Applied and numerical aspects for nonlocal initial value problems

Anders Bahrami Diego Rubio Dianlun (Jennifer) Luo Zhekai Liu

Polymath Jr. REU

Abstract

Nonlocal operators have recently been used successfully to approximate classical derivatives. A natural question is formulating a well-posed nonlocal counterpart to initial value problems. Constructing such a system which admits unique solutions is nontrivial and it is highly dependent on the choice of the interaction kernel. We will present some modeling aspects behind the selection of the kernel, as well as numerical studies showcasing the sensitivity of nonlocal solutions concerning initial input and the convergence of nonlocal solutions to their classical counterparts. This poster is a companion to the poster "Theoretical aspects for nonlocal initial value problems".

Problem Setup

Problem 1: How the nonlocal models can successfully approximate the local(classical) derivatives? From nonlocal perspective:

$$\mathcal{D}_{\mathcal{K}n}u(x_0) = \int_{-\infty}^{+\infty} [u(x_0+y) - u(x_0)]\, \mathcal{K}_n(y)\mathrm{d}y$$

What we do is find a proper kernel function $\mathcal{K}_n(x)$ such that $\mathcal{D}_{\mathcal{K}n}u(x_0) \to u'(x_0)$.

Problem 2: After finding a proper kernel function, we want to numerically study solutions to the *nonlocal differential equation* below and see if they approach the *classical solutions*.

$$\begin{cases} \mathcal{D}_{\mathcal{K}}u(x) = f(x, u(x)) \\ Initial\ Conditions \end{cases}$$

To do this, we define a *discrete* version of the nonlocal derivative:

$$\mathcal{D}_k u_i := \sum_{j=-m}^{m} k_j u_{i+j}$$

where $u_i = u(x_i)$ and $k(j) = \{k_{-m}, \dots k_m\}$ (x_i is the point in an evenly spaced partition/mesh of the real line).

Problem 1: The property of the kernel functions

We want a good function $\mathcal{K}$ where $\int_{-\infty}^{+\infty} \mathcal{K}(y)dy = 0$. Suppose we have the sequence $\{\mathcal{K}_i\}_{i\in\mathbb{N}}$ and $f(x)$ is a smooth function. Do a Taylor expansion of function f (at least 3 times differentiable):

$$u(x+y) = u(x) + u'(x)y + \frac{u''(x)}{2!}y^2 + \frac{u'''(\xi_x)}{3!}y^3$$

$$\begin{aligned}
\mathcal{D}_{\mathcal{K}_n}u(x) &= \int_{-\infty}^{+\infty} \left[u(x) + u'(x)y + \frac{u''(x)}{2!}y^2 + \frac{u'''(\xi_x)}{3!}y^3 - u(x)\right] \mathcal{K}_n(y)\mathrm{d}y \\
&= \int_{-\infty}^{+\infty} u'(x)y\mathcal{K}_n(y)\mathrm{d}y + \int_{-\infty}^{+\infty} \frac{u''(x)}{2!}y^2 k_n(y)\mathrm{d}y + \int_{-\infty}^{+\infty} \frac{u'''(\xi_n)}{3!}y^3\mathcal{K}_n(y)\mathrm{d}y \\
&= u'(x)\int_{-\infty}^{+\infty} y\mathcal{K}_n(y)\mathrm{d}y + \frac{u'''(\xi_n)}{3!}\int_{-\infty}^{+\infty} y^3\mathcal{K}_n(y)\mathrm{d}y \\
&\to u'(x) \quad (n \to +\infty)
\end{aligned}$$

The kernel functions can successfully approximate the derivative(The last equality holds) if and only if they satisfy these properties:

- $\lim_{n\to\infty} \int_{-\infty}^{+\infty} y\mathcal{K}_n(y)\mathrm{d}y = 1$
- $\lim_{n\to\infty} \int_{-\infty}^{+\infty} y^3\mathcal{K}_n(y)\mathrm{d}y = 0$
- $\int_{0}^{+\infty} y^2\mathcal{K}_n(y)\mathrm{d}y < +\infty$

Example of kernel functions

We can compare the kernel function $\mathcal{K}(y) = \frac{1}{y^{0.5}}\chi_{(0,\alpha]} - \frac{1}{|y|^{0.5}}\chi_{[-\alpha,0)}$ (where $\alpha = (0.75)^{\frac{2}{3}}$with Step kernel function $\mathcal{K}_n(y) = \frac{1}{h^2}\chi_{[0,h]} - \frac{1}{h^2}\chi_{[-h,0]}$ (where $n = 2,\ h = \frac{1}{n}$) with respect to $u(x) = sin(x)$, and the exact derivative is $cos(x)$:

