Recent advances in the numerical solution of fractional differential equations

Luigi Brugnano
University of Florence, Italy
https://people.dimai.unifi.it/brugnano/

Collaborations

- Kevin Burrage, QUT, Australia
- Pamela Burrage, QUT, Australia
- Gianmarco Gurioli, University of Florence, Italy
- Felice Iavernaro, University of Bari, Italy
- Mikk Vikerpuur, University of Tartu, Estonia

Overview

- Numerical solution of ODEs by using a local expansion of the vector field
- Hamiltonian Boundary Value Methods (HBVMs)
- HBVMs as spectral methods in time
- Extension to Fractional Differential Equations (FDEs)
- Fractional HBVMs (FHBVMs)
- Numerical results taken from the FDE-Testset
- Conclusions

ODE-IVPs

Assume we want to solve the ODE-IVP:

$$y'(t) = f(y(t)), \qquad t \in [0, T], \qquad y(0) = y_0 \in \mathbb{R}^m,$$

whose solution is very well known to be given by

$$y(t) = y_0 + \int_0^t f(y(x))\mathrm{d}x \equiv y_0 + I^1 f(y(t)), \qquad t \in [0, T],$$

where in general, for $\alpha > 0$.

$$I^\alpha g(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-x)^{\alpha-1} g(x)\mathrm{d}x.$$

Hereafter, we shall assume f to be enough regular.

Discrete mesh

We shall also assume that the following mesh is given,

$$t_0 = 0, \qquad t_n = t_{n-1} + h_n, \qquad n = 1, \ldots, N, \qquad t_N = T,$$

and we denote by

$$y_n(ch_n) = y(t_{n-1} + ch_n), \qquad c \in [0, 1].$$

Consequently, the original IVP can be formally rewritten as the following sequence of local problems:

$$\begin{aligned} y_n'(ch_n) &= f(y_n(ch_n)), \qquad c \in [0,1], \qquad n = 1, \ldots, N, \\ y_1(0) &= y_0 \in \mathbb{R}^m. \end{aligned}$$

Local problem

As is clear, the solution of the nth local problem is given by:

$$\begin{aligned} y_n(ch_n) &= y(t_{n-1}+ch_n) = y_0 + \int_0^{t_{n-1}+ch_n} f(y(x))\mathrm{d}x \\ &= y(t_{n-1}) + \int_0^{ch_n} f(y_n(x))\mathrm{d}x \\ &= \phi_n^1(ch_n) + I^1 f(y_n(ch_n)), \qquad c \in [0,1], \end{aligned}$$

having set

$$\phi_n^1(ch_n) \equiv y(t_{n-1}), \qquad c \in [0,1], \qquad n = 1,\dots,N.$$

Local expansion

Next, let us consider the expansion of the nth vector field along the Legendre orthonormal polynomial basis:

$$P_i \in \Pi_i, \qquad \int_0^1 P_i(c)P_j(c)\mathrm{d}c = \delta_{ij}, \qquad i,j = 0,1,\dots.$$

Consequently,

$$f(y_n(ch_n)) = \sum_{j\geq 0} P_j(c)\gamma_j(y_n), \qquad c \in [0,1],$$

with the Fourier coefficients defined by:

$$\gamma_j(y_n) = \int_0^1 P_j(c) f(y_n(ch_n))\mathrm{d}c, \qquad j = 0,1,\dots.$$

Equivalent form

Therefore,

$$
\begin{aligned}
y_n'(ch_n) &= \sum_{j\geq 0} P_j(c)\gamma_j(y_n), \qquad c\in[0,1], \qquad n=1,\ldots,N,\\
y_1(0) &= y_0\in\mathbb{R}^m,
\end{aligned}
$$

thus providing the local solutions formally given by:

$$
\begin{aligned}
y_n(ch_n) &= \phi_n^1(ch_n) + h_n\sum_{j\geq 0} I^1 P_j(c)\gamma_j(y_n), \qquad c\in[0,1],\\
n &= 1,\ldots,N.
\end{aligned}
$$

Polynomial approximation

We can derive a piecewise polynomial approximation, $\sigma(t) \approx y(t)$, s.t.:

$$\sigma_n(ch_n) \approx y_n(ch_n), \qquad c \in [0,1], \qquad n = 1,\dots,N,$$

by truncating the infinite series to finite sums:

$$\begin{aligned} \sigma_n'(ch_n) &= \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma_n), \qquad c \in [0,1], \qquad n = 1,\dots,N, \\ \sigma_1(0) &= y_0 \in \mathbb{R}^m, \end{aligned}$$

so that, for $n = 1,\dots,N$,

$$\sigma_n(ch_n) = \phi_n^{1,s}(ch_n) + h_n \sum_{j=0}^{s-1} I^1 P_j(c)\gamma_j(\sigma_n), \qquad c \in [0,1],$$

having set $\phi_n^{1,s}(ch_n) \equiv \sigma(t_{n-1}), \qquad c \in [0,1].$

Spectral accuracy

The local problem

$$\sigma_n'(ch_n) = \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma_n)$$

$$\equiv \sum_{j=0}^{s-1} P_j(c)\int_0^1 P_j(\tau) f(\sigma_n(\tau h_n))\mathrm{d}\tau, \qquad c \in [0,1],$$

is clearly equivalent to requiring the residual,

$$r_n(ch_n) := \sigma_n'(ch_n) - f(\sigma_n(ch_n)), \qquad c \in [0,1],$$

be orthogonal to all polynomials in Π_{s-1} w.r.t. the $L_2[0,1]$ product.

