IOCFF 2026 Conference

The 1st International Online Conference on Fractal and Fractional

13–15 April 2026 | Online

MDPI

Spectrally accurate collocation methods for fractional differential equations

Luigi Brugnano [1], Gianmarco Gurioli [1], Felice Iavernaro [2], Mikk Vikerpuur [3]

[1] Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli Studi di Firenze (Italy), e-mail: luigi.brugnano@unifi.it - gianmarco.gurioli@unifi.it
[2] Dipartimento di Matematica, Università degli Studi di Bari (Italy), e-mail: felice.iavernaro@uniba.it
[3] Institute of Mathematics and Statistics, University of Tartu (Estonia), e-mail: mikk.vikerpuur@ut.ee

INTRODUCTION & AIM

Reference Initial Value Problems for Fractional Differential Equations (IVP-FDEs)

$$\begin{cases} y^{(\alpha)}(t) = f(y(t)), & t \in [0,T] \\ y^{(j)}(0) = y_0^j \in \mathbb{R}^m, & j = 0,\dots,\ell-1 \end{cases},$$

with $\alpha \in (\ell-1,\ell)$, $T>0$, $\ell \in \mathbb{N}$, $\ell \geq 1$, $m \in \mathbb{N}$, $f\colon \mathbb{R}^m \to \mathbb{R}^m$ a suitably regular function and

$$y^{(\alpha)}(t) = \frac{1}{\Gamma(\ell-\alpha)} \int_0^t (t-x)^{\ell-1-\alpha} y^{(\ell)}(x)\mathrm{d}x$$

the *Caputo fractional derivative*.

Solution

$$y(t) = T_\ell(t) + I^\alpha f(y(t)) := \sum_{j=0}^{\ell-1} \frac{t^j}{j!} y_0^j + \frac{1}{\Gamma(\alpha)} \int_0^t (t-x)^{\alpha-1} f(y(x))\mathrm{d}x,$$

with $I^\alpha f(y(t))$ the *Riemann-Liouville integral* and the *Euler Gamma function* $\Gamma(\alpha)$.

Challenge

Such IVP-FDEs are widely spreading in many application contexts, but their numerical resolution can be challenging because of the *nonlocality* of the differential operator and the persistency of the *memory requirement* in the first solution addend, with the risk of long-time simulations being computationally demanding.

Proposal

Fractional Hamiltonian Boundary Value Methods (FHBVMs), a class of *spectrally accurate* and efficiently implementable Runge-Kutta type *collocation methods*, are proposed to mitigate the aforementioned issues [1,2,3,4].

METHOD

- Consider the reference time interval $[0,h]$. The core idea is to adopt a *truncated Fourier expansion* of the *vector field* along the $\{P_j\}_{j\geq 0}$ shifted and scaled *Jacobi polynomials*, that are *orthonormal* with respect to the inner product defined by the *weighting function* $\omega(c) = \alpha(1-c)^{\alpha-1}$, i.e.,

$$\int_0^1 \omega(c) P_i(c) P_j(c)\mathrm{d}c = \delta_{ij}, \quad i,j \in \mathbb{N}.$$

- In so doing, the problem and the corresponding *approximation* σ of the solution y take the form:

$$\begin{cases} \sigma^{(\alpha)}(ch) = \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma), & c \in [0,1] \\ \sigma^{(j)}(0) = y_0^j \in \mathbb{R}^m, & j = 0,\dots,\ell-1 \end{cases} \Rightarrow \quad \begin{aligned} \sigma(ch) &= T_\ell(ch) + h^\alpha \sum_{j=0}^{s-1} I^\alpha P_j(c)\gamma_j(\sigma) \\ \sigma(h) &= T_\ell(h) + \frac{h^\alpha}{\Gamma(\alpha+1)}\gamma_0(\sigma) \end{aligned}.$$

- The *Fourier coefficients*

$$\gamma_j(\sigma) = \int_0^1 \omega(\tau) P_j(\tau) f(\sigma(\tau h))\mathrm{d}\tau, \quad j = 0,\dots,s-1$$

are then approximated via the *Gauss-Jacobi quadrature* of order 2k, having abscissae placed at the zeros of the Jacobi polynomial of degree k [2].

