The code `fde12`

This code is described in [1], and is available at the URL [2]. It implements a predictor-corrector method for approximating the solution of the problem (for sake of brevity, we omit t as an argument of the vector field):

$$y^{(\alpha)} = f(y), \qquad t \in [0,T], \qquad y^{(i)}(0) = y_0^i \in \mathbb{R}^m, \qquad i = 0,\dots,\lceil\alpha\rceil - 1,$$

at the grid points $t_j = jh$, $j = 0,1,\dots,N$, with $h = T/N$ the used timestep. By setting, as is usual, $y_j \approx y(t_j)$ and $f_j = f(y_j)$, one then obtains, for $n = 1,\dots,N$:

$$\begin{aligned}
y_n^0 &= T_\alpha(t_n) + \frac{h^\alpha}{\Gamma(\alpha+1)} \sum_{j=0}^{n-1} b_{n-j-1} f_j,\\
\Psi_n &= T_\alpha(t_n) + \frac{h^\alpha}{\Gamma(\alpha+2)} \left(a_n^0 f_0 + \sum_{j=1}^{n-1} a_{n-j} f_j \right),\\
y_n^i &= \Psi_n + \frac{h^\alpha}{\Gamma(\alpha+2)} f(y_n^{i-1}) \qquad i = 1,\dots,\mu,
\end{aligned}$$

with the coefficients

$$\begin{aligned}
b_n &= (n+1)^\alpha - n^\alpha,\\
a_n^0 &= (n-1)^{\alpha+1} - n^\alpha(n-\alpha-1),\\
a_n &= (n+1)^{\alpha+1} - 2n^{\alpha+1} + (n-1)^{\alpha+1}, \qquad n = 1,2,\dots.
\end{aligned}$$

This method can be regarded as a generalization of a particular Adams predictor-corrector pair to the case of FDEs, using μ correction iterations. It is worth noticing that the convolutions are computed via a fast FFT algorithm [3]. The calling sequence of the function is:

```
[t,y] = fde12(alpha,fdefun,t0,tfinal,y0,h,param,mu,mutol)
```

In output `t and y` contain the computed solution (`y` is stored by columns), whereas, in input:

- `alpha` is the order α of the derivative;
- `fdefun` is the identifier of the function evaluating the vector field (`fdefun(t,y)`) which returns a column vector;
- `[t0,tfinal]` is the integration interval;
- `y0` is a matrix of dimension $m \times \lceil\alpha\rceil$ with the initial conditions;
- `h` is the used stepsize, assumed constant;
- `param` (optional) contains possible parameters for `fdefun`;
- `mu` is the number of corrector iterations. The default value is $\mu = 1$, corresponding to the classical PECE implementation;
- `mutol` is the tolerance for testing the convergence of the corrector iteration (the default value is 10^{-6}).

[1] Garrappa, R. On linear stability of predictor-corrector algorithms for fractional differential equations. *Internat. J. Comput. Math.* **2010**, *87*, 2281–2290. https://doi.org/10.1080/00207160802624331

[2] http://www.mathworks.com/matlabcentral/fileexchange/32918 (accessed on March 24, 2025).

[3] Hairer, E.; Lubich, Ch.; Schlichte, S. Fast numerical solution of nonlinear Volterra convolution equations *SIAM J. Sci. Statist. Comput.* **1985**, *6*, 532–541. https://doi.org/10.1137/0906037