Efficient Numerical Method for Models Driven by Lévy Process via Hierarchical Matrices

Abstract

The modeling via the fractional partial differential equation or the Lévy process has been an active area of research and has many applications. However, lack of efficient numerical computation method impede people from adopting such a modeling tool. We proposed an efficient solver for convection diffusion equation whose operator is the infinitesimal generator of a Lévy process based on $\mathcal{H}$-matrix technique. The proposed Crank Nicolson scheme is unconditionally stable and has theoretical $\mathcal{O}(h^2 + \Delta t^2)$ convergence rate. The $\mathcal{H}$-matrix technique has theoretical $\mathcal{O}(N)$ space and computational complexity compared to $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$ respectively for the direct method. Numerical experiments demonstrate the efficiency of the new algorithm.

1   Introduction

Over the last years anomalous diffusion or nonlocal modeling have seen a tremendous increase in popularities in many fields. Of particular interest is the fractional partial differential equations (FPDE) arising from many disciplines such as ... One extensively studied fractional operator is the fractional Laplacian

$$(-\Delta)^s u(\mathbf{x}) = \text{r.v.} \int_{\mathbb{R}^d} c_{d,s} \frac{u(\mathbf{x} + \mathbf{y}) - u(\mathbf{x})}{|\mathbf{y}|^{d+2s}} d\mathbf{x} \tag{1}$$

which is considered as the natural generalization of the Laplacian operator to the normal Laplacian operator. However, the numerical computation of the FPDE with such operators exhibits particular difficulties: (1) the kernel function $\frac{c_{d,s}}{|\mathbf{y}|^{d+2s}}$ has singularities which must be dealt with special care; (2) the kernel function is nonlocal, and therefore the corresponding coefficient matrix is typically dense. The second difficulty impedes people from using the new modeling tool due to its prohibitable computational requirement. There are some efforts to speed up the computation, mainly through analyzing its special structure or modifying the definition.

Taking another point of view, the fractional partial differential equation with the Laplacian operator (and many others) can be derived from the infinitesimal generator of the Lévy process. Indeed, the forward equation (or Fokker Planck equation in physics) will have the form

$$u_t = au_{xx} + bu_x + cu + \mathcal{L}u \quad x \in \mathbb{R}, t \in (0, 1) \tag{2}$$

where $a \geq 0$, $c \leq 0$, $b \in \mathbb{R}$, and

$$\mathcal{L}u = \int_{\mathbb{R}} (u(x+y) - u(x) - u'(x)\mathbf{1}_{0<|y|<1}(y)y)\nu(y)dy \qquad (3)$$

Here $\nu(y)$ will be a proper Lévy measure. The fractional Laplacian is just a special case where (d=1) $\nu(y) = \frac{c_{1,s}}{|y|^{1+2s}}$. The model eq. (2) incorporates a much richer structure and has a broader of applications. For example, in recent years, the modeling of financial markets by Lévy process has become an active area of research. The numerical difficulties are similar to that of FPDE.

In this paper, we aim at solving eq. (2) efficiently based on the well-established $\mathcal{H}$-matrix technique. In principal, our algorithm can work for any $\nu(y)$ under mild assumptions, including singular or slow decaying Lévy measure. We focus on the efficiency of the operator since the ability to efficient store data and solve is the main bottleneck for today's applications. In particular, the algorithm will equivalently work for many FPDE models, on condition it can be written in the form of eq. (2).

The advantage of adopting the $\mathcal{H}$-matrix is its high efficiency. If direct method is used, which results in a dense coefficient matrix, the storage complexity will be $\mathcal{O}(N^2)$ while the computation complexity will be $\mathcal{O}(N^3)$ (LU factorization). However, theoretically, $\mathcal{H}$-matrix can achieve nearly optimal $\mathcal{O}(N)$ storage and computation complexity. Although in practice, the optimal rate is hard to achieve (due to the increasing rank during LU factorization), we will demonstrate with careful implementation, the new algorithm has great advantage over the direct method for medium and large scale problems.

We mention that there are other approaches to solve FPDE. One of the main numerical methods is the Monte Carlo methods, which is based on the probabilistic interpretation of the model. Let X_t be a Lévy process with the Lévy measure $\nu(y)$ and appropriate diffusion and drift coefficients, under certain assumptions, the solution to eq. (2) can be written as

$$u(x) = \mathbb{E}(u(X_t)|X_0 = x) \qquad (4)$$

and Monte Carlo method can be applied thereafter. Although Monte Carlo might be the only way possible to numerically compute the solution in high dimensions, it suffers from slow convergence and therefore impractical for some cases. The grid-based method, such as the one we proposed in the paper, will enjoy fast convergence (and we will prove that the convergence order is $\mathcal{O}(\Delta t^2 + h^2)$).

To end this section, we summarize our major contributions of the paper

- Proposed and analyzed an unconditional stable Crank Nicolson scheme for the model problem eq. (2). The theoretical error is $\mathcal{O}(\Delta t^2 + h^2)$.

- Proposed and implemented an efficient solver for eq. (2) based on $\mathcal{H}$-matrix technique. The theoretical space and computation complexity is $\mathcal{O}(N)$.

- Proposed a trick for computation involving Lévy measure that has singular and slow decaying Lévy measure.

2 Lévy Process

Consider a given probability space $(\Omega, \mathcal{F}, P)$. A Lévy process $\{X_t\}_{t \geq 0}$ taking values in $\mathbb{R}^d$ is defined as a stochastic process with stationary and independent increments. In addition, we assume $X_0 = 0$ with probability 1.

By independent, we mean for any distinct time $0 \leq t_1 < t_2 < \ldots < t_n$, we have $X_{t_1}, X_{t_2-t_1}, \ldots, X_{t_n} - X_{t_{n-1}}$ are all independent.

By stationary, for any $0 \leq s < t < \infty$, the probability distribution of $X_t - X_s$ is the same as X_{t-s}.

One remarkable property of the Lévy process is that any Lévy process has a specific form of the characteristic function, called *Lévy-Khintchine formula*

$$\mathbb{E}(e^{\mathrm{i}(\xi, X_t)}) = e^{t\eta(\xi)} \tag{5}$$

where

$$\eta(\xi) = \mathrm{i}(b, \xi) - \frac{1}{2}(\xi, a\xi) + \int_{\mathbb{R}^d \setminus \{0\}} \left[e^{\mathrm{i}(\xi, y)} - 1 - \mathrm{i}(\xi, y)\mathbf{1}[0 < |y| < 1](y) \right] dy \tag{6}$$

here $b \in \mathbb{R}^d$, a is a positive definite symmetric matrix in $\mathbb{R}^{d \times d}$, and ν is a Lévy measure which satisfies

$$\int_{\mathbb{R}^d \setminus \{0\}} \min\{1, |y|^2\} \nu(dy) < \infty \tag{7}$$

In the case $\nu \equiv 0$, we obtain the Gaussian process. In the case $\int_{\mathbb{R}^d} \left[e^{\mathrm{i}(\xi, y)} - 1 \right] dy$ is well defined, we can omit the term $\mathrm{i}(\xi, y)\mathbf{1}[0 < |y| < 1](y)$.

In the case $\nu < \infty$, the Lévy process has the decomposition

$$X_t = bt + \sqrt{a}B_t + \sum_{0 \leq s \leq t} J_s \tag{8}$$

where J_s is the jump at time s. To be precise, define

$$N(t, A) = \#\{0 \leq s \leq t : J_s \in A\} \tag{9}$$

if t and A is fixed, $N(t, A)$ is a random variable; if t and $w \in \Omega$ is fixed, $N(t, \cdot)(w)$ is a measure; if A is fixed, $N(\cdot, A)$ is a Poisson process with intensity $\nu(A)$. Therefore, we can also write

$$\sum_{0 \leq s \leq t} J_s = \int_{\mathbb{R}^d - \{0\}} x N(t, dx) \tag{10}$$

To end this section, we provide a third view of the Lévy process. Consider the semigroup

$$(T_t f)(x) = \mathbb{E}(f(X_t + x)) \tag{11}$$

Then the infinitesimal generator will have the form

$$(Af)(x) = b^i (\partial_i f)(x) + \frac{1}{2} a^{ij} (\partial_i \partial_j f)(x) +$$

$$\int_{\mathbb{R}^d \setminus \{0\}} [f(x+y) - f(x) - y^i(\partial_i f)(x)\mathbf{1}_{0 < |y| < 1}(y)]\nu(dy) \tag{12}$$

Remark 1. Another definition of the infinitesimal generator is through the Fourier transform

$$(Af)(x) = \lim_{t \to 0+} \frac{P_t f - f}{t} \tag{13}$$

where $P_t f = f \star p_t$ and $\mathcal{F} p_t(\xi) = e^{-t\eta(\xi)}$.

