Numerical Methods in Holonomic Gradient Method (HGM)

Nobuki Takayama (Kobe University)

• TYZ[10] N.Takayama, T.Yaguchi, Y.Zhang, Comparison of Numerical Solvers for Differential Equations for Holonomic Gradient Method in Statistics,
`https://arxiv.org/abs/2111.10947`
• OpenXM-hgm[7]
`http://www.math.kobe-u.ac.jp/OpenXM/Math/hgm/ref-hgm.html`
• chebfun[2] `https://chebfun.org`
• `http://www.math.kobe-u.ac.jp/OpenXM/Math/defusing/ref.html`
Sample codes.

Numerical solver for ODE's

Numerical evaluation of
integrals at a few points

Holonomic systems
(ODE's) for them

Definite integrals in
physics and statistics

What is a difficulty in numerical solver in HGM?

The ODE may contain solutions $f(t)$ such that

$$f(t) \gg Z(t)(\text{normalizing constant, ...})$$

Example 1

$$\frac{dY}{dt} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & -\lambda_2 \end{pmatrix} Y$$

$\lambda_1 > 0 > -\lambda_2$. We assume $Z(t) = Y_1(t) + Y_2(t) \sim \exp(-\lambda_2 t)$.
A small numerical error ε in the initial condition

$$Y(0) = (\varepsilon, 1)^T$$

gives the solution $Y(t) = (\varepsilon \exp(\lambda_1 t), \exp(-\lambda_2 t))^T$ and then

$$Y_1(t) + Y_2(t) = \varepsilon \exp(\lambda_1 t) + \exp(-\lambda_2 t)$$

Example 2

(Airy function, running example 1)

$$\frac{d^2y}{dt^2} - ty = 0 \tag{1}$$

$$\begin{aligned} \mathrm{Ai}(t) &\sim \frac{1}{2\sqrt{\pi}t^{1/4}} \exp\left(-\frac{2}{3}t^{3/2}\right) O(1) \\ \mathrm{Bi}(t) &\sim \frac{1}{\sqrt{\pi}t^{1/4}} \exp\left(\frac{2}{3}t^{3/2}\right) O(1) \end{aligned}$$

`https://en.wikipedia.org/wiki/Airy_function`

The initial value problem to obtain $\mathrm{Ai}(t)$ will have the difficulty.

Example 3

($H_n^k(x,y)$, running example 2) Let n and k be positive integers.
(OpenXM/Math/defusing/Hkn/19-a19-n-pf.rr)

$$\begin{aligned} H_n^k(x,y) &= \int_0^x t^k \exp(-t)\,{}_0F_1(;n;yt)dt \qquad (2) \\ &= \frac{\Gamma(n)}{\sqrt{\pi}\Gamma(n-1/2)} \int_{D(x)} t^k (1-s^2)^{n-3/2} \exp(-t-2s\sqrt{yt})dtds \\ & \quad \text{where } D(x) = \{(t,s) \in [0,x] \times [-1,1]\} \qquad (3) \end{aligned}$$

This function appears in studies of the outage probability of MIMO WiFi systems KA[6]. The function $H_n^k(x,y)$ is annihilated by the following ordinary differential operator w.r.t y.

$$\begin{aligned} & y^2\partial_y^4 + (-y+2n+2)y\partial_y^3 \\ & \quad +(-yx+(-k-n-3)y+n(n+1))\partial_y^2 \\ & \qquad +((y-n)x-n(k+2))\partial_y + (k+1)x \qquad (4) \end{aligned}$$

Initial value problem.

1. Runge-Kutta methods work in a short range. Implicit Runge-Kutta methods work in a longer range, but are not enough.
2. Geometric integrators like simpletic methods cannot be applied in most cases.

Boundary value problem.

1. A naive approaches do not work well.

Sparse interpolation/extrapolation methods

A few evaluations of the integral (value data) and ODE L → values on a longer interval

1. Chebyshef function method Trefethen[12], chebfun[2].
2. Minimizing $\int_D |Lf|^2 d\mu(t)$ with constraints by value data[1]

[1] It is standard in functional and numerical analysis, but it is useful in HGM. Then, we name it the sparse interpolation/extrapolation method.