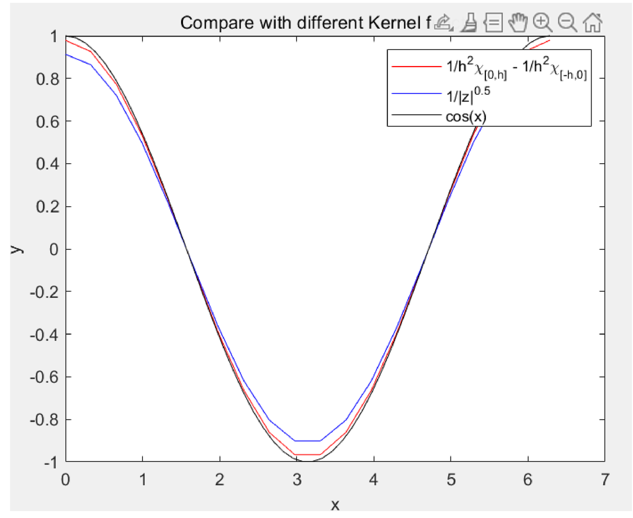


Figure 1. Approximation with different kernels

Remark: Nowadays, in the Natural Language Processing (NLP) field, there are lots of derivative-free situations (DALL · E, GPT-4, GPT-4o, etc.), which means that we cannot get the true gradient from the APIs except for the object function's value. So how to optimize the loss function in the black box tuning situation becomes more intractable. We want to use the nonlocal derivative to estimate the true derivative (1-dimensional) which gives us a new perspective on how to take cognitive derivatives. Zeroth order method in black-box optimization is really useful and the main idea is using a **Gaussian kernel** to approximate gradient.

Problem 2: Solve the differential equations

Combining $\mathcal{D}_{\mathcal{K}}u(x) = f(x,u)$ with $\mathcal{D}_{\mathcal{K}}u_i = \sum_{j=-m}^{m} k_j u_{i+j}$, we get a system of N equations:

$$\begin{cases}
\mathcal{D}_{\mathcal{K}}u_0 = f(u_0, x_0) & = k_{-m,0}u_{-m} \cdots + k_{0,0}u_0 \cdots + k_{m,0}u_m \\
\mathcal{D}_{\mathcal{K}}u_1 = f(u_1, x_1) & = k_{-m,1}u_{-m+1} \cdots + k_{0,1}u_1 \cdots + k_{m,1}u_{m+1} \\
 & \vdots \\
\mathcal{D}_{\mathcal{K}}u_{N-1} = f(u_{N-1}, x_{N-1}) & = k_{-m,N-1}u_{-m+N} \cdots + k_{0,N-1}u_N \cdots + k_{m,N-1}u_{m+N-1}
\end{cases}$$

Solution to above system "approximates" the classical ODE solution, and we need $u_{i_1} \dots u_{i_{2m}}$ as initial conditions for solvability. (For example, the initial bounded conditions: $u_{-m}, \dots, u_{-1}, u_N, \dots, u_{N-1+m}$)

Question: What kernel/coefficients $\{\{k_{i,j}\}_{i=-m}^{m+N-1}\}_{j=0}^{N-1}$ should we use?

Nonlocal method

After tuning lots of times, we would like to use this proper kernel:

$$k(x,y) = \frac{1}{\delta^2}\chi_{[0,\delta]} - \frac{1}{\delta^2}\chi_{[-\delta,0]} \approx \{\{k_{i,j}\}_{i=-m}^{m+N-1}\}_{j=0}^{N-1}$$

Where: $\alpha = 0.4$, fixed $\delta = (0.75)^{\frac{2}{3}}$, and $m = 2$. The results:

