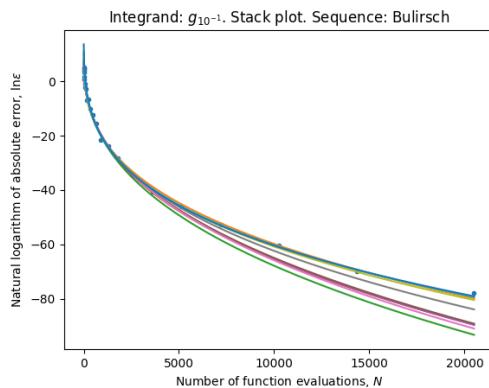
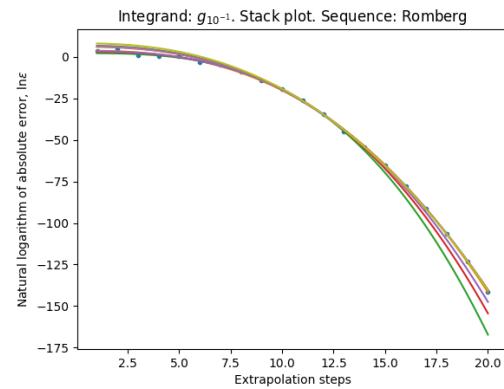
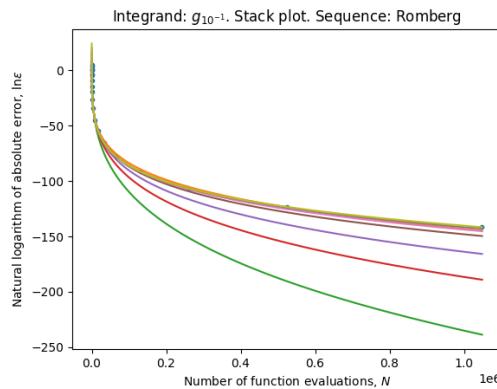


Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Harmonic	3.771	0.02895	1.793	0.0001375	0.4963	$6.495 \cdot 10^{-6}$	0.1014	$3.209 \cdot 10^{-7}$
Romberg	$4.114 \cdot 10^{37}$	5.211	46.94	0.1026	0.135	0.03489	$1.548 \cdot 10^5$	0.0005732
Bulirsch	$8.199 \cdot 10^{22}$	4.987	24.22	0.1103	0.2236	0.04529	$3.921 \cdot 10^5$	0.001355
Harmonic	0.6596	0.04548	1.277	0.0002625	0.9918	$1.093 \cdot 10^{-5}$	0.2565	$1.067 \cdot 10^{-6}$
Romberg	3.747	2.921	0.4181	0.02976	2.079	0.0008172	0.2401	$7.665 \cdot 10^{-5}$
Bulirsch	1.191	1.72	0.2048	0.105	2.062	0.00252	0.3619	$9.058 \cdot 10^{-5}$

We see that the harmonic sequence performs best, then Bulirsch and then Romberg. In standard double precision arithmetic, we get down to machine level precision using Romberg or Bulirsch, but we are like 5 digits from there, using the harmonic sequence.

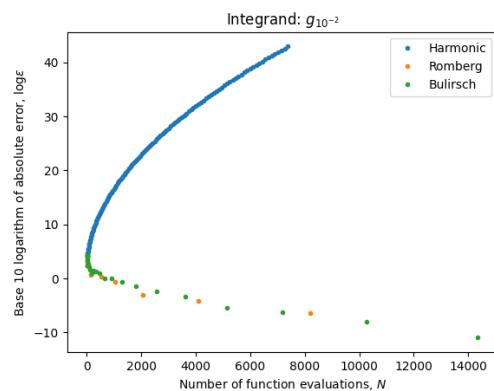
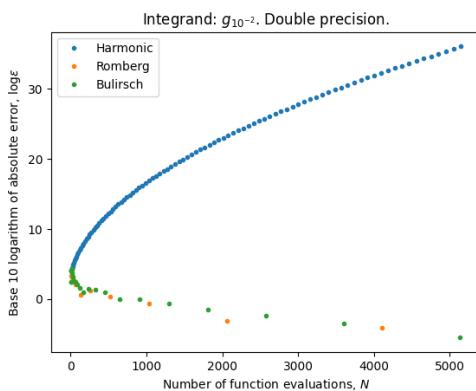
Here we clearly have exponential convergence in the number of evaluations for the Harmonic sequence and hence also in the number of steps. We do not have exponential convergence in the number of evaluations for Romberg and Bulirsch but the model for exponential convergence in the number of steps fits moderately well for those sequences.

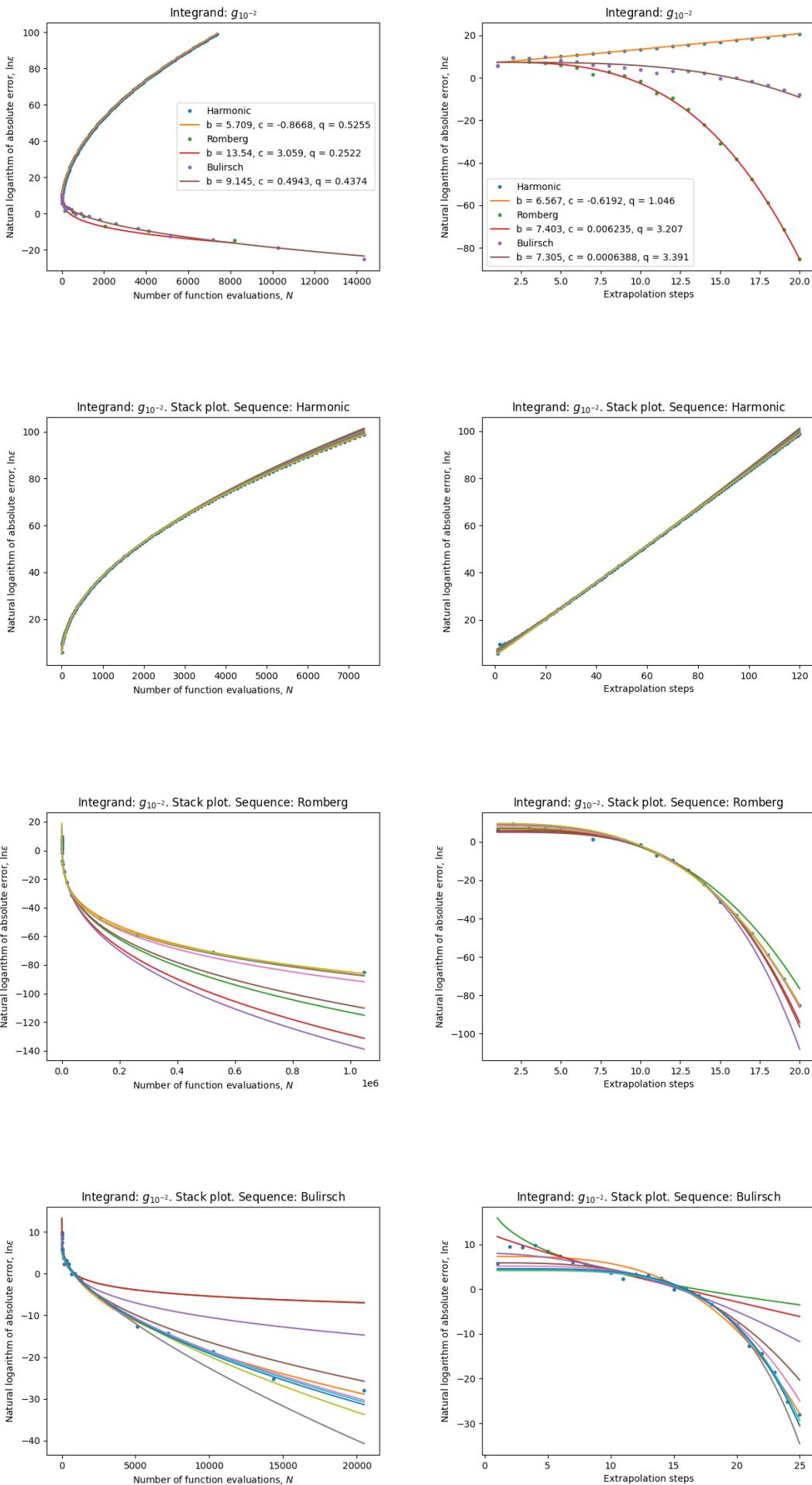




Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\ln}$	$\rho_{\ln}$
Harmonic	5.231	0.0773	-0.7178	0.001803	0.5101	$9.026 \cdot 10^{-5}$	0.01647	$8.663 \cdot 10^{-6}$
Romberg	$7.533 \cdot 10^{22}$	5.999	14.35	0.3322	0.2047	0.07855	974.3	0.001405
Bulirsch	$8.963 \cdot 10^7$	7.604	2.995	0.186	0.3602	0.0129	33.54	0.001672
Harmonic	10.97	0.05032	-0.5086	0.001492	1.019	$6.47 \cdot 10^{-5}$	0.01168	$6.684 \cdot 10^{-6}$
Romberg	727.7	2.108	0.06624	0.1819	2.623	0.005292	0.521	0.0001729
Bulirsch	32.46	2.026	0.01381	0.7483	2.764	0.006506	0.5118	0.0008637

Here we get divergence for the harmonic sequence, but convergence for the other sequences, fastest for Bulirsch. In standard double precision arithmetic, we get down to machine level precision using Romberg or Bulirsch. The model for exponential convergence in number of evaluations does not fit for Bulirsch nor Romberg but it is hard to tell whether we have exponential convergence in the number of steps.





Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Harmonic	122.7	0.8001	-0.9641	0.007205	0.5155	0.0003682	0.07171	$1.955 \cdot 10^{-5}$
Romberg	$8.898 \cdot 10^{10}$	5.785	3.021	0.6343	0.2928	0.05447	2.317	0.00266
Bulirsch	.	.	2876	4.874	0.4042	0.3708	0.7208	0.01268
Harmonic	323.1	0.5753	-0.682	0.006766	1.029	0.0002999	0.05485	$1.857 \cdot 10^{-5}$
Romberg	4384	1.505	0.005443	0.4455	3.388	0.0103	0.76	0.001069
Bulirsch	$1.73 \cdot 10^{11}$	8	1.498	6.41	3.17	0.272	0.8139	0.01705

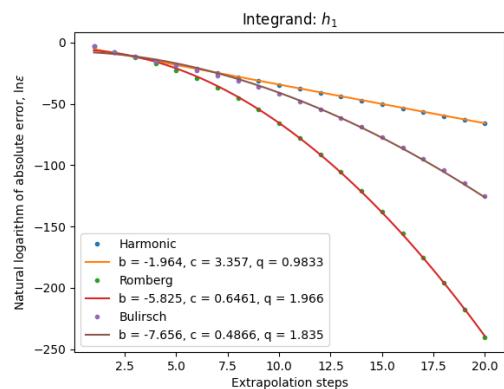
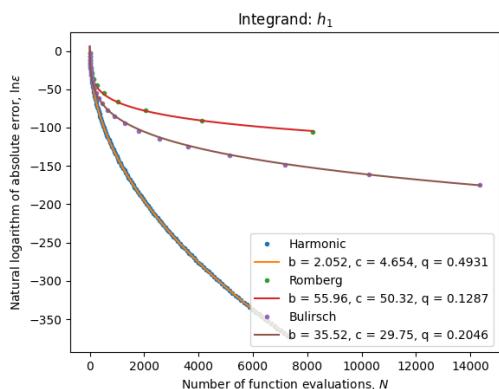
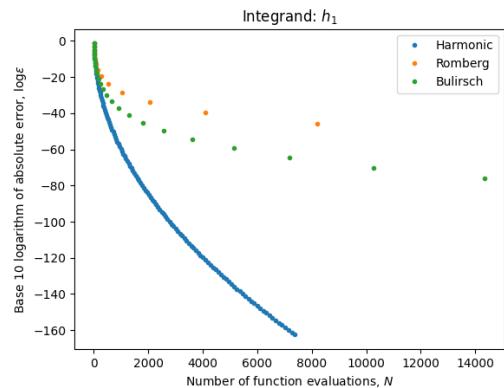
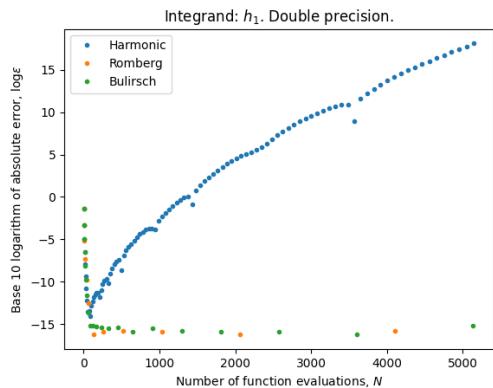
Here the same comments apply as for  $a = 10^{-1}$ , except that now the Romberg sequence performs better than the Bulirsch sequence and the model fitting is worse.

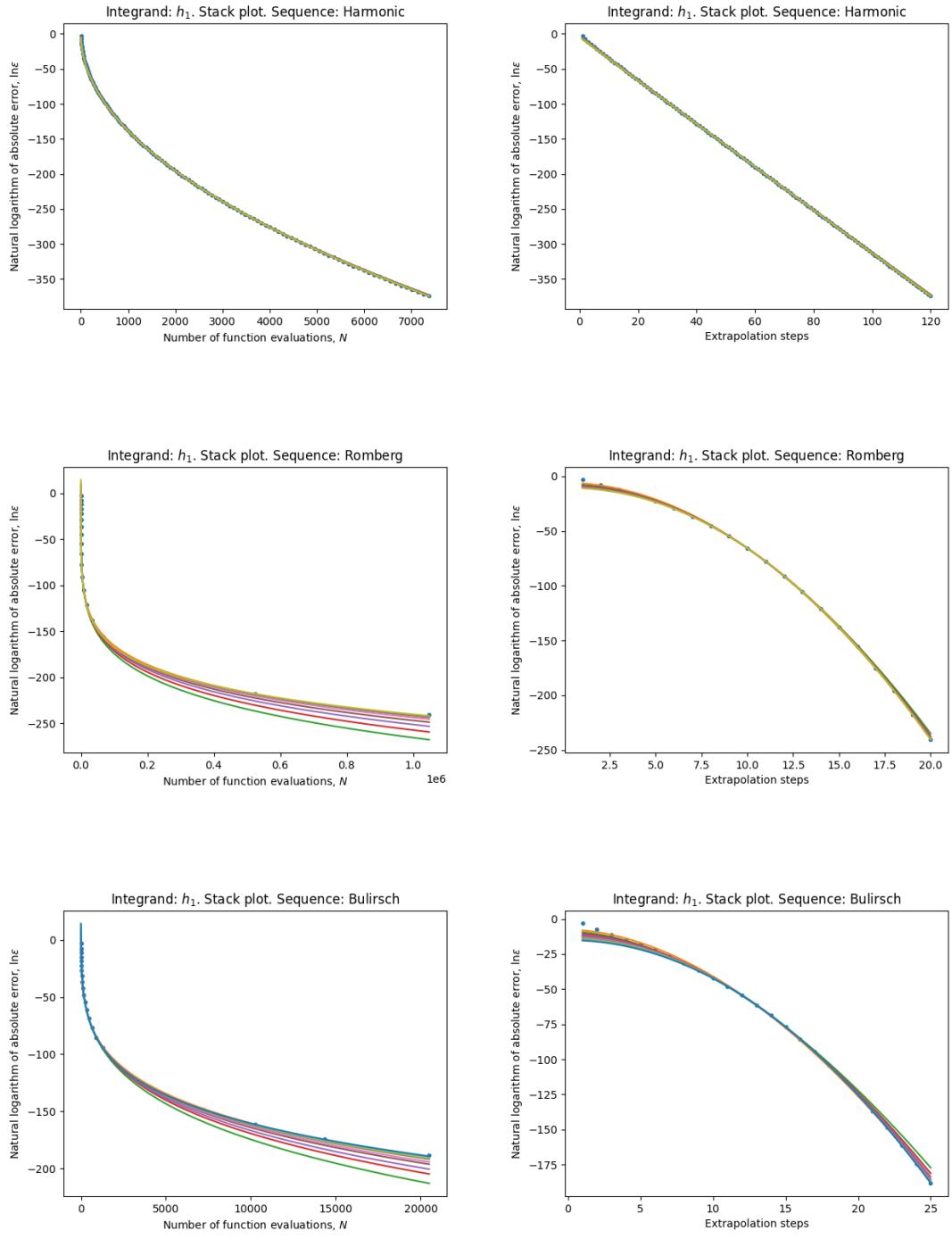
### 2.2.3 Logarithm

Now we will consider the following function

$$h_a : [0, 1] \rightarrow \mathbb{R}, \quad h_a(x) := \ln(a + x), \quad a > 0.$$

This function is analytic on neighbourhood about the interval but we have a singularity at the horizontal ray from  $-a$  to  $-\infty$ .

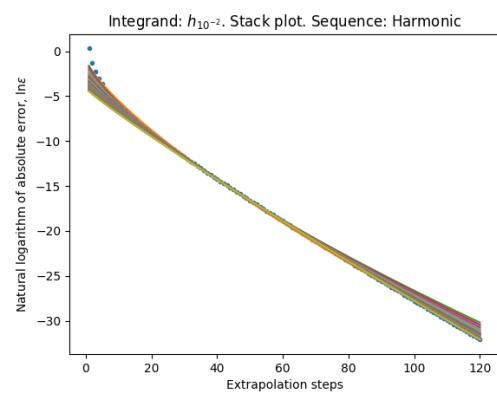
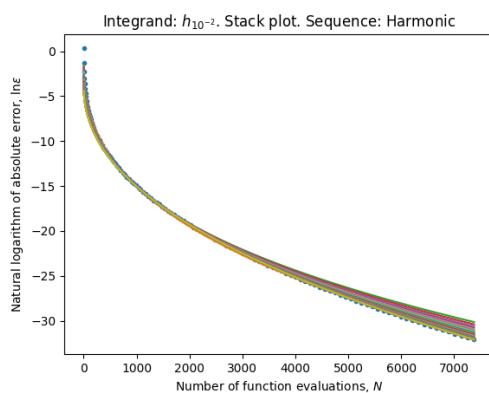
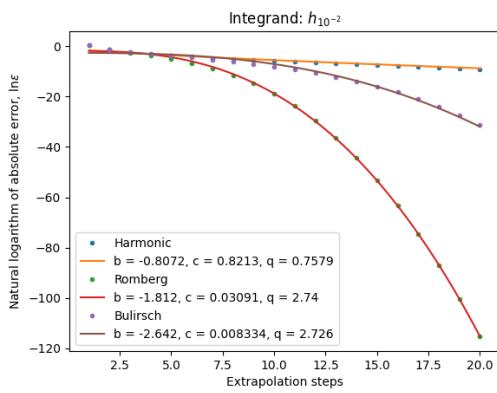
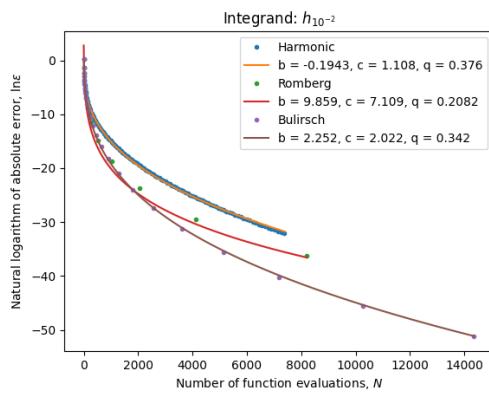
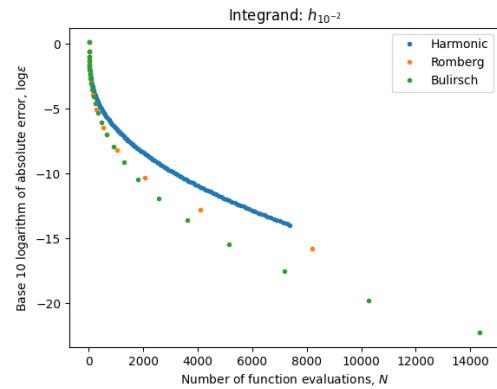
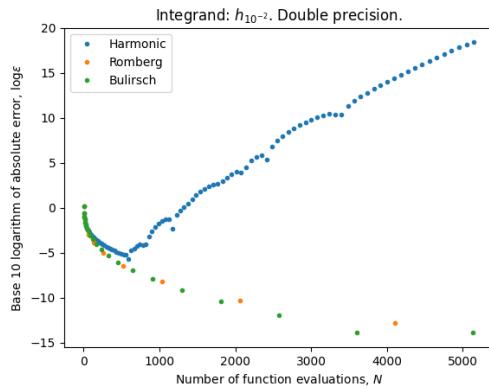


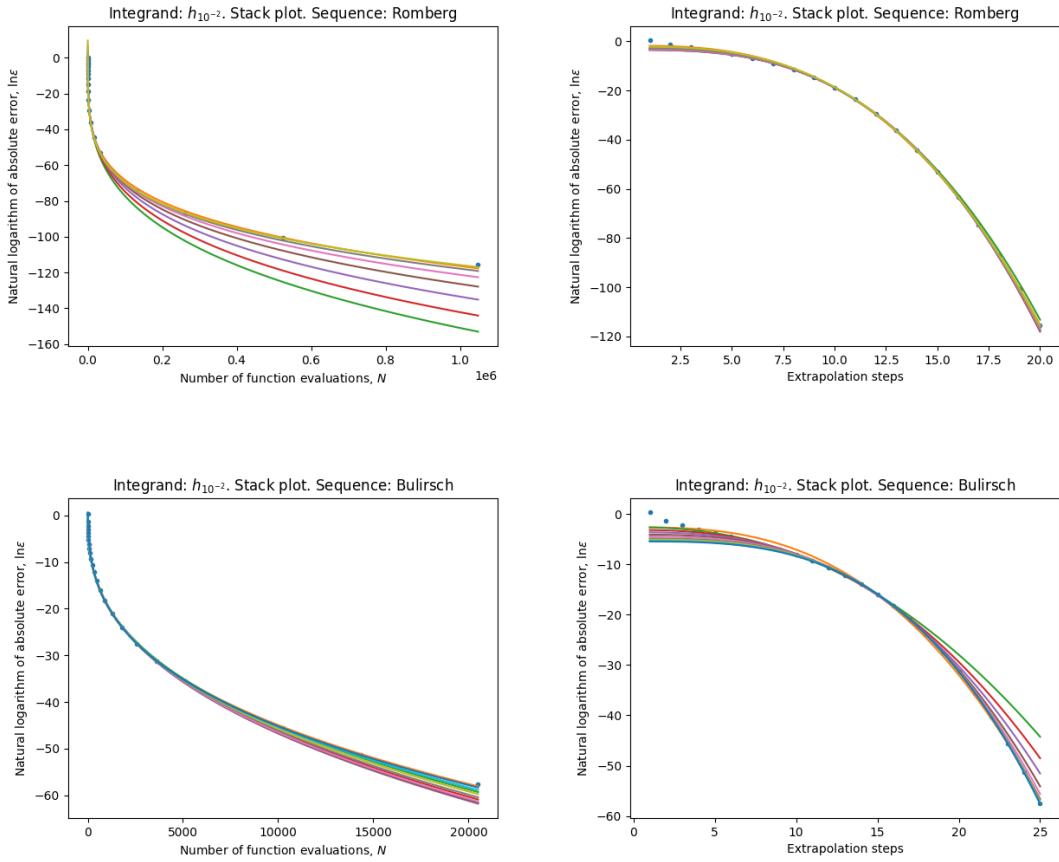


Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Harmonic	3.072	0.3632	4.545	0.000229	0.4954	$1.085 \cdot 10^{-5}$	0.5183	$6.335 \cdot 10^{-7}$
Romberg	$1.538 \cdot 10^{43}$	6	56.82	0.06603	0.1269	0.0163	4054	0.0002381
Bulirsch	$1.55 \cdot 10^{31}$	7.894	36.84	0.08131	0.1955	0.0187	7380	0.0003219
Harmonic	0.03919	0.5477	3.237	0.0003865	0.9901	$1.608 \cdot 10^{-5}$	0.7691	$1.584 \cdot 10^{-6}$
Romberg	0.0002063	1.214	0.5431	0.009515	2.02	0.000263	0.9235	$7.907 \cdot 10^{-5}$
Bulirsch	$2.293 \cdot 10^{-5}$	3.407	0.3579	0.06839	1.929	0.001656	0.9854	0.0002632

We see that the harmonic sequence performs best, then Bulirsch and then Romberg. In standard double precision arithmetic, we get down to machine level precision using Romberg or Bulirsch, but we are like 2 digits from there, using the harmonic sequence.

Here we clearly have exponential convergence in the number of evaluations for the harmonic sequence and hence also in the number of steps. For Romberg and Bulirsch we seem to have exponential convergence in the number of steps though the fitting is not as nice as in the case of the harmonic sequence.