To see this, consider the 1D case and without the adjustment term $-y^i(\partial_i f)(x)\mathbf{1}_{0<|y|<1}(y)$. By taking the Fourier transform of $(Af)(x)$, we have

$$\mathcal{F}(Af)(\xi) = \mathrm{i}\xi b \hat{f}(\xi) - \frac{1}{2} a\xi^2 \hat{f}(\xi) + \int_{-\infty}^{+\infty} (\hat{f}(\xi)e^{\mathrm{i}y\xi} - \hat{f}(\xi))\nu(y)dy$$

$$= \left(\mathrm{i}\xi b - \frac{1}{2} a\xi^2 + \int_{-\infty}^{+\infty} (e^{\mathrm{i}y\xi} - 1)\nu(y)dy \right) \hat{f}(\xi) \tag{14}$$

this is exactly the expression we see in eq. (6).

One the other hand,

$$\mathcal{F}\left[\lim_{t \to 0+} \frac{P_t f - f}{t} \right] = \lim_{t \to 0+} \frac{e^{\eta(\xi)t} \hat{f}(\xi) - \hat{f}(\xi)}{t} = \eta(\xi)$$

which coincides with eq. (14).

Again, we can drop the extra term $y^i(\partial_i f)(x)\mathbf{1}_{0<|y|<1}(y)$ if $\int_{\mathbb{R}^d}[f(x+y) - f(x)]\nu(dy)$ is well defined. We call the Lévy measure is *regular* if this condition is fulfilled.

Let T_t be the semigroup associated with Lévy process, and the associated infinitesimal generator is

$$(Af)(x) = c(x)f(x) + b^i(\partial_i f)(x) + \frac{1}{2} a^{ij}(\partial_i \partial_j f)(x) +$$

$$\int_{\mathbb{R}^d \setminus \{0\}} [f(x+y) - f(x) - y^i(\partial_i f)(x)\mathbf{1}_{0<|y|<1}(y)]\nu(dy) \tag{15}$$

we consider the transition measures $p_t(x)$ associated with T_t. Here p_t is absolutely continuous with respect to Lebesgue measure. Define the adjoint operator A^* of A, which satisfies

$$\int_{\mathbb{R}^d} (Af)(y)p_t(y)dy = \int_{\mathbb{R}^d} f(y)A^* p_t(y)dy \tag{16}$$

for all $f \in C_c^\infty(\mathbb{R}^d)$.

In general, there is no nice form for A^*. However, in the case $c(x)$, $b(x)$ and $a(x)$ are all constant, we have

$$A^* p_t(x) = c(\partial_i p_t)(x) - b^i(\partial_i p_t)(x) + \frac{1}{2} a^{ij} \partial_i \partial_j p_t +$$

$$\int_{\mathbb{R}^d} \left[p_t(x-y) - p_t(x) + y^i(\partial_i p_t)(x)\mathbf{1}_{0<|y|<1}(y) \right] \nu(dy) \tag{17}$$

The Fokker-Planck equation, or Kolmogorov forward equation, is

$$\frac{\partial p_t(x)}{\partial t} = A^* p_t(x) \quad p_0(x) = \delta(x) \tag{18}$$

4

3 $\mathcal{H}$ Matrix

For completeness, we discuss the hierarchical matrix technique. Especially we give a detailed description on the storage format, construction, fast matrix vector multiplication routine and LU decomposition. We later show how to construct the $\mathcal{H}$ matrix from kernels.

The discretization of the jump diffusion part $\int_{\mathbb{R}}(u(x+y)-u(y))dy$ will usually lead to a dense matrix, which typically requires $\mathcal{O}(N^2)$ storage and has $\mathcal{O}(N^2)$ complexity for matrix-vector multiplication, $\mathcal{O}(N^3)$ for LU decomposition. Many techniques, such as the panel clustering methods and the fast multipole methods were developed. Later $\mathcal{H}$-matrix was considered by W. Hackbusch and many variations of hierarchical matrices have been intensively studied by researchers. $\mathcal{H}$-matrices can reduces the storage and arithmetics to nearly optimal complexity $\mathcal{O}(N)$ up to $\log N$ scaling. It relies on the fact that the kernel functions are smooth in the off diagonal.

3.1 Construction and Storage

The construction of the $\mathcal{H}$ matrices can be best described in terms of matrix indices and the geometric points. Each entry A_{ij} represents the interaction between two nodes x_i and x_j. Let I, $J \subset \mathbb{N}$ be row and column index sets, then $A_{IJ} = (a_{ij})_{i \in I, j \in J}$ describes the interaction between a cluster $X_I = \{x_i\}_{i \in I}$ and another cluster $X_J = \{x_j\}_{j \in I}$. The interaction kernel function $k(x, y)$ is assumed to be smooth for sufficiently large $|x - y|$.

Typically, it requires $\mathcal{O}(|I||J|)$ complexity to store the interaction data. However, if we assume that $I \subset J = \emptyset$ and geometrically the clusters X_I, X_J are separate in the sense of admissibility

Definition 3.1. For two sets of indices I and J and the associated cluster X_I, X_J; assume that the kernel is asymptotically smooth, the admissibility condition is given by

$$\min\{\text{diam}(X_I), \text{diam}(X_J)\} \le \eta \text{dist}(X_I, X_J) \tag{19}$$

where $\text{diam}(X_I) = \max_{x_i, x_j \in X} |x_i - x_j|$ AND $\text{dist}(x_i, x_j) = \min_{x_i \in X_I, x_j \in X_J} |x_i - x_j|$. If the condition eq. (19) is not satisfied, we say X_I and X_J or I and J are inadmissible.

In our numerical examples, we use $\eta = 1$, which indicates adjacent clusters are inadmissible since the distance is always zero.

The admissible blocks usually have low rank structures. This is best illustrated by an example. Suppose $k(x, y) = \frac{1}{|x-y|^2}$, and further assume $x \in X_I$, $y \in X_J$. Assume X_I and Y_J are inadmissible, and $\bar{x} \in \mathcal{X}$, where $\mathcal{X}$ is the convex hull of X_I. Then we have

$$\frac{1}{|x-y|^2} = \frac{1}{|x-\bar{x}-(y-\bar{x})|^2} = \frac{1}{|y-\bar{x}|^2 \left|\frac{x-\bar{x}}{y-\bar{x}} - 1\right|^2}$$

Since

$$\left|\frac{x-\bar{x}}{y-\bar{x}}\right| < 1$$

we have

$$\frac{1}{|y-\bar{x}|^2 \left| \frac{x-\bar{x}}{y-\bar{x}} - 1 \right|^2} = \frac{1}{|y-\bar{x}|^2} \left(1 + \frac{x-\bar{x}}{y-\bar{x}} + \left(\frac{x-\bar{x}}{y-\bar{x}} \right)^2 + \ldots \right)$$

which is in the form of

$$\frac{1}{|x-y|^2} = \sum_{n=0}^{\infty} \alpha_n(x-\bar{x})\beta_n(y-\bar{x}) \tag{20}$$

Then series is convergent and therefore the residual term will decay. It is possible to approximate $\frac{1}{|x-y|^2}$ with a few terms

$$\frac{1}{|x-y|^2} \approx \sum_{n=0}^{r} \alpha_n(x-\bar{x})\beta_n(y-\bar{x}) \tag{21}$$

And therefore the interaction matrix for the cluster A_{IJ} is

$$A_{IJ} = \left(\frac{1}{|x_i - x_j|^2} \right)_{i \in I, j \in J} = \left(\sum_{n=0}^{r} \alpha_n(x_i - \bar{x})\beta_n(x_j - \bar{x}) \right)_{i \in I, j \in J} = UV' \tag{22}$$

where
$$U = \begin{bmatrix} \alpha_0(x_i - \bar{x}) & \alpha_1(x_i - \bar{x}) & \ldots & \alpha_r(x_i - \bar{x}) \end{bmatrix} \tag{23}$$
$$V = \begin{bmatrix} \beta_0(x_i - \bar{x}) & \beta_1(x_i - \bar{x}) & \ldots & \beta_r(x_i - \bar{x}) \end{bmatrix} \tag{24}$$

If $r \ll |I| \wedge |J|$, we have achieved matrix compression using a low rank representation.

The idea of the hierarchical matrix is then to classify each block A_{IJ} into three types

- Full matrix. In this case, A_{IJ} is represented using fully populated matrices.