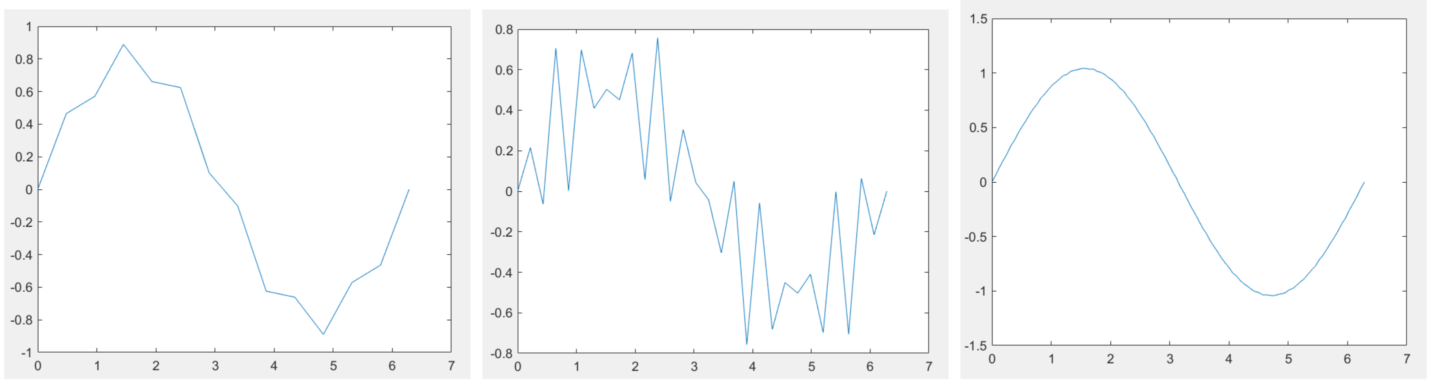


Figure 2. $N = 14, 30, 100$ from left to right

Finite Difference Method

However, you may find that the nonlocal method may not be stable, we find it is equivalent to finding the proper kernel and coefficients, so why not find the coefficients directly? We have proposed new method:

Using Taylor Expansion:

$$u(x+jh) = u(x) + jhu'(x) + \frac{(jh)^2}{2!}u''(x) + \frac{(jh)^3}{3!}u'''(x) + \cdots$$

Substituting this into the nonlocal derivative expression:

$$D_k[u](x) = \frac{1}{h}\left(\sum_{j=-m}^{m} a_j u(x) + \sum_{j=-m}^{m} ja_jhu'(x) + \cdots\right)$$

- $\sum_{j=-m}^{m} a_j = 0$
- $\sum_{j=-m}^{m} ja_j = 1$
- $\sum_{j=-m}^{m} j^2a_j = 0 \cdots \sum_{j=-m}^{m} j^{2m}a_j = 0$

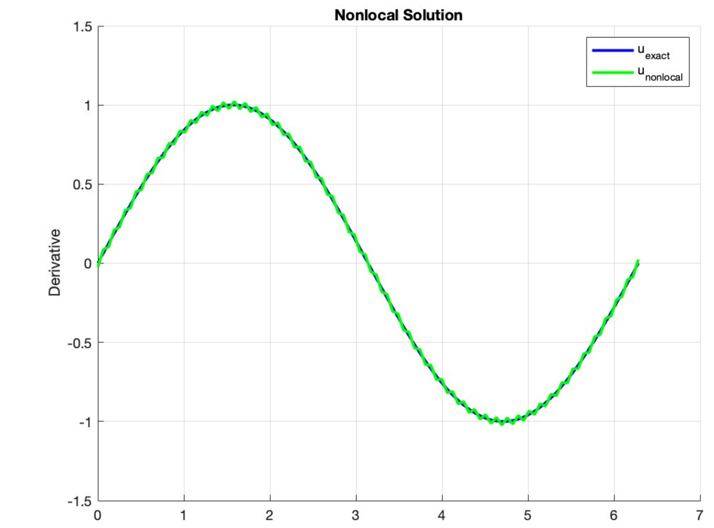
System of Equations:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ -m & -(m-1) & \cdots & m \\ m^2 & (m-1)^2 & \cdots & m^2 \\ \vdots & \vdots & \ddots & \vdots \\ m^{2m} & (m-1)^{2m} & \cdots & m^{2m} \end{pmatrix} \begin{pmatrix} a_{-m} \\ a_{-(m-1)} \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Solvability: The matrix is a square Vandermonde matrix and is invertible if all entries are distinct.

Accuracy: The method's theoretical order of accuracy is $O(h^{2m})$.

Finite Difference Method's experiment results and $L2$ error



Nonlocal Solution for $N = 100, m = 10$

Spatial Discretization $N = 5000$	
m	L2 Error
1	4.67e-07
2	1.53e-13
3	2.58e-03
5	1.50e-03
6	1.28e-03
7	1.25e-03
9	1.13e-03
10	1.09e-03
11	1.06e-03
13	9.70e-02
>=14	Results no longer make sense

Nonlocal Models vs Finite Difference Methods

1. Weights:
   - Finite Difference Methods: Use fixed weights derived from Taylor series expansions around a point.
   - Nonlocal Models: Allow flexibility by using different weights, often determined by specific kernels.
2. Horizon and Symmetry:
   - Finite Difference Methods: The set of points for approximation is usually fixed and symmetric around the point of interest.
   - Nonlocal Models: Can choose different horizons for the kernel, not necessarily symmetric.
3. Coefficients:
   - Finite Difference Methods: Coefficients are generally computed to be optimal for a broad class of functions, typically all smooth functions.
   - Nonlocal Models: Can compute optimal coefficients for specific classes of functions, such as odd functions or polynomials.