Chebyshef function method, chebfun[2]

The chebfun project was initiated in 2002 by Lloyd N. Trefethen and his student Zachary Battles.
`https://en.wikipedia.org/wiki/Chebfun`.
The n-th Chebyshef function (polynomial) is

$$T_n(x) = \cos(n\theta), \quad x = \cos\theta \tag{5}$$

The extreme points of the curve $y = T_n(x)$ in $[-1, 1]$, which we mean points that take the value $y = 1$ or $y = -1$, are called Chebyshef points (of the second kind) of T_n. For example,
$T_2(x) = 2x^2 - 1$, the Chebyshef points are $\{-1, 0, 1\}$.
$T_3(x) = 4x^3 - 3x$, $\{-1, -0.5, 0.5, 1\}$.

Chebyshef interpolant

Let $f(x)$ be a function. Fix the set of Chebyshef points for $T_n(x)$. Let the value of f at Chebyshef point x_j be f_j. The Chebyshev interpolant is

$$p(x) = {\sum_{j=0}^{n}}' \frac{(-1)^j f_j}{x - x_j} \Big/ {\sum_{j=0}^{n}}' \frac{(-1)^j}{x - x_j} \tag{6}$$

The primes on the summation signes signify that the terms $j = 0$ and $j = n$ are multiplied by $1/2$.

$p(x_j) = f_j$. Degree n polynomial.

Convergence rate

$a_k = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}dx$. $f = \sum_{k=0}^{\infty} a_k T_k(x)$ when f is Lipschitz continuous.

Theorem 4

(Bernstein 1911, 1912. See, e.g., Th 8.2, Th 8.3 in Trefethen[12])
If f is analytic on $[-1,1]$, its Chebyshef coefficients a_k decrease geometrically. If f is analytic and $|f| \le M$ in the Bernstein ρ-ellipse [2] about $[-1,1]$, then $|a_k| < 2M\rho^{-k}$. The degreee n Chebyshev interpolant has accuracy $O(M\rho^{-n})$ by the sup norm.

[2] The radius ρ circle in the z-plane. Map it by $x = (z + z^{-1})/2$ and then we obtain the Bernstein ρ-ellipse:
`https://www.chebfun.org/examples/approx/Entire.html`

chebmat $M(n-m,n;s)$ chebfun[2]

Let X be the set of the n Chebyshef points (of the second kind) for the Chebyshef function T_{n-1}.
$\ell_j(X;t)$: the j-th polynomial of the Lagrange interpolation for X.
Let Y be the set of the $(n-m)$ Chebyshef points where $m \geq 0$ [3].

Definition 5

chebfun[2] $M(n-m,n;s)$: $(n-m)\times n$ matrix with (i,j) entries

$$\sum_{k=0}^{n-m-1} \ell_k(Y;Y_i)\ell_j^{(s)}(X;Y_k) \tag{7}$$

When $f(t)$ is the Chebyshef interpolant w.r.t. X,

$$f^{(s)}(Y_i) = (i\text{-th row of } M(n-m,n;s))\cdot(f_0,\ldots,f_{n-1})^T$$

[3]We approximate $f(t)$ by the values at Y, which is called "down-sampling" in DH2016[3].

From ODE to a (dense) matrix equation

Example 6

The Airy equation

$$f'' - tf = 0$$

Symbolically, we solve

$$\left(M(n-2,n;2) - \mathrm{diag}(Y)M(n-2,n;0)\right)F = 0 \qquad (8)$$

where $F = [f_0, f_1, \ldots, f_{n-1}]^T$ with giving, e.g., values of f_0 and f_{n-1} (boundary values) or values of f_0 and the first entry of $M(n-2,n;1)(f_0,\ldots,f_n)^T$ (initial values f and f').
See `https://www.chebfun.org/examples/ode-linear/SpectralDisc.html`.

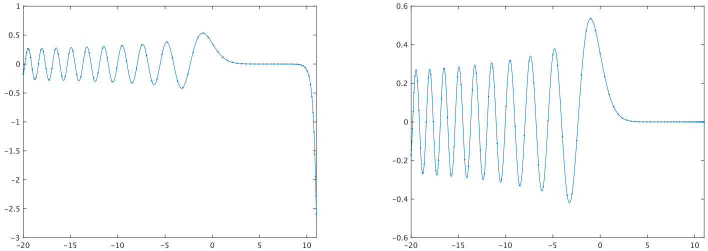


Figure: Solving the Airy differential equation by chebfun

Initial value problem for Airy $\mathrm{Ai}(t)$. (OpenXM/Math/defusing/intro/y2023_07_16_airy_initial_value.m)
$\mathrm{Ai}(-20) = -0.17640612707798468959019229221 9$,. $\mathrm{Ai}'(-20) = 0.89286285673647123839840993411 4$
Chebfun gives reasonable values[4] upto $t = 9$, but divergent values appear when t is larger than 9. See the left graph of Figure 1.