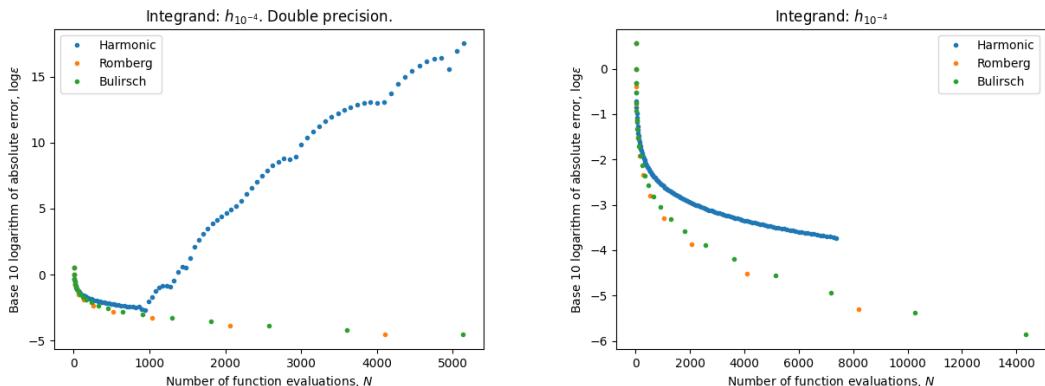


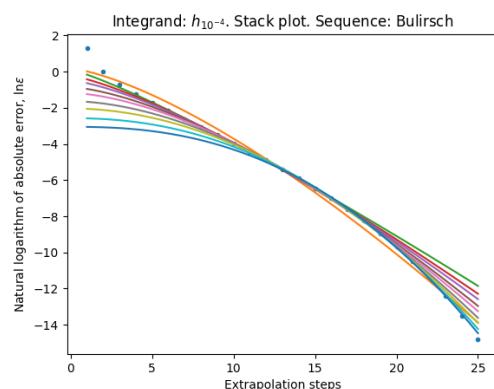
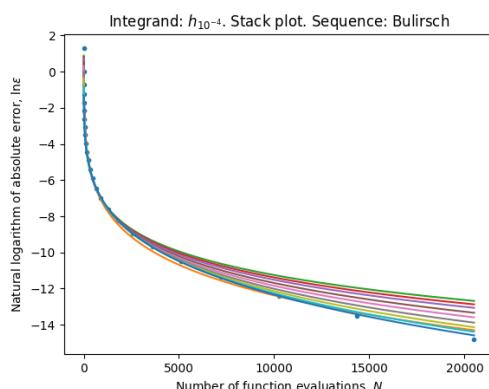
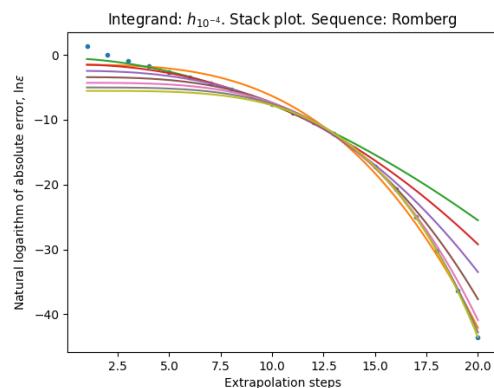
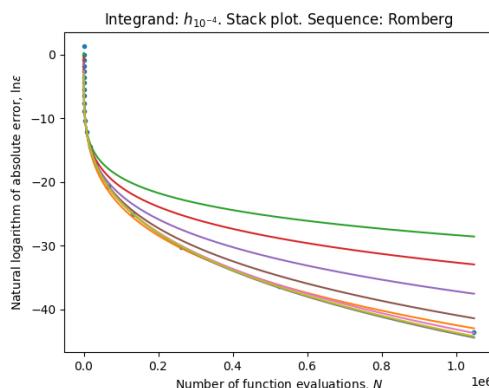
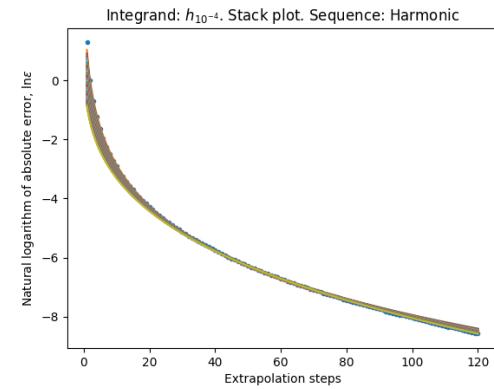
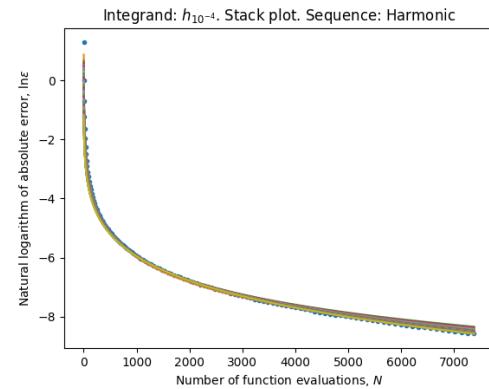
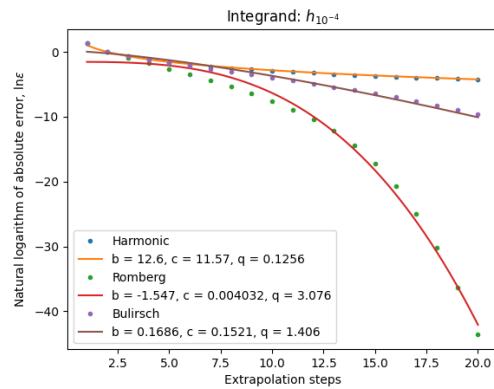
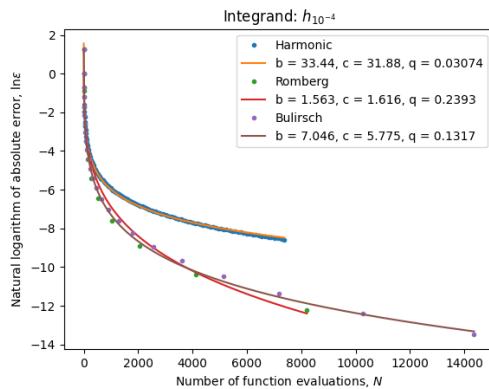


Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Harmonic	0.1351	2.455	0.6631	0.1084	0.4296	0.004684	0.7131	0.0001955
Romberg	$3.388 \cdot 10^9$	5.969	6.696	0.2315	0.2286	0.02614	7.917	0.0005581
Bulirsch	6.859	1.956	1.626	0.0311	0.3675	0.002076	0.2014	0.0001172
Harmonic	0.07535	1.798	0.4822	0.1106	0.8628	0.0042	0.7117	0.0001939
Romberg	0.05205	0.2618	0.02252	0.01836	2.848	0.0002742	0.7571	0.0001742
Bulirsch	0.0214	1.016	0.00882	1.141	2.822	0.01164	0.8803	0.00138

We see that we can not attain high precision using the harmonic sequence and standard double precision. It is hard to tell which sequence performs best in the long run, though we can say that Bulirsch performs better than Romberg.

For Romberg, we get a moderately good fit for exponential convergence in the number of steps. The fitting is quite good for the harmonic sequence but it is quite unstable for Bulirsch.





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\ln}$	$\rho_{\ln}$
Harmonic	$1.328 \cdot 10^{13}$	55.82	8.14	0.5474	0.09568	0.0936	0.1034	$9.631 \cdot 10^{-5}$
Romberg	106.1	5.445	2.037	0.7267	0.2364	0.06414	0.6139	0.001346
Bulirsch	$1.888 \cdot 10^5$	5.399	6.394	0.3315	0.1362	0.1274	0.1687	0.0008794
Harmonic	$1.395 \cdot 10^4$	17.13	5.261	0.2049	0.2157	0.04738	0.04431	$4.038 \cdot 10^{-5}$
Romberg	0.1362	2.091	0.02426	2.55	3.039	0.07013	0.8591	0.004444
Bulirsch	0.4121	0.6973	0.09479	0.8435	1.695	0.05417	0.4717	0.003557

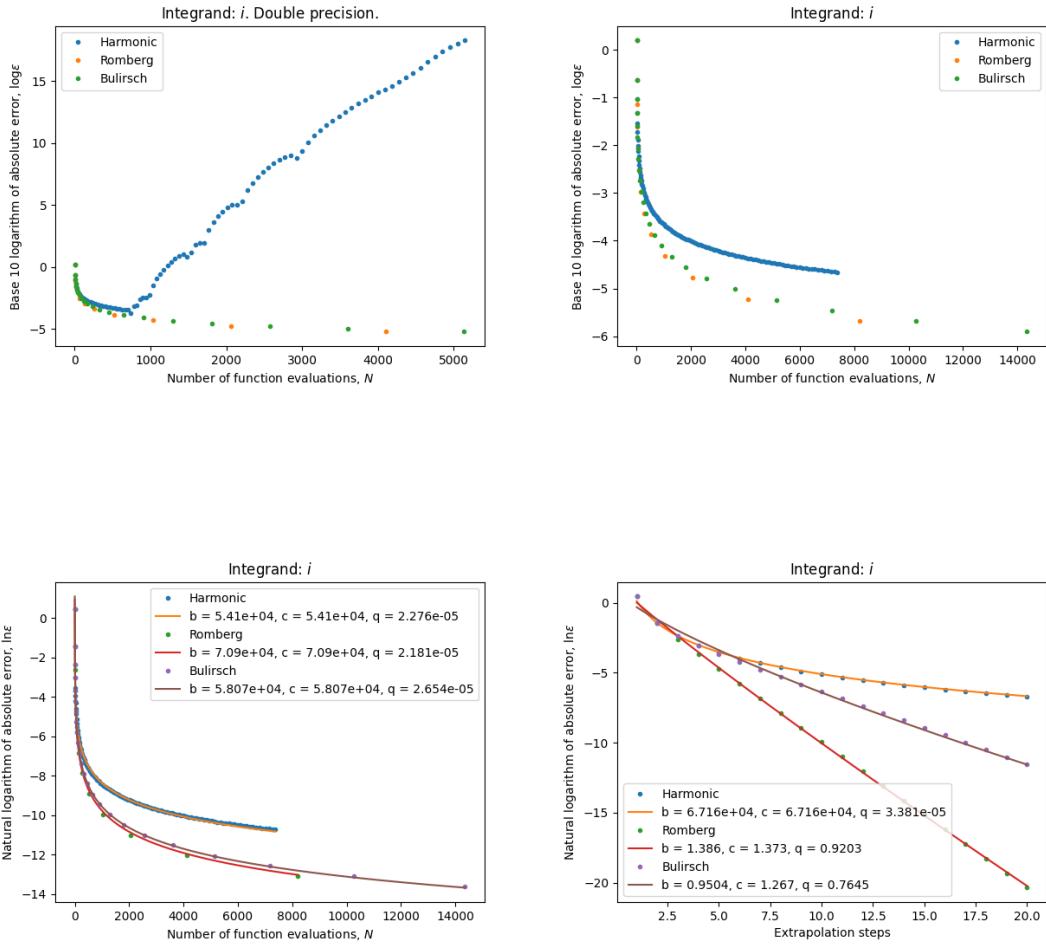
Here again, we do not attain high precision when using the Harmonic sequence in double precision arithmetic. It is hard to say which sequence performs best. We get reasonably good fit for the harmonic sequence but not for Romberg and Bulirsch.

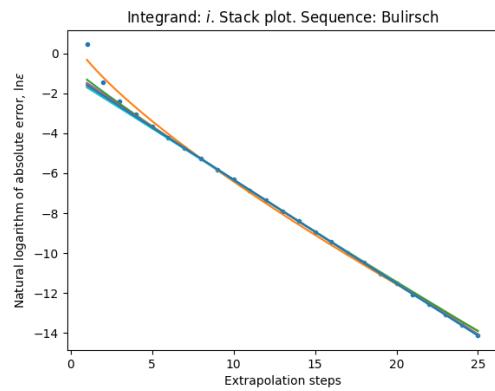
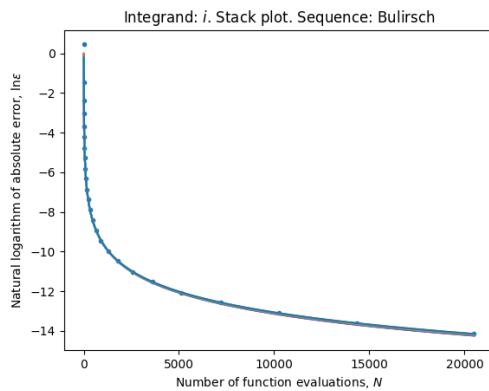
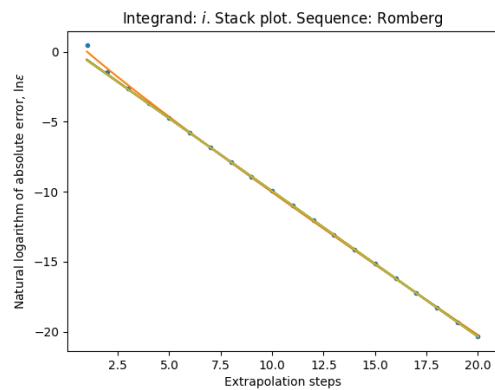
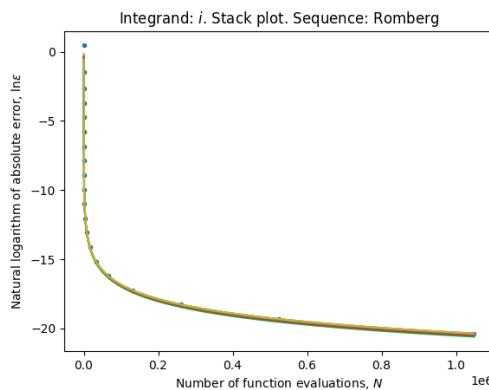
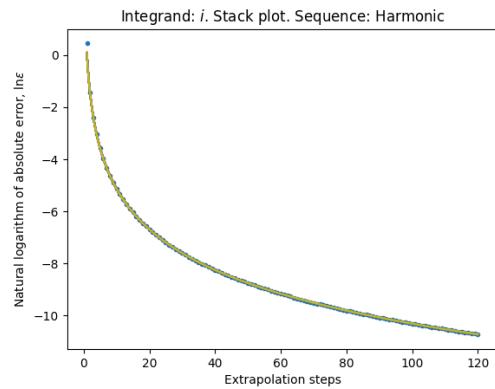
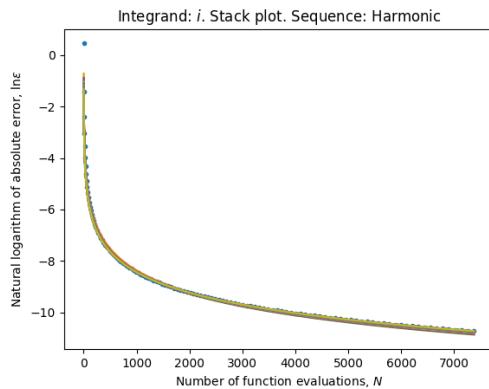
### 2.2.4 Area of half circle

Now we will try the following function:

$$i : [-1, 1] \rightarrow \mathbb{R}, \quad i(x) := \sqrt{1 - x^2}.$$

This function is analytic inside the interval of definition but not at the endpoints. Its derivative has singularities at the endpoints.





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Harmonic	.	.	$5.358 \cdot 10^4$	0.003851	$2.176 \cdot 10^{-5}$	0.00346	.	0.0002451
Romberg	.	.	$7.348 \cdot 10^4$	0.005727	$2.071 \cdot 10^{-5}$	0.005124	.	0.0002178
Bulirsch	.	.	$5.763 \cdot 10^4$	0.0003158	$2.647 \cdot 10^{-5}$	0.0002409	.	0.0001299
Harmonic	.	.	$6.694 \cdot 10^4$	0.0002388	$3.364 \cdot 10^{-5}$	0.0002515	.	$1.351 \cdot 10^{-5}$
Romberg	1.581	0.0005141	1.046	$5.395 \cdot 10^{-5}$	0.9983	$4.956 \cdot 10^{-6}$	0.1245	0.0001394
Bulirsch	0.375	0.03054	0.5595	0.00836	0.9807	0.000633	0.2822	0.0007036

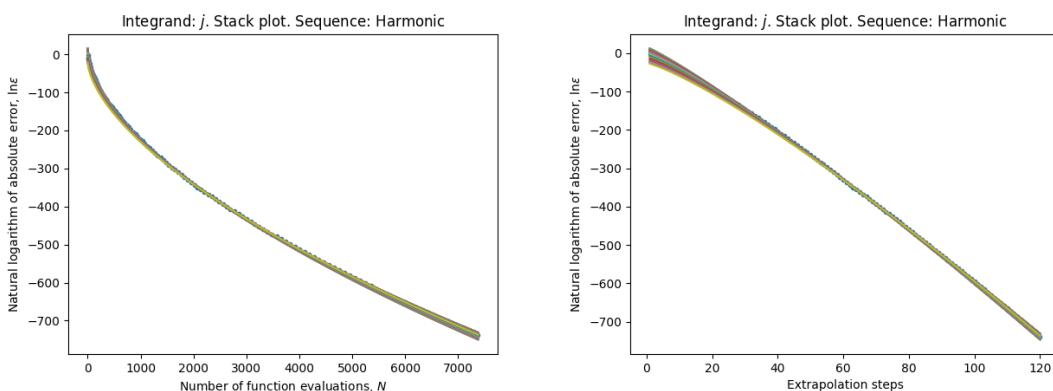
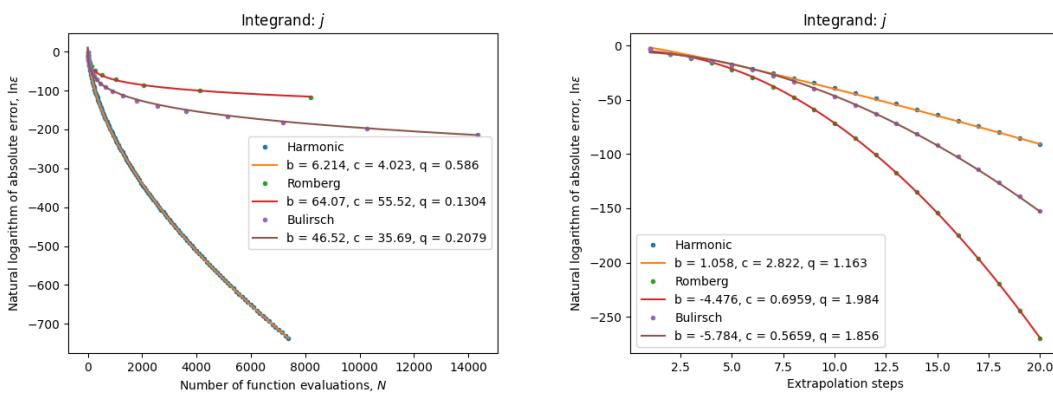
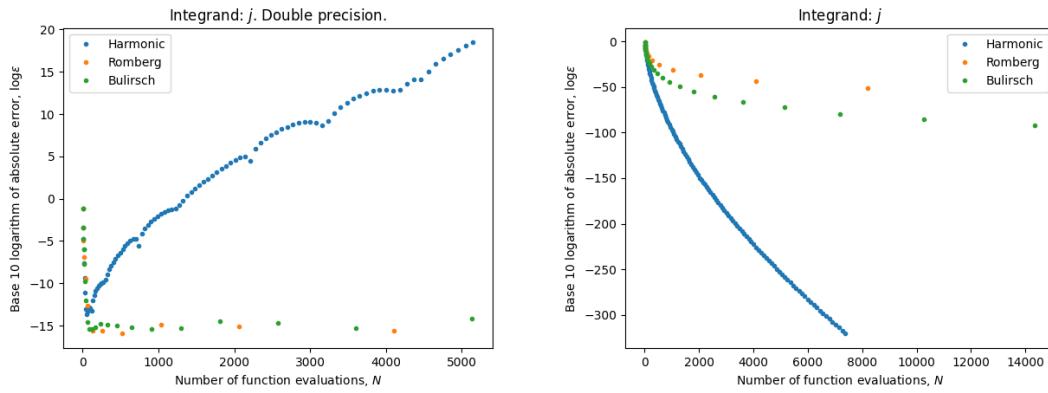
We see that we do not get high precision using double precision arithmetic, independent of sequence. The Romberg and Bulirsch sequence seem to perform similarly well but the harmonic sequence seems to be slowest.

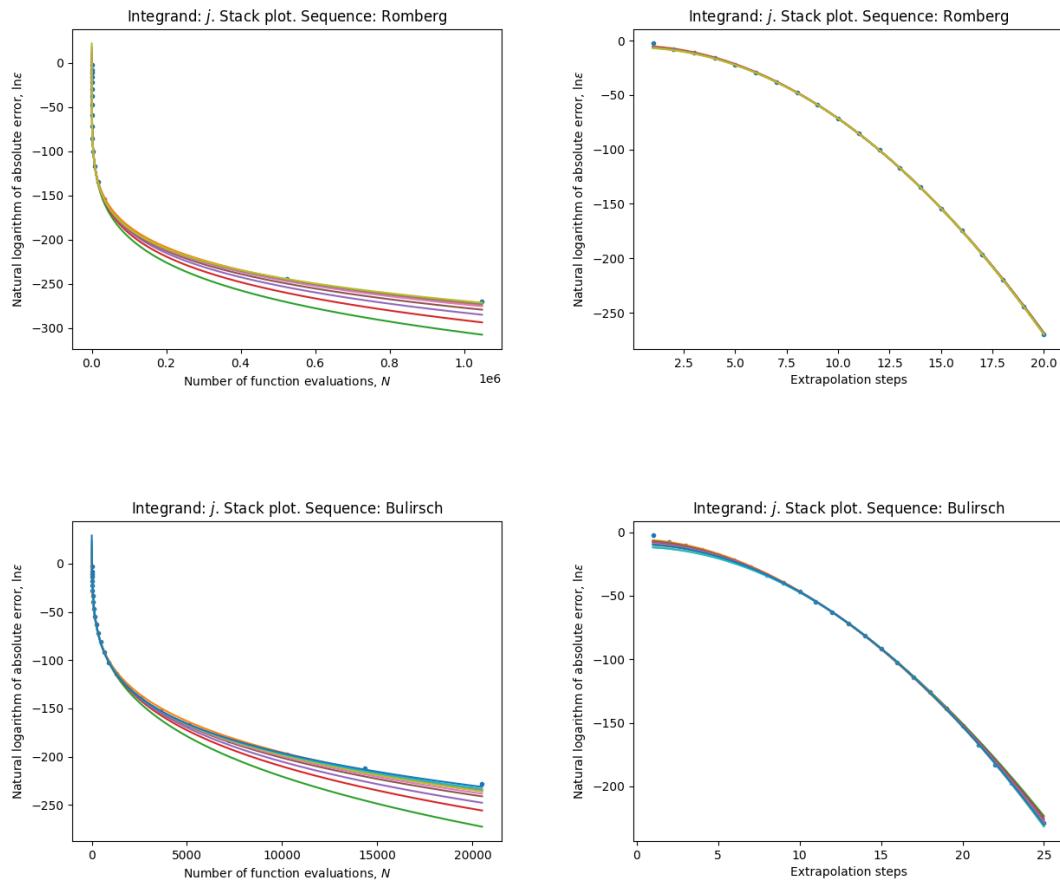
For the harmonic sequence we reject the fitting because we get unreasonale values of the parameters. The model for exponential convergence in the number of steps fits well in the case of the Romberg sequence and Bulirsch.

### 2.2.5 Gaussian

Finally we will consider the Gaussian function

$$j : [0, 1] \rightarrow \mathbb{R}, \quad k(x) := \frac{2}{\sqrt{\pi}} e^{-x^2}.$$





Sequence	A-mean	A-var	c-mean	c-var	q-mean	q-var	$\rho_{\ln}$	$\rho_{\ln}$
Harmonic	lin- $\ln$ evals-error	$1.037 \cdot 10^9$	5.523	4.257	0.02142	0.5814	0.0007265	
Romberg	lin- $\ln$ evals-error	$1.398 \cdot 10^{55}$	6	68.42	0.08326	0.1248	0.02265	
Bulirsch	lin- $\ln$ evals-error	$1.356 \cdot 10^{51}$	8	49.55	0.1212	0.1931	0.02978	
Harmonic	lin- $\ln$ steps-error	$6.922 \cdot 10^5$	5.362	2.939	0.02212	1.157	0.0006628	
Romberg	lin- $\ln$ steps-error	0.003928	0.1462	0.659	0.0008259	2	$2.366 \cdot 10^{-5}$	
Bulirsch	lin- $\ln$ steps-error	0.0003995	1.479	0.4704	0.02518	1.913	0.0007081	

In double precision arithmetic we get down to machine level precision using Romberg or Bulirsch, but we get down to like 2 digits from there, using the harmonic sequence. The harmonic sequence performs best, then Bulirsch and then Romberg.

For the Harmonic sequence it is hard to tell whether we have exponential convergence because we get very large value for  $A$ . For Romberg we seem to have exponential convergence in the number steps and also for Bulirsch.

# Chapter 3

## Extrapolation of difference quotients

### 3.1 The algorithm

Let  $a \in \mathbb{R}$ ,  $\varepsilon > 0$  and  $f : ]a - \varepsilon, a + \varepsilon[ \rightarrow \mathbb{R}$  be differentiable at  $a$ . We are interested in estimating  $f'(a)$ . Assume that  $f$  is  $2k + 1$  times differentiable at  $a$ . Then by Taylor's theorem we have

$$f(a + h) = f(a) + f'(a)h + \frac{f''(a)}{2}h^2 + \dots + \frac{f^{(2k)}(a)}{(2k)!}h^{2k} + \frac{f^{(2k+1)}(\xi)}{(2k+1)!}h^{2k+1} \quad (3.1)$$

where  $a < \xi < a + h$ . Now plug  $-h$  instead of  $h$  in (3.1):

$$f(a - h) = f(a) - f'(a)h + \frac{f''(a)}{2}h^2 - \dots + \frac{f^{(2k)}(a)}{(2k)!}h^{2k} - \frac{f^{(2k+1)}(\eta)}{(2k+1)!}h^{2k+1} \quad (3.2)$$

where  $a - h < \eta < a$ . If we subtract (3.2) from (3.1) and divide by  $2h$  we get:

$$f'(a) = D_f(h) + \frac{f'''(a)}{3!}h^2 + \dots + \frac{f^{(2k-1)}(a)}{(2k-1)!}h^{2k-2} + \frac{f^{(2k+1)}(\xi) + f^{(2k+1)}(\eta)}{2 \cdot (2k+1)!}h^{2k} \quad (3.3)$$

where

$$D_f(h) := \frac{f(a + h) - f(a - h)}{2h} \quad (3.4)$$

is the *symmetric difference quotient* of  $f$  at  $a$ . Note that  $\frac{1}{2}(f^{(2k+1)}(\xi) + f^{(2k+1)}(\eta))$  is in the image of  $f^{(2k+1)}$  so we can rewrite (3.3) as

$$f'(a) = D_f(h) + \frac{f'''(a)}{3!}h^2 + \dots + \frac{f^{(2k-1)}(a)}{(2k-1)!}h^{2k-2} + \frac{f^{(2k+1)}(\zeta)}{(2k+1)!}h^{2k} \quad (3.5)$$

where  $a - h < \zeta < a + h$ . Formula (3.5) tells us that the symmetric difference quotient method has asymptotic expansion in  $h^2$  of order  $2k - 2$  if  $f$  is  $2k + 1$  times differentiable. Thus we can use the following scheme to extrapolate the symmetric difference quotient method:

1.  $D_{i1} := D_f(h_i)$  for  $i = 1, \dots, k$ .
2.  $D_{ij} := D_{i,j-1} + \frac{D_{i,j-1} - D_{i-1,j-1}}{\left(\frac{h_{i-j+1}}{h_i}\right)^2 - 1}$  for  $2 \leq j \leq i$ .