- Low rank matrix. In the case I and J are admissible, we can store the block A_{IJ} in the form of low rank matrices. This will help us save storage and computational cost.

- $\mathcal{H}$-matrix. For the blocks that are neither low rank matrix nor small enough to become a full matrix, it is further divided into subblocks (for example, via quadtree structure).

The $\mathcal{H}$-matrix will be stored in a hierarchical format.

3.2 Matrix Vector Multiplication

One advantage of the $\mathcal{H}$ matrix is that the matrix vector multiplication is cheap. The matrix vector multiplication of $\mathcal{H}$-matrix can be described through the rule of the operator for three different kinds of subblocks

- Full matrix. In this case, the normal dense matrix vector multiplication is used.

- Low rank matrix. The operator can be carried out quite efficiently via

$$(UV')x = U(V'x) \tag{25}$$

  note $V'x$ is a $r \times 1$ vector.

- $\mathcal{H}$-matrix. If the subblock is

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{26}$$

  the matrix vector multiplication will be carried out recursively, i.e.

$$Bx = \begin{bmatrix} B_{11}x_1 + B_{12}x_2 \\ B_{21}x_1 + B_{22}x_2 \end{bmatrix} \tag{27}$$

3.3 LU Decomposition

$\mathcal{H}$-LU can be done in $\mathcal{H}$-matrix format and recursively in computational cost $\mathcal{O}(N)$ up to a $\log N$ scaling compared to dense LU in $\mathcal{O}(N^3)$.

We need to define a triangular solver which solves $AX = B$ for lower triangular matrix or $XA = B$ or upper triangular matrix. The matrices are either $\mathcal{H}$-matrix or full matrix. We only need to consider the lower triangular cases since in the latter case by transposition $A'X' = B'$, we reduce the problem to former.

The triangular solver will work differently for different situations.

- If B is a full matrix, then X is a full matrix and $X = A^{-1}B$. Here A is converted to a full matrix.

- If B is a low rank matrix, $B = B_1 B_2'$, then X is also a low rank matrix $X = (A^{-1}B_1)B_2'$.

- If A and B are both hierarchical matrices

$$\begin{bmatrix} A_{11} & \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

  Then we will first solve $A_{11}X_{11} = B_{11}$ and $A_{11}X_{12} = B_{12}$. Then we solve

$$A_{22}X_{21} = B_{21} - A_{21}X_{11} \quad A_{22}X_{22} = B_{22} - A_{21}X_{12} \tag{28}$$

The LU decomposition also works differently for different types of matrices. Again only full matrices and $\mathcal{H}$ matrices are considered.

For full matrices, the standard dense LU is adopted. For $\mathcal{H}$ matrices,

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & U_{22} \end{bmatrix}$$

The algorithm will works as follows

- LU decomposition of $A_{11} = L_{11}U_{11}$

- Triangular solve $L_{11}U_{12} = A_{12}$ (lower triangular, U_{12} is the unknown)

- Triangular solve $L_{21}U_{11} = A_{21}$ (upper triangular, L_{21} is the unknown)

- LU decomposition of $A_{22} - L_{21}U_{12} = L_{22}U_{22}$

The LU decomposition can also be performed in an in-place way, which will save storage.

4 Crank Nicolson Scheme Based on $\mathcal{H}$-matrix

4.1 Model Problem

We will consider the forward or backward equation driven by the Lévy process, where the model problem in 1D can be stated as a convection-diffusion integro-differential equation

$$u_t = au_{xx} + bu_x + cu + \mathcal{L}u \quad x \in \mathbb{R}, t \in (0,1) \tag{29}$$

where $a \geq 0$, $c \leq 0$, $b \in \mathbb{R}$, and

$$\mathcal{L}u = \int_{\mathbb{R}} (u(x+y) - u(x) - u'(x)\mathbf{1}_{0<|y|<1}(y)y)\nu(y)dy \tag{30}$$

4.2 Numerical Scheme

We consider the case where $\nu(y) < \infty$ and thus the term $u'(x)\mathbf{1}_{0<|y|<1}(y)y$ is not needed. In addition, we assume $\nu(y)$ is semi-heavy. The case $\nu(y)$ might grow to infinity at $y = 0$ and decays algebraically will be discussed in Section 4.4.3. To compute numerically the integral term, we need to restrict the computational domain to a bounded interval Ω

$$\mathcal{L}u \approx \int_{B_l}^{B_r} (u(x+y) - u(x) - u'(x)\mathbf{1}_{0<|y|<1}(y)y)\nu(y)dy \tag{31}$$

In fact, it is proved in ??? that if $\nu(dy)$ is semi-heavy, i.e., there exists $\alpha_r, \alpha_l > 0$, such that $\int_1^\infty e^{(1+\alpha_r)y}\nu(dy) < \infty$, and $\int_{-\infty}^{-1} |y|e^{\alpha_l|y|}\nu(dy) < \infty$, the solution $\tilde{u}(x,t)$ obtained using the truncated integral will satisfy

$$|u(x,t) - \tilde{u}(x,t)| = \mathcal{O}(e^{-\alpha_l|B_l|} + e^{-\alpha_r|B_r|}) \tag{32}$$

Therefore, the discretization scheme for $\mathcal{L}$ using trapezoidal rule on uniform grid will be

$$(\mathcal{L}u)(jh) \approx \sum_{j \in \mathcal{I}} u_{i+j}\nu_j w_j - u_i \lambda w_j \quad \lambda = \sum_{j \neq 0, j \in \mathcal{I}} \nu_j \tag{33}$$

Here $\nu_j = \nu(jh)$,
$$\mathcal{I} = \{i : ih \in \Omega\} \tag{34}$$

w_j is the weight for the trapezoidal rule and

$$w_j = \begin{cases} h & j \text{ is not the endpoint of } \mathcal{I} \\ \frac{h}{2} & j \text{ is the endpoint of } \mathcal{I} \end{cases} \tag{35}$$

We define the discrete operator δ_L

$$(\delta_L u)_j = \sum_{j=-\infty}^{\infty} (u_{i+j} - u_i)\nu_j h = \sum_{j \in \mathcal{I}, j=-\infty}^{\infty} u_{i+j}\nu_j h - u_i \lambda h \quad \lambda = \sum_{j \neq 0, j \in \mathcal{I}} \nu_j \tag{36}$$

Then Crank-Nicolson discretization of eq. (29) on a uniform grid with spacing h and timestep Δt is

$$(I + \frac{1}{2}\Delta t A)u_j^{n+1} = (I - \frac{1}{2}\Delta t A)u_j^n \tag{37}$$

8

where

$$A = -a\delta_x^2 - b\delta_{2x} - c - \delta_L \tag{38}$$

here δ_x^2 and δ_{2x} are the standard second difference and central first difference. Therefore, we have

$$A_{ij} = \begin{cases} \frac{2a}{h^2} - c + \lambda w_{j-i} & i = j \\ -\frac{a}{h^2} + \frac{b}{2h} - \nu_{-1}w_{j-i} & j = i - 1 \\ -\frac{a}{h^2} - \frac{b}{2h} - \nu_1 w_{j-i} & j = i + 1 \\ -\nu_{j-i}w_{j-i} & |j - i| \geq 2 \end{cases} \tag{39}$$

4.3 $\mathcal{H}$-matrix Construction

For simplicity, assume $a = b = c = 0$; according to eq. (39), these coefficients only contribute to the first off-diagonal parts of the coefficient matrix. We consider the matrix $A_\pm = I \pm \frac{1}{2}A$. Note since the operator δ_x^2, δ_{2x} only contributes to the tridiagonal, any nonzero entry in $A_\pm$ in the off-diagonal more than one entry away from the diagonal must be $\mp\frac{1}{2}\nu_j h$ according to eq. (36).

We define the kernel associated with each Lévy measure by

$$K(x, y) = \nu(y - x) \tag{40}$$

then we have $A_{i'j'} = K(x_i, x_j)$ for $|i - j| \geq 2$, where i' and j' corresponds to x_i, x_j.

The numerical scheme lead us to the following task

$$\text{Given } \mathbf{x}, \text{ compute } \mathbf{y}: \mathbf{y} = (I + \lambda A)\mathbf{x} \tag{41}$$

$$\text{Given } \mathbf{y}, \text{ solve for } \mathbf{x}: \mathbf{y} = (I + \lambda A)\mathbf{x} \tag{42}$$

here $\lambda > 0$ is a constant. We illustrate here the application of the $\mathcal{H}$-matrix technique using the example of A generated by Merton matrix. Assume $a = c = 0$ in eq. (39), then we can see that

$$A_{ij} = h^d k(x_i, x_j) \tag{43}$$

for some kernel function $k(x, y)$.