Boundary value problem for Airy $\mathrm{Ai}(t)$. (OpenXM/Math/defusing/intro/y2023_07_16_airy_boundary_value.m)

$\mathrm{Ai}(-20) = -0.176406127077984689590192292219$, $\mathrm{Ai}(11) = 4.22627586496035959129883545080 \times 10^{-12}$.

Divergent values do not appear. See the right graph of Figure 1.

[4] Values are compared with Mathematica.

$H_n^k(x, y)$

Example 7

Boundary value problem for $H_n^k(x, y)$ for $x = 1$ and $y \in [10^8, 10^8 + 2 \times 10^5]$.

We give the boundary values of $H_1^{10}(1, y)$ and $\frac{\partial H_1^{10}}{\partial y}(1, y)$ at $y = 10^8$ and $y = 10^8 + 2 \times 10^5$. We apply the chebfun package for this boundary value problem.

(OpenXM/Math/defusing/Hkn/y2023_07_25_hkn_valid10power8.m)

To check the accuracy, we compare the values by the chebfun package and by the numerical integration by Mathematica at $y = 10^8 + 200$. The chebfun package keeps 4 digits accuracy at the point and the ODE is solved in 1.66s[5]. On the otherhand, the numerical integration by Mathematica (2022) (OpenXM/Math/defusing/Hkn/2023-07-09-hkn-int.m) took 23.58s[6].

[5] Apple M1, 2020, Matlab 2022b

[6] AMD EPYC 7552 48-Core Processor, 1499.534MHz

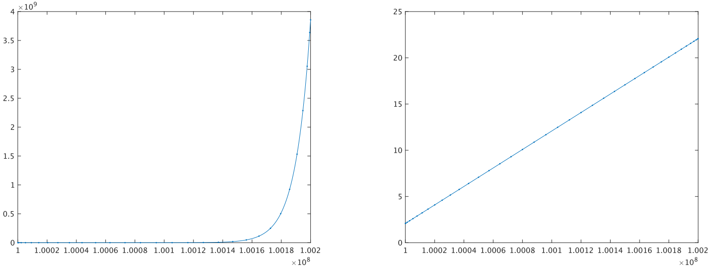


Figure: Left: $H_1^{10}(1, y)$. Right: $\log H_1^{10}(1, y)$. Values should be magnified by 10^{8678}.

Sparse interpolation/extrapolation TYZ[10]

Known: $Lf = 0$ (ODE), $f(p_i) = q_i$ for some points p_i's. $\{e_j\}$: a set of basis functions. Put $f(t) = \sum_{k=0}^{M} f_k e_k(t)$ (unknown contants f_j's). Minimize

$$\int_a^b |Lf(t)|^2 d\mu(t), \ f(p_i) = q_i, i = 1, 2, \ldots. \tag{9}$$

A numerical integration for a function g:

$$I_N(g) = \sum_{j=0}^{N} T_j g(t_j) \tag{10}$$

where $t_0 = a < t_1 < \cdots < t_{N-1} < t_N = b$ and $T_j \in \mathrm{R}_{\geq 0}$. Fix it. Then, the loss function is

$$\begin{aligned} \ell(\{f_k\}) &:= \sum_{j=0}^{N} |(Lf)(t_j)|^2 T_j \\ &= \sum_{j=0}^{N} \left| \sqrt{T_j} \sum_{k=0}^{M} f_k (Le_k)(t_j) \right|^2 \end{aligned} \tag{11}$$

We minimize it under $f(p_i) = q_i$ (least square for the data $(Le_k)(t_j)\sqrt{T_j}$).

Chebyshef function method as a sparse interpolation

The Chebyshef function method can be regarded as a special case of this method. The numerical integration scheme of the Chebyshef quadrature:

$$\int_{-1}^{1} \sqrt{1-t^2} g(t) dt \sim \sum_{i=1}^{k-1} T_i g(Y_i) \tag{12}$$

where Y is the set of the Chebyshef points for T_k and the weight T_i is

$$T_i = \frac{\pi}{k} \sin^2 \left(\frac{i}{k} \pi \right)$$

Put $g(t) = |Lf|^2$ and $d\mu(t) = \sqrt{1-t^2} dt$ in (9). Since the left hand side of (8) are values at the set of Chebyshef points Y, assuming that it is equal to the zero vector is equivalent to that the integral by the Chebyshef quadrature over Y is equal to zero.