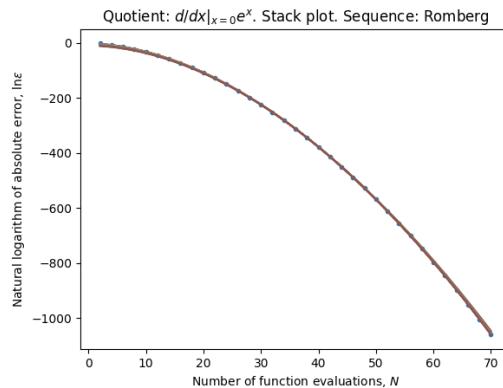
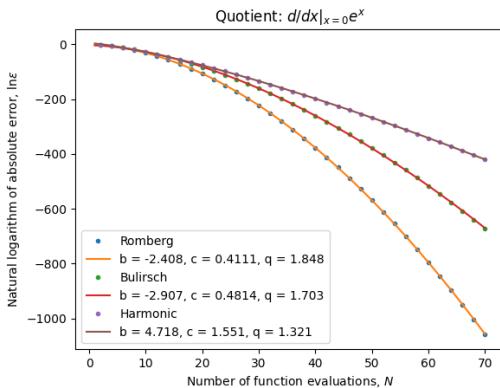
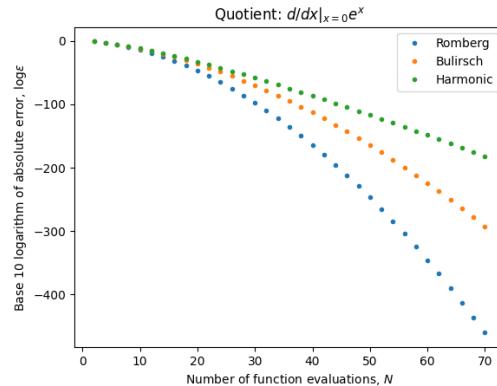
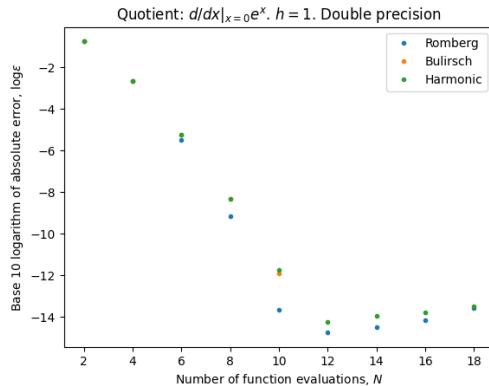
## 3.2 Numerical experiments

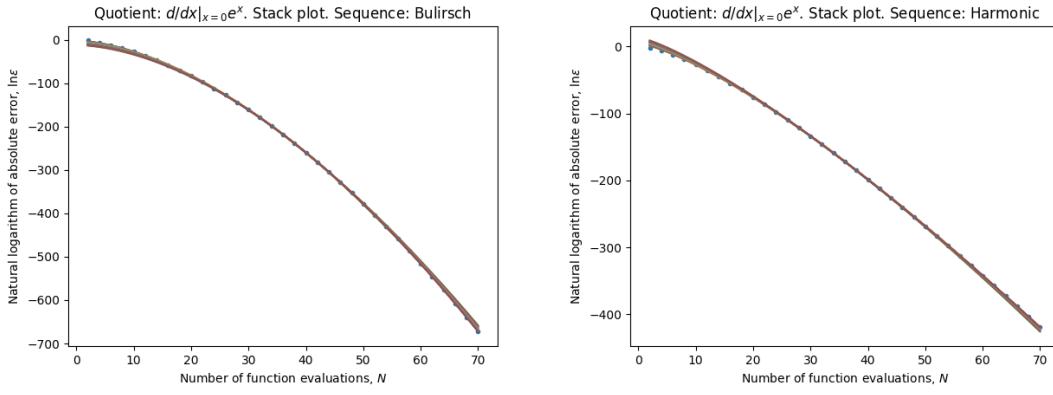
In this section we are going to extrapolate the symmetric difference quotient for approximating the derivative of a function at a given point. Let  $h > 0$  be some number,  $f : [a - \varepsilon, a + \varepsilon] \rightarrow \mathbb{R}$  a function differentiable at  $a$  and  $n_1 < n_2 < \dots$  a sequence of integers. Let  $h_i := h/n_i$ . Let  $D_{ij}$  be the extrapolation table that we get from extrapolating in  $h^2$  using the points  $(h_1^2, D_f(h_1)), (h_2^2, D_f(h_2)), \dots$ , as we described in the first chapter. We let  $\varepsilon_i := |X_{ii} - f'(a)|$ . We want to analyse how  $\varepsilon_i$  as  $i$  increases and we also want to do similar efficiency analysis as in the chapter on Romberg quadrature and check whether we have exponential convergence. We will do the computations with precision up to 500 significant digits and also using standard double precision arithmetic.

Now we will consider the results of the experiments.

### 3.2.1 The exponential function

We begin by considering the derivative of the exponential function at zero.





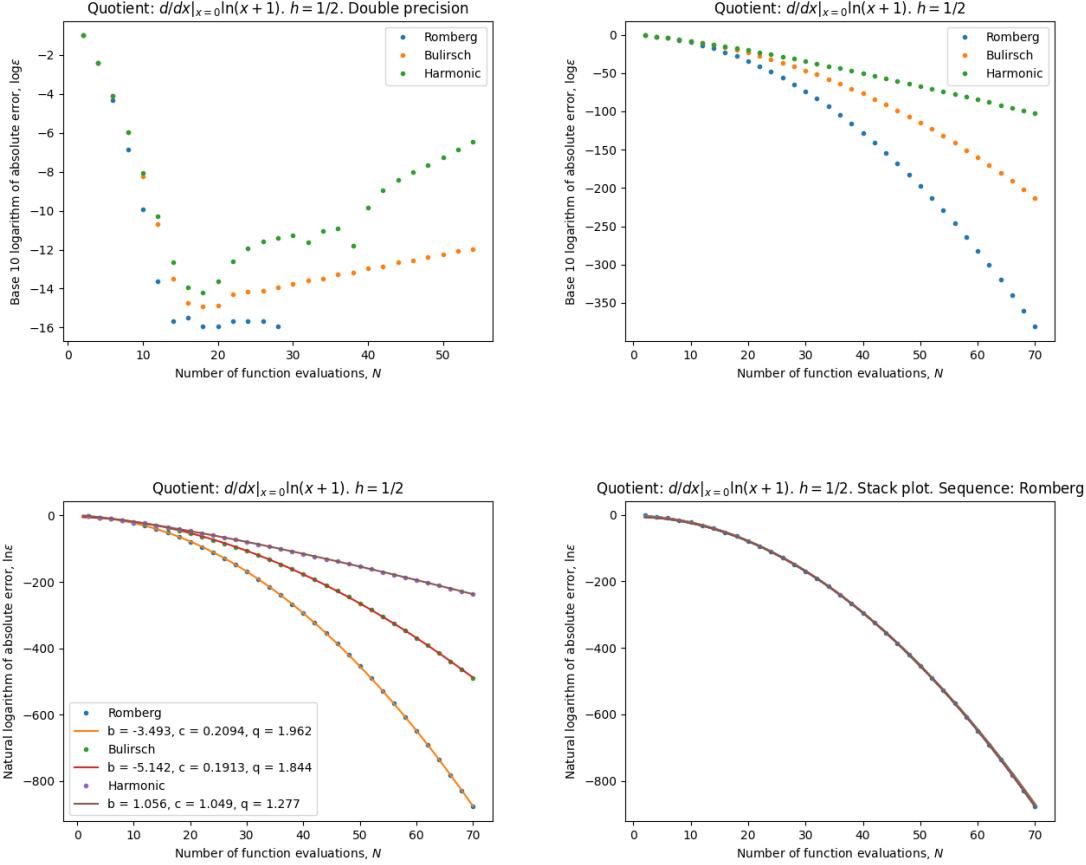
Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	0.03687	2.584	0.4004	0.004019	1.853	$6.702 \cdot 10^{-5}$	0.7797	$3.025 \cdot 10^{-6}$
Bulirsch	0.01953	3.508	0.4544	0.0119	1.716	0.0002258	0.8741	$9.934 \cdot 10^{-6}$
Harmonic	$1.122 \cdot 10^5$	3.293	1.732	0.007202	1.299	0.0002359	150	$1.225 \cdot 10^{-5}$

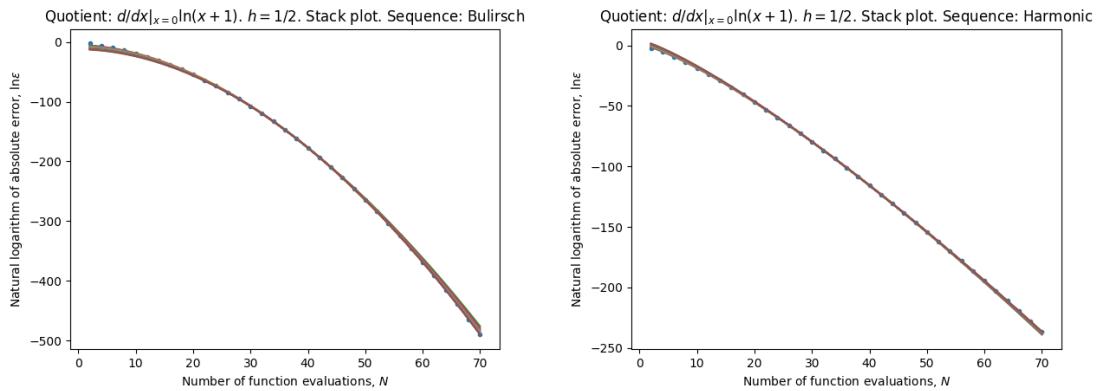
In standard floating point arithmetic, we get down to machine level precision using any sequence. The Romberg sequence works best, then Bulirsch and then the harmonic. The model seems to fit well in all cases.

### 3.2.2 Logarithm

Now we will consider the derivative at zero of the function

$$g(x) := \ln(x + 1).$$





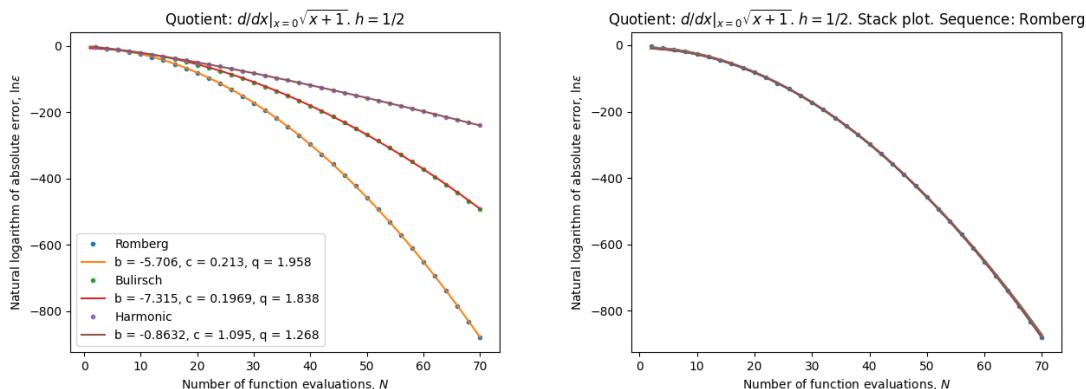
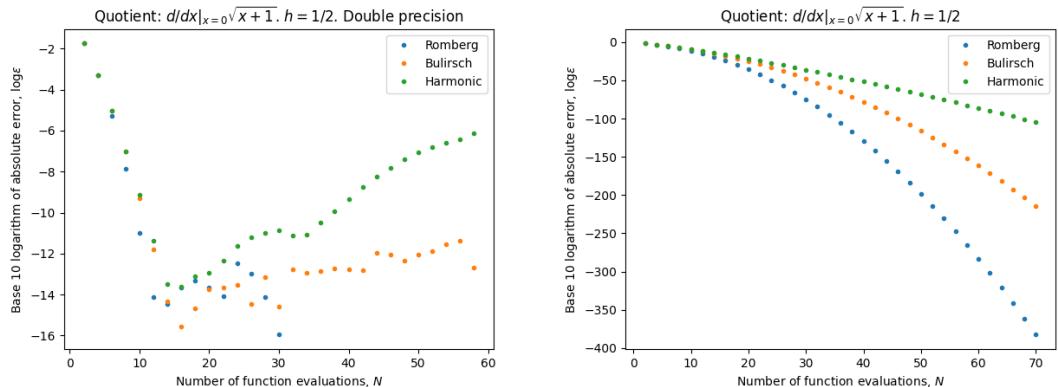
Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	0.009098	0.9823	0.2041	0.001354	1.967	$2.052 \cdot 10^{-5}$	0.7453	$2.051 \cdot 10^{-6}$
Bulirsch	0.001179	3.144	0.1757	0.01703	1.863	0.0002721	0.9409	$2.144 \cdot 10^{-5}$
Harmonic	27.62	0.9622	1.126	0.003487	1.263	0.0001174	1.666	$4.495 \cdot 10^{-6}$

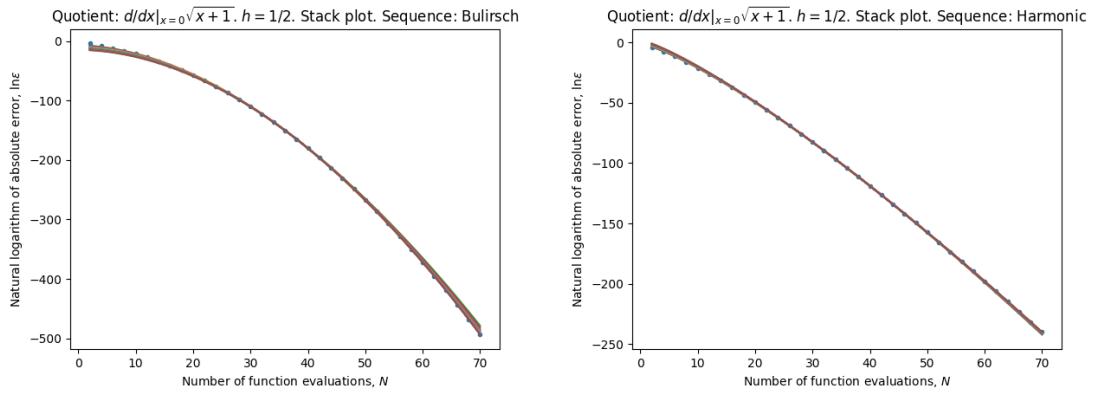
We get down to machine level precision using any sequence, Romberg performs best, then Bulirsch and then the Harmonic one. The model fits well in all cases.

### 3.2.3 Square root

Now we shall consider the derivative at zero of the following function:

$$h(x) := \sqrt{1+x}$$





Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	0.0008261	1.263	0.2064	0.001755	1.965	$2.659 \cdot 10^{-5}$	0.8417	$2.946 \cdot 10^{-6}$
Bulirsch	0.0001188	3.504	0.1792	0.01927	1.859	0.0003075	0.9629	$2.551 \cdot 10^{-5}$
Harmonic	2.598	0.8154	1.155	0.002659	1.257	$8.929 \cdot 10^{-5}$	0.5049	$2.483 \cdot 10^{-6}$

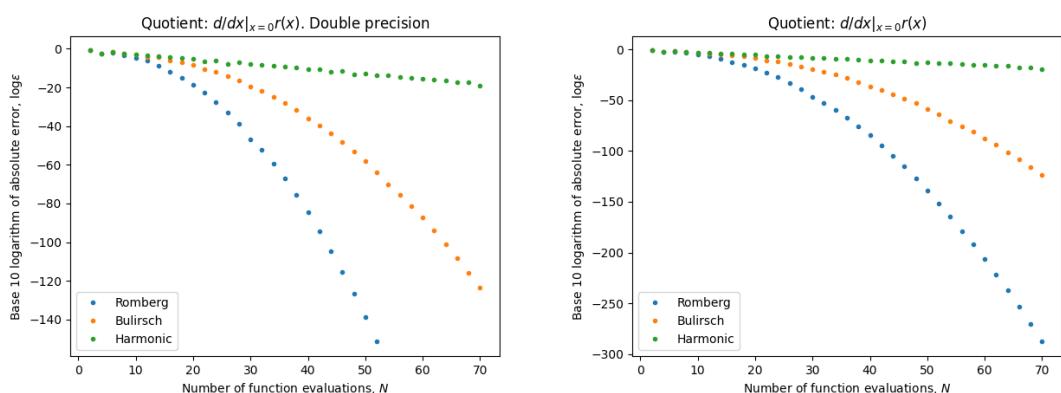
In standard double precision floating point arithmetic we get down to machine level precision using any sequence. The model fits well in all cases.

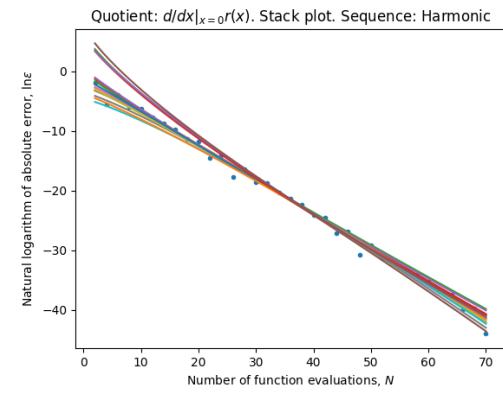
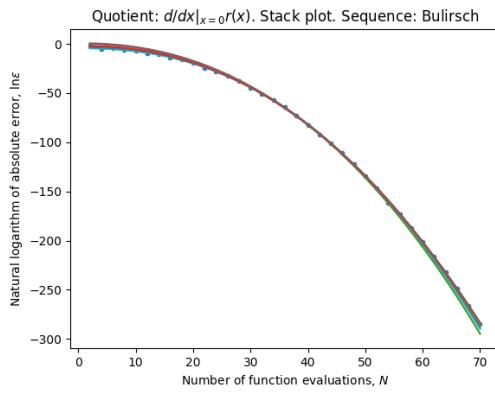
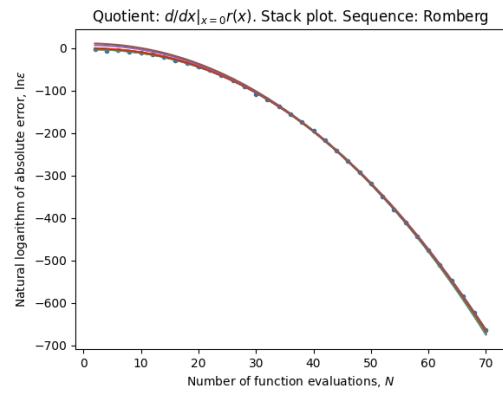
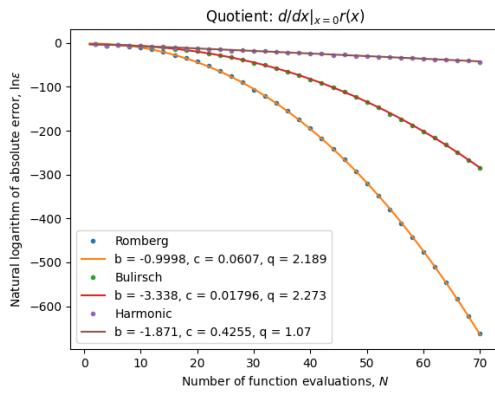
### 3.2.4 Smooth but not analytic function

Now we will consider the derivate at zero of the following function:

$$r(x) := \begin{cases} e^{-1/x} & \text{if } x > 0 \\ 0 & \text{else} \end{cases}$$

which is smooth but not analytic.





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	7305	12.55	0.06014	0.02339	2.195	0.0002104	1.666	$2.198 \cdot 10^{-5}$
Bulirsch	0.2728	4.406	0.0178	0.03004	2.281	0.000316	0.5886	$3.574 \cdot 10^{-5}$
Harmonic	287.5	4.85	0.818	0.4753	0.9831	0.02252	0.3061	0.001189

Romberg performs best and the harmonic sequence worst.

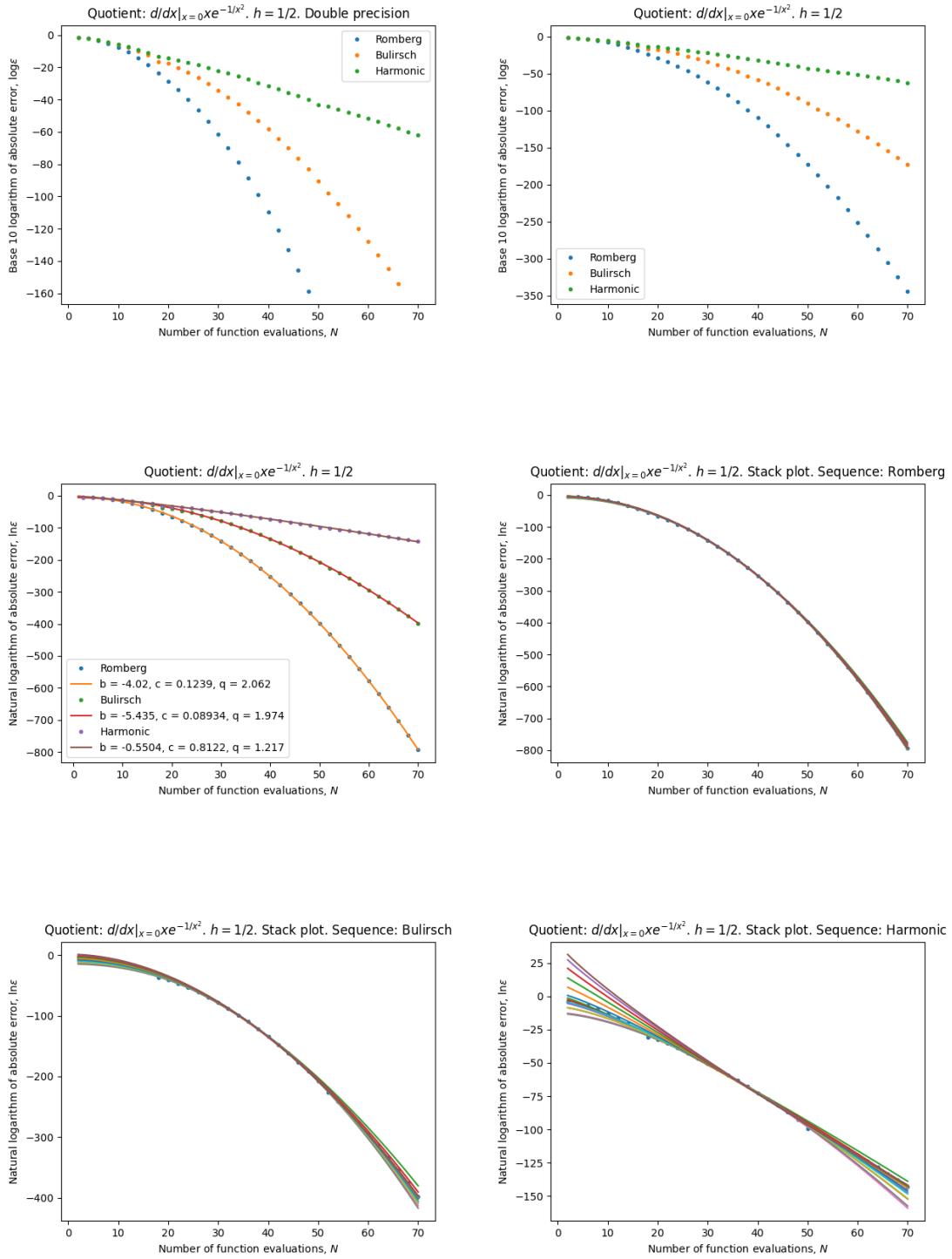
The model seems to fit reasonably well for the Romberg and Bulirsch sequence but the fitting is not so nice for the harmonic sequence.

### 3.2.5 Another smooth but not analytic function

Now we will consider the derivative at zero of the following function:

$$i(x) := \begin{cases} xe^{-1/x^2} & \text{if } x \neq 0 \\ 0 & \text{else} \end{cases}$$

which is smooth but not analytic.



Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	0.009202	1.693	0.1182	0.00696	2.073	$9.612 \cdot 10^{-5}$	0.1993	$4.392 \cdot 10^{-6}$
Bulirsch	0.4381	5.98	0.07965	0.09556	2.015	0.001606	0.6906	$4.969 \cdot 10^{-5}$
Harmonic	$2.061 \cdot 10^{16}$	12.84	1.483	1.04	1.216	0.03736	12.73	0.0001837

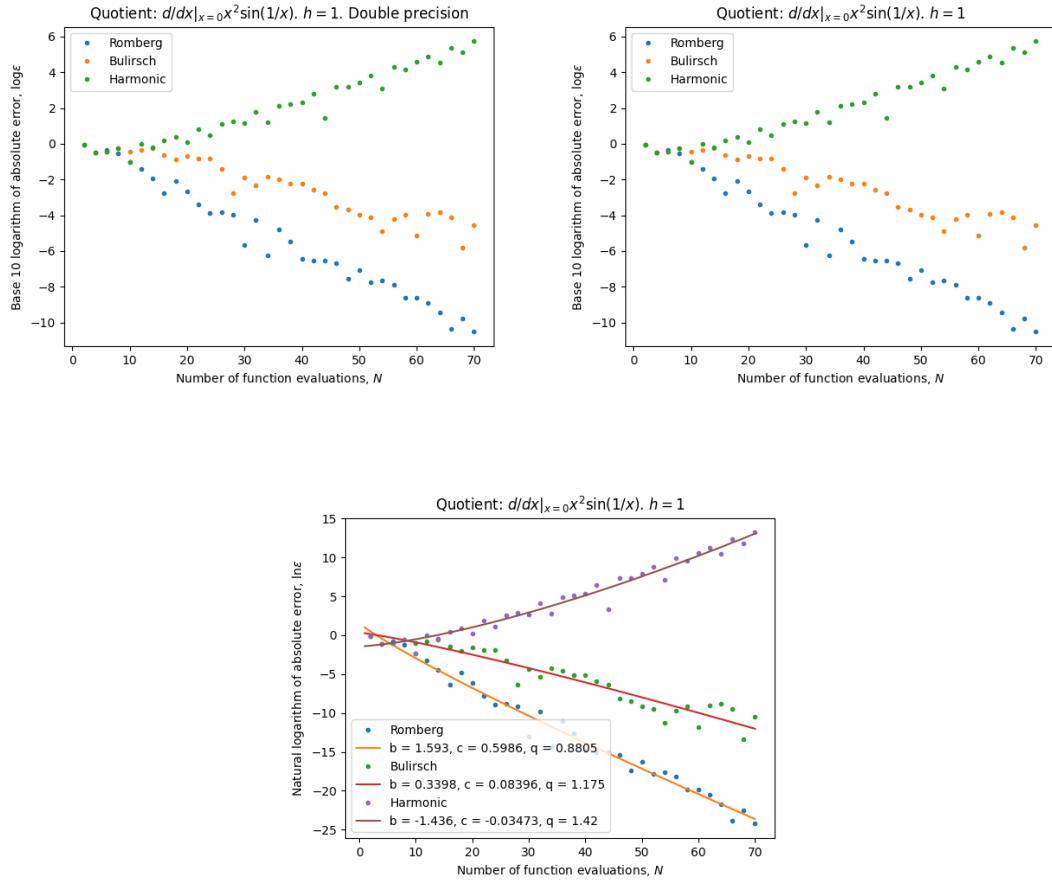
Here Romberg performs best and the harmonic sequence worst.

We seem to have nice fit for the Romberg sequence, moderately good for the Bulirsch sequence but not so good for the harmonic sequence.