- The scheme generates the method *FHBVM(k,s)*, k ≥ s, of *order 2s*, with the following properties [2]:
  - the method admits a *k-stage Runge-Kutta formulation*;
  - the *underlying discrete problem* to be solved to recover the approximation of the solution at the end time can be proved to have *block dimension s* and to be *efficiently solvable* using a fixed-point / simplified Newton / blended implementation; for insight on such implementation issues, we refer to [4].
  - FHBVM(s,s) are *spectrally accurate in time collocation methods*, for sufficiently large values of s.

- In practical implementations, the time interval [0,T] is partitioned basing on the specific features of the problem, by automatically selecting either a *uniform* or a *graded* time mesh (Matlab code *fhbvm*) [3,4].

  To handle problems with nonsmooth solution / vector field at the starting time, but eventually periodic solution, the novel Matlab code *fhbvm2* provides a variant combining an initial graded mesh with a subsequent uniform one [1,2].

- Both the resulting Matlab codes *fhbvm* and *fhbvm2* are *publicly available* at https://people.dimai.unifi.it/brugnano/fhbvm/

RESULTS & DISCUSSION

Fractional variant of the Brussellator problem

As a representative numerical illustration, consider the following *fractional* variant of the 2-dimensional *Brussellator problem*, arising in *physical chemistry* [1]:

$$\begin{cases} y_1^{(0.7)} = 1 - 4y_1 + y_1^2 y_2, \quad y_2^{(0.7)} = 3y_1 - y_1^2 y_2, \quad t \in [0,200] \\ y(0) = (1.2,\ 2.8)^\top \end{cases},$$

whose solution / vector field are nonsmooth at the starting time, with an eventually periodic solution.

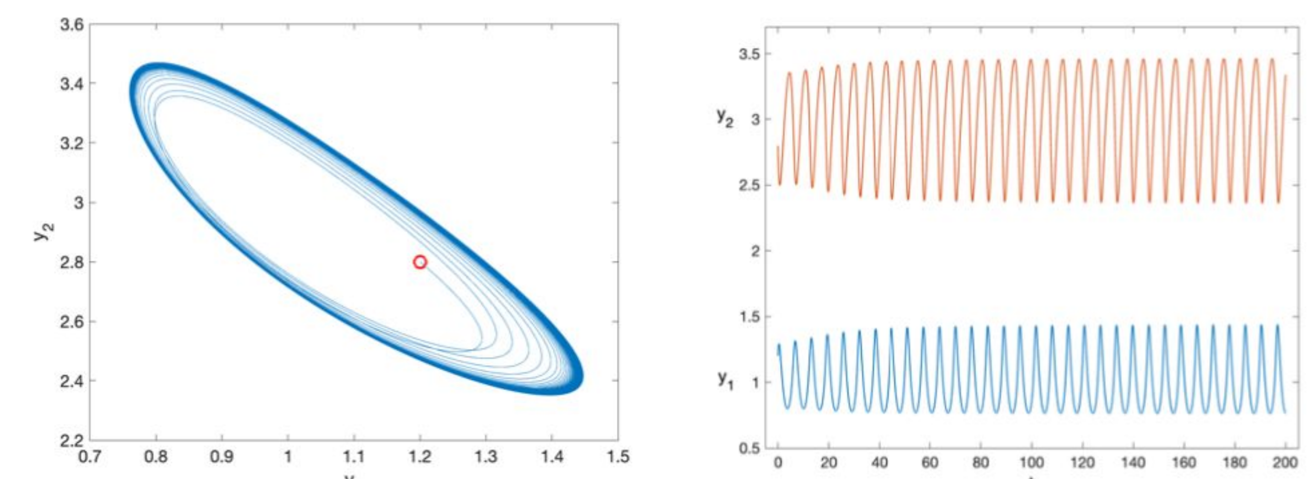

Figure. Orbit (left) and solution components (right) of the Brussellator problem.