Consequently, a spectrally accurate solution in time can be expected, when $s \gg 1$.

HBVM(k, s)

Further, by approximating the Fourier coefficients through the Gauss-Legendre formula of order $2k$,

$$\gamma_j(\sigma_n) \approx \sum_{i=1}^{k} b_i P_j(c_i) f(\sigma_n(c_i h_n)) =: \gamma_j^n, \qquad j = 0, \ldots, s-1,$$

with $P_k(c_i) = 0$, $i = 1, \ldots, k$, then gives a Hamiltonian Boundary Value Method with parameters (k, s). In short HBVM(k, s).

For all $k \geq s$:

- formally a symmetric k-stage Runge-Kutta method of order $2s$;
- when $k = s$ it reduces to the s-stage Gauss-collocation method;
- when used for solving Hamiltonian problems, the Hamiltonian error is at most $O(h_n^{2k+1})$ (and possibly 0, for k large enough).

Discrete problem

Though formally a k-stage Runge-Kutta method, one can recast the discrete problem in terms of the s approximate Fourier coefficients:

$$\gamma_j^n = \sum_{i=1}^{k} b_i P_j(c_i) f\left(\phi_n^{1,s}(c_i h_n) + h_n \sum_{\ell=0}^{s-1} I^1 P_\ell(c_i)\gamma_\ell^n\right), \qquad j = 0, \ldots, s-1,$$

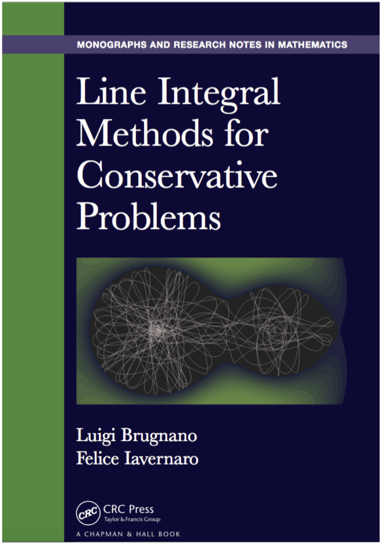
with the new approximation given by

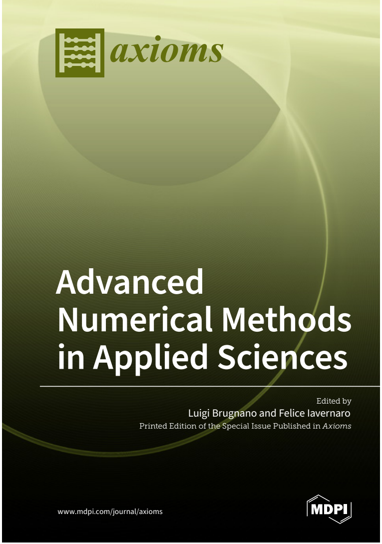
$$\sigma(t_{n+1}) \;=\; \phi_n^{1,s}(h_n) + h_n \gamma_0^n.$$

An extremely effective Newton-type iteration is available for its solution (the leading term in the cost being independent of s).

This allows the use of $k \geq s \gg 1$, thus allowing the use of the methods as spectral methods in time (particularly effective for semilinear problems).

References





The Matlab© code hbvm.m is available at the website of the book.

FDE-IVPs

Assume we want now to solve the IVP of fractional differential equations:

$$y^{(\alpha)}(t) = f(y(t)), \qquad t \in [0, T], \qquad y(0) = y_0 \in \mathbb{R}^m,$$

with, for $\alpha \in (0,1)$ and assuming y absolutely cointinuous in $[0, T]$,

$$y^{(\alpha)}(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-x)^{-\alpha} y'(x) \mathrm{d}x$$

the Caputo fractional derivative of y.

Under regularity assumption on f, it is known that its solution is given by:

$$y(t) = y_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-x)^{\alpha-1} f(y(x)) \mathrm{d}x \equiv y_0 + I^\alpha f(y(t)), \qquad t \in [0, T].$$

Discrete mesh

As in the ODE case, let us assume that the following mesh is given,

$$t_0 = 0, \qquad t_n = t_{n-1} + h_n, \qquad n = 1, \dots, N, \qquad t_N = T,$$

and denote by

$$y_n(ch_n) = y(t_{n-1} + ch_n), \qquad c \in [0, 1].$$

Consequently, the original IVP can be formally rewritten as the following sequence of local problems:

$$\begin{aligned} {y_n}^{(\alpha)}(ch_n) &= f(y_n(ch_n)), \qquad c \in [0,1], \qquad n = 1, \dots, N, \\ y_1(0) &= y_0 \in \mathbb{R}^m. \end{aligned}$$

Local problem

As is clear, the solution of the nth local problem is now given by:

$$\begin{aligned}
y_n(ch_n) &= y(t_{n-1}+ch_n) = y_0 + I^\alpha f(y(t_{n-1}+ch_n)) \\
&= y_0 + \frac{1}{\Gamma(\alpha)}\int_0^{t_{n-1}+ch_n}(t_{n-1}+ch_n-x)^{\alpha-1} f(y(x))\mathrm{d}x \\
&= \underbrace{y_0 + \frac{1}{\Gamma(\alpha)}\int_0^{t_{n-1}}(t_{n-1}+ch_n-x)^{\alpha-1} f(y(x))\mathrm{d}x}_{=:\,\phi_n^\alpha(ch_n)} \\
&\quad + \frac{1}{\Gamma(\alpha)}\int_0^{ch_n}(ch_n-x)^{\alpha-1} f(y_n(x))\mathrm{d}x \\
&= \phi_n^\alpha(ch_n) + I^\alpha f(y_n(ch_n)), \qquad c\in[0,1],
\end{aligned}$$

with $\phi_n^\alpha(ch_n)$ a memory term.