### 3.2.6 Only once differentiable function

Finally we will consider the derivate at zero of the following function which is only once differentiable at that point:

$$j(x) := \begin{cases} x^2 \sin \frac{1}{x} & \text{if } x \neq 0 \\ 0 & \text{else} \end{cases}.$$



Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$1.99 \cdot 10^{27}$	13	6.139	3.832	0.8069	0.397	0.7138	0.004549
Bulirsch	.	.	2316	4.918	1.24	0.4409	0.3599	0.0283
Harmonic	0.3968	1.653	-0.3656	6.72	1.307	0.09291	0.09287	0.01539

Here the model simply does not fit. Note that we do not have the asymptotic expansion for the derivate here, since the function is only once differentiable.

# Chapter 4

## Initial Value Problems

### 4.1 The explicit midpoint rule

Let  $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a smooth mapping and consider the initial value problem

$$y'(t) = f(t, y(t)), \quad y(a) = y_a, \quad t \in [a, b]. \quad (4.1)$$

The explicit midpoint methods is a method for computing an approximation to the solution of (4.1), and it goes as follows: Let  $n \geq 1$  be an integer and  $h := (b - a)/2n$ . We then define recursively

$$\xi_h(a) := y_a, \quad \xi_h(a + h) := \xi_h(a) + h f(a, \xi_h(a))$$

and

$$\xi_h(a + (i + 1)h) := \xi_h(a + (i - 1)h) + 2h f(a + ih, \xi_h(a + ih)).$$

Then  $\xi_h$  is an approximate solution to (4.1) defined at  $a, a + h, \dots, b$ . We are interested in the value  $X_f(h) := \xi_h(b)$ . It is possible to show that  $X_f(h)$  has an asymptotic expansion in  $h^2$ . We have the following implementation in Python of the explicit midpoint rule for computing  $X_f(h)$ .

```
class ExplicitMidpointRule(Scheme):

    def __init__(self):
        super(ExplicitMidpointRule, self).__init__(2)

    def apply(self, ivp, n):
        h = (ivp.b - ivp.a) / (2 * n)
        y_s1 = ivp.y0
        y_l = ivp.y0 + h * ivp.f(ivp.a, ivp.y0)

        for i in range(1, 2 * n):
            tmp = y_l
            y_l = y_s1 + 2 * h * ivp.f(ivp.a + i * h, y_l)
            y_s1 = tmp

        return y_l
```

### 4.2 Numerical experiments

In this section we are going to extrapolate the explicit midpoint rule and analyze the convergence of the approximations as we extrapolate more often. Consider the initial value

problem (4.1). Let  $n_1 < n_2 < \dots$  be some sequence of integers and  $h_i := (b - a)/n_i$ . Let  $X_{ij}$  the extrapolation table which we get from extrapolating in  $h^2$ , using the points  $(h_i, X_f(h_i))$ . Let  $\varepsilon_i := |X_{ii} - y(b)|$  be the absolute error. We are going to do the same convergence and efficiency analysis as in the two previous chapters. We will both do the computations using high precision arithmetic with 500 correct digits and also in standard double precision.

In those cases where we do not have an analytic solution to the equations, we computed a reference solution up to high precision. We did that by using extrapolation with the harmonic sequence and estimating the error as the difference between successive terms in the sequence of approximations.

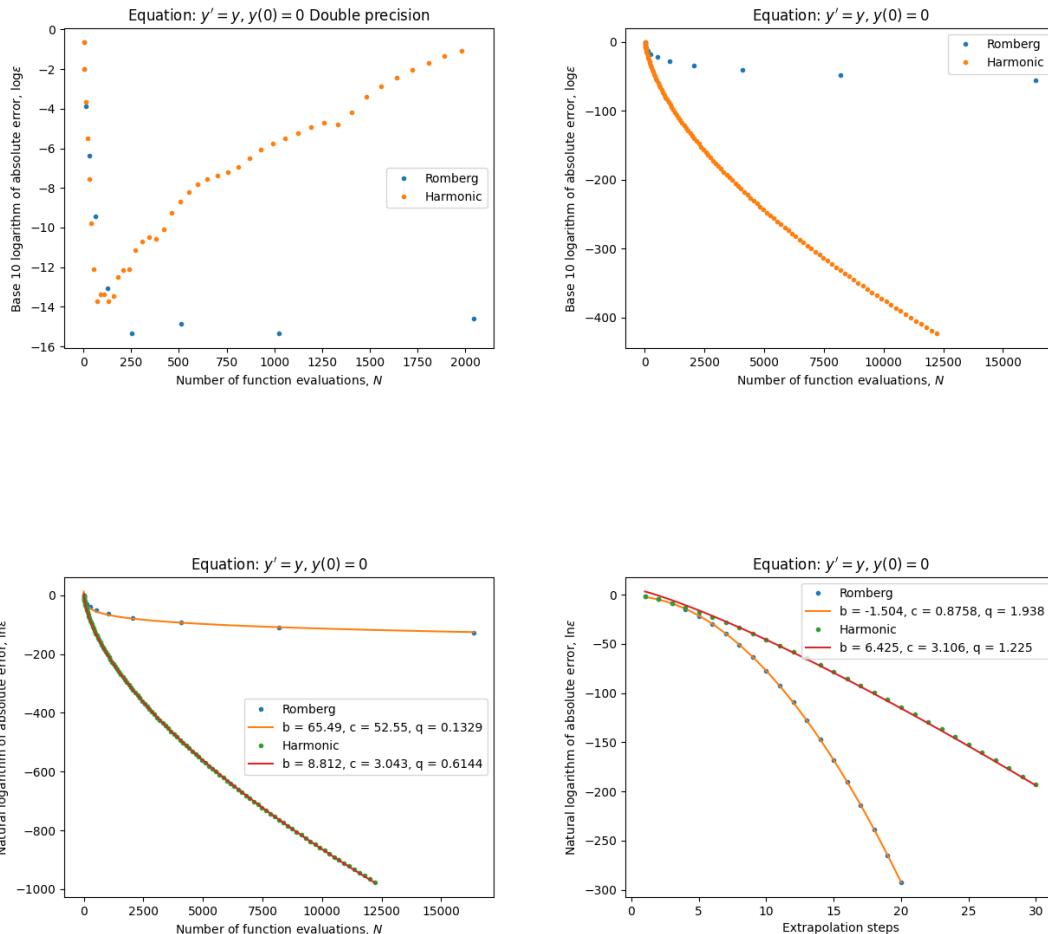
Now we will consider the results of the experiments.

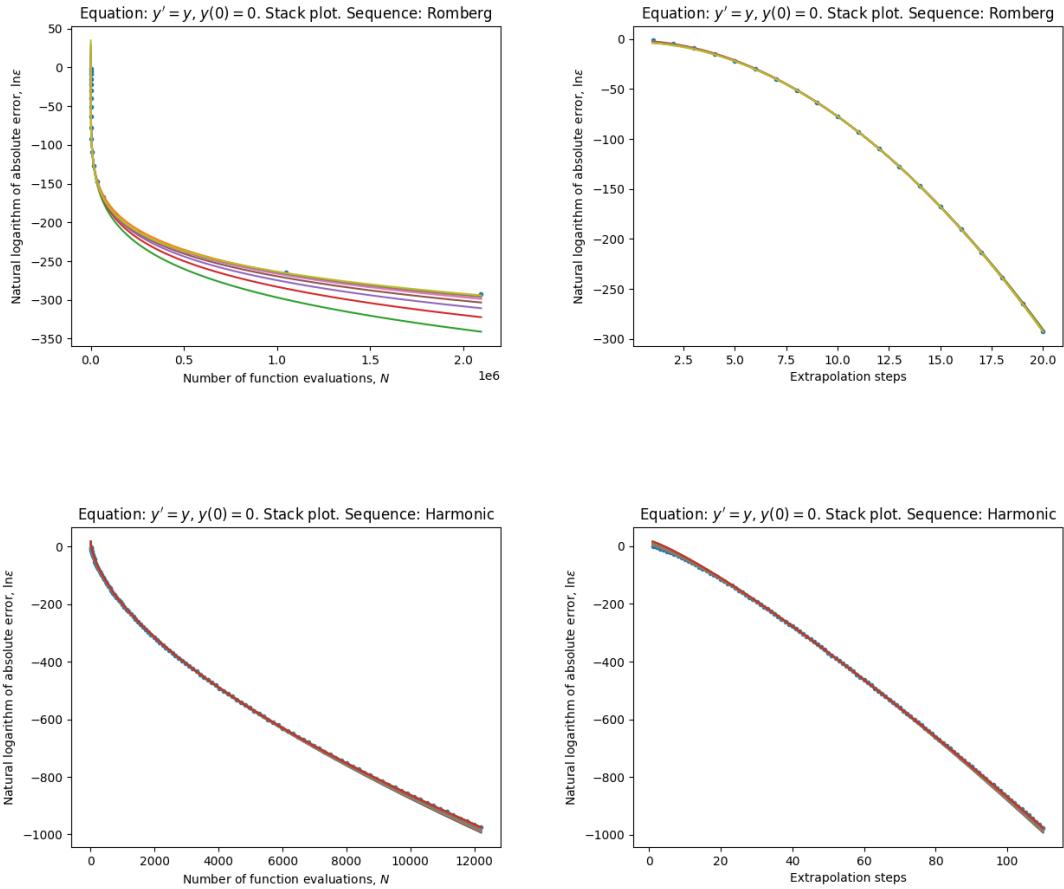
### 4.2.1 Exponential growth

First we will consider the following initial value problem:

$$y'(x) = y(x), \quad y(0) = 0, \quad x \in [0, 1] \quad (4.2)$$

whose solution is the analytic function  $y(x) = e^x$ .





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$6.231 \cdot 10^{65}$	6	71.92	0.1174	0.1236	0.03122	$1.415 \cdot 10^8$	0.0006922
Harmonic	$1.204 \cdot 10^9$	6.365	3.264	0.005221	0.6081	0.0001653	$8.428 \cdot 10^4$	$5.564 \cdot 10^{-6}$
Romberg	0.0873	0.2223	0.841	0.0009203	1.95	$2.797 \cdot 10^{-5}$	0.3309	$4.499 \cdot 10^{-6}$
Harmonic	$3.805 \cdot 10^7$	6.024	3.309	0.004502	1.214	0.0001425	$1.574 \cdot 10^4$	$4.503 \cdot 10^{-6}$

The Harmonic sequence performs better. We get down to machine level precision using either sequence in double precision arithmetic.

We clearly have exponential convergence in the number of steps for the Romberg sequence and the fitting for the Harmonic sequence seems nice but we though have very big values for  $A$ .

### 4.2.2 Logistic curve

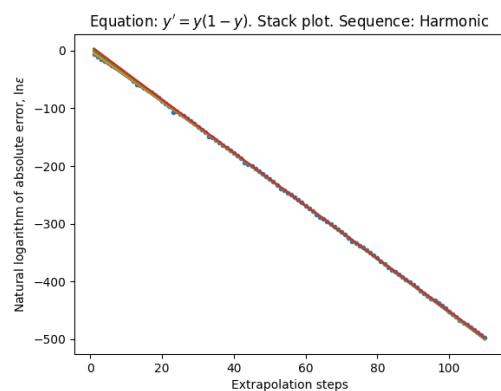
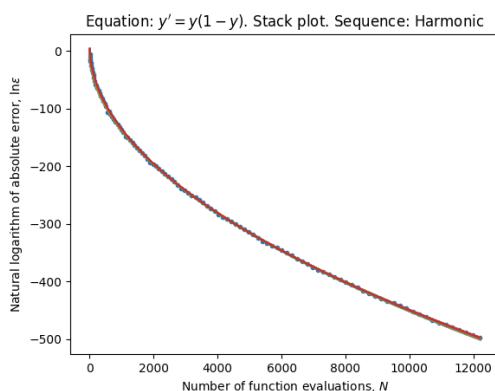
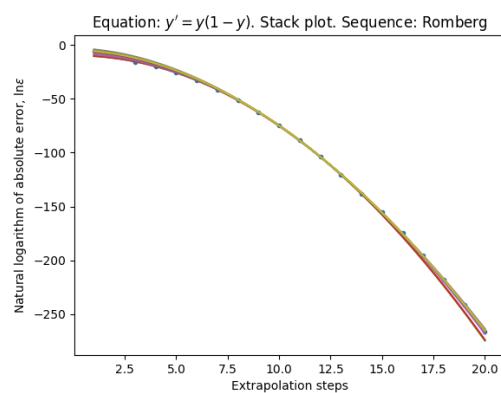
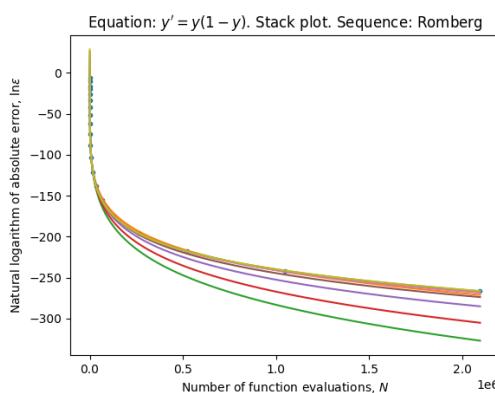
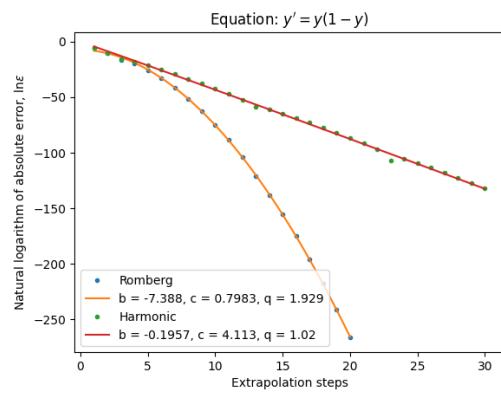
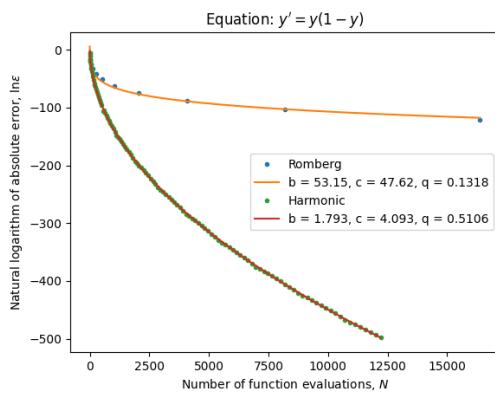
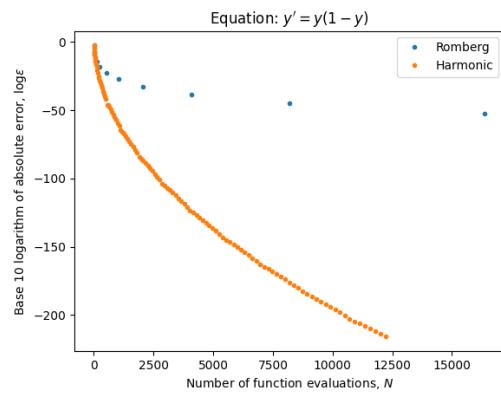
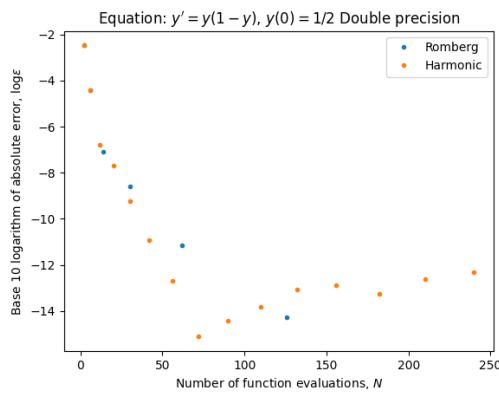
Then we will consider the following initial value problem

$$y'(x) = y(x)(1 - y(x)), \quad y(0) = 1/2, \quad x \in [0, 1] \quad (4.3)$$

whose solution is the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

which is analytic.



Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$8.848 \cdot 10^{63}$	6	70.69	0.1952	0.1226	0.0637	$6.36 \cdot 10^5$	0.0006121
Harmonic	2510	9.384	4.249	0.00357	0.5073	0.0001329	18.59	$1.332 \cdot 10^{-5}$
Romberg	0.01026	1.856	0.814	0.03122	1.932	0.001091	0.839	$4.568 \cdot 10^{-5}$
Harmonic	257.8	9.223	4.256	0.003464	1.014	0.0001294	9.348	$1.304 \cdot 10^{-5}$

The harmonic sequence performs better and we get down to machine level precision using either sequence, in double precision.

We seem to have exponential convergence in the number of steps for the Romberg sequence and the models fit well for the harmonic sequence.

### 4.2.3 Tangens

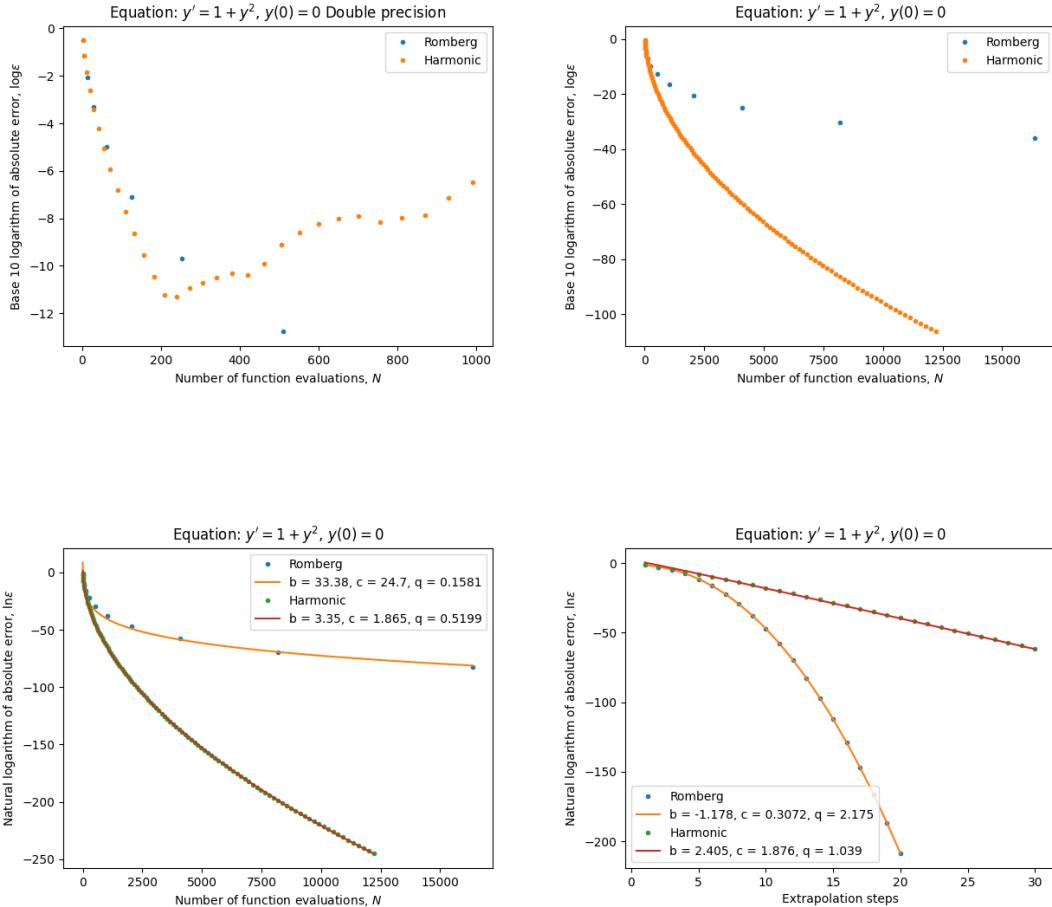
Now we will consider the following equation

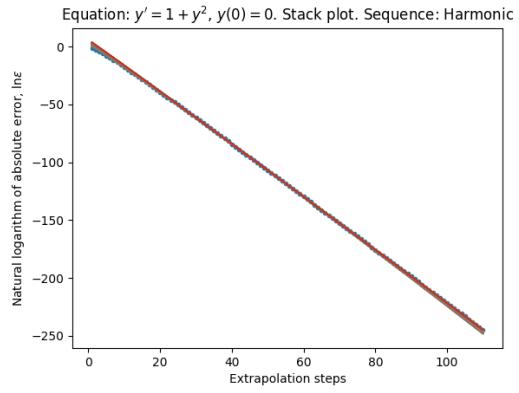
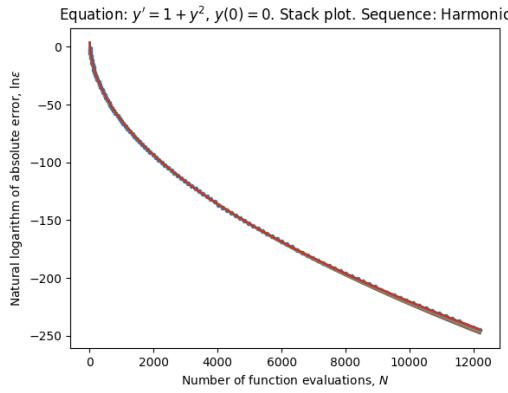
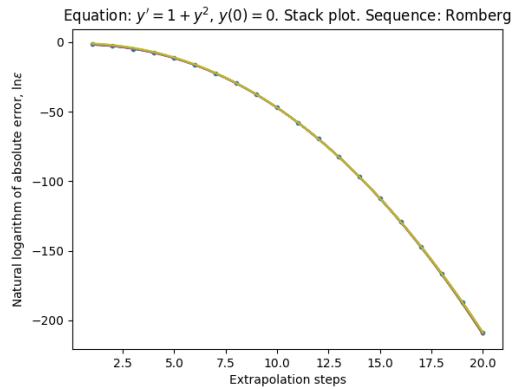
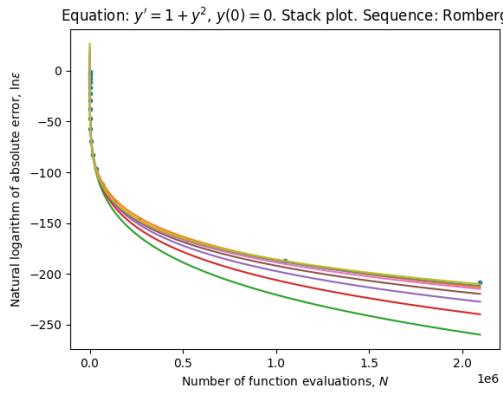
$$y'(x) = 1 + y(x)^2, \quad y(0) = 0, \quad x \in [0, 1] \quad (4.4)$$

whose solution is

$$y(x) := \tan(x)$$

which is meromorphic and we are quite far from singularities.





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	$9.496 \cdot 10^{36}$	6	33.25	0.1689	0.1523	0.03648	$1.13 \cdot 10^6$	0.0009519
Harmonic	358.8	0.5652	2.017	0.002488	0.5127	0.000109	27.95	$5.92 \cdot 10^{-6}$
Romberg	0.3476	0.1097	0.3039	0.001043	2.18	$2.579 \cdot 10^{-5}$	0.06654	$1.294 \cdot 10^{-6}$
Harmonic	121.5	0.5373	2.021	0.002309	1.025	0.0001013	19.67	$5.328 \cdot 10^{-6}$

The harmonic sequence performs better and we get down to machine level precision in double precision arithmetic, using either sequence.

Here we clearly have exponential convergence in the number of steps for the Romberg sequence and the fit is also very nice for the harmonic sequence.