Consider the Merton model with Gaussian jumps, i.e., the Lévy density can be represented as

$$\nu(x) = e^{-\varepsilon^2 x^2} \tag{44}$$

We consider the kernel function $k(x, y)$ associated with the density

$$k(x, y) = \nu(x - y) = e^{-\varepsilon^2 (x-y)^2} \tag{45}$$

Assume that $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $\mathcal{X} \cap \mathcal{Y} = \emptyset$, and let $\bar{x} \in \mathcal{X}$. Denote $t_0 = x - \bar{x}$, and $t = y - \bar{x}$, then by assumption we have $|t| > |t_0|$. From Taylor expansion we have

$$e^{-\varepsilon^2(x-y)^2} = e^{-\varepsilon^2(t-t_0)^2} \tag{46}$$

$$= e^{-\varepsilon^2 t^2 \left(1 - \frac{t_0}{t}\right)^2} = e^{-\varepsilon^2 t^2 - \varepsilon^2 t_0^2 + 2\varepsilon^2 t_0 t} \tag{47}$$

$$= e^{-\varepsilon^2 t^2 - \varepsilon^2 t_0^2} \left(1 + 2\varepsilon^2 t_0 t + \frac{(2\varepsilon^2 t_0 t)^2}{2} + \frac{(2\varepsilon^2 t_0 t)^3}{3!} + \dots\right) \tag{48}$$

Thus we have

$$\alpha_n(t) = \frac{2^n \varepsilon^{2n} e^{-\varepsilon^2 t^2} t^n}{n!} \qquad \beta_n(t) = e^{-\varepsilon^2 t^2} t^n \tag{49}$$

we will have

$$e^{-\varepsilon^2 (x-y)^2} = \sum_{n=0}^{\infty} \alpha_n(t_0) \beta_n(t) \tag{50}$$

Lemma 1. Assume $\mathcal{X}$, $\mathcal{Y}$ are two disjoint set in Ω and $\mathrm{diam}(\Omega) = D$. Let $\delta > 0$ be any positive constant, then if

$$r > \max\left\{ \log_2\left(\frac{e^{2\varepsilon^2 D^2}}{\delta}\right) - 1, 12\varepsilon^2 D^2 - 1 \right\} \tag{51}$$

we have

$$\left| e^{-\varepsilon^2 (x-y)^2} - \sum_{n=0}^{r} \alpha_n(x - \bar{x}) \beta_n(y - \bar{x}) \right| < \delta \tag{52}$$

for any $\bar{x} \in \mathcal{X}$.

Proof. Note that

$$\left| e^{-\varepsilon^2 (x-y)^2} - \sum_{n=0}^{r} \alpha_n(x - \bar{x}) \beta_n(y - \bar{x}) \right| \tag{53}$$

$$= e^{-\varepsilon^2 t^2 - \varepsilon^2 t_0^2} \left(\frac{(2\varepsilon^2 t_0 t)^{n+1}}{(n+1)!} + \frac{(2\varepsilon^2 t_0 t)^{n+2}}{(n+2)!} + \dots \right) \tag{54}$$

$$\leq e^{-\varepsilon^2 t^2 - \varepsilon^2 t_0^2} \frac{(2\varepsilon^2 t_0 t)^{n+1}}{(n+1)!} e^{2\varepsilon^2 t_0 t} \leq e^{2\varepsilon^2 D^2} \frac{(2\varepsilon^2 D^2)^{n+1}}{(n+1)!} \tag{55}$$

We invoke the basic estimate

$$n! > \left(\frac{n}{3}\right)^n \tag{56}$$

and obtain

$$\left| e^{-\varepsilon^2 (x-y)^2} - \sum_{n=0}^{r} \alpha_n(x-\bar{x}) \beta_n(y-\bar{x}) \right| \leq e^{2\varepsilon^2 D^2} \frac{(2\varepsilon^2 D^2)^{n+1}}{\left(\frac{n+1}{3}\right)^{n+1}} = \left(\frac{6\varepsilon^2 D^2}{n+1}\right)^{n+1} e^{2\varepsilon^2 D^2} \tag{57}$$

Since we have eq. (51), which indicates

$$\frac{6\varepsilon^2 D^2}{n+1} < \frac{1}{2} \tag{58}$$

and therefore

$$\left| e^{-\varepsilon^2 (x-y)^2} - \sum_{n=0}^{r} \alpha_n(x - \bar{x}) \beta_n(y - \bar{x}) \right| < \left(\frac{1}{2}\right)^{n+1} e^{2\varepsilon^2 D^2} < \delta \tag{59}$$

the last equation is due to the assumption eq. (51). $\qquad\square$

Remark 2. In practice, the estimate eq. (51) is quite conservative and smaller r can actually work very well.

By using the $\mathcal{H}$-matrix, the storage complexity is reduced to $\mathcal{O}(N)$ which is demonstrated in fig. 1. The construction time is also reduced to $\mathcal{O}(N)$ compared to $\mathcal{O}(N^2)$ for full matrices.

In 2D, the Merton model with Gaussian jumps read

$$\nu(\mathbf{x}) = \exp(-\varepsilon^2 \|\mathbf{x}\|^2) \tag{60}$$

with the kernel function

$$k(\mathbf{x}, \mathbf{y}) = \nu(\mathbf{x} - \mathbf{y}) = \exp(-\varepsilon^2 \|\mathbf{x} - \mathbf{y}\|^2) \tag{61}$$

Let $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathcal{Y}$ and $\mathcal{X} \cap \mathcal{Y} = \emptyset$, and assume that $\bar{x} \in \mathcal{X}$,

$$t_1 = \mathbf{x}_1 - \bar{\mathbf{x}}_1 \quad t_2 = \mathbf{x}_2 - \bar{\mathbf{x}}_2 \quad s_1 = \mathbf{y}_1 - \bar{\mathbf{y}}_1 \quad s_2 = \mathbf{y}_2 - \bar{\mathbf{y}}_2 \tag{62}$$

we have

$$k(\mathbf{x}, \mathbf{y}) = \sum_{m,n=0}^{\infty} \left[\frac{(2\varepsilon^2)^{m+n}}{m!n!} s_1^m t_1^n \exp\left(-\varepsilon^2(t_1^2 + s_1^2)\right) \right] \left[s_2^m t_2^n \exp\left(-\varepsilon^2(t_2^2 + s_2^2)\right) \right] \tag{63}$$

Let

$$\alpha_{m,n} = \frac{(2\varepsilon^2)^{m+n}}{m!n!} s_1^m t_1^n \exp\left(-\varepsilon^2(t_1^2 + s_1^2)\right) \tag{64}$$

$$\beta_{m,n} = s_2^m t_2^n \exp\left(-\varepsilon^2(t_2^2 + s_2^2)\right) \tag{65}$$

Similar to eq. (52), we can approximate the kernel using low rank summation

$$k(\mathbf{x}, \mathbf{y}) \approx \sum_{m,n=0}^{r} \alpha_{m,n}(s_1, t_1) \beta_{m,n}(s_2, t_2) \tag{66}$$

Using the storage strategy in Section 3.1, we are able to construct the $\mathcal{H}$-matrix directly. Figure 1 shows the construction time as well as the storage consumption. Particularly, we compare the construction time of the $\mathcal{H}$-matrix with that of the dense matrix. We can see that the construction of $\mathcal{H}$-matrix is quite efficient, both in terms of storage consumption and time consumption: they both achieves an approximately linear asymptotic rate with respect to the problem size N. For 16k×16k matrices, the speed of constructing an $\mathcal{H}$-matrix is almost 600 times faster than the direct approach.