Local expansion

Next, let us consider the expansion of the nth vector field along the following Jacobi orthonormal polynomial basis:

$$P_i \in \Pi_i, \qquad \int_0^1 \omega(c) P_i(c) P_j(c) \mathrm{d}c = \delta_{ij}, \qquad i,j = 0,1,\ldots,$$

with $\omega(c) = \alpha(1-c)^{\alpha-1}$. Consequently,

$$f(y_n(ch_n)) = \sum_{j\geq 0} P_j(c)\gamma_j(y_n), \qquad c \in [0,1],$$

with the Fourier coefficients defined by:

$$\gamma_j(y_n) = \int_0^1 \omega(c) P_j(c) f(y_n(ch_n)) \mathrm{d}c, \qquad j = 0,1,\ldots.$$

Equivalent form

Therefore,

$$\begin{aligned} {y_n}^{(\alpha)}(ch_n) &= \sum_{j\ge 0} P_j(c)\gamma_j(y_n), \qquad c\in[0,1], \qquad n=1,\dots,N,\\ y_1(0) &= y_0\in\mathbb{R}^m, \end{aligned}$$

thus providing the local solutions formally given by:

$$\begin{aligned} y_n(ch_n) &= \phi_n^\alpha(ch_n) + h_n\sum_{j\ge 0} I^\alpha P_j(c)\gamma_j(y_n), \qquad c\in[0,1],\\ n &= 1,\dots,N. \end{aligned}$$

Polynomial approximation

We can derive a piecewise approximation, $\sigma(t) \approx y(t)$, s.t.:

$$\sigma_n(ch_n) \approx y_n(ch_n), \qquad c \in [0,1], \qquad n = 1, \dots, N,$$

by truncating the infinite series to finite sums:

$$\begin{aligned} {\sigma_n}^{(\alpha)}(ch_n) &= \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma_n), \qquad c \in [0,1], \qquad n = 1, \dots, N, \\ \sigma_1(0) &= y_0 \in \mathbb{R}^m. \end{aligned}$$

Consequenly, for $n = 1, \dots, N$,

$$\sigma_n(ch_n) = \phi_n^{\alpha,s}(ch_n) + h_n \sum_{j=0}^{s-1} I^\alpha P_j(c)\gamma_j(\sigma_n), \qquad c \in [0,1],$$

having set $\phi_n^{\alpha,s}(ch_n)$ the corresponding truncated memory term.

Variational interpretation

Alike the ODE case,

$$\begin{aligned}
{\sigma_n}^{(\alpha)}(ch_n) &= \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma_n) \\
&\equiv \sum_{j=0}^{s-1} P_j(c) \int_0^1 \omega(\tau) P_j(\tau) f(\sigma_n(\tau h_n))\mathrm{d}\tau, \qquad c \in [0,1],
\end{aligned}$$

is clearly equivalent to require that the residual,

$$r_n(ch_n) := {\sigma_n}^{(\alpha)}(ch_n) - f(\sigma_n(ch_n)), \qquad c \in [0,1],$$

be orthogonal to all polynomials in Π_{s-1} w.r.t. the $L_2[0,1]$ weighted product.

Consequently, a spectrally accurate solution in time can be expected, when $s \gg 1$, also in the FDE case.

FHBVM(k, s)

Further, by approximating the Fourier coefficients through the Gauss-Jacobi formula of order $2k$,

$$\gamma_j(\sigma_n) \approx \sum_{i=1}^{k} b_i P_j(c_i) f(\sigma_n(c_i h_n)) =: \gamma_j^n, \qquad j = 0, \ldots, s-1,$$

with $P_k(c_i) = 0$, $i = 1, \ldots, k$, then gives a Fractional Hamiltonian Boundary Value Method with parameters (k, s). In short FHBVM(k, s).

For all $k \geq s$:

- formally a k-stage Runge-Kutta type method;
- when $k = s$ it becomes a collocation method at the Jacobi abscissae;
- when a uniform mesh is allowed, error bounded by $O(h^{s+\alpha-1})$;
- when the graded mesh $h_n = r h_{n-1}$, $n = 1, \ldots, N$, $r > 1$ is used, error bounded by $O(h_1^{2\alpha} + h_N^{s+\alpha})$.

