The code `tsfcoll`

This code is described and is published in [1]. It implements a two-step collocation method, aimed at increasing the order of accuracy w.r.t. the code `fcoll`. As this latter code, it only solves scalar problems:

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

In more details, given the mesh

$$t_0 = 0, \qquad t_j = t_{j-1} + h_j, \qquad j = 1,\dots,N,$$

and q collocation points

$$t_{ji} \equiv t_{i-1} + \eta_i h_j \in [t_{j-1}, t_j], \qquad i = 1,\dots,q,$$

the used method looks for approximations

$$z_{ji} \approx D^{(\alpha)}y(t_{ji}), \qquad i = 1,\dots,q,$$

such that, at first, the code `fcoll` (actually, a slight modification of it) is used to advance the solution from $t_0 = 0$ to $t_1 = h_1$ (i.e., the code `fcoll` is the so called *starting procedure* of the two-step method). After that, for $j = 2,\dots,N$, to advance the solution from t_{j-1} to t_j, one solves a collocation problem in the form

$$z_{jk} = f\left(t_{jk},\, \hat{\phi}_j(t_{jk}) + \sum_{i=1}^{q}\left(z_{ji} I^{\alpha}\hat{\ell}^j_{q+i}(t_{jk}) + z_{j-1,i} I^{\alpha}\hat{\ell}^j_i(t_{jk})\right)\right), \qquad k = 1,\dots,q,$$

with $\hat{\ell}^j_i(t) \in \Pi_{2q-1},\ i = 1,\dots,2q,$ the ith Lagrange polynomial defined on the $2q$ abscissae $\{t_{j-1,1},\dots,t_{j-1,q}, t_{j1},\dots,t_{jq}\}$, $I^{\alpha}\hat{\ell}^j_i(t)$ its Riemann-Liouville integral, and the *memory* term given by

$$\hat{\phi}_j(t) := \sum_{i=0}^{\lceil\alpha\rceil-1} \frac{t^i}{i!} y_0^i + \sum_{\mu=2}^{j-1}\sum_{i=1}^{q}\left(z_{\mu i} I^{\alpha}\hat{\ell}^{\mu}_{q+i}(t) + z_{\mu-1,i} I^{\alpha}\hat{\ell}^{\mu}_i(t)\right), \qquad t \in [t_{j-1}, t_j].$$

This system of equations is solved by using the Matlab© function `fsolve` (by using very tiny tolerances) and, moreover, a graded mesh in the form

$$t_j = \left(\frac{j}{N}\right)^r, \qquad j = 0,1,\dots,N, \qquad \text{with} \quad r \geq 1,$$

may be considered, to cope with possible nonsmooth vector fields at the starting point. The calling sequence of the code `tsfcoll` is:

```
[t,y] = tsfcoll(f,b,gam,alpha,eta,r,N)
```

In output `t` and `y` contain the computed solution, whereas, in input:

- `f` is the identifier of the function implementing the vector field (`f(t,y)`);
- `b` is the final integration time (the starting time is 0);
- `gam` is a column vector containing the initial conditions;
- `alpha` is the order of the fractional derivative;
- `eta` is a column vector of the normalized collocation abscissae defined above,

$$\eta_i = \frac{t_{ji} - t_{j-1}}{h_j}, \qquad i = 1\dots,q;$$

- `r` is the parameter for the graded mesh;
- `N` is the number of mesh points.

[1] Cardone, A; Conte, D.; Paternoster, B. A MATLAB Code for Fractional Differential Equations Based on Two-Step Spline Collocation Methods. In: *Cardone A. et al. (eds.), Fractional Differential Equations – INDAM 2021*. Springer INdAM Series *50*, **2023** pp. 121–146. https://doi.org/10.1007/978-981-19-7716-9_8