4.4 Error Analysis

4.4.1 Stability

We carry out the stability analysis using the Fourier transform pair

$$u_j^n = \frac{1}{2\pi h} \int_{-\pi}^{\pi} \hat{u}^n(x) \exp(ijx) dx \tag{67}$$

$$\hat{u}_j^n = h \sum_{-\infty}^{\infty} u_j^n \exp(-ijx) dx \tag{68}$$

For simplicity, we assume that the domain is not truncated, i.e. $\mathcal{I} = \mathbb{Z}$. We need the following formula for
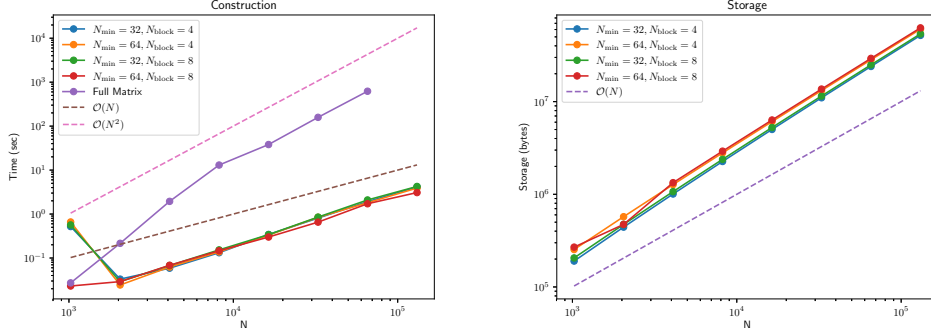
11

Figure 1: The construction time and the storage consumption of $\mathcal{H}$-matrix. We compare the construction time of the $\mathcal{H}$-matrix with that of the dense matrix. We can see that the construction of $\mathcal{H}$-matrix is quite efficient, both in terms of storage consumption and time consumption: they both achieves an approximately linear asymptotic rate with respect to the problem size N.

Lemma 2. Let

$$\eta_h(\theta) = \sum_{j=-\infty}^{\infty} (e^{\mathrm{i}jh\theta} - 1)v_j h \tag{69}$$

then we have

$$\delta_L e^{\mathrm{i}\theta x} = \eta_h(\theta)e^{\mathrm{i}\theta x} \tag{70}$$

where $\nu_j = \nu(jh)$. In the case ν_j is symmetric, we have

$$\eta_h(\theta) = -2 \sum_{j=-\infty}^{\infty} \sin^2\left(\frac{j\theta h}{2}\right) v_j h \leq 0 \tag{71}$$

Proof. By definition, we have

$$\delta_L e^{\mathrm{i}\theta x} = \int_{-\infty}^{\infty} \left(e^{\mathrm{i}(x+y)\theta} - e^{\mathrm{i}x\theta} \right) \nu(y)dy = \eta_h(\xi)e^{\mathrm{i}\theta x}$$

The other properties easily follow. $\qquad\square$

The Fourier transform of the numerical scheme gives

$$\hat{u}_i^{n+1} = \frac{1 - a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} + \frac{b\Delta t}{2h}\mathrm{i}\sin\theta + \frac{c\Delta t}{2} + \frac{\Delta t\eta_h(\theta)}{2}}{1 + a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} - \frac{b\Delta t}{2h}\mathrm{i}\sin\theta - \frac{c\Delta t}{2} - \frac{\Delta t\eta_h(\theta)}{2}}\hat{u}_i^n \tag{72}$$

Note we have

$$\left| 1 - a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} + \frac{b\Delta t}{2h}\mathrm{i}\sin\theta + \frac{c\Delta t}{2} + \frac{\Delta t\eta_h(\theta)}{2} \right|^2 =$$

$$\left| 1 - a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} + \frac{c\Delta t}{2} + \frac{\Delta t\eta_h(\theta)}{2} \right|^2 + \left| \frac{b\Delta t}{2h}\sin\theta \right|^2 \tag{73}$$

$$\left| 1 + a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} - \frac{b\Delta t}{2h}\mathrm{i}\sin\theta - \frac{c\Delta t}{2} - \frac{\Delta t\eta_h(\theta)}{2} \right| =$$

$$\left| 1 + a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} - \frac{c\Delta t}{2} - \frac{\Delta t\eta_h(\theta)}{2} \right|^2 + \left| \frac{b\Delta t}{2h}\sin\theta \right|^2 \quad (74)$$

Since we have $a \geq 0$, $c \leq 0$, $\eta_h(\theta) \leq 0$, we always have

$$\left| 1 + a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} - \frac{c\Delta t}{2} - \frac{\Delta t\eta(\theta)}{2} \right| \geq \left| 1 - a\frac{\Delta t}{h^2}\sin^2\frac{\theta}{2} + \frac{c\Delta t}{2} + \frac{\Delta t\eta(\theta)}{2} \right| \quad (75)$$

Therefore, the model of the ratio in eq. (72) is always no greater than 1. Thus all the wave modes $e^{\mathrm{i}\theta x}$ will not grow in magnitude if we carry out the Crank-Nicolson scheme.

4.4.2 Consistency

In consideration of eq. (32), we assume that

$$\nu(y) = 0 \quad y > B_r \text{ or } y < B_l \quad (76)$$

The consistency is a direct result of the Crank Nicolson scheme. Note that eq. (36) is the trapezoidal discretization of the nonlocal operator, we have

$$\int_{B_l}^{B_r} (u(x_j + y) - u(x_j))\nu(y)dy = (\delta_L u)_j + \mathcal{O}(h^2) \quad (77)$$

and therefore

$$\frac{(\delta_L u)_j^n + (\delta_L u)_j^{n+1}}{2}$$
$$= \int_{B_l}^{B_r} \left(\frac{u(x_{i+j} + y, t_n) + u(x_{i+j} + y, t_{n+1})}{2} - \frac{u(x_i + y, t_n) + u(x_i + y, t_{n+1})}{2} \right)\nu(y)dy + \mathcal{O}(h^2)$$
$$= \int_{B_l}^{B_r} \left(u\left(x_{i+j} + y, t_{n+\frac{1}{2}}\right) - u\left(x_i + y, t_{n+\frac{1}{2}}\right) \right)\nu(y)dy + \mathcal{O}(\Delta t^2 + h^2)$$
$$= \mathcal{L}u\left(x_i, t_{n+\frac{1}{2}}\right) + \mathcal{O}(\Delta t^2 + h^2)$$

$$(78)$$

It is classical results that

$$\frac{(a\delta_x^2 + b\delta_{2x} + c)u(x_i, t_n) + (a\delta_x^2 + b\delta_{2x} + c)u(x_i, t_{n+1})}{2}$$
$$= au_{xx}\left(x_i, t_{n+\frac{1}{2}}\right) + bu_x\left(x_i, t_{n+\frac{1}{2}}\right) + cu\left(x_i, t_{n+\frac{1}{2}}\right) + \mathcal{O}(h^2 + \Delta t^2) \quad (79)$$

and that

$$\frac{u(x_i, t_{n+1}) - u(x_i, t_n)}{\Delta t} = \mathcal{O}(\Delta t^2)$$

therefore combining eqs. (78) and (79) we have

Lemma 3 (Consistency). Assume eq. (76) holds. Then the truncation error for the numerical scheme eq. (37)

$$T_i^n := \frac{u(x_i, t_{n+1}) - u(x_i, t_n)}{\Delta t} - \frac{(a\delta_x^2 + b\delta_{2x} + c + \delta_L)u(x_i, t_n) + (a\delta_x^2 + b\delta_{2x} + c + \delta_L)u(x_i, t_{n+1})}{2}$$

$$(80)$$

satisfies

$$T_i^n = \mathcal{O}(\Delta t^2 + h^2) \quad (81)$$

13

4.4.3 Convergence

Finally, we are in a position to prove the convergence of the numerical scheme eq. (37).

Theorem 1. Assume that $\nu(y) \in C(\mathbb{R})$ and the condition in lemma 3 is satisfied. Let u_i^n be the numerical solution at x_i and time t_n, and $u(x,t)$ be the exact solution. Then the numerical scheme eq. (37) is unconditionally stable and

$$|u(x_i, t_n) - u_i^n| = \mathcal{O}(\Delta t^2 + h^2) \quad \Delta t \to 0, h \to 0 \tag{82}$$

Proof. The theorem is a direct result of the stability and consistency. $\square$

5 Singular and Slow Decaying Lévy Measure

We now consider the general case where $\nu(y)$ can behave very badly. One such example is the fractional Laplacian where the Lévy measure is

$$\nu(y) = \frac{c_{1,s}}{|y|^{1+2s}} \tag{83}$$

where $s \in (0,1)$ and $c_{1,s}$ is a positive constant. Note in this case, $\int_{\mathbb{R}} (u(x + y) - u(x))\nu(y)dy$ must be understood in the principal value integration. The corresponding stochastic process associated with the fractional Laplacian is the α-stable process.

Consider the general singular integral operator

$$I(x) = \int_{\mathbb{R}} (u(x + y) - u(x) - \rho(y)u'(x)y)\nu(y)dy \tag{84}$$

where $\rho(y)u'(x)$ is a drift term to remove small activity from the jumps. $\rho(y)$ is a radial symmetric window function, satisfying

$$\begin{cases} 1 - \rho(y) \sim \mathcal{O}(y^4) & y \to 0 \\ \rho(y) = 0 & |y| \geq r \end{cases} \tag{85}$$

where $r > 0$ is a positive number.