Discrete problem

Though formally a k-stage Runge-Kutta type method, one can recast the discrete problem in terms of the s approximate Fourier coefficients:

$$\gamma_j^n = \sum_{i=1}^{k} b_i P_j(c_i) f\left(\phi_n^{\alpha,s}(c_i h_n) + h_n \sum_{\ell=0}^{s-1} I^\alpha P_\ell(c_i)\gamma_\ell^n\right), \qquad j = 0,\ldots,s-1,$$

with the new approximation given by

$$\sigma(t_{n+1}) \;=\; \phi_n^{\alpha,s}(h_n) + \frac{h_n}{\Gamma(\alpha+1)}\gamma_0^n.$$

As for HBVMs, an extremely effective Newton-type iteration is available for its solution (the leading term in the cost being independent of s).

This allows the use of $k \geq s \gg 1$, thus allowing the use of the methods as spectral methods in time (particularly effective for semilinear problems).

References

1. L.Brugnano, K.Burrage, P.Burrage, F.Iavernaro. A spectrally accurate step-by-step method for the numerical solution of fractional differential equations. *J. Sci. Comput.* 99 (2024) 48. `https://doi.org/10.1007/s10915-024-02517-1`

2. L.Brugnano, G.Gurioli, F.Iavernaro. A shooting-Newton procedure for solving fractional terminal value problems. *Appl. Math. Comput.* 489 (2025) 129164. `https://doi.org/10.1016/j.amc.2024.129164`

3. L.Brugnano, G.Gurioli, F.Iavernaro. Numerical solution of FDE-IVPs by using Fractional HBVMs: the `fhbvm` code. *Numer. Algorithms* 99 (2025) 463-489. `https://doi.org/10.1007/s11075-024-01884-y`

4. L.Brugnano, G.Gurioli, F.Iavernaro. Solving FDE-IVPs by using Fractional HBVMs: some experiments with the `fhbvm` code. *J. Comput. Methods Sci. Eng.* 25(1) (2025) 1030-1038. `https://doi.org/10.1177/14727978251321328`

5. LBrugnano, G.Gurioli, F.Iavernaro, M.Vikerpuur. Analysis and implementation of collocation methods for fractional differential equations. *arXiv:2503.17719* [math.NA] `https://doi.org/10.48550/arXiv.2503.17719`

https://people.dimai.unifi.it/brugnano/fhbvm/

people.dimai.unifi.it

Posta in arrivo (... | Editorial Manage... | esoneri - Googl... | Matlab FDE-Test... | FHBVM Matlab... | similar - Diziona... | B006819 (B032... | Pubblicazioni -... | Program - Go20... | Acquista iPad W... | Research | go20 con

Fractional HBVMs

Matlab® Software

Main programs:

- fhbvm.m (Rel. 2025-05-04) [2,3,4]
- fhbvm2.m (Rel. 2025-04-08) [1,2,3,4] (improved version of the code fhbvm.m)

Examples from reference 1.:

- ex1.m
- osci.m
- vdp.m
- ex4.m

Examples from reference 2.:

- ex1.m
- ex2.m
- ex3.m
- ex4.m

Examples from reference 3.:

- exe1.m
- exe2.m
- exe3.m
- exe4.m

References:

1. L.Brugnano, G.Gurioli, F.Iavernaro, M.Vikerpuur. **Analysis and implementation of collocation methods for fractional differential equations.** arXiv:2503.17719 [math.NA] https://doi.org/10.48550/arXiv.2503.17719
2. L.Brugnano, G.Gurioli, F.Iavernaro. **Numerical solution of FDE-IVPs by using Fractional HBVMs: the fhbvm code.** *Numer. Algorithms* **99** (2025) 463-489. https://doi.org/10.1007/s11075-024-01884-y
3. L.Brugnano, G.Gurioli, F.Iavernaro. **Solving FDE-IVPs by using Fractional HBVMs: some experiments with the fhbvm code.** *J. Comput. Methods Sci. Eng.* **25**(1) (2025) 1030-1038. https://doi.org/10.1177/14727978251321328 (preprint on arXiv.2407.11460 [math.NA])
4. L.Brugnano, K.Burrage, P.Burrage, F.Iavernaro. **A spectrally accurate step-by-step method for the numerical solution of fractional differential equations.** *J. Sci. Comput.* **99** (2024) 48. https://doi.org/10.1007/s10915-024-02517-1

The Matlab© codes fhbvm and fhbvm2

fhbvm:

- implements a FHBVM(22,20) method
- automatically recognizes which mesh has to be used (uniform or graded)
- possibility of error estimation on a doubled mesh
- almost no tuning parameter is needed

fhbvm2:

- implements a FHBVM(22,22) method
- possibility of combining an initial graded mesh with a subsequent uniform one
- very few tuning parameters are needed

Numerical tests

We compare the following codes available on the web, or published in the literature:

- fde12 (fde12 and fde12-10)
- flmm2 (flmm2-1, flmm2-2, and flmm2-3)
- fcoll (fcoll-s-r, only for scalar problems)
- tsfcoll (tsfcoll-n-r, only for scalar problems)
- fhbvm
- fhbvm2

when solving some problems taken from the FDE-Testset.

https://people.dimai.unifi.it/brugnano/FDEtestset/

fractal and fractional

an Open Access Journal by MDPI

FDE-Testset: Comparing Matlab© Codes for Solving Fractional Differential Equations of Caputo Type

Luigi Brugnano; Gianmarco Gurioli; Felice Iavernaro; Mikk Vikerpuur

Fractal Fract. **2025**, Volume 9, Issue 5, 312

https://doi.org/10.3390/fractalfract9050312

FDE-Testset

It compares the previous codes, in terms of Work-Precision Diagrams (WPD) on a set of 10 problems:

- linear or nonlinear ones
- stiff or nonstiff ones
- scalar or vector ones
- with a stable or an oscillatory solution

in order to have a wide range of dynamic behavior.