#### 4.2.4 The logarithm

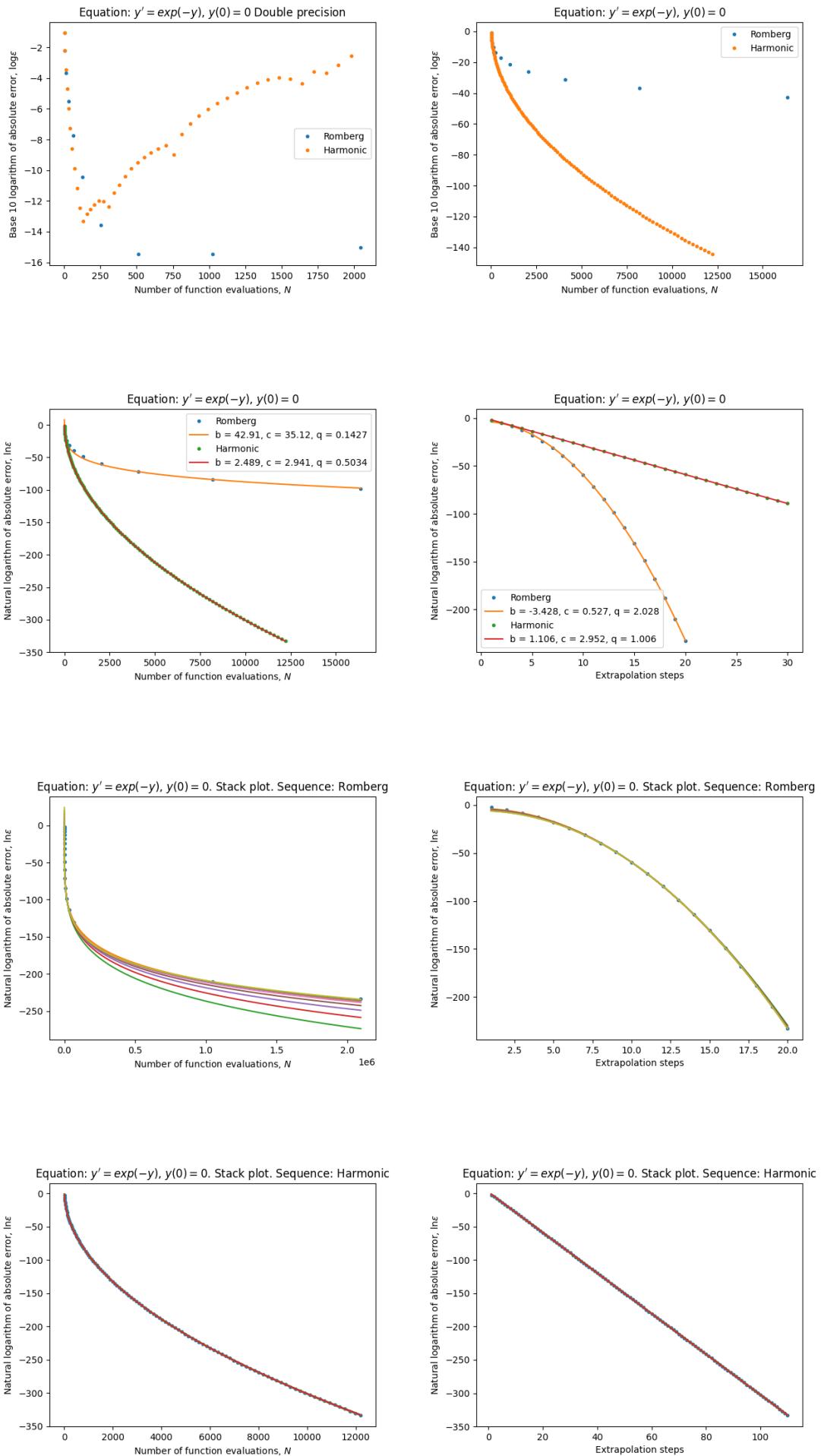
Now we will consider the following initial value problem:

$$y'(t) = \exp(-y(t)), \quad y(0) = \ln(a), \quad t \in [0, 1]. \quad (4.5)$$

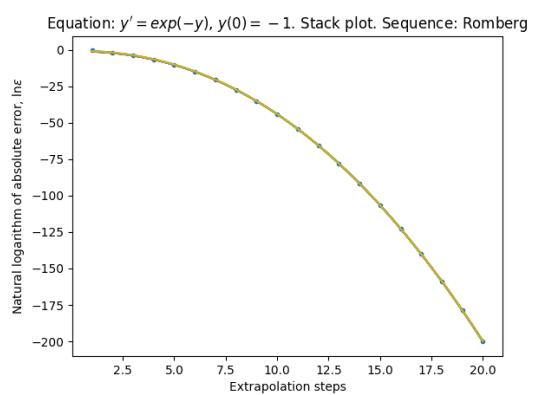
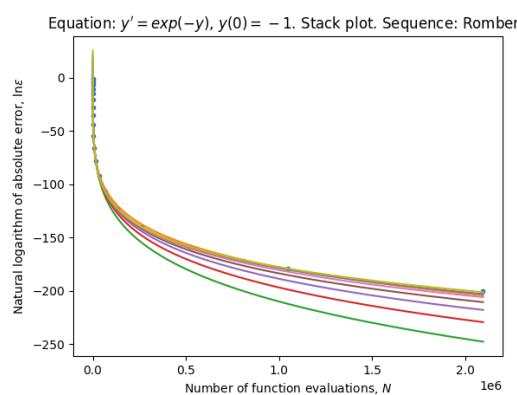
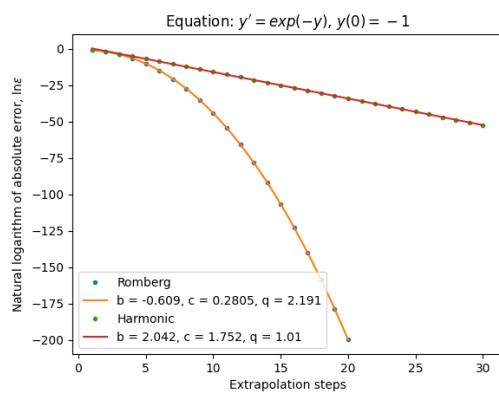
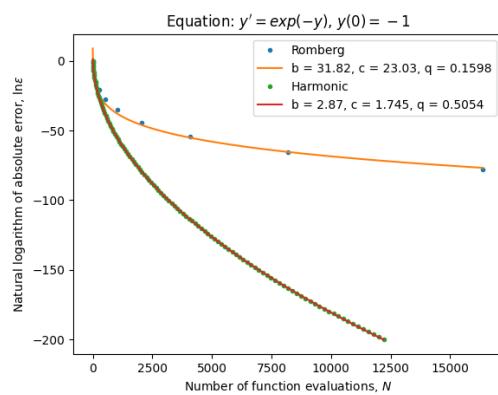
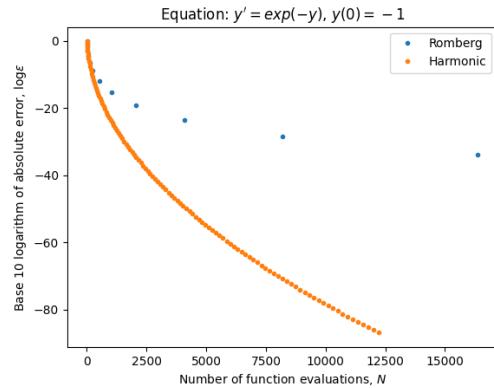
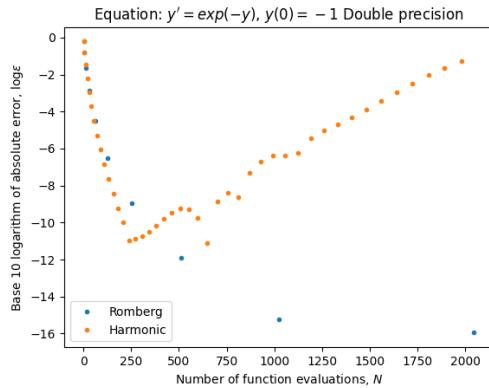
whose solution is

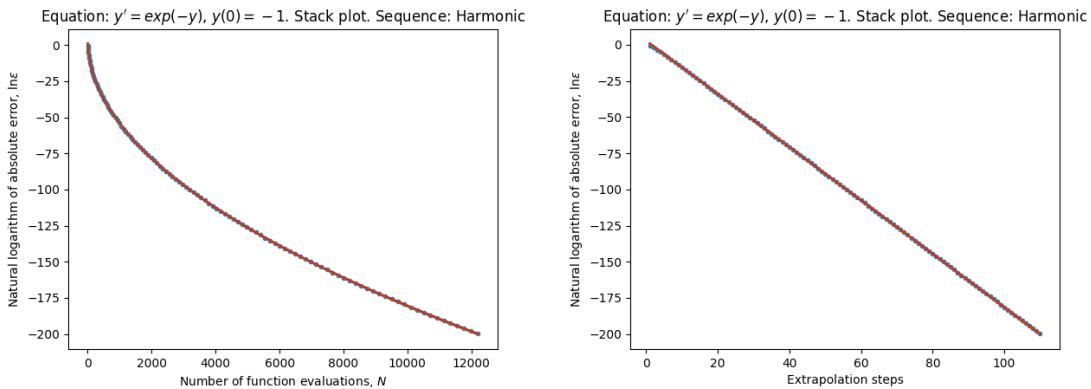
$$y(t) = \ln(a + t).$$

The solution is analytic on but with a singularity on the closed horizontal ray from  $-a$  to  $-\infty$ .

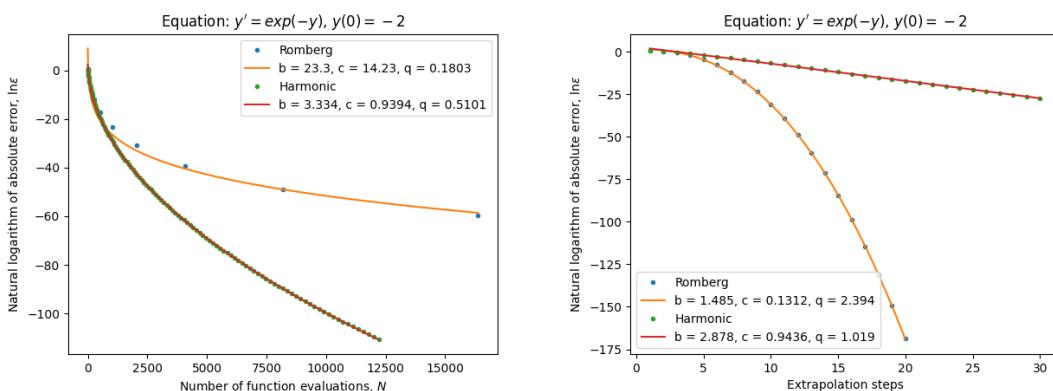
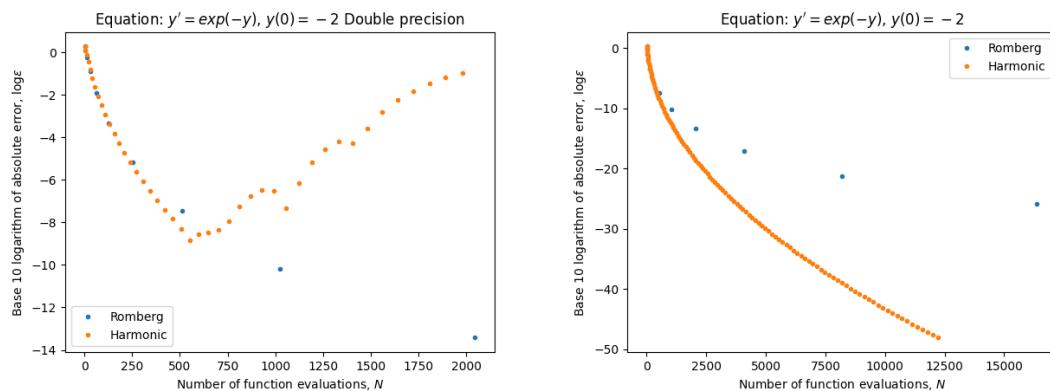


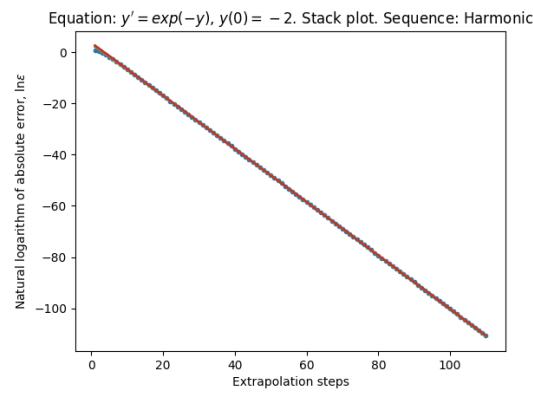
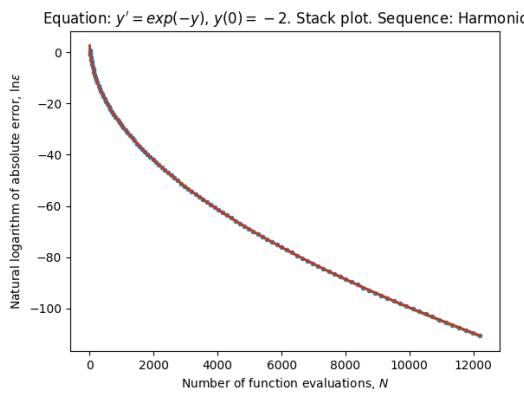
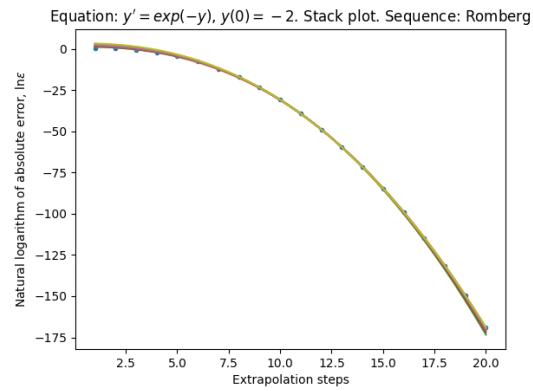
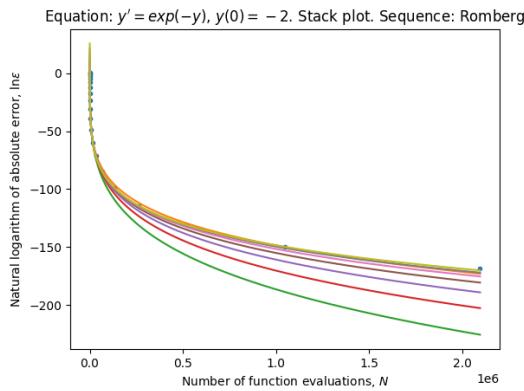
Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$4.885e + 42$	6	45.32	0.1197	0.1365	0.02779	$5.241e + 05$	$0.0006028$
Harmonic	21.43	0.04237	2.986	$6.169e - 05$	0.5019	$2.669e - 06$	1.32	$2.916e - 07$
Romberg	0.007492	0.413	0.4813	0.003324	2.056	$9.134e - 05$	0.6043	$2.141e - 05$
Harmonic	4.89	0.03478	2.989	$4.918e - 05$	1.004	$2.126e - 06$	0.6747	$1.905e - 07$



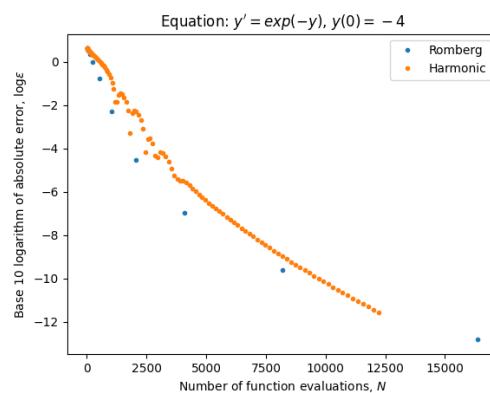
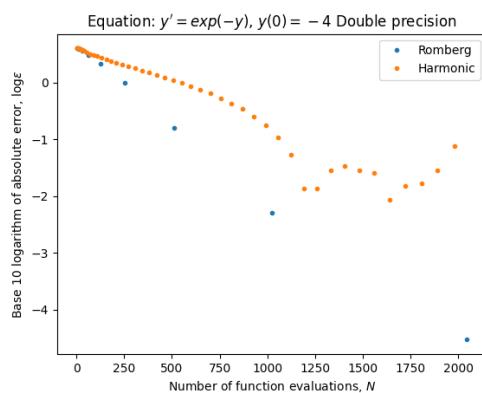


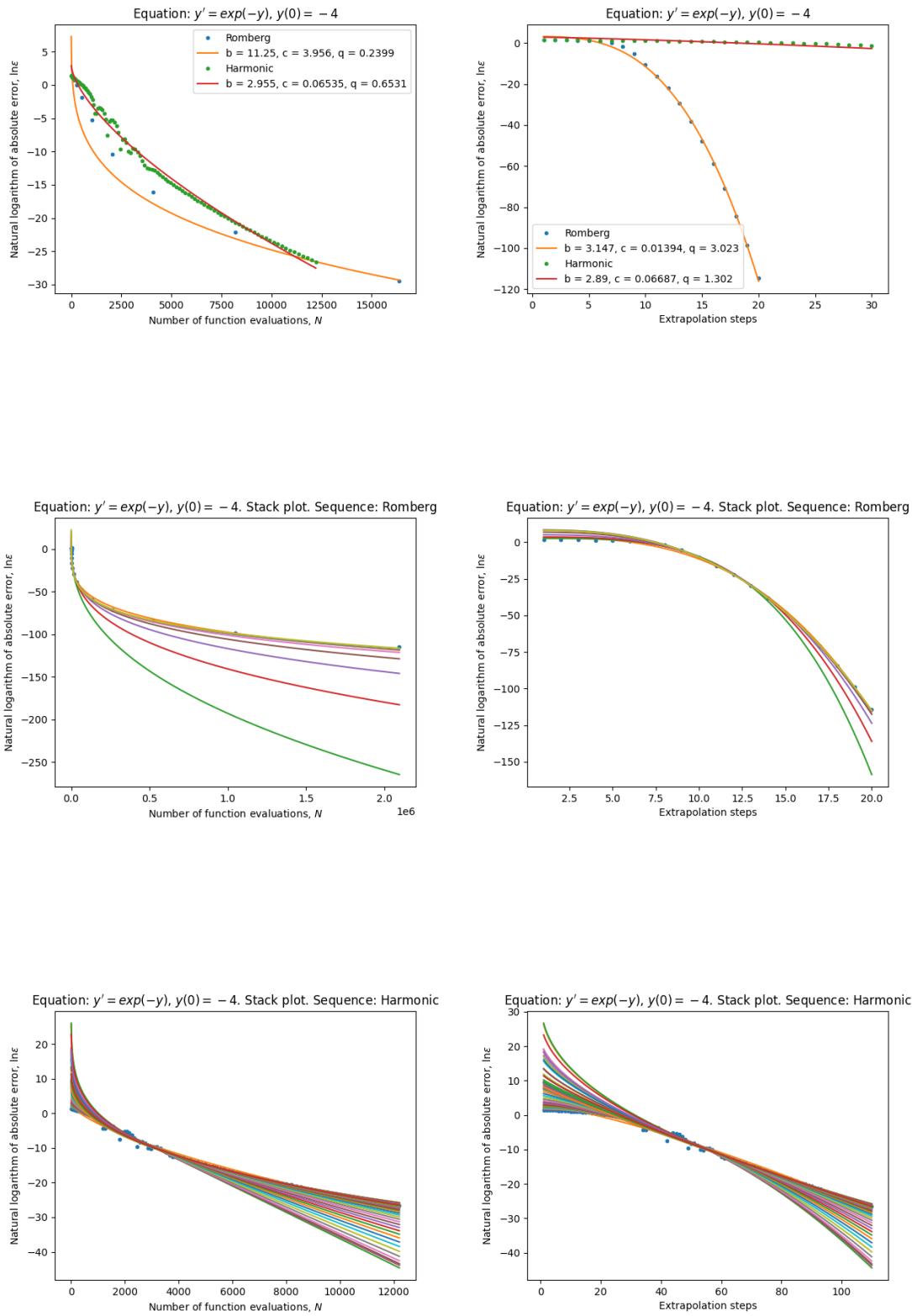
Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	$1.253e + 34$	6	30.35	0.163	0.1548	0.03379	$4.863e + 05$	0.0009135
Harmonic	30.61	0.03646	1.788	0.0001346	0.5031	$5.761e - 06$	1.864	9.42e - 07
Romberg	0.4234	0.02508	0.2721	0.0001868	2.202	$4.231e - 06$	0.1032	$1.729e - 06$
Harmonic	12.56	0.03136	1.79	0.0001139	1.006	$4.872e - 06$	1.294	$7.415e - 06$



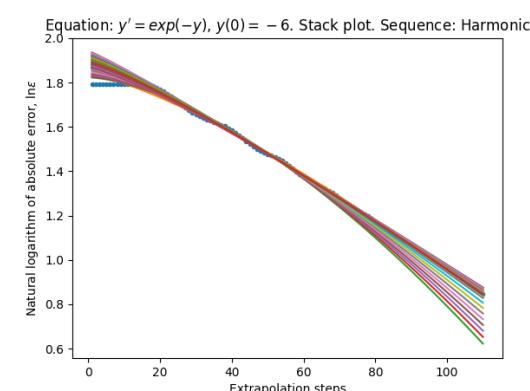
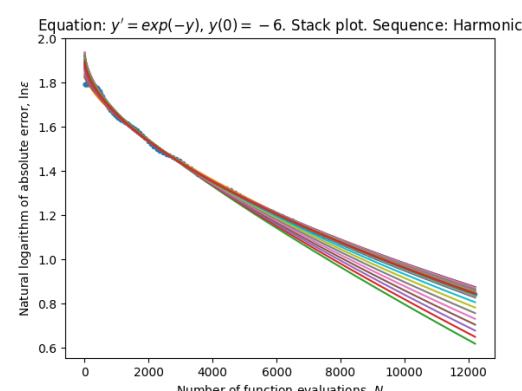
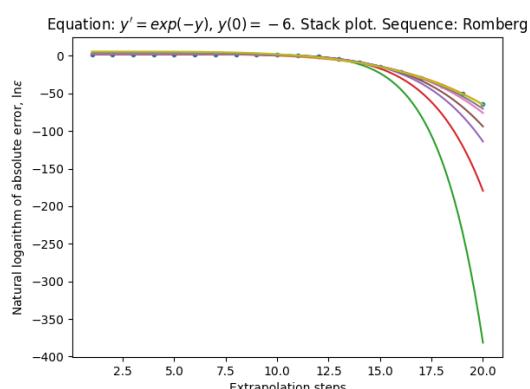
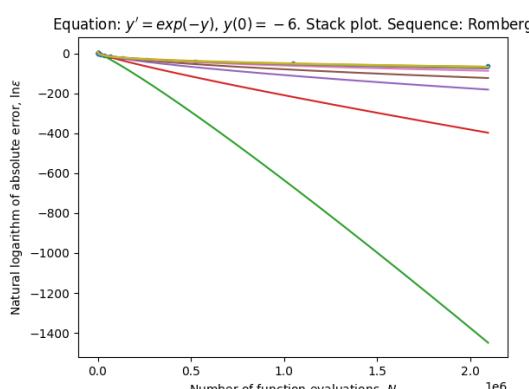
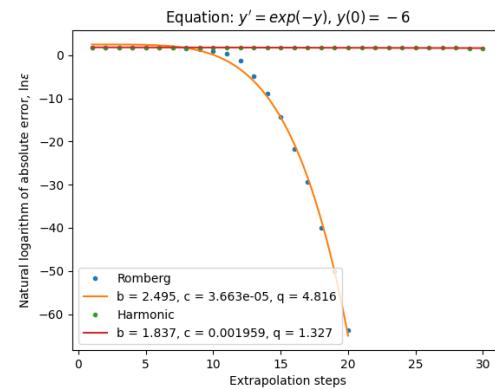
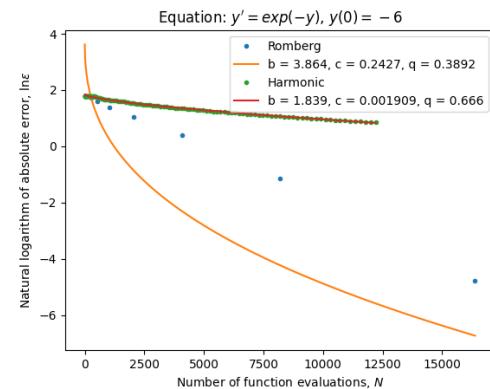
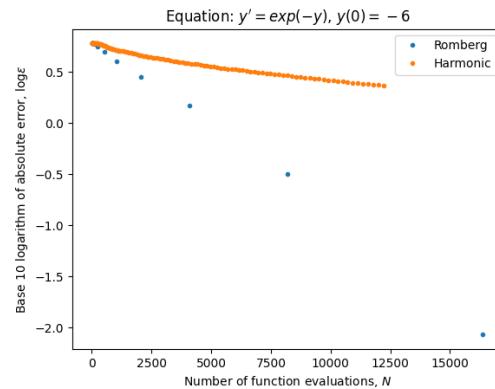
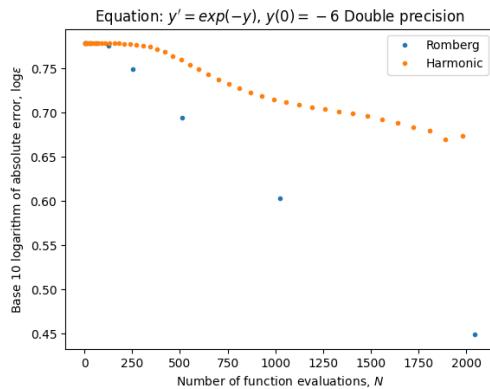


Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$6.703e + 26$	6	19.26	0.2229	0.1773	0.04097	$2.919e + 05$	0.001426
Harmonic	52.85	0.01529	0.9931	0.0001445	0.5047	$6.004e - 06$	5.014	$7.709e - 06$
Romberg	13.84	0.5067	0.138	0.01003	2.383	0.0002168	0.7745	$2.073e - 05$
Harmonic	32.1	0.01319	0.9945	0.0001227	1.009	$5.087e - 06$	4.245	$7.068e - 06$





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\text{ln}}$
Romberg	$1.768e + 15$	5.856	6.076	0.5152	0.2574	0.1078	7022	0.003923
Harmonic	$3.118e + 13$	28.2	1.047	2.401	0.5476	0.1796	3.262	0.003283
Romberg	1740	1.019	0.02087	0.366	3.03	0.01318	10.56	0.0007478
Harmonic	$1.45e + 13$	28.09	1.027	2.361	1.093	0.1757	3.112	0.003259



Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$1.103e + 04$	5.839	0.1695	2.457	0.6307	0.1706	6.594	0.006067
Harmonic	6.659	0.0009203	0.004057	0.1522	0.6042	0.00972	0.0003433	0.0001041
Romberg	54.13	3.259	$2.868e - 05$	3.836	6.419	0.06706	0.662	0.002097
Harmonic	6.636	0.0008806	0.004099	0.1488	1.206	0.00939	0.0003374	0.0001024

### 4.2.5 Equation with singularity

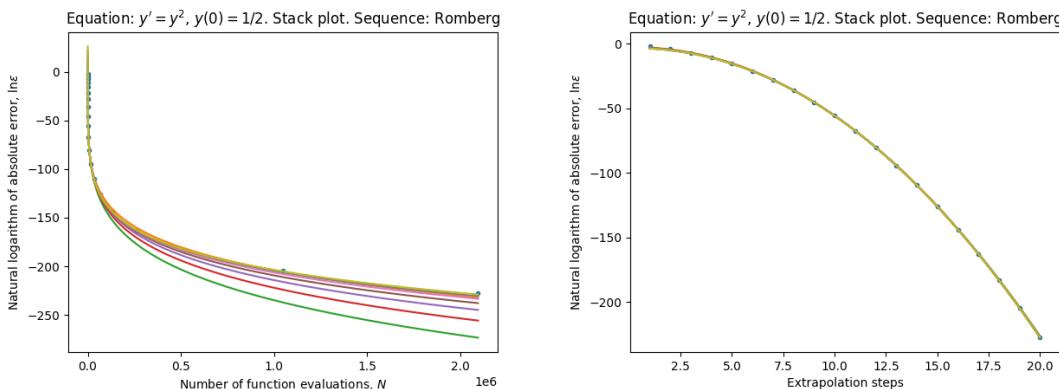
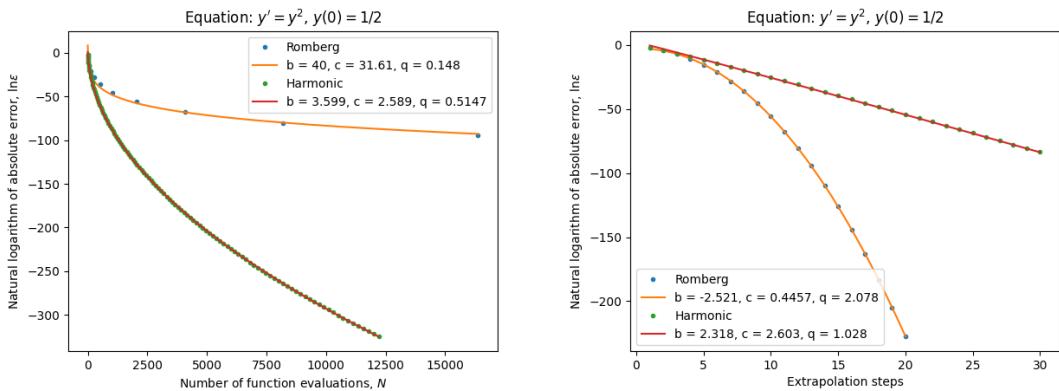
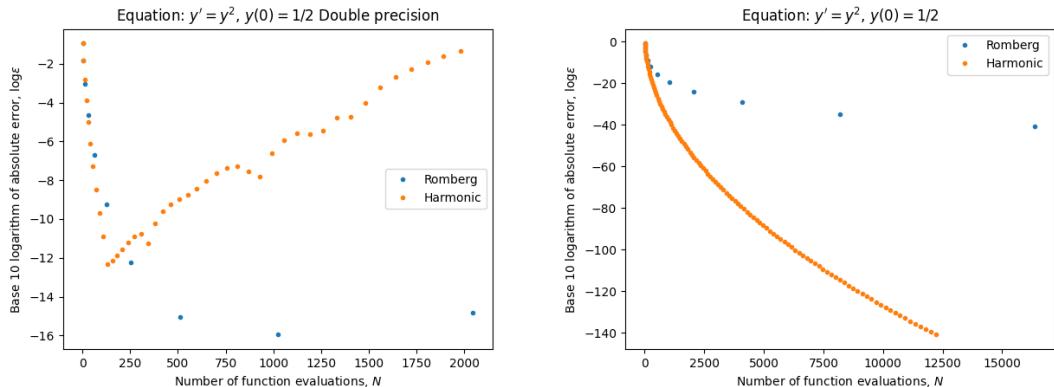
Now we will consider the following initial value problem:

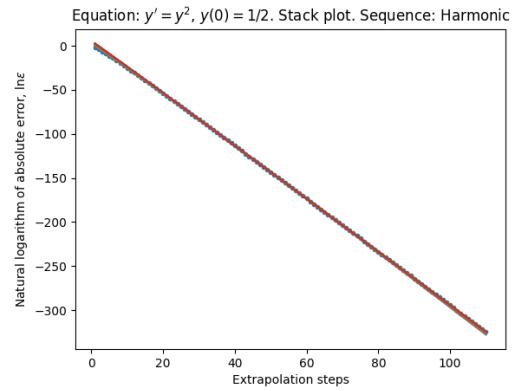
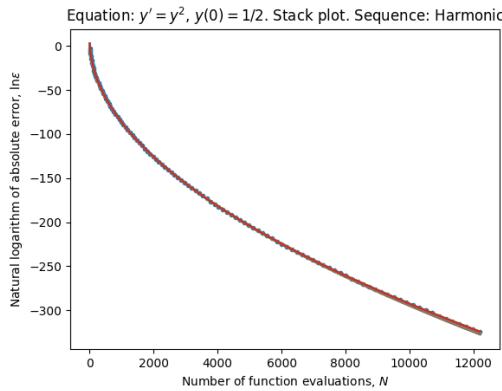
$$y'(t) = y^2(t), \quad y(0) = 1/(1+a), \quad t \in [0, 1] \quad (4.6)$$

whose solution is

$$y(t) = \frac{1}{1 - (t - a)}.$$

The solution is meromorphic with a pole at  $1 + a$ .