As a reminder, we require $\nu(y)$ to satisfy the following conditions

$$\int_{-r}^{r} y^2 \nu(y)dy < \infty, \quad \int_{|y| \geq r} \nu(y)dy < \infty \tag{86}$$

The choice of $\rho(y)$ doesn't matter. In fact, if $\tilde{\rho}(x)$ is another window function that satisfies eq. (85), we have

$$\int_{\mathbb{R}} (u(x + y) - u(x) - \tilde{\rho}(y)u'(x)y)\nu(y)dy \tag{87}$$

$$= \int_{\mathbb{R}} (u(x + y) - u(x) - \rho(y)u'(x)y)\nu(y)dy + u'(x)\int_{\mathbb{R}} (\rho(y) - \tilde{\rho}(y)y)\nu(y)dy \tag{88}$$

we can add the second drift term into the model.

The first condition in eq. (85) is designed to take into consideration of the heavy tail case, where $\nu(y)$ can decay like $\mathcal{O}(1/|y|^{1+2s})$ for some $s \in (0,1)$. For example, in the special case

$$\nu(y) = c_{1,s} \frac{1}{|y|^{1+2s}} \tag{89}$$

we obtain the fractional Laplacian. Then $\int_{\mathbb{R}} (u(x+y) - u(x))\nu(y)dy$ is not well defined but only in the principle value integration, and it is easy to see

$$\text{p.v.} \int_{\mathbb{R}} (u(x+y) - u(x))\nu(y)dy = \int_{\mathbb{R}} (u(x+y) - u(x) - \rho(y)u'(x)y)\nu(y)dy \tag{90}$$

for any valid window function $\rho(y)$ thanks to the cancellation of the drift term due to symmetry.

Although $\int_{\mathbb{R}} (u(x+y) - u(x) - \rho(y)u'(x)y)\nu(y)dy$ is well-defined in this case, the integrand will behave like

$$(u(x+y) - u(x) - \rho(y)u'(x)y)\nu(y) = \mathcal{O}\left(\frac{1}{|y|^{2s-1}}\right) \tag{91}$$

in the case $s \to 0+$, we will have numerical difficulty if a direct numerical integration is applied. In the following, we will propose a numerical discretization for eq. (84) targeting at the most numerical challenging case described above

$$\nu(y) = \frac{n_0(y)}{|y|^{1+2s}} \tag{92}$$

where $n_0(y)$ is a bounded continuous function.

We make two assumptions on $u(x)$

- $u \in C(\mathbb{R})$

- Local smoothness. $u \in C^4([x - \delta, x + \delta])$ for some $\delta > 0$, i.e., u has fourth order derivative near the location where we want to evaluate $I(x)$.

- Far field asymptotic limit. The far field contribution

$$f_x(y) = \int_{\mathbb{R}} u(x+y)\nu(y)dy \tag{93}$$

  is well defined. In the case $\frac{u(x+y)}{u(y)} \to f(y)$, this term can be approximated by $\int_{\mathbb{R}} f(y)dy$

The strategy is to singularity subtraction. We subtract a local diffusion term from eq. (84)

$$I(x) = \int_{\mathbb{R}} (u(x+y) - u(x) - \rho(y)u'(x)y - \frac{1}{2}\rho(y)u''(x)y^2)\nu(y)dy + \frac{1}{2}u''(x)\int_{\mathbb{R}} \rho(y)\nu(y)y^2 dy \tag{94}$$

We can immediately split the first integral into two parts

$$I_1(x) = \int_{|y| \leq L_W} (u(x+y) - u(x) - \rho(y)u'(x)y - \frac{1}{2}\rho(y)u''(x)y^2)\nu(y)dy \tag{95}$$

15

and

$$I_2(x) = \int_{|y|>L_W} (u(x+y) - u(x) - \rho(y)u'(x)y - \frac{1}{2}\rho(y)u''(x)y^2)\nu(y)dy \quad (96)$$

$$= \int_{|x|>L_W} (u(x+y) - u(x))\nu(y)dy = f_x^{L_W} - u(x)\int_{|y|>L_W} \nu(y)dy \quad (97)$$

where L_W is a mask window and satisfies

$$L_W > r \quad (98)$$

By Taylor expansion it is easy to see

$$u(x+y) - u(x) - \rho(y)u'(x)y - \frac{1}{2}\rho(y)u''(x)y^2 = \mathcal{O}(|y|^3) \quad (99)$$

and therefore $I_1(x)$ will behave like $\mathcal{O}(|y|^{2-2s})$ near the origin. We will prove this fact in the appendix. That indicates the integrand becomes continuous near the origin. Thus $I_1(x)$ is well defined.

As $y \to \infty$, the term terms in $I_2(x)$ are both well defined according to the assumptions eqs. (86) and (93).

The second term in eq. (94)

$$I_3(x) = \frac{1}{2}u''(x)\int_{|y|\leq r} \nu(y)y^2 dy \quad (100)$$

is a local diffusion term and the coefficient is well defined according to eq. (86).

We now focus on the numerical discretization of $I_1(x)$, $I_2(x)$ and $I_3(x)$. We divide the mask window into $2N$ uniform subintervals and consider the grid $\{ih : i \in \mathbb{Z}\}$, where $h = L_W/N$. We denote $u_i = u(ih)$.

Since the integrand in $I_1(x)$ is continuous, we can use a simple trapezoidal quadrature rule to approximate the integral. Assume the quadrature weights are w_j given by $w_{-N} = w_N = \frac{h}{2}$ and $w_j = h$, $j = -N+1, -N+2, \ldots, N-2, N-1$.

$$I_1(x_i) \approx \sum_{j=-N}^{N} u_{i+j}n(jh)w_j - u_i \sum_{j=-N}^{N} n(jh)w_j$$

$$- \frac{u_{i+1} - u_{i-1}}{h} \sum_{j=-N}^{N} \rho(jh)n(jh)jh - \frac{u_{i+1} + u_{i-1} - 2u_i}{2h^2} \sum_{j=-N}^{N} \rho(jh)(jh)^2 n(jh) \quad (101)$$

where $\sum_{j=-N}^{N}$ denotes the summation excluding $j = 0$.

For $I_2(x)$, $f_x^{L_W}$ is either provided as an input or computed using a numerical quadrature and so is $\int_{|x|>L_W} \nu(y)dy$. We will see how these terms are obtained in the examples below.

$$I_2(x_i) \approx f_{x_i}^{L_W} - u(x_i)\int_{|y|>L_W} \nu(y)dy \quad (102)$$

For I_3, a central difference scheme is applied to the second order derivative term.

$$I_3(x_i) \approx \frac{u_{i+1} + u_{i-1} - 2u_i}{2h^2} \int_{|y|\leqslant r} n(y)y^2 dy \quad (103)$$

16

and the integral can either be computed analytically or numerically.

In practice, we want to compute $I(x)$ for $x \in [-L, L]$, according to **??**, we need to know the values of $u(x)$ on $[-L - L_W, L + L_W]$ and its corres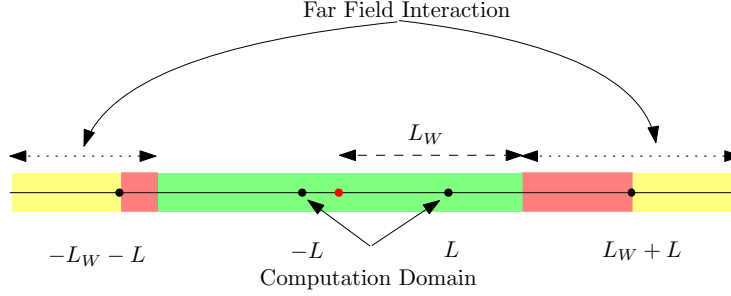ponding far-field interactions. Figure 2 visualizes the relationship. To compute $I(x_i)$, we need to compute the near field interaction and local interaction using values of $u(x)$ from the green area. The values of $u(x)$ are provided in the green and red area for computing $I(x)$, $x \in [-L, L]$.

Figure 2: To compute $I(x_i)$, we need to compute the near field interaction and local interaction using values of $u(x)$ from the green area. The values of $u(x)$ are provided in the green and red area for computing $I(x)$, $x \in [-L, L]$.

Although we have used a different formula for the evaluation of the integral, we should soon realize that in the far-away off-diagonal parts, the entries are still $\nu_j h$ (except on the boundary), which the $\mathcal{H}$ matrix construction routine can still work.