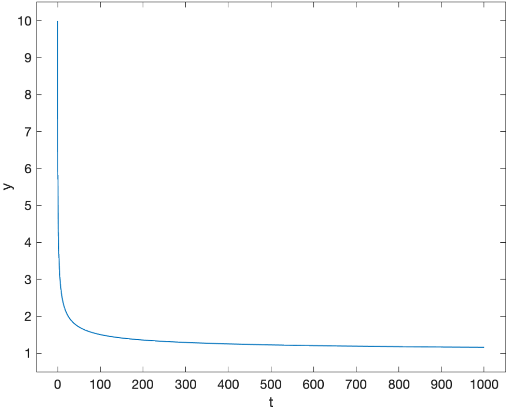
Accuracy is measured in terms of mescd,

$$-\log_{10} \max_{n} \|(y_n - y(t_n))./(1 + |y(t_n)|)\|_\infty$$

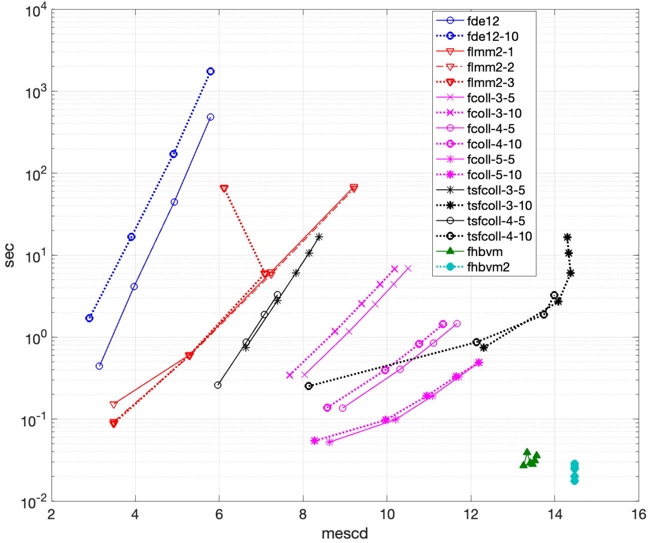
according to the TestSet for IVP-Solvers.

Problem 1

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



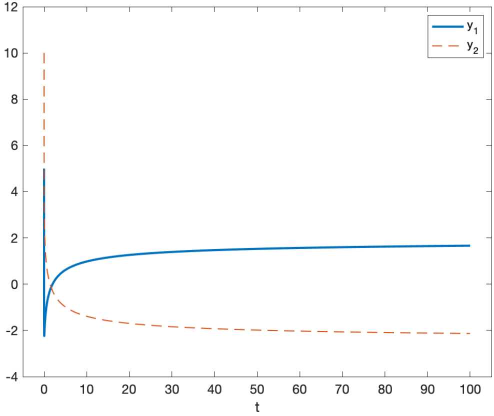
Probem 1 WPD



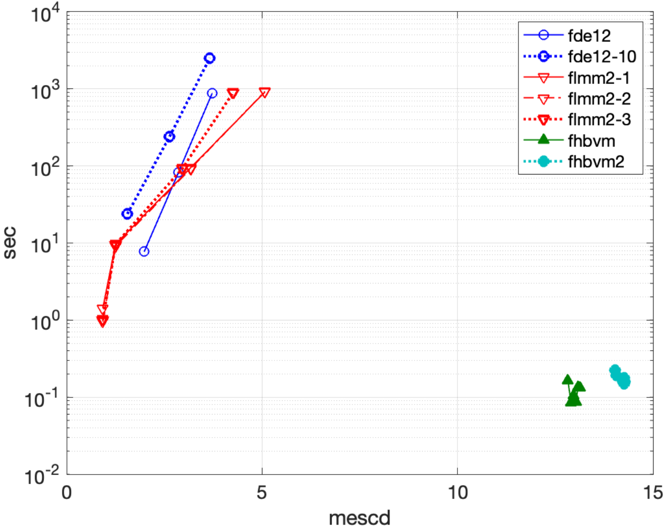
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
sec
2
4
6
8
10
12
14
16
mescd
fde12
fde12-10
flmm2-1
flmm2-2
flmm2-3
fcoll-3-5
fcoll-3-10
fcoll-4-5
fcoll-4-10
fcoll-5-5
fcoll-5-10
tsfcoll-3-5
tsfcoll-3-10
tsfcoll-4-5
tsfcoll-4-10
fhbvm
fhbvm2

Problem 2

$$y^{(0.5)} = \frac{1}{5}\begin{pmatrix} -92 & -87 \\ -58 & -63 \end{pmatrix} y - \frac{1}{10}\begin{pmatrix} 67 \\ 83 \end{pmatrix}, \quad t \in [0, 10^2], \quad y(0) = \begin{pmatrix} 5 \\ 10 \end{pmatrix}.$$