For further numerical tests, please visit https://people.dimai.unifi.it/brugnano/FDEtestset/

Numerical simulation

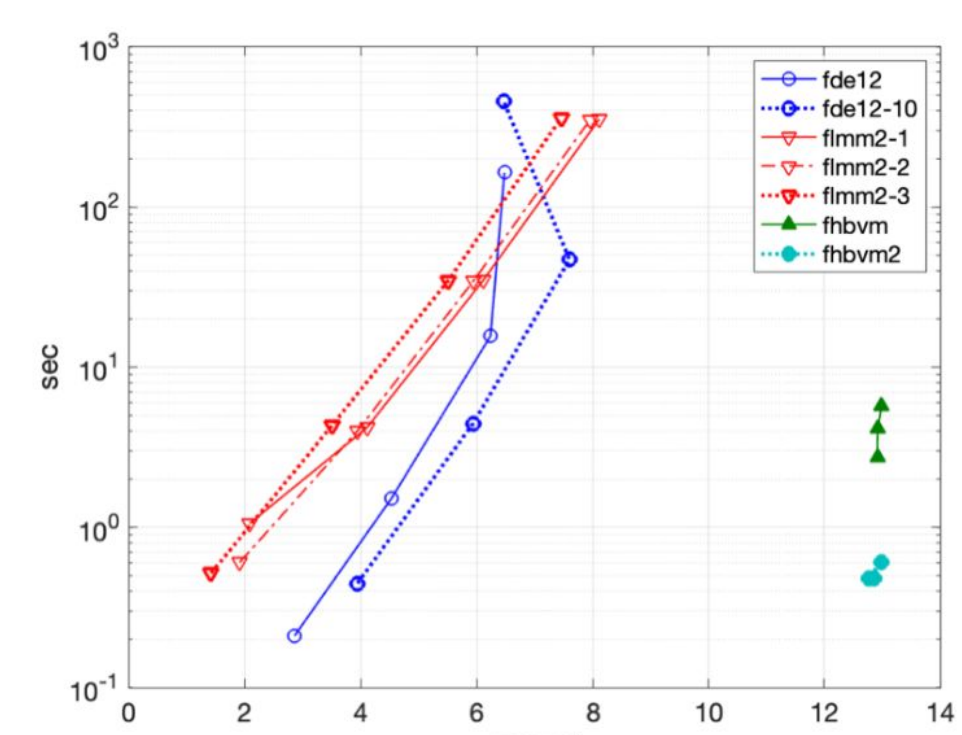

Figure. Execution time vs. number of exact significant digits [1,2] in the solution. Codes: *fhbvm2*, *fhbvm*, 2nd order *fractional linear multisteps flmm2-i* [1] (i=1: trapezoidal rule, i=2 Newton-Gregory rule, i=3: 2nd order BDF), *predictor-corrector fde12-m* [1] (with *m* corrections).

Interpretation of the results

While all the other methods reach at most *8* exact significant digits, *fhbvm* and *fhbvm2* achieve a uniform accuracy of *13* digits, with a *much lower* computation time (the new *fhbvm2* outperforms *fhbvm*).

CONCLUSION

FHBVMs are a class of *spectrally accurate collocation methods* for the *efficient solution* of *IVP-FDEs* of the *Caputo type*. They are *implemented* in *freely available Matlab codes*. The implementation *benefits* from the *recent release fhbvm2*, adopting an initial graded time mesh followed by a uniform one, for problems with nonsmooth solution/vector field at the starting time, but an eventually periodic solution.

FUTURE WORK / REFERENCES

Extension of the methods to solve *systems of FDEs* showing *different orders* of the *fractional derivatives*.

[1] L. Brugnano, G. Gurioli, F. Iavernaro, M. Vikerpuur. "*FDE-Testset: Comparing Matlab Codes for Solving Fractional Differential Equations of Caputo Type*". Fractal and Fractional **9**(5), 312, 2025.

[2] L. Brugnano, G. Gurioli, F. Iavernaro, M. Vikerpuur. "*Analysis and implementation of collocation methods for fractional differential equations*". Journal of Scientific Computing **104**(3), 92, 2025.

[3] L. Brugnano, G. Gurioli, F. Iavernaro. "*Solving FDE-IVPs by using Fractional HBVMs: Some experiments with the fhbvm code*". Journal of Computational Methods in Sciences and Engineering **25**(1), 1030-1038, 2025.

[4] L. Brugnano, G. Gurioli, F. Iavernaro. "*Numerical solution of FDE-IVPs by using fractional HBVMs: the fhbvm code.*" Numerical Algorithms **99**(1), 463-489, 2025.