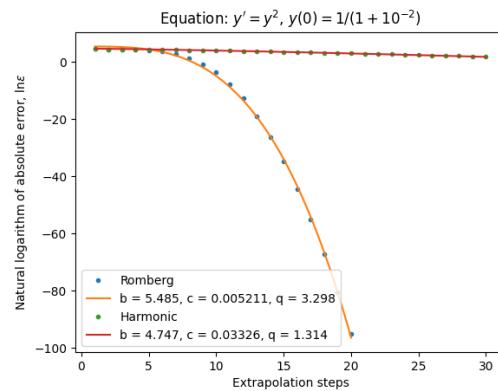
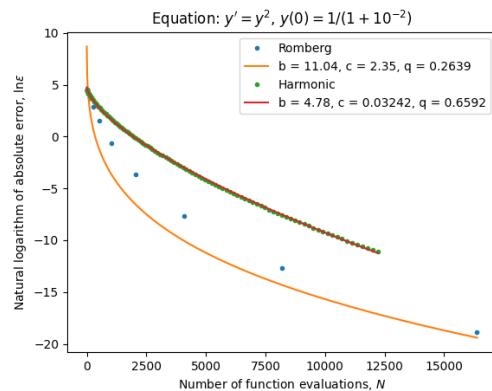
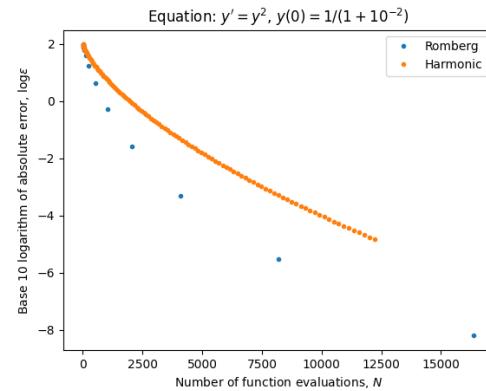
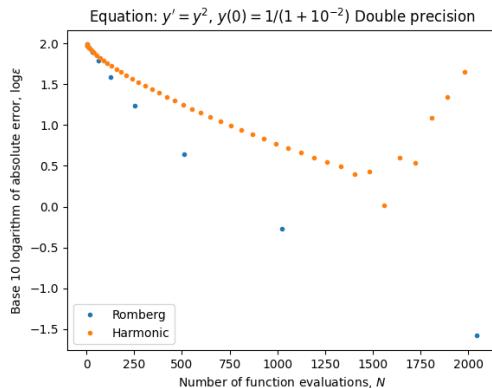


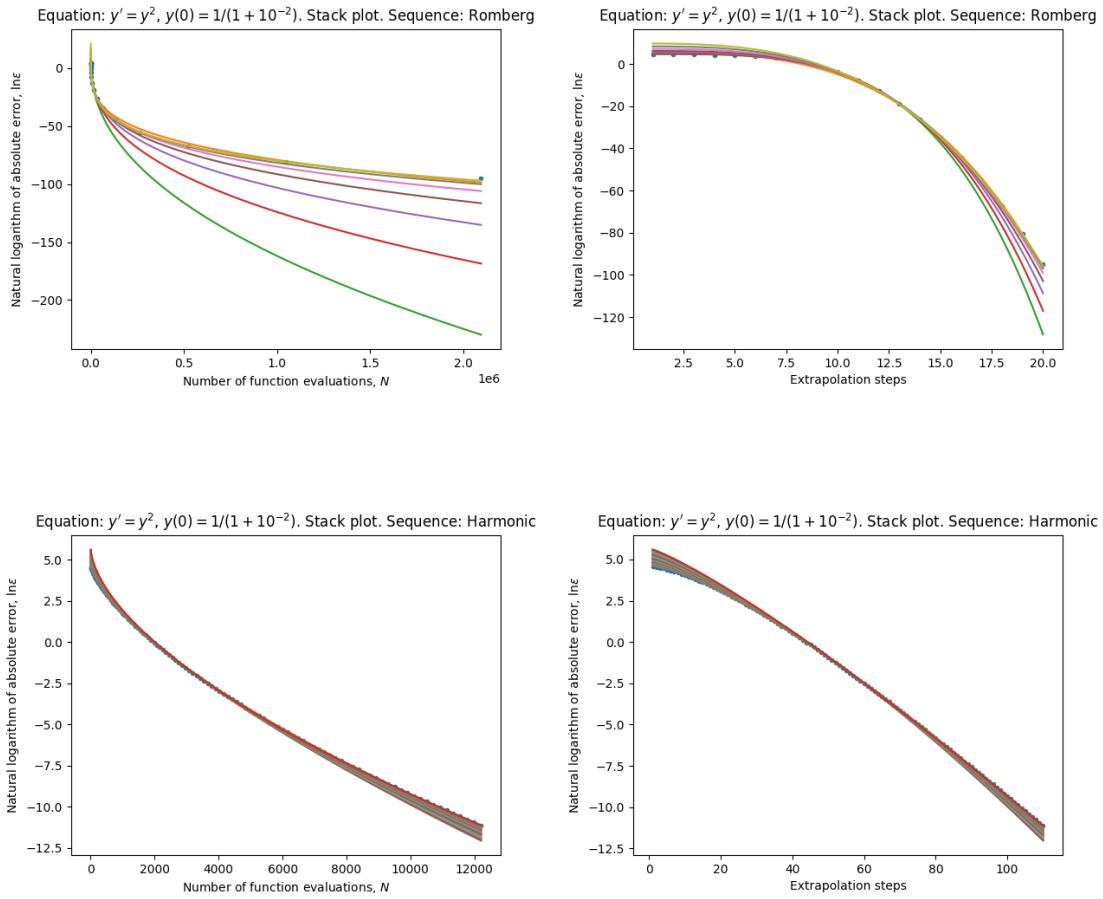


Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\text{lin}}$	$\rho_{\text{ln}}$
Romberg	$1.418 \cdot 10^{42}$	6	42.1	0.1387	0.141	0.03195	$1.734 \cdot 10^6$	0.0007659
Harmonic	440	0.5196	2.747	0.001256	0.5092	$5.472 \cdot 10^{-5}$	52.08	$3.588 \cdot 10^{-6}$
Romberg	0.04186	0.03016	0.4274	0.0002961	2.091	$8.21 \cdot 10^{-6}$	0.2758	$4.731 \cdot 10^{-6}$
Harmonic	104.3	0.4903	2.752	0.001151	1.018	$5.022 \cdot 10^{-5}$	34.01	$3.153 \cdot 10^{-6}$

The harmonic sequence performs better and we get almost down to machine level precision using standar double precision floating point arithmetic, with either sequence.

We have very nice fit for the exponential convergence in the number of steps for the Romberg sequence and also for the harmonic sequence.

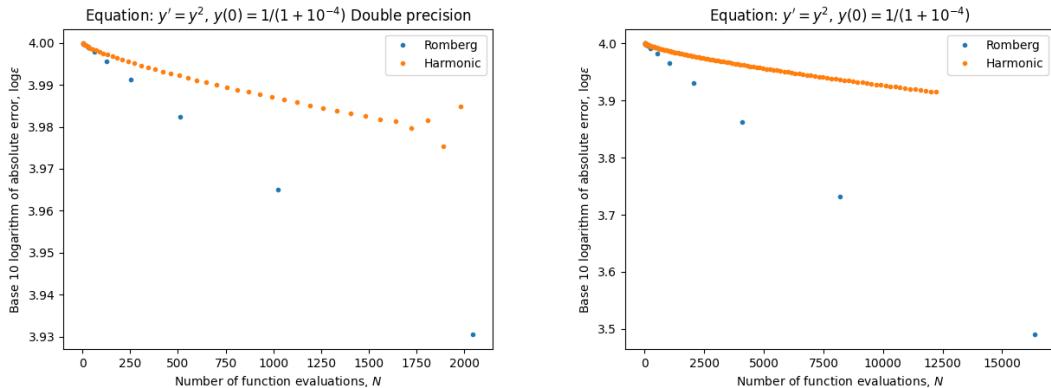


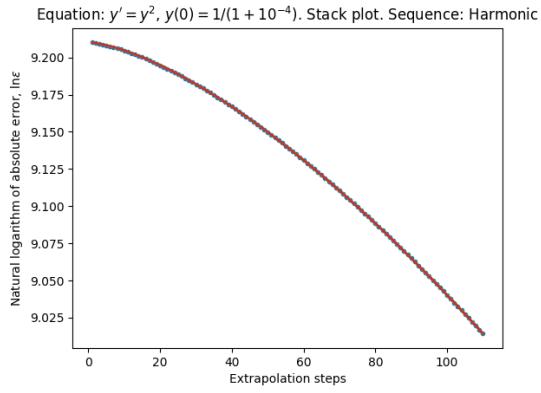
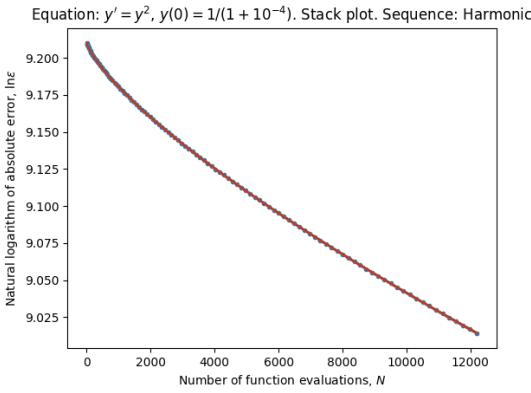
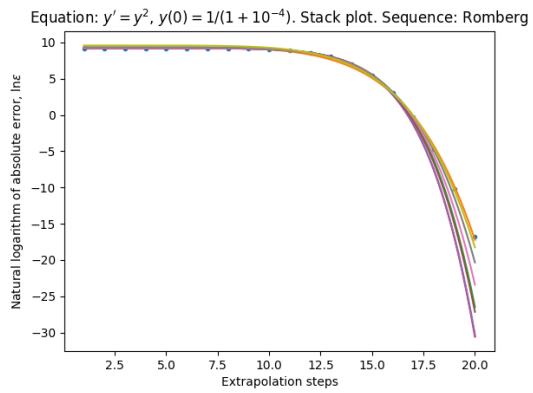
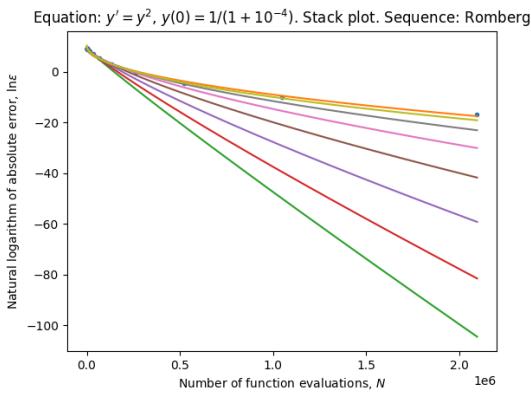
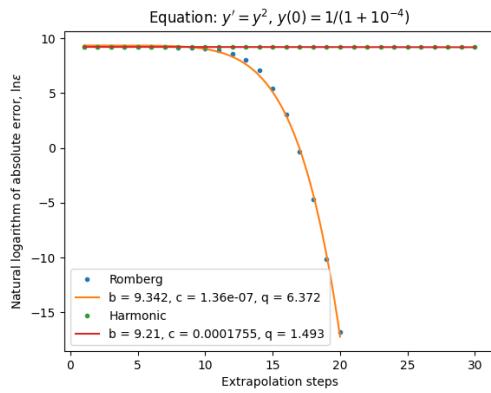
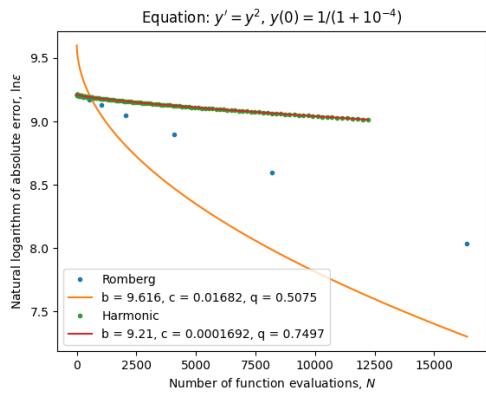


Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	$1.249 \cdot 10^{12}$	5.984	2.77	0.7656	0.3067	0.08315	400.9	0.003622
Harmonic	173.2	0.1158	0.03983	0.08603	0.6452	0.002375	0.009795	0.0001024
Romberg	3540	2.885	0.005212	0.6307	3.461	0.00941	1.498	0.0005041
Harmonic	165.2	0.1095	0.04047	0.08154	1.287	0.002248	0.008497	$9.476 \cdot 10^{-5}$

Here Romberg works better and we do not attain as high precision using the harmonic as when using Romberg, in standard double precision arithmetic.

Here the model fits moderately well for Romberg sequence, when considering exponential convergence in the number of steps. We also have moderate fit for the harmonic sequence.





Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{lin}$	
Romberg	$1.462 \cdot 10^4$	0.2298	0.004335	2.285	0.7393	0.04196	0.1052	0.0
Harmonic	$1 \cdot 10^4$	$5.52 \cdot 10^{-9}$	0.0001699	$5.806 \cdot 10^{-5}$	0.7494	$1.194 \cdot 10^{-6}$	$4.151 \cdot 10^{-10}$	4.75
Romberg	$1.087 \cdot 10^4$	0.02206	$1.977 \cdot 10^{-8}$	3.201	7.604	0.006289	0.01786	0.0
Harmonic	9997	$3.302 \cdot 10^{-10}$	0.0001749	$5.29 \cdot 10^{-6}$	1.494	$1.204 \cdot 10^{-7}$	$2.077 \cdot 10^{-10}$	2.20

Here we clearly do not have exponential convergence in the number of evaluations for the Romberg sequence. We have not so good fit for exponential convergence in the number of steps.

Regarding the harmonic sequence, we note that we have extremely slow convergence, so the  $x$  and  $y$  values differ by many orders of magnitude, so the results are maybe not so reliable.

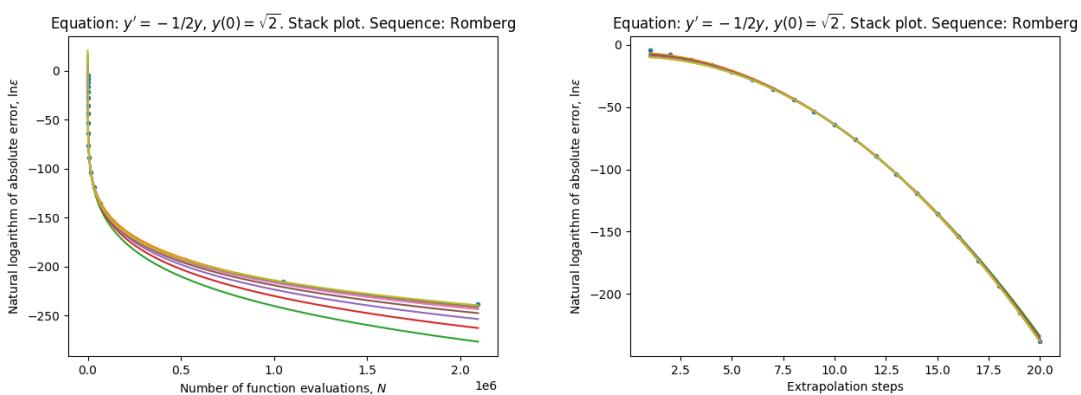
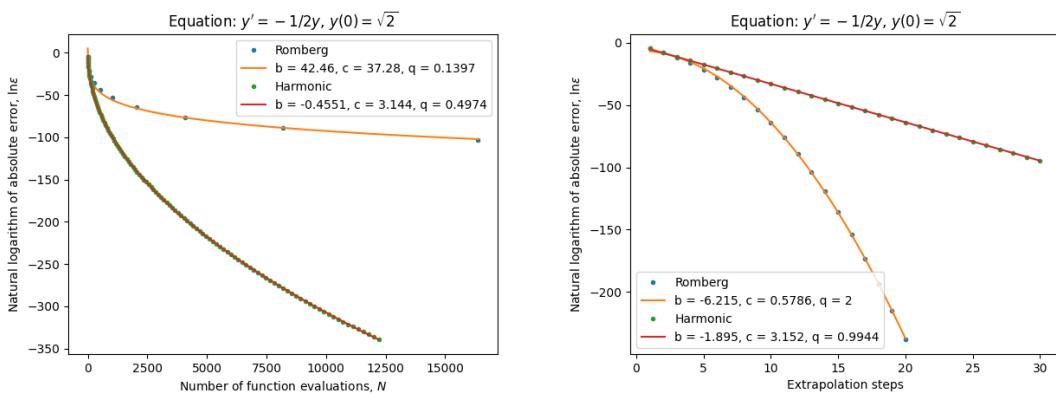
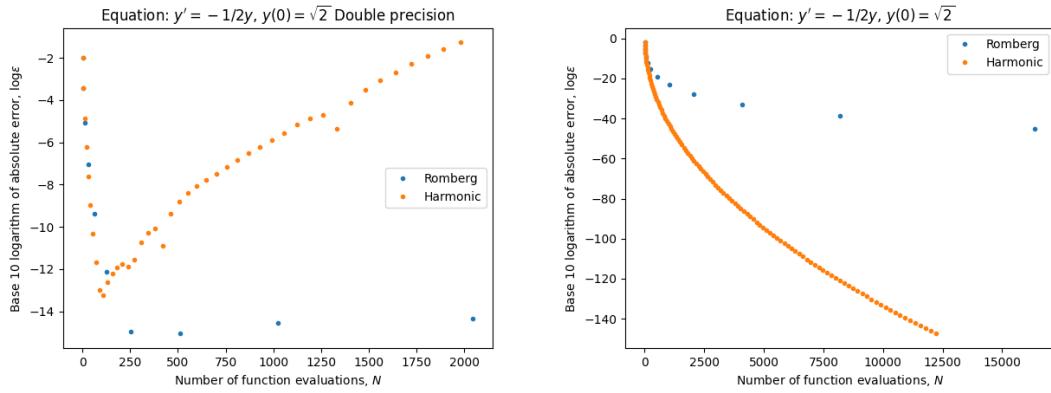
### 4.2.6 Equation with moderate singularity

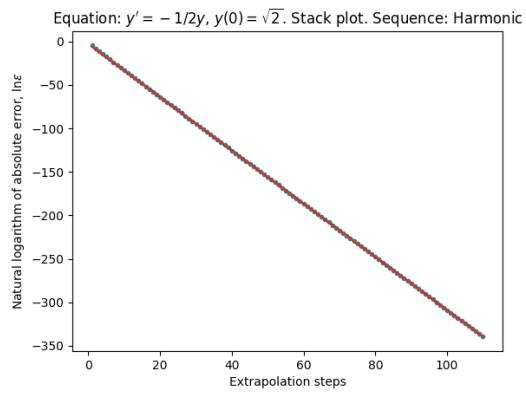
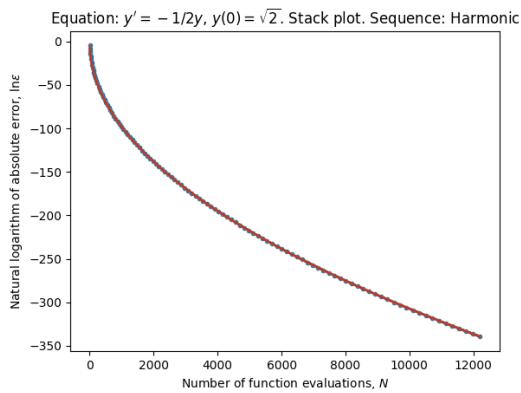
Now we will consider the following initial value problem

$$y'(t) = -\frac{1}{2y}, \quad y(0) = \sqrt{1+a}, \quad t \in [0, 1] \quad (4.7)$$

whose solution is

$$y(t) = \sqrt{1 - (t - a)}$$

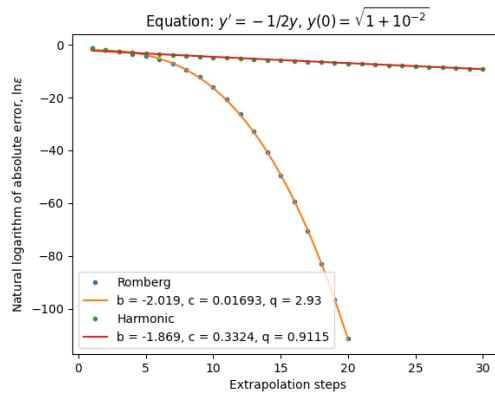
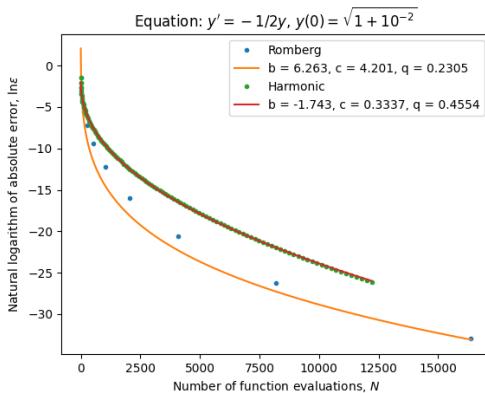
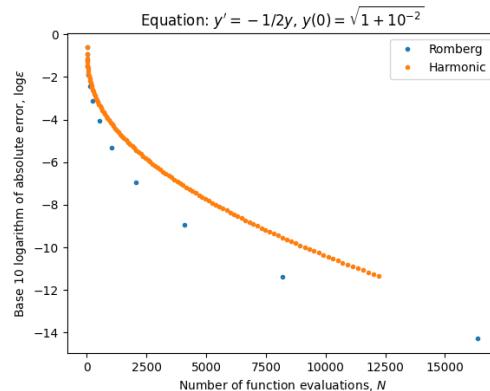
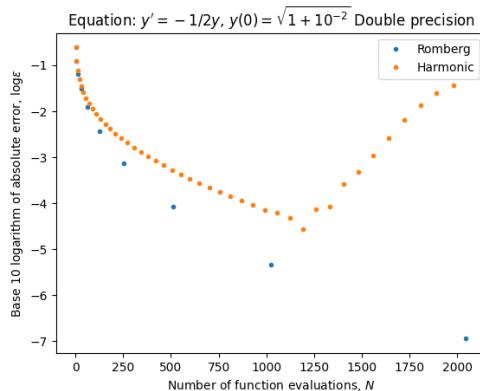


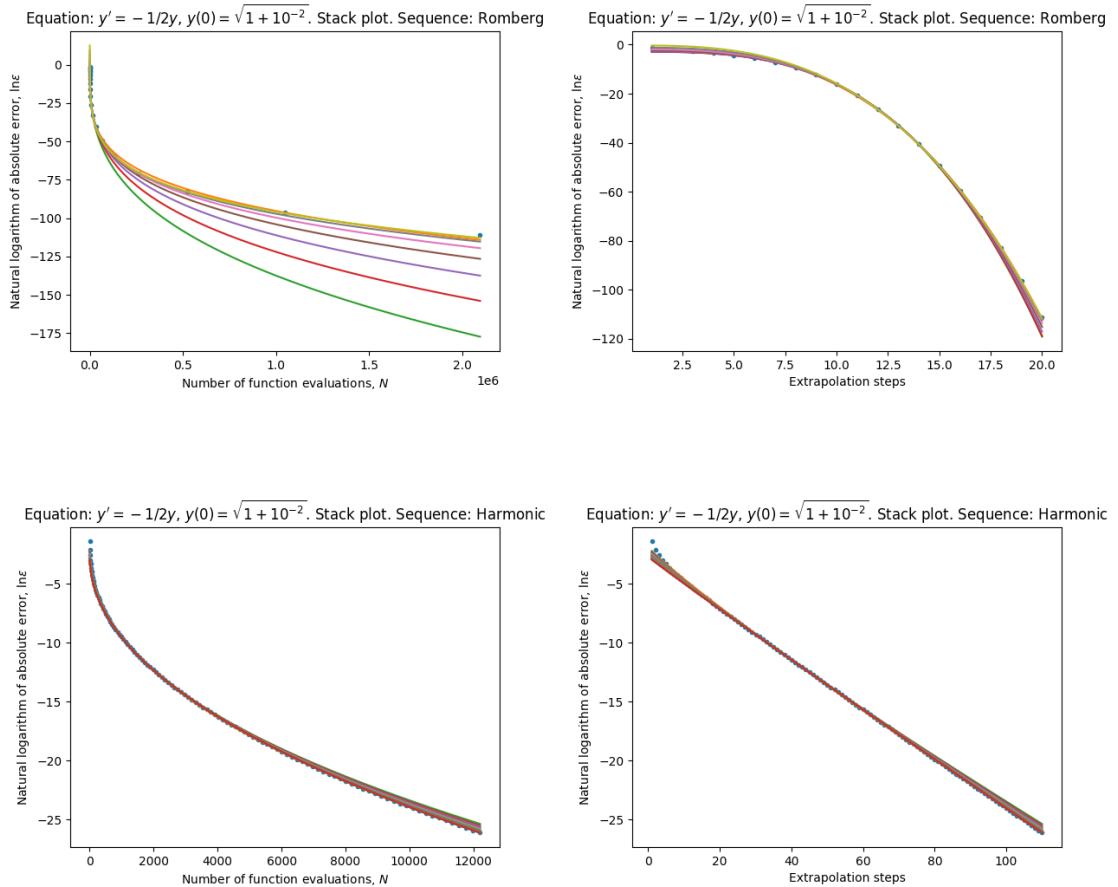


Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$8.335 \cdot 10^{41}$	6	47.04	0.1111	0.1343	0.02593	$1.543 \cdot 10^5$	0.0004902
Harmonic	0.454	0.03631	3.115	$3.797 \cdot 10^{-5}$	0.4983	$1.623 \cdot 10^{-6}$	0.06823	$6.992 \cdot 10^{-8}$
Romberg	0.0002771	0.7384	0.5104	0.0058	2.038	0.0001599	0.7906	$4.025 \cdot 10^{-5}$
Harmonic	0.1009	0.0414	3.117	$4.412 \cdot 10^{-5}$	0.9965	$1.9 \cdot 10^{-6}$	0.1341	$1.235 \cdot 10^{-7}$

Here the harmonic sequence works better than Romberg and we get down to machine level precision using either sequence, in standard double precision floating point arithmetic.