Probem 2 WPD

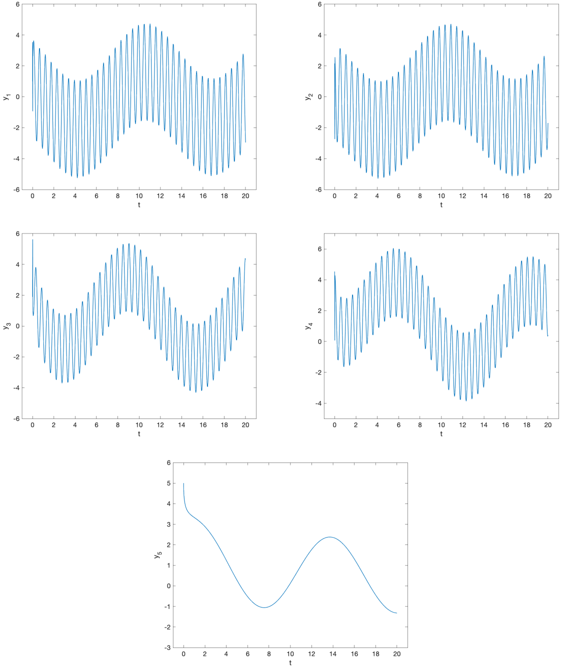


10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
sec
0
5
10
15
mescd
fde12
fde12-10
flmm2-1
flmm2-2
flmm2-3
fhbvm
fhbvm2

Problem 3

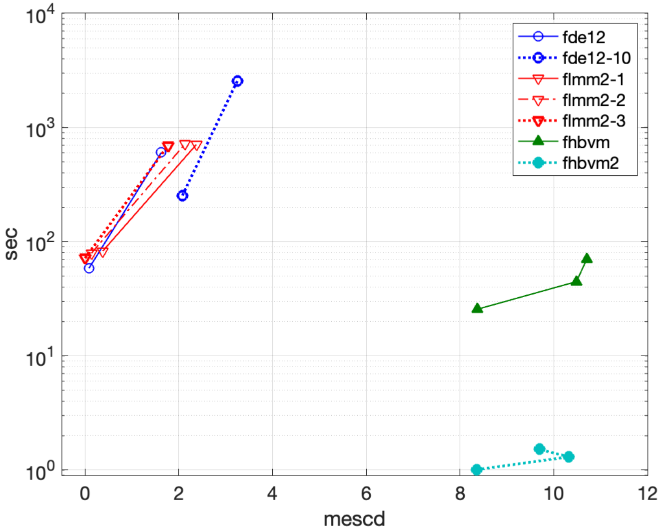
$$y^{(0.5)} = \frac{1}{8}\begin{pmatrix} 41 & 41 & -38 & 40 & -2 \\ -79 & 81 & 2 & 0 & -2 \\ 20 & -60 & 20 & -20 & -8 \\ -22 & 58 & -24 & 20 & -4 \\ 1 & 1 & -2 & -4 & -2 \end{pmatrix} y, \qquad t \in [0, 20],$$

$$y(0) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}.$$



y_1
y_2
y_3
y_4
y_5
t

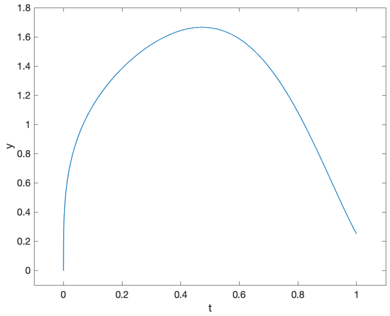
Problem 3 WPD

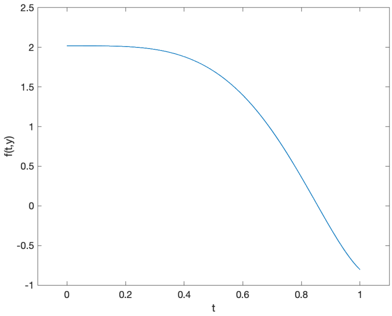


Problem 4

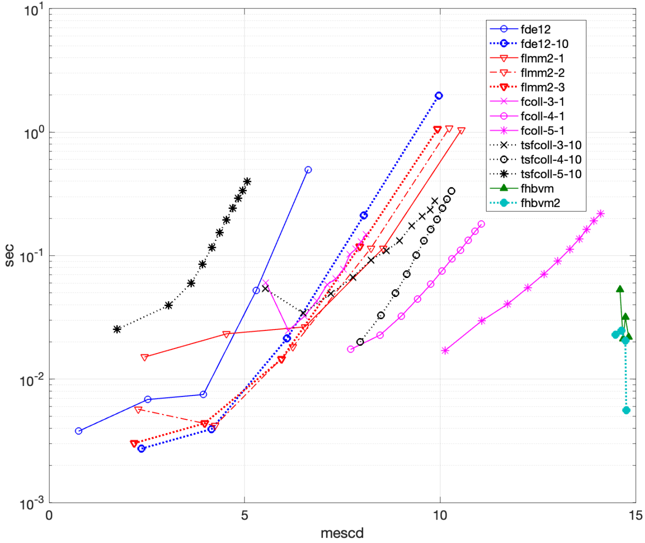
$$y^{(\alpha)} = -y^{3/2} + \frac{8!}{\Gamma(9-\alpha)} t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)} t^{4-\alpha/2} + \left(\frac{3}{2} t^{\alpha/2} - t^4\right)^3 + \frac{9}{4}\Gamma(\alpha+1), \qquad t \in [0,1], \qquad y(0) = 0.$$

We choose $\alpha = 0.3$.



