Problem 1

This is a scalar problem with a slowly decaying mode:

$$y^{(0.5)} = -y + 1, \qquad t \in [0, 10^3], \qquad y(0) = 10.$$

One verifies that $y(t) \to 1$, as $t \to \infty$. Its solution is depicted in Figure 1. The width of the integration interval is chosen in order to observe the convergence to the limit point.

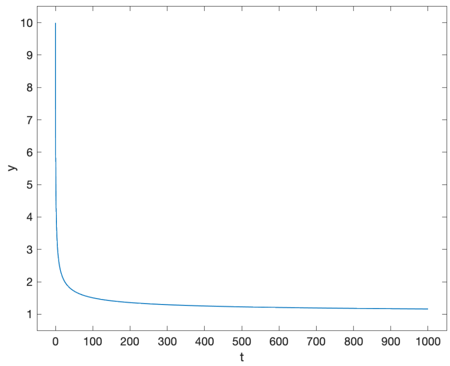

Figure 1. Solution of Problem 1.

For building the WPD for this problem, we used the following parameters for the codes:

- `fde12`, `fde12-10`: $\texttt{h} = 10^{-i}$, $i = 2, \ldots, 5$;
- `flmm2-1`, `flmm2-2`, `flmm2-3`: $\texttt{h} = 10^{-i}$, $i = 1, \ldots, 4$;
- `fcoll-3-5`, `fcoll-3-10`, `tsfcoll-3-5`, `tsfcoll-3-10`: $\texttt{N} = 500i$, $i = 1, \ldots, 5$;
- `fcoll-4-5`, `fcoll-4-10`, `tsfcoll-4-5`, `tsfcoll-4-10`: $\texttt{N} = 250i$, $i = 1, \ldots, 5$;
- `fcoll-5-5`, `fcoll-5-10`, `tsfcoll-5-5`, `tsfcoll-5-10`: $\texttt{N} = 100i$, $i = 1, \ldots, 5$;
- `fhbvm`: $\texttt{M} = 5, \ldots, 10$;
- `fhbvm2`: $\texttt{N} = \texttt{n} = 1$, $\texttt{nu} = 25i$, $i = 2, \ldots, 6$;

We observe that `tsfcoll-5-5` and `tsfcoll-5-10` exhibit convergence problems, so that we do not report the corresponding results. In Figure 2 we report the WPD obtained for this problem. From the reported plots, one infers that:

- `fde12` is the less efficient code (about 500 sec are needed to obtain less than 6 digits of accuracy), and additional iterations of the corrector are not effective, since they only increase the execution time, without improving accuracy;
- `flmm2` is more efficient that `fde12`, but less effective than the other codes. Moreover `flmm2-3` achieves slightly more than 7 digits accuracy, whereas `flmm2-1` and `flmm2-2` can achieve slightly less than 10 significant digits in about 70 sec;
- `fcoll` is more efficient than `flmm2`, and the higher the number of collocation points, the more accurate the solution, which can reach more than 12 digits in 0.5 sec. Moreover, the choice of the parameter `r` seems not affecting that much the performance, even though $\texttt{r} = 5$ is slightly better;
- `tsfcoll` performs quite differently, depending on the choice of the parameter `r`. In particular, the choice $\texttt{r} = 10$ is much more effective than $\texttt{r} = 5$. In fact, for the same value of `N`, the execution time is approximately the same but the accuracy is pretty different: with $\texttt{r} = 10$, `tsfcoll` can reach about 14 digits of accuracy in 6 sec by using 3 collocation points, which are approximately the double of those obtained by using $\texttt{r} = 5$. Further, the code appears to be more effective when using 3 collocation points only;

- `fhbvm` and `fhbvm2` are the most effective codes, with a uniform accuracy of about 14 digits, and a negligible execution time (of less than $5 \cdot 10^{-2}$ sec).

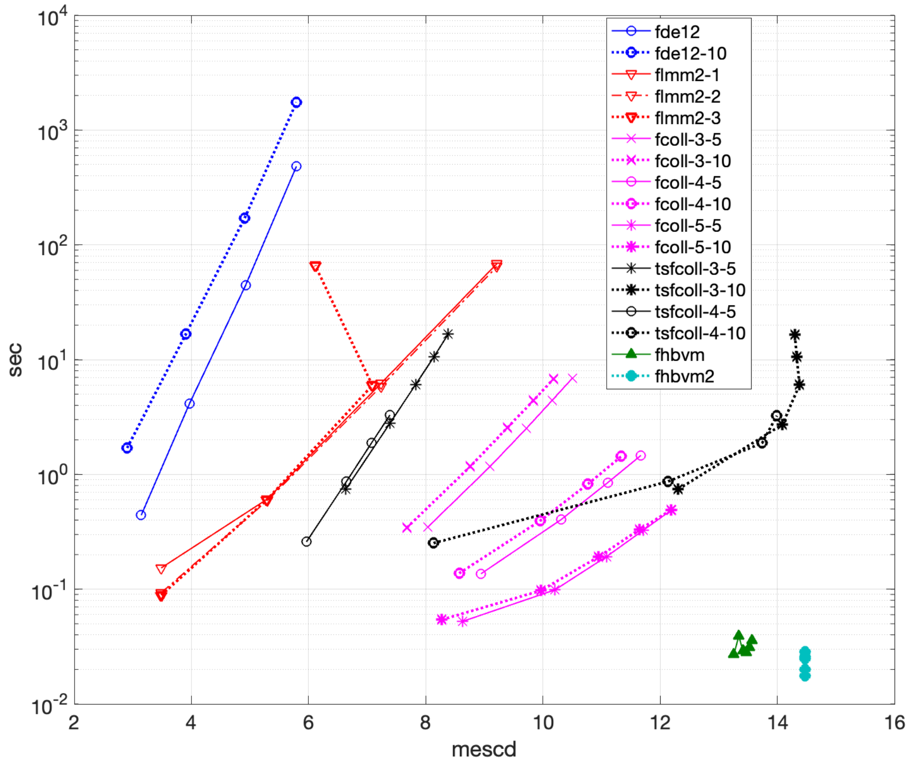

Figure 2. WPD for Problem 1.