Todo

A different solver with validation and Chebyshef functions is proposed in BBJ2018[1][7].

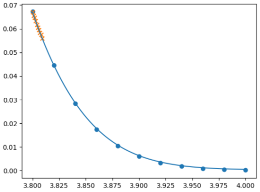
The advantage of the method is that matrices in the solver are banded and validation is given. We will test this method for the HGM as a next try.

[7] F.Bréhard, N.Brisebarre, M.Jolders, Validated and numerically efficient Chebyshev spectral methods for linear ordinary differential equations

Example 8

$E[\chi(M_x)]$, TJKZ2020[11] (Expectation of Euler characteristic of random manifolds).

Extrapolation of some values near $t = 4.8$ by the sparse interpolation/extrapolation method; The degree 29 polynomial and the rectangle integration is used for a rank 11 ODE (26KB).
`https://colab.research.google.com/drive/1XhysmF1DMZfAhTtl0tc9A7tFYRBeI6tI?usp=sharing` See Figure 3.



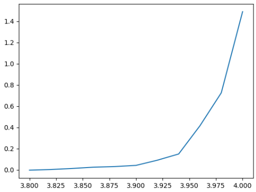


Figure: The graph of $F_{29}(t)$ and simulation values in the left and relative errors in the right. The data points are marked with 'x'.

Defusing method (filter method) for initial value problems

The initial value problem of the ODE

$$\frac{dF}{dt} = P(t)F \qquad (13)$$

$$F(t_0) = F_0^{\mathrm{true}} \in \mathrm{R}^r \qquad (14)$$

where $P(t)$ is an $r \times r$ matrix, $F(t)$ is a column vector function of size r, and F_0^{true} is the initial value of F at $t = t_0$.

Situation 1

1. The initial value has at most 3 digits of accuracy. We denote this initial value F_0.
2. The property $|F| \to 0$ when $t \to +\infty$ is known, e.g., from a background of the statistics.
3. There exists a solution $\tilde{F}$ of (13) such that $|\tilde{F}| \to +\infty$ or non-zero finite value when $t \to +\infty$.

Defusing method

Numerical schemes such as the Runge-Kutta method obtain a numerical solution by the recurrence

$$F_{k+1} = Q(k,h)F_k \tag{15}$$

from F_0 where $Q(k,h)$ is an $r \times r$ matrix determined by a numerical scheme and h is a small number. The vector F_k is an approximate value of $F(t)$ at $t = t_k = t_0 + hk$. Let N be a suitable natural number and put

$$Q = Q(N-1,h)Q(N-2,h)\cdots Q(1,h)Q(0,h) \tag{16}$$

We call Q the *matrix factorial* of $Q(k,h)$. The matrix Q approximates the fundamental solution matrix of the ODE.

Project F_0 to eigenspaces of non-positive eigenvalues.

Defusing method — algorithm

Algorithm 1

1. *Obtain eigenvalues* $\lambda_1 > \lambda_2 > \cdots > \lambda_r > 0$ *of* Q *and the corresponding eigenvectors* $v_1, \ldots, v_r$.
2. *Let* λ_m *be the first non-positive eigenvalue.*
3. *Express the initial value vector* F_0 *containing errors in terms of* v_i*'s as*
$$F_0 = f_1 v_1 + \cdots + f_r v_r, \quad f_i \in \mathrm{R} \tag{17}$$
4. *Choose a constant* c *such that* $F_0' := c(f_m v_m + \cdots + f_r v_r)$ *approximates* F_0.
5. *Determine* F_N *by* $F_N = QF_0'$ *with the new initial value vector* F_0'.

Example 9

Solving Airy differential equation by the defusing method. (OpenXM/Math/defusing/intro/2023-07-21-airy.rr) Give initial values at $t = -20$ as
F0=[-0.17640612707798468959,0.89286285673647123840]
(Ai$[-20]$ and Ai$'[-20]$).