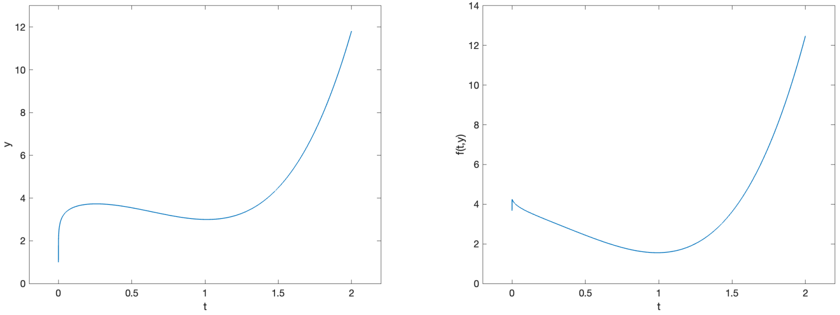


Problem 4 WPD

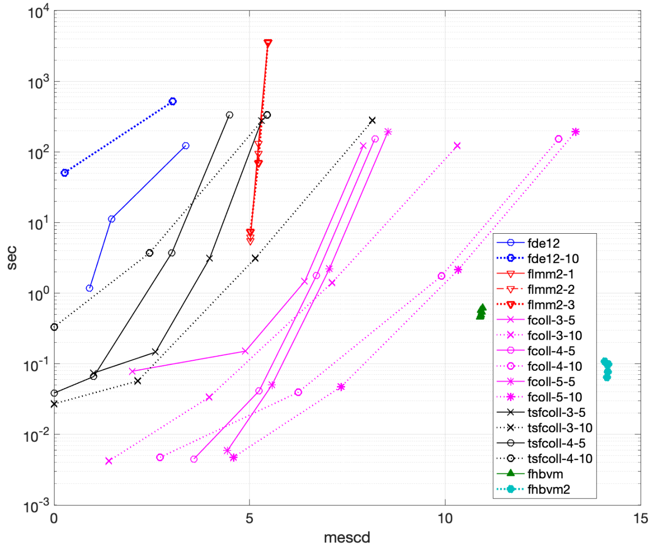


Problem 5

$$y^{(0.2)} = \left((1-t^2)^2 + (4+2t^{0.1}-3t^{0.3})t^{0.2}\right)^2 - y^2 + \frac{24}{\Gamma(4.8)}t^{3.8} - \frac{4}{\Gamma(2.8)}t^{1.8}$$
$$-3\frac{\Gamma(1.5)}{\Gamma(1.3)}t^{0.3} + 2\frac{\Gamma(1.3)}{\Gamma(1.1)}t^{0.1} + 4\Gamma(1.2), \qquad t \in [0,2], \qquad y(0) = 1.$$



Problem 5 WPD

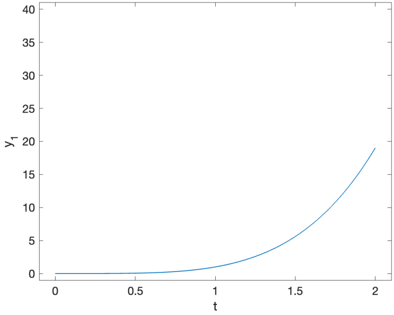


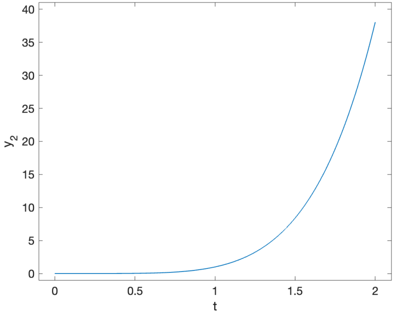
Problem 6

$$y_1^{(\alpha)} = \frac{\Gamma(4+\alpha)}{6} t^3 - t^{8+2\alpha} + y_2^2, \qquad y_2^{(\alpha)} = \frac{\Gamma(5+\alpha)}{24} t^4 + t^{3+\alpha} - y_1,$$

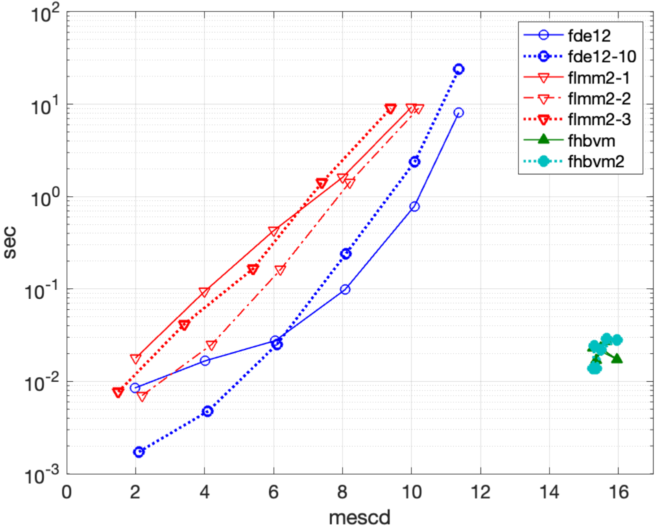
$$t \in [0,2], \qquad y_1(0) = y_1'(0) = y_2(0) = y_2'(0) = 0.$$

We choose $\alpha = 1.25$.