Here we clearly have exponential convergence in the number of steps and evaluations for the harmonic sequence. Regarding Romberg, we seem to have exponential convergence in the number of steps.

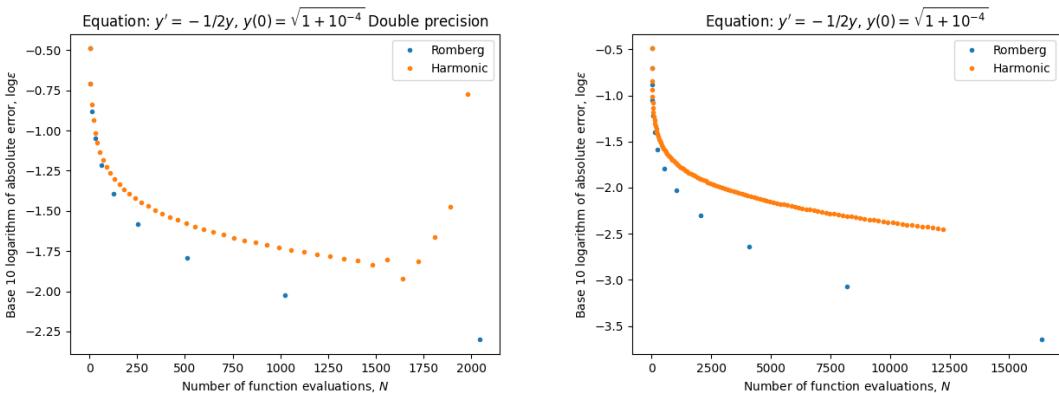


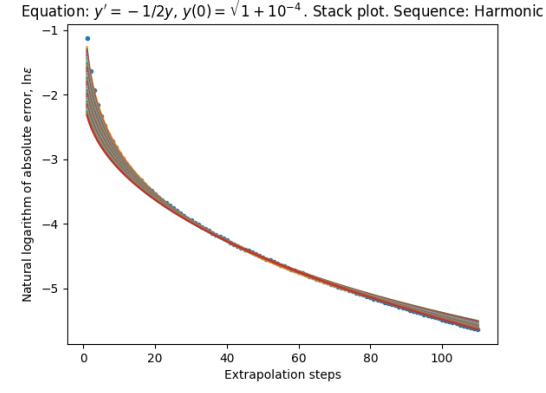
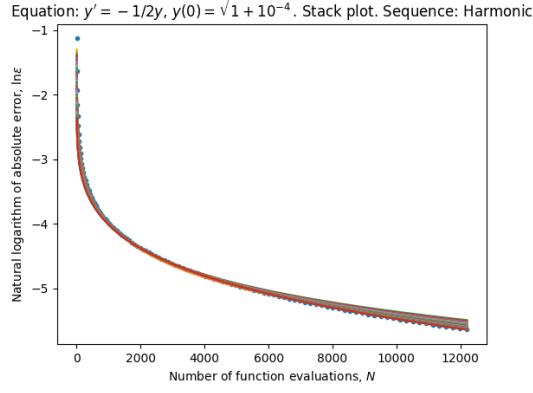
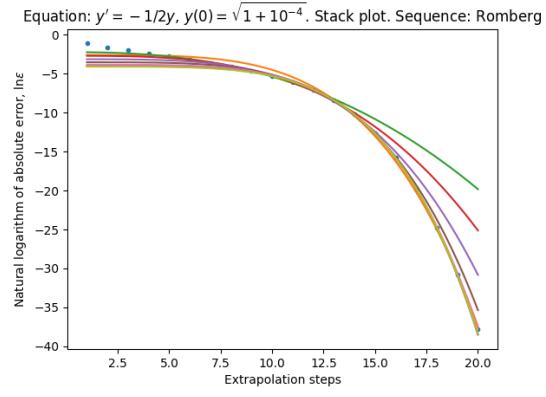
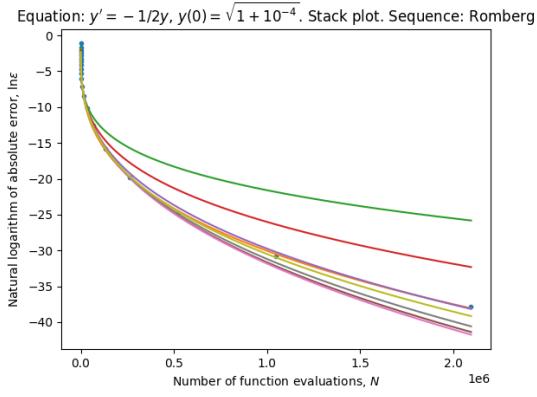
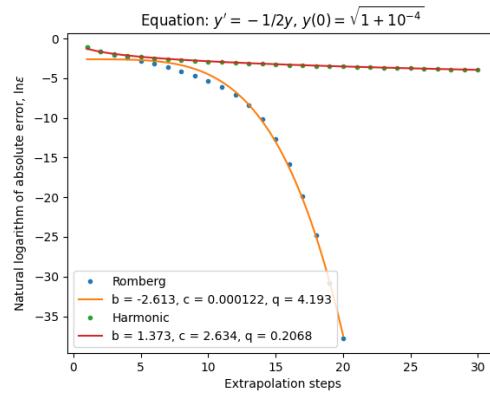
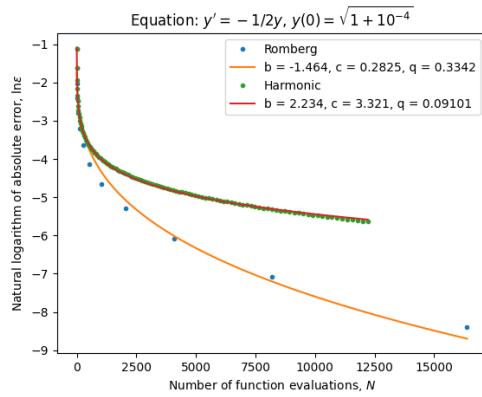


Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$3.095 \cdot 10^9$	5.978	4.521	0.4282	0.2537	0.04639	166.8	0.001351
Harmonic	0.09273	0.0545	0.263	0.01142	0.4785	0.0004549	0.228	$5.083 \cdot 10^{-7}$
Romberg	0.2097	1.241	0.01393	0.1057	3.022	0.001297	0.1724	$5.672 \cdot 10^{-5}$
Harmonic	0.08262	0.05039	0.2624	0.01092	0.9574	0.0004388	0.2281	$5.092 \cdot 10^{-5}$

Here Romberg performs better and we do not attain as high precision using the harmonic sequence in standard double precision floating point arithmetic.

For the Romberg sequence, the model seems to fit moderately well when considering exponential convergence in the number of evaluations. We also have reasonably good fit for the harmonic sequence.





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	0.1398	0.4846	0.2489	0.4625	0.3506	0.01881	0.1737	0.0004512
Harmonic	1.282	3.403	1.258	0.3894	0.1631	0.05472	0.0121	$4.935 \cdot 10^{-5}$
Romberg	0.0429	0.5116	0.001785	3.769	4.002	0.04408	0.4745	0.00176
Harmonic	0.8136	1.714	1.146	0.311	0.3342	0.04634	0.007362	$3.466 \cdot 10^{-5}$

Here, we do not have any clear fit.

#### 4.2.7 Circular rotation

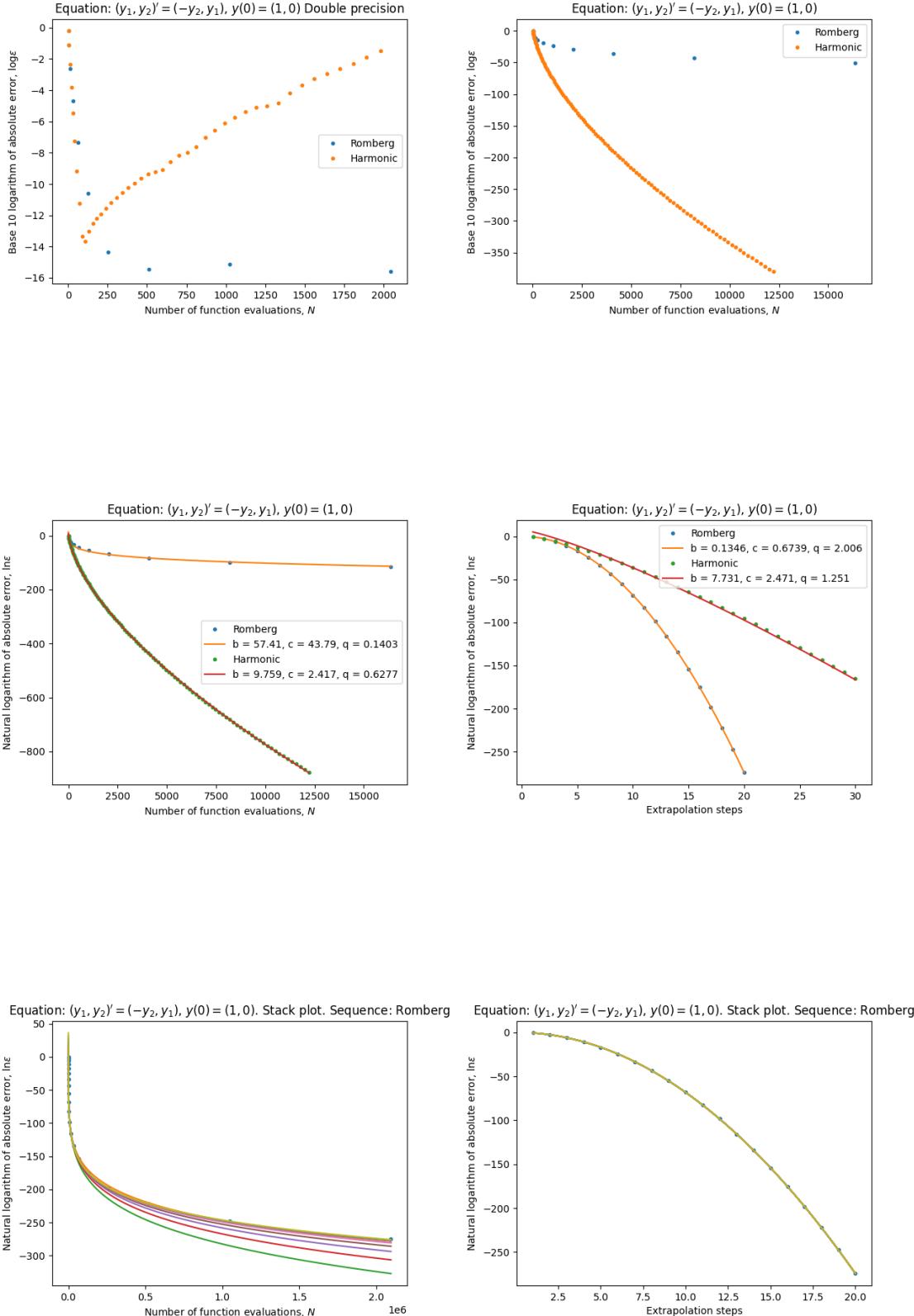
Now we will consider the following system of equations:

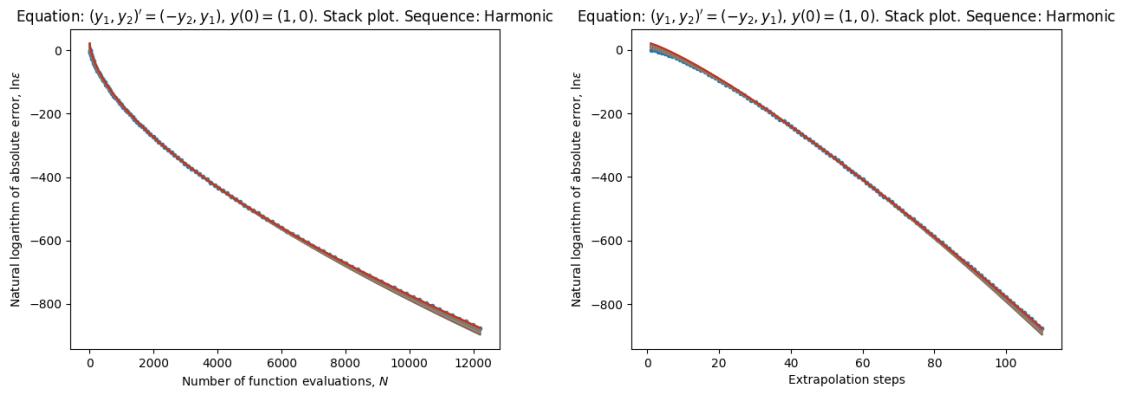
$$(y_1(t), y_2(t))' = (-y_2(t), y_1(t)), \quad y(0) = (1, 0), \quad t \in [0, \pi/2] \quad (4.8)$$

whose solution is

$$(y_1(t), y_2(t)) = (\cos t, \sin t)$$

which is entire.





Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$1.215 \cdot 10^{60}$	6	61.05	0.1358	0.1308	0.03453	$2.289 \cdot 10^8$	0.0008594
Harmonic	$9 \cdot 10^9$	6.841	2.636	0.007896	0.62	0.0002457	$4.413 \cdot 10^5$	$8.933 \cdot 10^{-6}$
Romberg	1.083	0.000101	0.6717	$4.963 \cdot 10^{-7}$	2.007	$1.48 \cdot 10^{-8}$	0.002923	$1.972 \cdot 10^{-8}$
Harmonic	$4.276 \cdot 10^8$	6.508	2.675	0.006923	1.237	0.0002153	$9.526 \cdot 10^4$	$7.494 \cdot 10^{-6}$

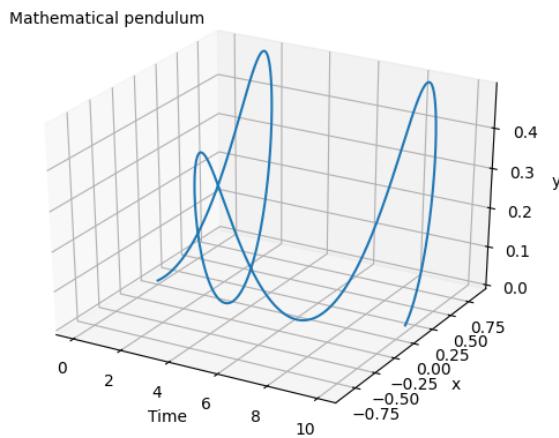
The harmonic sequence works better than Romberg and we get down to machine level precision using either sequence when using standard floating point arithmetic.

We clearly have exponential convergence in the number of steps for the Romberg sequence. For the harmonic sequence, we seem to have exponential convergence, but though we must note that the mean value of the  $A$  coefficient is suspiciously large.

#### 4.2.8 Mathematical pendulum

Now we will consider the mathematical pendulum equation:

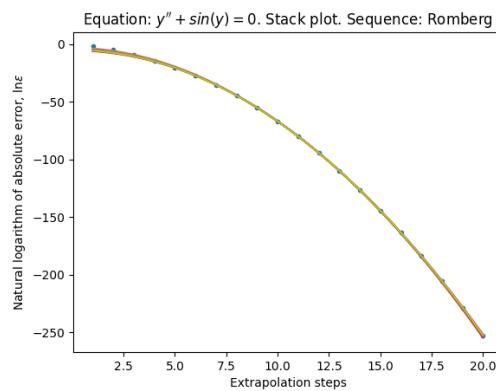
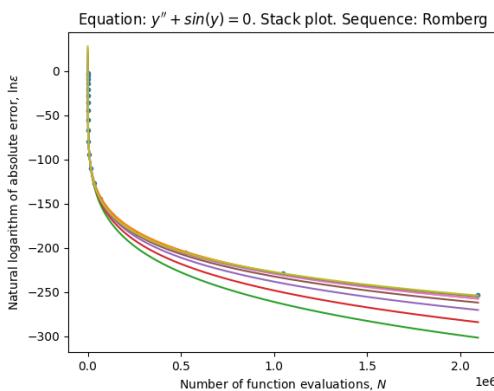
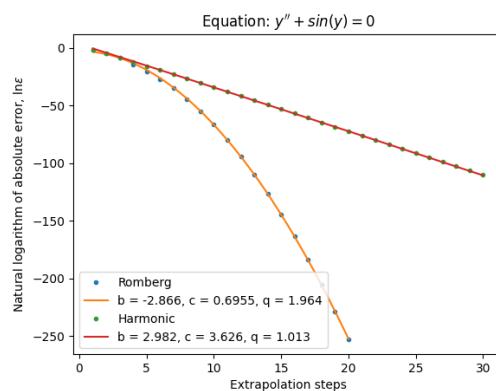
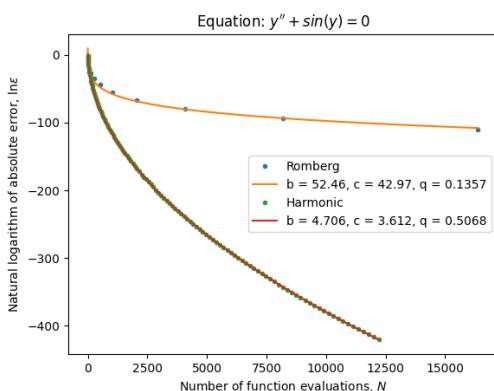
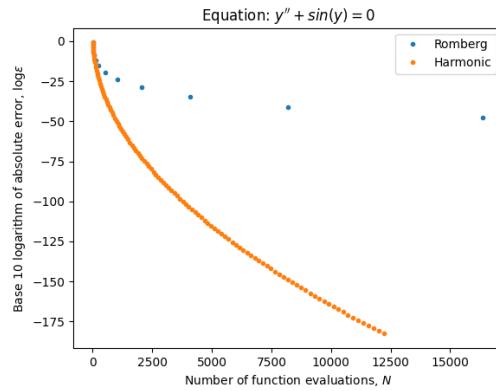
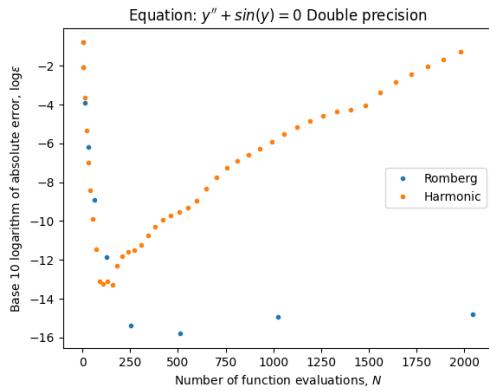
$$y''(t) + \sin y(t) = 0, \quad y(0) = 0, \quad y'(0) = 1, \quad t \in [0, 1]. \quad (4.9)$$

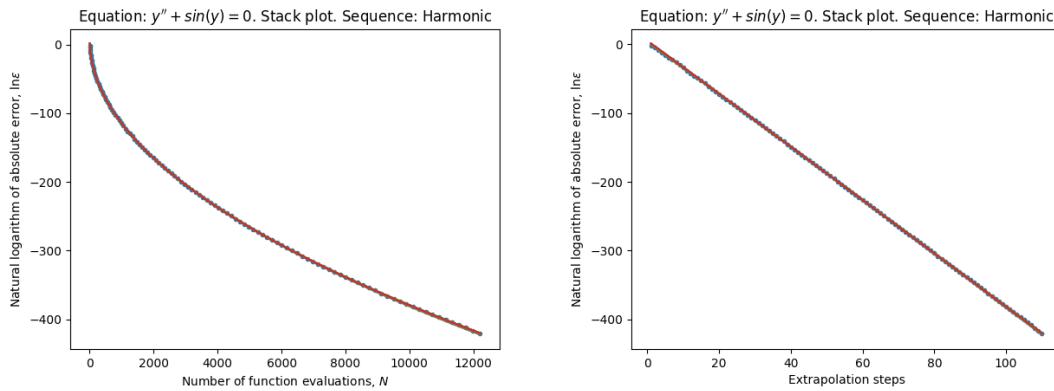


We use the following solution as a reference solution:

$$y(1) = 0.8477986816771168446636783318809189285204869487805839233894518237264 \\ 0630434688566338353630224868585976468140927461204156022529742699705231816982332$$

8758573465006731337868899164920465794464433365734542344489127172608253083611368  
 1006527088120002980265012291946040048160269697235789023333048631065165528080437  
 9429035809638912058441220815487971711586081657146212645642189375079112255277643  
 9852179622939101197041300976161428914224324373911458462193934898208401314255177  
 19029786828908788598595270867026923369,

$$y'(1) = 0.5685689980951714899417351630224588908163828902980003832061653705295  
 9766024841080148281818712844186765300851429899969930131242230797649624727126014  
 4817345395779299526094029847932513513414050406462412114396118139738629227789263  
 8171260577815198979573088520212213199643020386321347981028859823073409833268487  
 4465028063198951013936721519329558610769737497582995347991161206624419371276898  
 6489889976919046237233718506150977278781928633388966603053245457050361914127647  
 64458195896384935977161329263091278345$$




Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	$3.245 \cdot 10^{52}$	6	56.9	0.1423	0.1295	$0.03859$	$1.42 \cdot 10^6$	0.0005633
Harmonic	471.5	0.2744	3.713	0.0002618	0.5042	$1.13 \cdot 10^{-5}$	9.558	$1.093 \cdot 10^{-6}$
Romberg	0.01384	0.3333	0.6305	0.002076	1.995	$6.207 \cdot 10^{-5}$	0.6747	$2.822 \cdot 10^{-5}$
Harmonic	72.54	0.2499	3.718	0.0002297	1.008	$9.918 \cdot 10^{-6}$	5.231	$9.151 \cdot 10^{-7}$

Here the harmonic sequence works better and we get down to machine level precision in standard double precision floating point arithmetic, using either sequence.

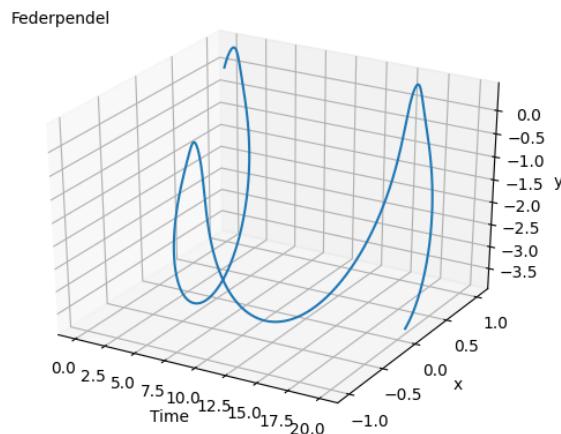
We have nice fit for the exponential convergence in the number of steps for the Romberg sequence. We also have very nice fit for the harmonic sequence.

#### 4.2.9 Federpendel

Now we will consider the equation of motion for das Federpendel or the spring pendulum:

$$\mathbf{p}' = -(|\mathbf{q}| - 1) \frac{\mathbf{q}}{|\mathbf{q}|} - \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{q}' = \mathbf{p}$$

where  $\mathbf{p}$  and  $\mathbf{q}$  are two dimensional vectors. We will consider it with the initial condition  $\mathbf{q}(0) = (1, 0)$  and  $\mathbf{p}(0) = (0, 1)$  and try to both estimate the solution at time  $t = 1$  and time  $t = 2$ .