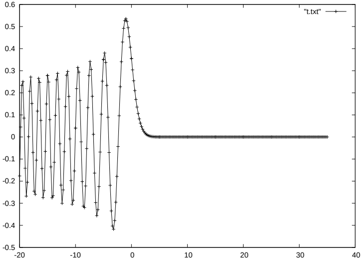


Figure: Solving initial value problem, $t \in [-20, 30]$

Example 10

We implement the defusing method in `tk_ode_by_mpfr.rr` [8] for the Risa/Asir [9]. It generates C codes utilizing the MPFR [8] for bigfloat and the GSL [4] for eigenvalues and eigenvectors. We apply the defusing method for initial value problem to $H_1^{10}(1, y)$ which is a solution of the ODE (4). We apply the defusing method for a transformed ODE with a gauge function $\exp(y) y^{1-n+k}$ to make the target solution decrease to 0 when $y \to \infty$. We use the step size $h = 10^{-3}$ and the bigfloat of 30 digits of accuracy.
(OpenXM/Math/defusing/asir-tmp/tk-ode-assert.rr (code generation), tk-ode-assert.hkn1(), tk-ode-assert.hkn2()) The Figure 5 shows that the adaptive Runge-Kutta method of GSL [4] fails before y becomes 30. The Figure 6 presents the relative error of values by the defusing method and exact values. It shows that the defusing method works even when $y = 10^3$.

[8] `http://www.math.kobe-u.ac.jp/OpenXM/Current/doc/asir-contrib/ja/tk_ode_by_mpfr-html/tk_ode_by_mpfr-ja.html`. Todo, English manual.

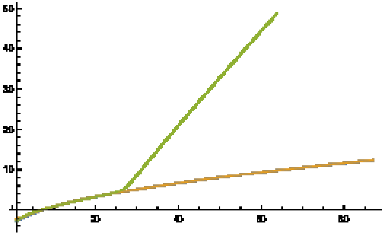


Figure: $\log H_1^{10}(1, y)$. Exact value (by numerical integration) and the value by our defusing method agree. The adaptive Runge-Kutta method with the initial relative error 10^{-20} (upper curve) does not agree with the exact value when y is larger than about 25.

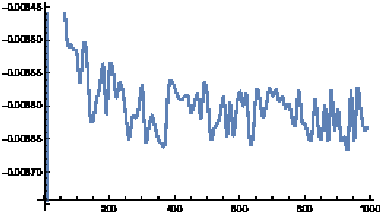


Figure: The relative error of $H_1^{10}(1, y)$ of our defusing method. The relative error is defined as $(H_d - H)/H$ where H_d is the value by the defusing method and H is the exact value.

Summary

1. The use of implicit Runge-Kutta method will be a good choice for solving ODE in a short range.
2. In order to solve unstable HGM initial value problem, the defusing method (filter method) will be a good choice.
3. In order to solve unstable HGM boundary value or sparse interpolation problem, the Chebyshef function method and other sparse interpolation method will be a good choice.
4. See also todo.

[1] F.Bréhard, N.Brisebarre, M.Jolders, Validated and numerically efficient Chebyshev spectral methods for linear ordinary differential equations, ACM Transasctions on Mathematical Software, 44 (2018), 1-42.
https://hal.science/hal-01526272v3

[2] https://chebfun.org

[3] T.A.Driscoll, N.Hale, Rectangular spectral collocation, IMA Journal of Numerical Analysis, 36 (2016), 108–132
https://doi.org/10.1093/imanum/dru062(2016)

[4] GSL, GNU scientific library,
https://www.gnu.org/software/gsl/

[5] E.Hailer, S.P.Norsett, G.Wanner, Solving Ordinary Differential Equations I, II. Springer, 1993.

[6] M. Kang, M. S. Alouini, Largest Eigenvalue of Complex Wishart Matrices and Performance Analysis of MIMO MRC Systems, IEEE Journal on Selected Areas in Communications 21 (2003), 418–426.

[7] References for HGM, `http://www.math.kobe-u.ac.jp/OpenXM/Math/hgm/ref-hgm.html`

[8] The GNU MPFR library, `https://www.mpfr.org/`

[9] Risa/Asir, a Computer algebra system, `http://www.openxm.org`

[10] N.Takayama, T.Yaguchi, Y.Zhang, Comparison of Numerical Solvers for Differential Equations for Holonomic Gradient Method in Statistics, `https://arxiv.org/abs/2111.10947`

[11] N.Takayama, L.Jiu, S.Kuriki, Y.Zhang, Computations of the Expected Euler Characteristic for the Largest Eigenvalue of a Real Wishart Matrix, Journal of Multivariate Analysis 179 (2020), 104642.
https://doi.org/10.1016/j.jmva.2020.104642

[12] L. N. Trefethen, Approximation Theory and Approximation Practice, 2020, SIAM.