Problem 6 WPD

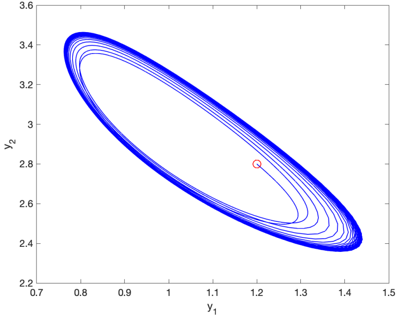


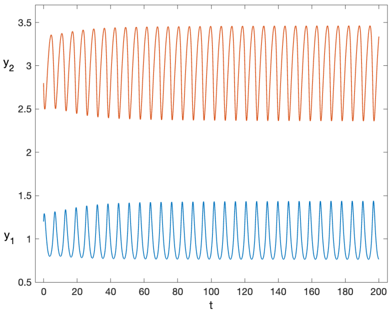
Problem 7

Fractional Brusselator:

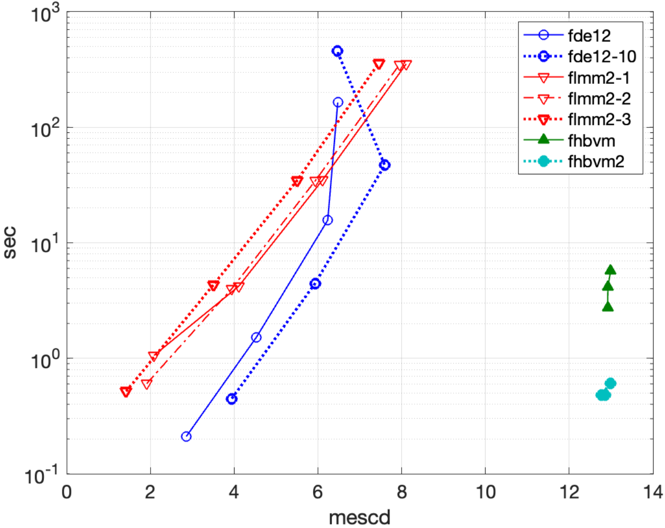
$$y_1^{(0.7)} = 1 - 4y_1 + y_1^2 y_2, \qquad y_2^{(0.7)} = 3y_1 - y_1^2 y_2,$$

$$t \in [0, 200], \qquad y(0) = (1.2,\ 2.8)^\top.$$





Problem 7 WPD

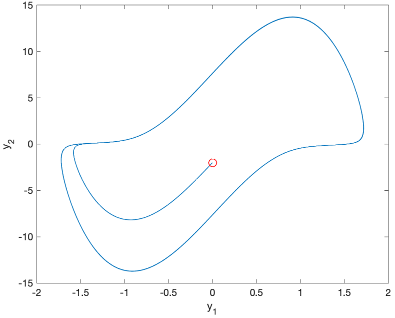


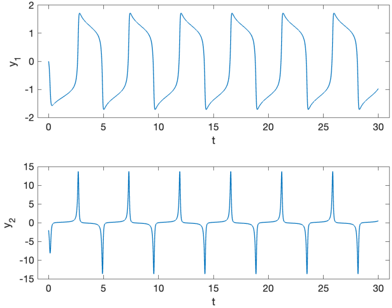
Problem 8

Fractional Van der Pol:

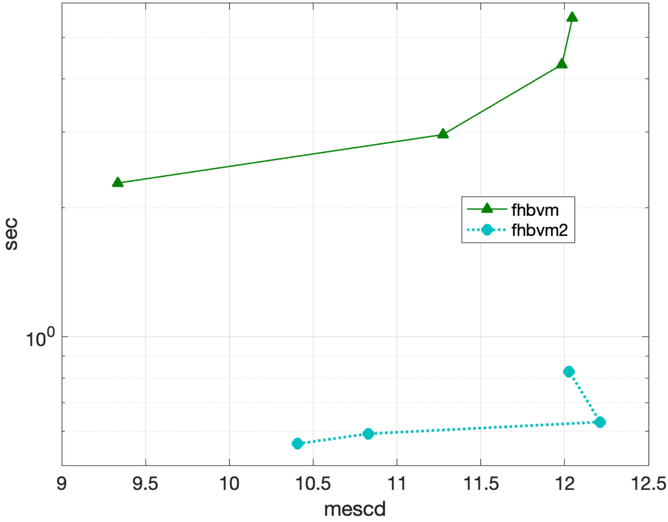
$$y_1^{(0.9)} = y_2, \qquad y_2^{(0.9)} = -y_1 - 10y_2(y_1^2 - 1),$$

$$t \in [0, 30], \qquad y(0) = (0, -2)^\top.$$





Problem 8 WPD



Conclusions

- Effective numerical methods can be devised for solving ODE-IVPs by expanding the vector field along the Legendre polynomial basis
- In this way, the class of HBVMs methods is derived
- Such methods are particularly effective for solving Hamiltonian problems, and can gain spectrally accurate solutions
- The approach can be extended to FDE-IVPs by expanding the vector field along a suitable Jacobi polynomial basis
- In this way, the class of FHBVMs methods is derived, able to gain spectrally accurate solutions as well
- Their effectiveness is confirmed by a number of examples taken from the FDE-Testset

THANK YOU!

-