6 Applications

In this section, we list several possible applications of the numerical scheme and fast algorithms. These applications are taken from literature which can be formulated as an integro-differential equation.

6.1 Option Pricing

Let S_t be the price of a financial asset which is modelled as a stochastic process under a martingale equivalence measure $\mathbb{Q}'$ and on a filtered probability space $(\Omega, \mathcal{F}, \mathcal{F}_t, \mathbb{Q})$.

One of the popular models is the exponential Lévy model which assumes

$$S_t = S_0 e^{rt + X_t} \tag{104}$$

where X_t is a Lévy process. Assume r is the interest rate. For a European call or put, the terminal payoff H_T at time T is associated with the underlying asset price S_T

$$H_T = H(S_T) \tag{105}$$

The value of the option is defined as a discounted conditional expection of H_T under the risk-adjusted martingale measure

$$C_t = \mathbb{E}[e^{-r(T-t)} H(S_T) | \mathcal{F}_t] = \mathbb{E}[e^{-r(T-t)} H(S_T) | S_t = S] \tag{106}$$

By introducing $\tau = T - t$, $x = \log\left(\frac{S}{S_0}\right)$, and define

$$u(x, \tau) = \mathbb{E}[h(x + Y_\tau)] \quad h(x) = H(S_0 e^x) \tag{107}$$

for sufficiently smooth u, by applying the Ito's formula for Lévy process we have the integro-differential equation

$$\frac{\partial u}{\partial \tau} = \frac{\sigma^2}{2} u_{xx} - (\sigma^2/2 - r + \alpha) u_x + \int_{\mathbb{R}} (u(x+y) - u(x) - u'(x)\mathbf{1}_{0<|y|<1}(y)y)\nu(y)dy \tag{108}$$

with initial condition

$$u(0, x) = h(x) \tag{109}$$

6.2 Quantum Mechanics

If the underlying stochastic process powering the random fluctuations is a Gaussian Brownian motion, we obtain the non relativistic Schrödinger's equation

$$i\hbar\partial_t \psi(x, t) = -\frac{\hbar^2}{2m} \partial_x^2 \psi(x, t) \tag{110}$$

In recent years, there are a growing interest in non Gaussian stochastic process, and particularly the Lévy process. One of the popular models is the fractional quantum mechanics, where the stable processes are used as the underlying stochastic process. The popularity of the stable process is justified by the properties of scaling and self-similarity displayed by the process. For any distribution with power-law decay $\frac{1}{|x|^{1+\alpha}}$, $0 < \alpha < 1$ the generalized central limit theorem guarantee that their sum scaled by $\frac{1}{n^{1/\alpha}}$ converge to the α-stable distribution. If the variance are finite, i.e., $\alpha \geq 2$, then the central limit theorem holds, where their sum scaled by $\frac{1}{n^{1/2}}$, properly centered, and identically distributed, converge to the Gaussian distribution. This leads to the fractional Schrödinger equation

$$i\hbar\partial_t \psi(x, t) = D_\alpha (-\hbar^2\Delta)^{\alpha/2} \psi(x, t) \tag{111}$$

where $(-\hbar^2\Delta)^{\alpha/2}$ is the fractional Laplacian which can be defined through

$$(-\hbar^2\Delta)^{\alpha/2} \psi(x, t) = \frac{1}{(2\pi\hbar)^3} \int |\xi|^\alpha \hat{\psi}(\xi, t) \exp(i(\xi, x)/\hbar) d\xi \tag{112}$$

https://pdfs.semanticscholar.org/1dc0/9fedfa72c9f051da4642ff753d2bfe9f7a5f.pdf?_ga=2.23

More generally, other Lévy measure can be used to develop the quantum mechanics. The more general Schrödinger equation reads

$$i\hbar\partial_t \psi(x, t) = -\frac{\hbar^2}{2m} \partial_x^2 \psi(x, t) - \hbar \int_{\mathbb{R}} [\psi(x + y, t) - \psi(x, t)]\nu(y)dy \tag{113}$$

https://arxiv.org/pdf/0805.0503.pdf
Some examples of the Lévy-Schrödinger equations are

- Relativistic.
$$i\hbar\partial_t \psi(x, t) = \sqrt{m^2 c^4 - c^2\hbar^2\partial_x^2} \psi(x, t) \tag{114}$$

- Variance-Gamma laws
$$i\hbar\partial_t \psi(x, t) = -\frac{\lambda\hbar}{\tau} \int_{\mathbb{R}} \frac{\psi(x + y, t) - \psi(x, t)}{|y|} e^{-|y|/\hbar} dy \tag{115}$$

18

6.3 Turbulence Flow

It is known that turbulence flow exhibits anomalous diffusion, i.e., the diffusion occurs over distance ξ may scale more than one half, $\xi \sim \mathcal{O}(t^{1/2})$. There are many efforts to model turbulence and capture these anomalies. One of the recent research is the the modeling of turbulence flow via the fractional Laplacian.

If we assume that the equilibrium probability distribution of particle speeds to be Lévy α-stable distributions instead of the Maxwell-Boltzmann distribution, we will arrive at the Navier-Stokes equation with the fractional Laplacian operator as a means to represent the mean friction force arising in a turbulence flow

$$\rho \frac{D\bar{u}}{Dt} = -\nabla p + \mu_\alpha \nabla^2 \bar{u} + \rho C_\alpha \int_{\mathbb{R}^3} \frac{\bar{u}(x',t) - \bar{u}(x,t)}{|x - x'|^{\alpha+3}} dx' \tag{116}$$

7 Numerical Examples

7.1 Efficiency of $\mathcal{H}$-Matrix: 1D Case

In this section, we show the efficiency of the arithmetic operations using the $\mathcal{H}$-matrix in 1D. In the experiment, the minimum block size is either 64 or 32. The matrix sizes tested are $2^5, 2^6, \ldots, 2^{17}$. The maximum block size for $2^n \times 2^n$ matrices is $2^{n-3} \times 2^{n-3}$. The rank for off-diagonal approximation is 10, which is quite accurate for the Merton kernel we considered. The author observed that for $2^{17} \times 2^{17}$ matrices, the dense LU will throw `OutOfMemory` error. The codes are implemented using `julia`.

We divide the interval $[-1, 1]$ into 2^n equal length intervals, $h = \frac{1}{2^{n-1}}$. Consider the following kernel function for the construction of $\mathcal{H}$-matrix

$$k(x, y) = \begin{cases} -\frac{10}{h} + e^{-|x-y|^2} & \text{if } x = y \\ e^{-|x-y|^2} & \text{otherwise} \end{cases} \tag{117}$$

the term $-\frac{10}{h}$ is used to adjust the condition number of the matrix (it also arises naturally from implicit scheme of the PDE derived from Lévy process). For admissibility condition, we use $\eta = 1$. For low-rank blocks, the rank is fixed to be 5. In effect, the rank can be chosen adaptively; however, we observe that the fixed rank strategy is practical for our cases.

A typical hierarchical matrix in 1D will have the skeleton shown in fig. 3. Here we use a different color for each block. The blue and green blocks represent low rank blocks while the yellow blocks represent dense blocks. The matrix is arranged into a hierarchical structure, from which $\mathcal{H}$-matrix got its name.
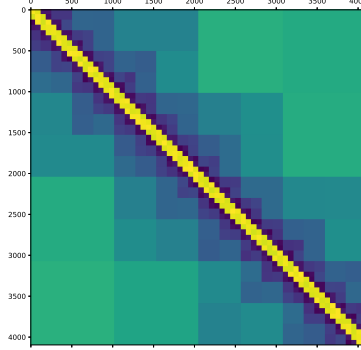
Figure 3: We use a different color for each block. The blue and green blocks represent low rank blocks while the yellow blocks represent dense blocks.

The key for maintaining optimal rates while the problem size N becomes large is to control the total number of dense blocks. In principal, the number of dense blocks should grow linearly with problem size, which can be demonstrated by looking at the compression ratio or the number of total blocks (full dense blocks as well as low rank blocks).