In the first case, we use the following solution as a reference solution:  $p_1(1) = 0.98196696582217844720319657954919759480867944130559325336717107885226840029194668537546599580718275493078481180650217303945645979298918706112255593505505626685102547536251828245902064758977315697810980003041775025607529367664638776665518639316652745962224476408233686057684574922030895683568627055060053379858290677222729974308155700102559653851623870181639231532304294556258406979390490261455173467397881497189867405108793688928536548109014102191774977544220410936638197206151636634436423913415562755$

$$p_2(1) = 0.49335546798350334626041620593851945416396626519837138591308850652625943297872531022435283639791643816215666656344985806930404094615344197083157552614944297128599991716468553830196779287602566748280602038872196712489924019695274403281608946318516254403134467824486021626022434745815153955369537221397337406711474039822233987947870868876535838241993296027192451258265454232143359143046007803845068321044214822314881129930911306463010391141587450151146417694046339447862830560090943990593856861344868184,$$

$$q_1(1) = -0.054497168417449092396564692225277150158654249263273506069580872211139539308332921017183899865929552758056806081203650952356671774365740816189687626311834428013129538817979728576002102542805967179984145909860049511182076050552585325721312328821818561702429270968402730822217441482557867420055841500117550908728647830863352423646797486489760901115782149061190926028828423329291592505638956227897810714981510350185421442915816513117559281274595080602397863560840902157944618028123455200134949086739518845,$$

$$q_2(1) = -0.0230420035518673961944722653237093488086826402838527559012373519448197015716127572871536446235004830049403120036677816647644884326953669673337599439534273474622647411298705630194650041584209786049441534330449870245692413879729859770729363840065407792791795827910921213951428336765797024949865664777018947487656008718079153902989561662350280790109128832725301766534037895182631610247518858508004712617753409972990264342834952054535399902351521870626261165278195016073168755255019773931010852710787326$$

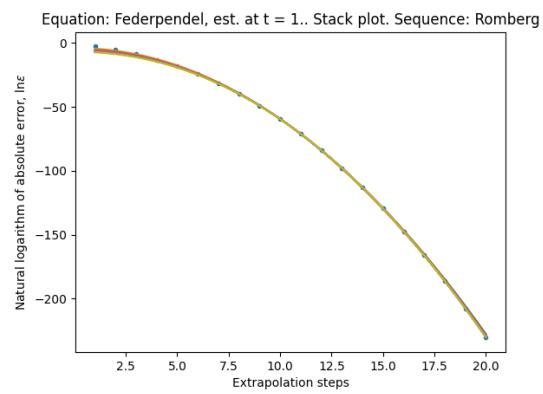
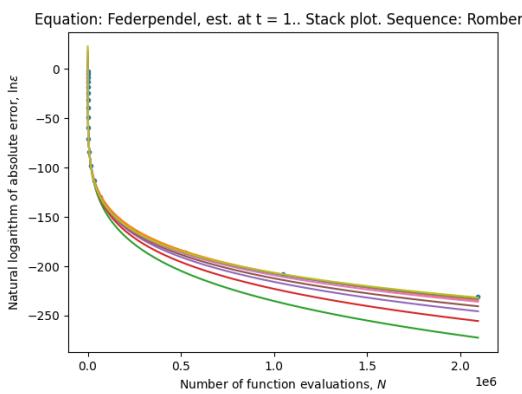
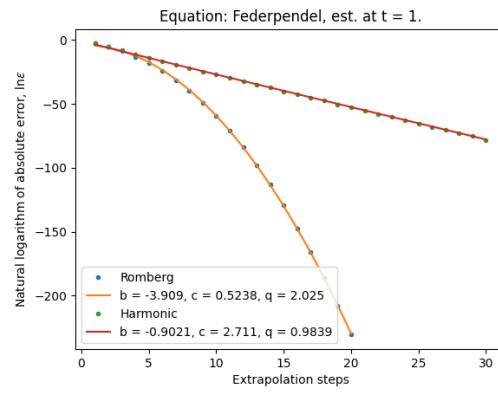
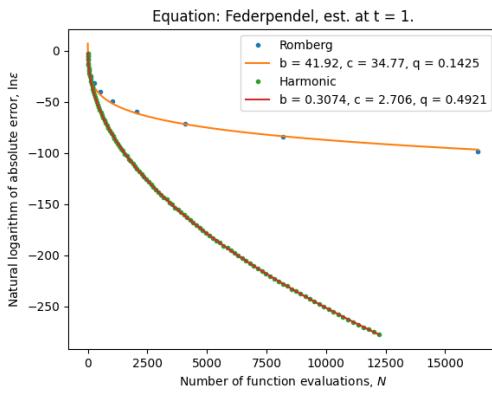
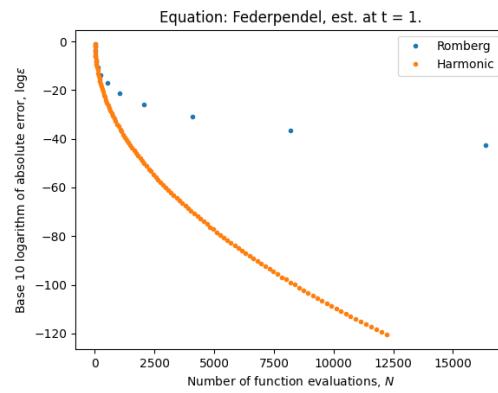
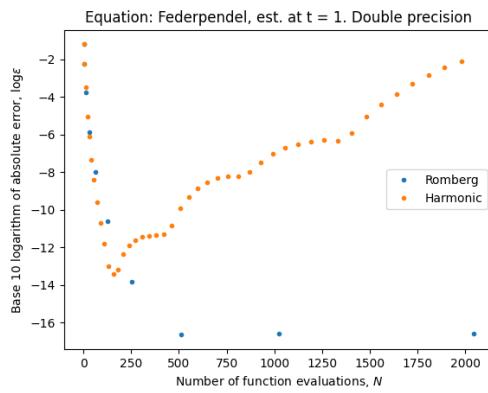
In the second case, we use the following solution as a reference solution:

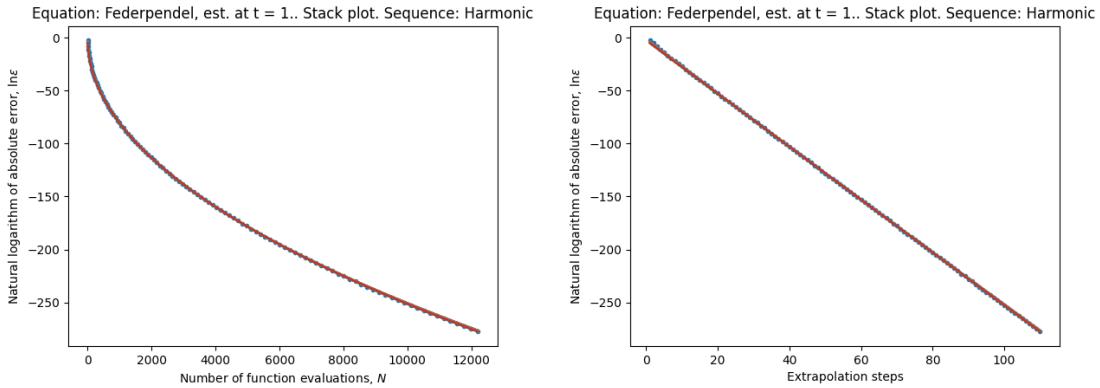
$$p_1(2) = 0.90848595139849884769603653291496062503270523806688097650559051463679059621717406765127072562974902443098692105491434882819713510498544780275140566262920923919094308879461423242023249006528802304016469019659450867666853516641897160581599858863899725312812829812684512320657317082138495267978162519228332646143785995651440296148577391748961328189085871243444861978503160776609750334301861509139734505417881523161797903118442562808951135430345052299394995970839644880196049228869133677552355615201639841,$$

$$p_2(2) = -0.0400127440187803370433611931197131969498274041987037275653260071358935781531162588844245590290263338843920016918602606157039461358161414116875822338011746054046268046036596809856610785677948220161247049818070416285194976844006296124477109268266833636190723006029713149111709247362257659423681770012878721823547658511754828372204936485071190516706898628581042433198667327558306613787976959627739409422402942858732160505254216934224179224762870110338678881856667907336082354739548076217771507305280004',$$

$$q_1(2) = -0.05789051934786593611832698201238105890228492456837661843208483670326635593885844470933229180067670016942393390771972203851305951263697059378236143483165809767496092414534534793820369871649278187183812343128435112189640180789574731700574531441815999883199187278530734796698704550921548528330382535994721752$$

5206610787068565140850708766935937312360126964825271205268059256423103439064362  
 6280120579422255358159426927623348189550558743531180566800761374684384778131649  
 51871956636577739297079502516623843152,  
 $q_2(2) = -103366327960599891968494267743085332548477514735768044836892501581121$   
 $860537920088974942190809167733252046858540709564361720865486118909151079825532$   
 $2936786498098603887151833920930228566895164749459579768574320635663289125546215$   
 $5603569899824920050367652458460591306616174920680036366502840459245685377224521$   
 $9693335386156005977279835893881529767415171560307613621573703750005039261045148$   
 $3325045547225290040950956466096120578479317086472139909045481432231784472145723$   
 $227375922414201905077415299034086814.$

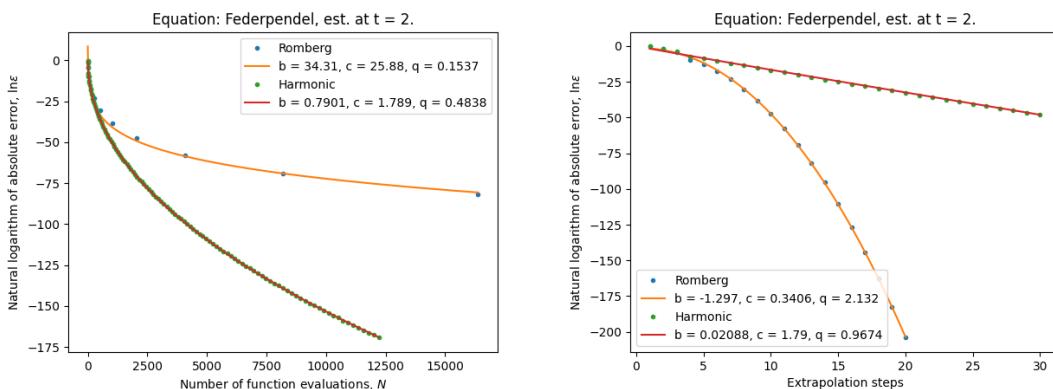
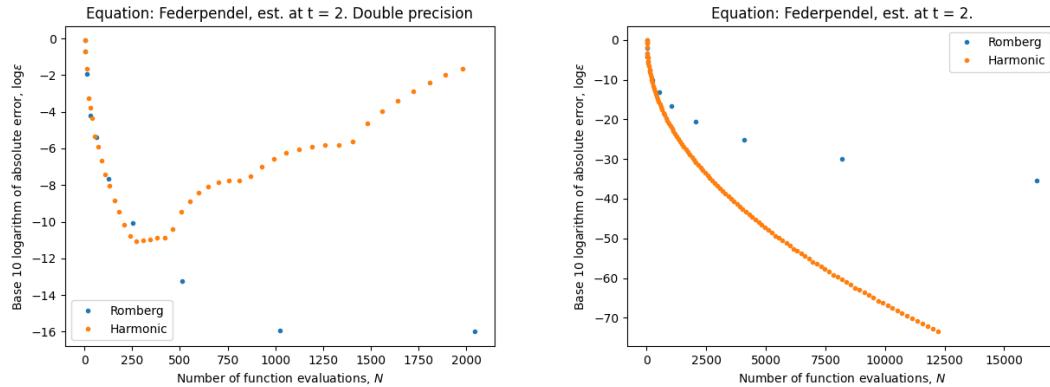


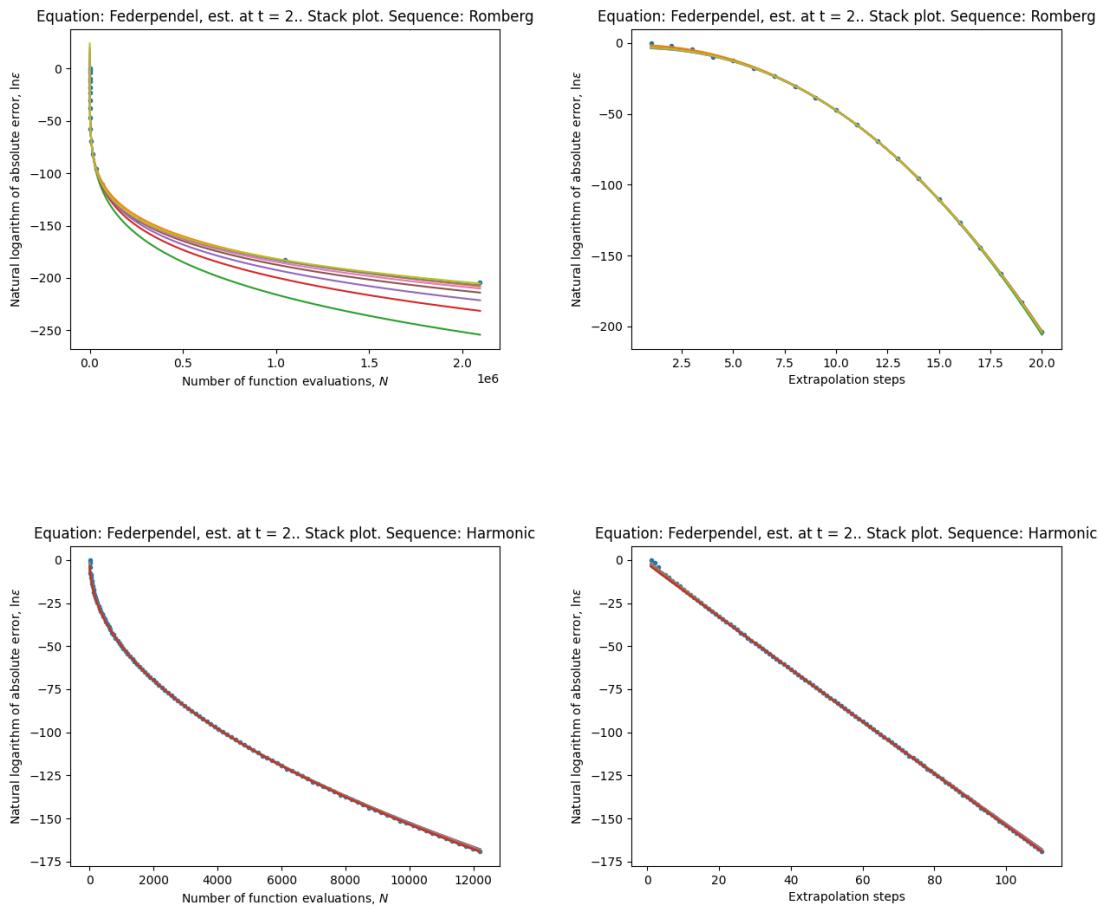


Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$8.241 \cdot 10^{40}$	6	43.86	0.1208	0.1376	0.02818	$3.087 \cdot 10^5$	0.0005385
Harmonic	0.5082	0.2486	2.612	0.0003409	0.4955	$1.457 \cdot 10^{-5}$	0.2652	$1.798 \cdot 10^{-6}$
Romberg	0.002791	0.3701	0.4615	0.003294	2.064	$8.941 \cdot 10^{-5}$	0.6517	$3.692 \cdot 10^{-5}$
Harmonic	0.1465	0.2485	2.613	0.0003471	0.9909	$1.492 \cdot 10^{-5}$	0.3215	$1.965 \cdot 10^{-6}$

Here the harmonic sequence works better and we get down to machine level precision in standard double precision floating point arithmetic, using either sequence.

We have nice fit for exponential convergence in the number of steps for the Romberg sequence. We also have nice fit for exponential convergence for the harmonic sequence.





Sequence	$A$ -mean	$A$ -var	$c$ -mean	$c$ -var	$q$ -mean	$q$ -var	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	$2.335 \cdot 10^{34}$	6	32.11	0.1569	0.1519	0.03495	$9.637 \cdot 10^4$	0.0006459
Harmonic	0.5245	0.3886	1.643	0.001271	0.492	$5.413 \cdot 10^{-5}$	0.5945	$1.112 \cdot 10^{-5}$
Romberg	0.05507	0.06164	0.2956	0.001162	2.178	$3.178 \cdot 10^{-5}$	0.5542	$5.079 \cdot 10^{-5}$
Harmonic	0.2422	0.3718	1.642	0.001252	0.984	$5.36 \cdot 10^{-5}$	0.6136	$1.149 \cdot 10^{-5}$

Here the harmonic sequence also works better and we attain high precision using either sequence in standard double precision floating point arithmetic.

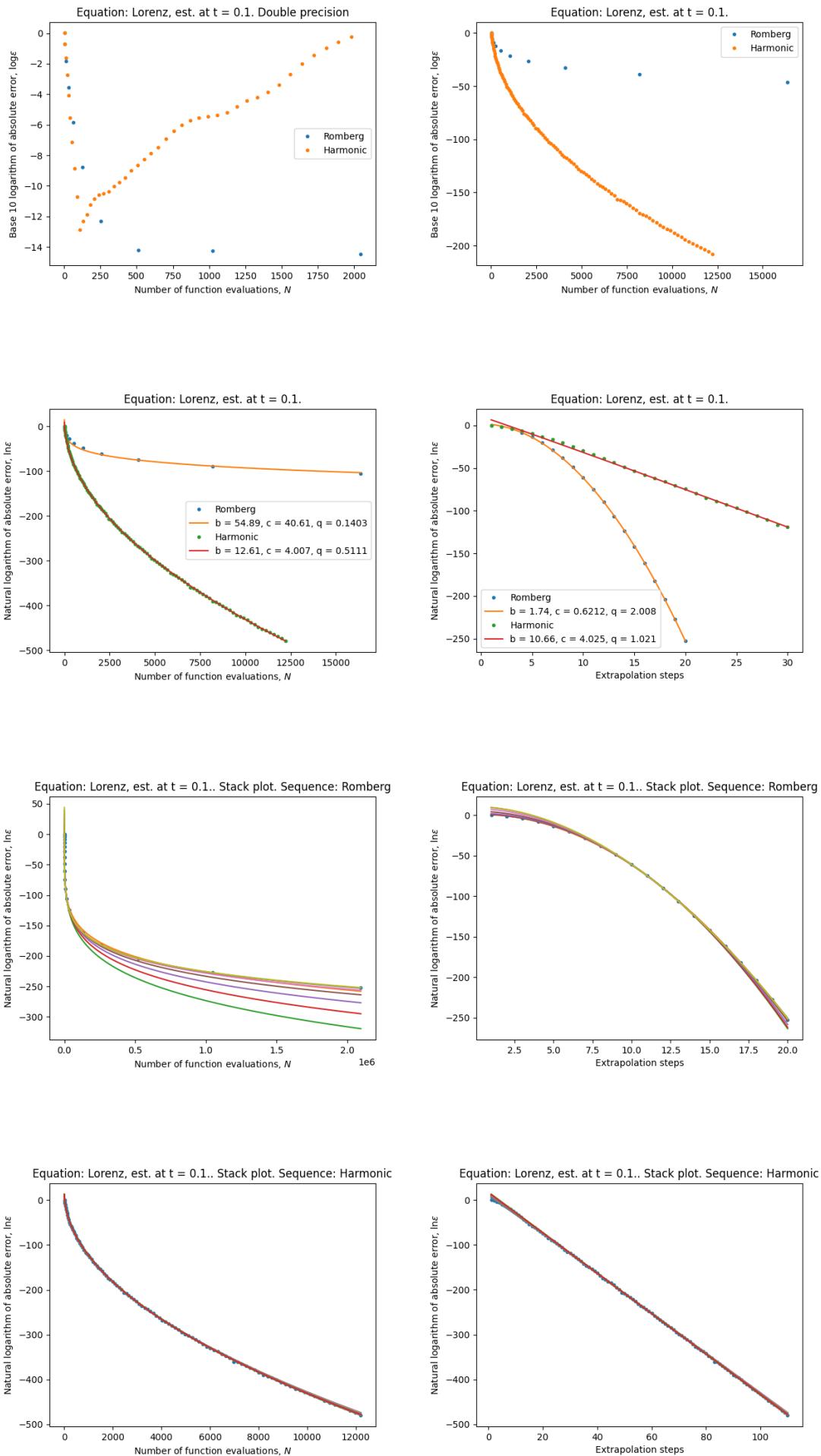
We seem to have very nice fit for exponential convergence in the number of steps for the Romberg sequence. We also have very nice fit for exponential convergence for the harmonic sequence.

#### 4.2.10 Lorenz equations

The Lorenz equations are the following system:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z$$

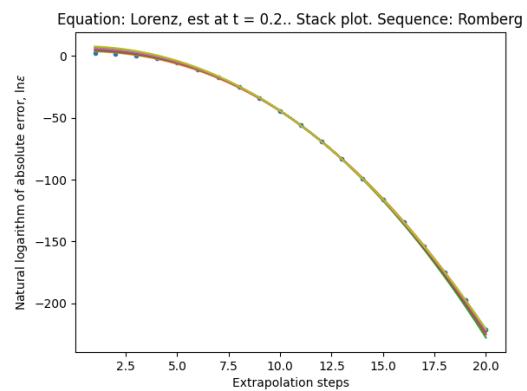
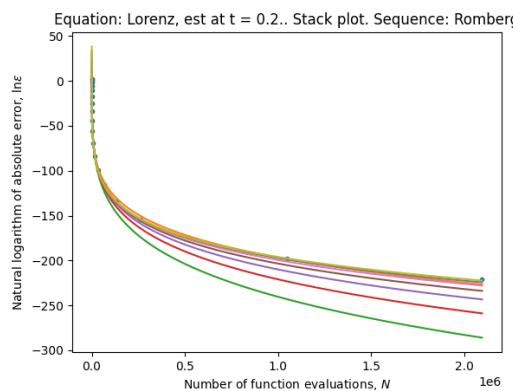
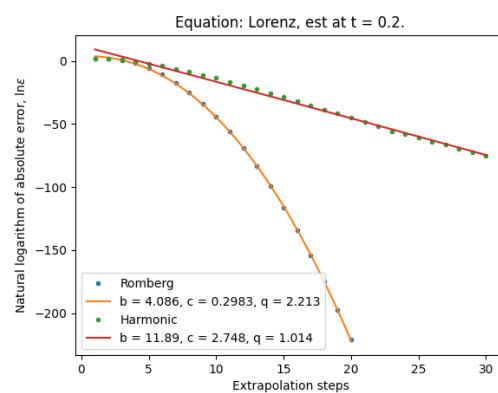
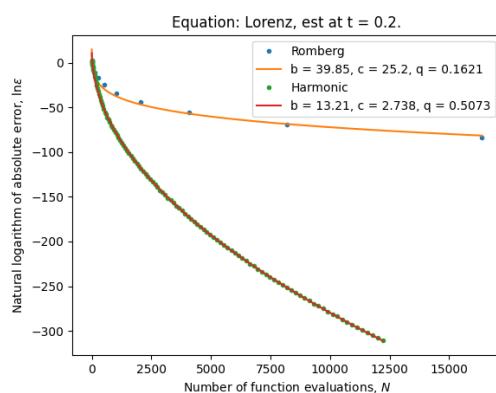
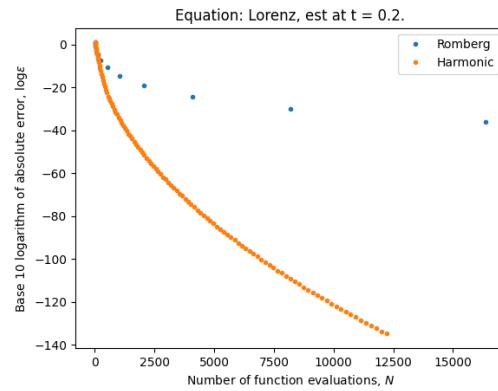
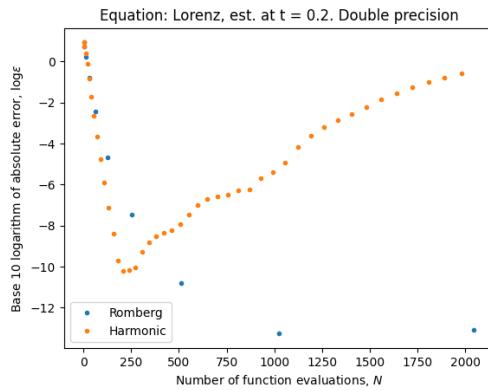
where  $\sigma$ ,  $\rho$  and  $\beta$  are constants. In our experiment, the constants are set to  $\sigma = 10$ ,  $\rho = 28$  and  $\beta = 8/3$ . The initial condition we will consider is  $(x(0), y(0), z(0)) = (1, 1, 1)$ .

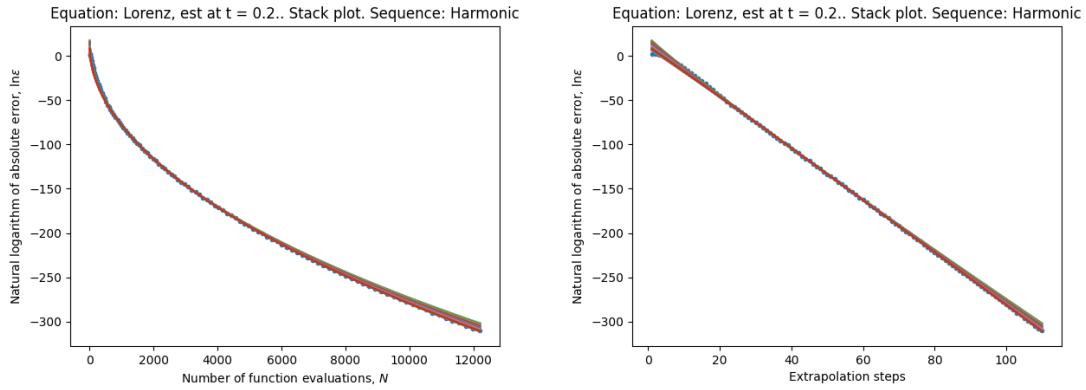


Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\ln}$	$\rho_{\ln}$
Romberg	$1.532 \cdot 10^{73}$	6	66.42	0.2606	0.1285	0.07315	$5.844 \cdot 10^8$	0.001216
Harmonic	$2.639 \cdot 10^7$	6.828	4.214	0.001814	0.5062	$7.152 \cdot 10^{-5}$	$9.241 \cdot 10^5$	$2.166 \cdot 10^{-5}$
Romberg	8932	2.321	0.7248	0.06447	1.976	0.001984	3.973	$3.557 \cdot 10^{-5}$
Harmonic	$2.903 \cdot 10^6$	6.603	4.22	0.001799	1.012	$7.141 \cdot 10^{-5}$	$5.369 \cdot 10^5$	$2.087 \cdot 10^{-5}$

Here the harmonic sequence works better and we get down to machine level precision in standard double precision arithmetic.

The model for exponential convergence in the number of steps fits moderately well for the Romberg sequence. We get nice fit for exponential convergence for the harmonic sequence.





Sequence	$A\text{-mean}$	$A\text{-var}$	$c\text{-mean}$	$c\text{-var}$	$q\text{-mean}$	$q\text{-var}$	$\rho_{\text{lin}}$	$\rho_{\ln}$
Romberg	$2.009 \cdot 10^{46}$	6	37.38	0.2052	0.1527	0.04644	$1.305 \cdot 10^8$	0.001529
Harmonic	$7.974 \cdot 10^8$	7.349	2.931	0.02946	0.5011	0.001063	$1.167 \cdot 10^6$	$5.043 \cdot 10^{-5}$
Romberg	1040	1.066	0.337	0.01327	2.18	0.0003518	13.23	$4.751 \cdot 10^{-5}$
Harmonic	$1.538 \cdot 10^8$	7.211	2.931	0.02863	1.002	0.001041	$8.184 \cdot 10^5$	$4.911 \cdot 10^{-5}$

Here we also get down to machine level precision in standard double precision floating point arithmetic. The harmonic sequence performs better.

The model for exponential convergence in the number of steps fits moderately well for the Romberg sequence. We get moderate fit for exponential convergence for the harmonic sequence.



# Bibliography

- [1] Perter Deuflhard and Andreas Hohmann. *Numerical Analysis in Modern Scientific Computing*, vol. 43 of Texts in Applied Mathematics, Springer, New York, 2003.
- [2] Konrad Knopp. *Theorie und Anwendung der unendlichen Reihen.*, Springer Verlag, Berlin, Heidelberg, New York, (5. Auflage) 1964.