In fig. 4, the left figure shows the compression ratio of the $\mathcal{H}$-matrix, which is defined as

$$\texttt{CompressRatio} = \frac{\#\texttt{Matrix Entries Stored}}{\#\texttt{Full Matrix Entries}} \tag{118}$$

compress ratio gives us an estimate of the efficiency of the $\mathcal{H}$-matrix storage. Small compress ratios for large matrices is critical for large scale problems. On the right hand size, we show that number of full subblocks and low-rank subblocks as N grows. The linear growth is easily read off the figure.
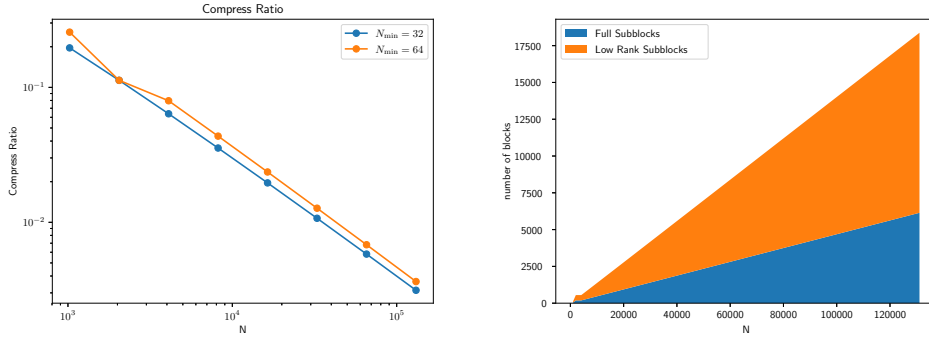
Figure 4: Compression ratio and number of blocks as the problem size grows.

Matrix Vector Multipliction   Figure 5 shows the complexity of the matrix vector multiplication for full matrices and $\mathcal{H}$-matrix. Compared to the dense

matrix vector multiplication, the $\mathcal{H}$-matrix structure lends us great speedup. This enables us to device highly efficient iterative solvers, such as preconditioned conjugate gradient method, which may requires many matrix vector productions during the iterations.
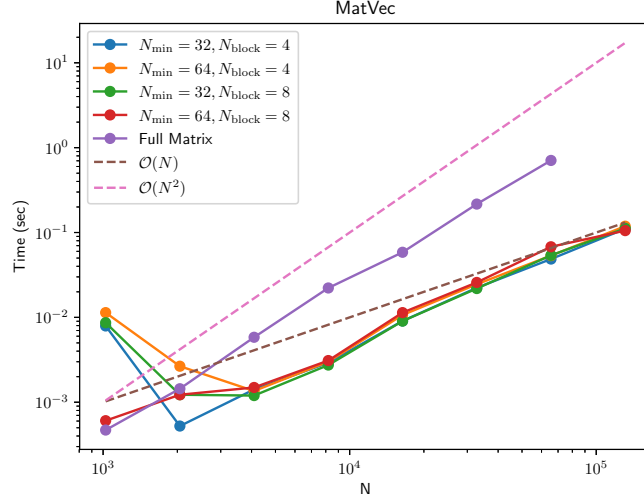

Figure 5: Matrix vector production is also much more efficient using the $\mathcal{H}$-matrices than using the dense matrix. It has the asymptotic complexity rate slightly above $\mathcal{O}(N)$, compared to $\mathcal{O}(N^2)$ for dense matrices.

LU Decomposition   We have already shown that the storage and construction complexity is $\mathcal{O}(N)$ in fig. 1. In fig. 6 we also show that the LU decomposition is also much more efficient using the $\mathcal{H}$-matrices. We see that the $\mathcal{H}$-LU has better asymptotic complexity than the dense LU, which has complexity $\mathcal{O}(N^3)$. Note the $\mathcal{H}$-LU decomposition is carried out using high accuracy and can serve as a direct solver for linear systems. We need to point out that although the $\mathcal{H}$-LU tends to beat dense LU in terms of time consumption for large scale problems, the constant in the asymptotic rate $\mathcal{O}(N)$ is still large, which is well-known in literatures. Also, since we have fixed ranks, the practical time consumption is not optimal and has a higher asymptotic rate than the linear one.

Solve   One crucial step for a successful implicit scheme is to solve the equation $A\mathbf{x} = \mathbf{y}$. We can of course use matrix-free solvers such as PCG. However, in the case that A is ill-conditioned we may require a good preconditioner. Finding such a preconditioner is not an easy task, especially for the dense matrices, which is not covered by literature as comprehensively as that of sparse counterparts. $\mathcal{H}$-LU lends us a generic way to construct preconditioners or direct solvers. In both cases, we need to factorize A as mentioned, and then solve $A\mathbf{x} = \mathbf{y}$.

   In this numerical experiment, we generate a random vector and record the solving time for both factorized $\mathcal{H}$-matrix and LU factorized dense matrix. In
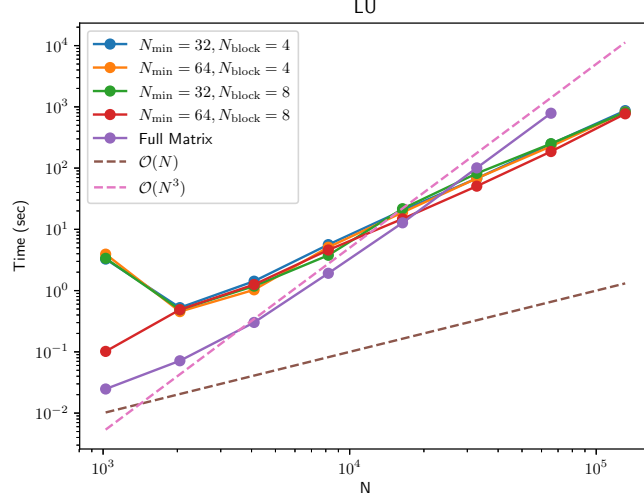
Figure 6: LU decomposition of $\mathcal{H}$-LU and the dense LU. The $\mathcal{H}$-LU has better asymptotic complexity than the dense LU, which has complexity $\mathcal{O}(N^3)$.

Figure 9 compare the solving time for both the dense matrix and the $\mathcal{H}$-matrix. We see that the $\mathcal{H}$-matrix solving is both faster and has better asymptotic rate than the dense one.

7.2 Efficiency of $\mathcal{H}$-Matrix: 2D Case

We briefly mention that the $\mathcal{H}$-matrix technique also works well in 2D. In this case, we divide $[-1,1]^2$ into $2^n \times 2^n$ equal size squares and let $h = \frac{1}{2^{n-1}}$. We use the kernel function

$$k(\mathbf{x}, \mathbf{y}) = \begin{cases} -\frac{10}{h^2} + e^{-|\mathbf{x}-\mathbf{y}|^2} & \text{if } \mathbf{x} = \mathbf{y} \\ e^{-|\mathbf{x}-\mathbf{y}|^2} & \text{otherwise} \end{cases} \tag{119}$$

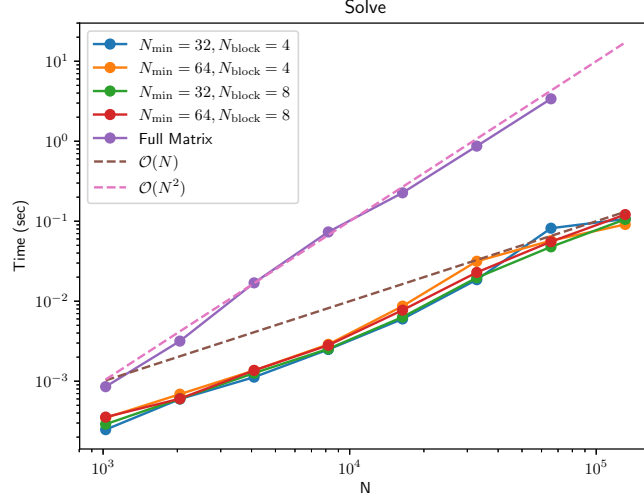We use the rank $r = 5$. A typical $\mathcal{H}$-matrix is shown in fig. 8.

Figure 7: Solving time for both the dense matrix and the $\mathcal{H}$-matrix. The $\mathcal{H}$-matrix solving is both faster and has better asymptotic rate than the dense one.
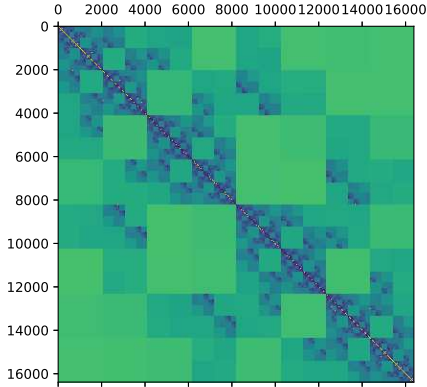

Figure 8: A typical $\mathcal{H}$-matrix in 2D constructed from eq. (119).

We perform the same comparison as that in the last section. The $\mathcal{H}$-matrix technique has better asymptotic rate than that of the dense matrices in terms of time consumption. $\mathcal{H}$-matrix will have great advantage over the dense matrices over the dense matrices for large scale problems.
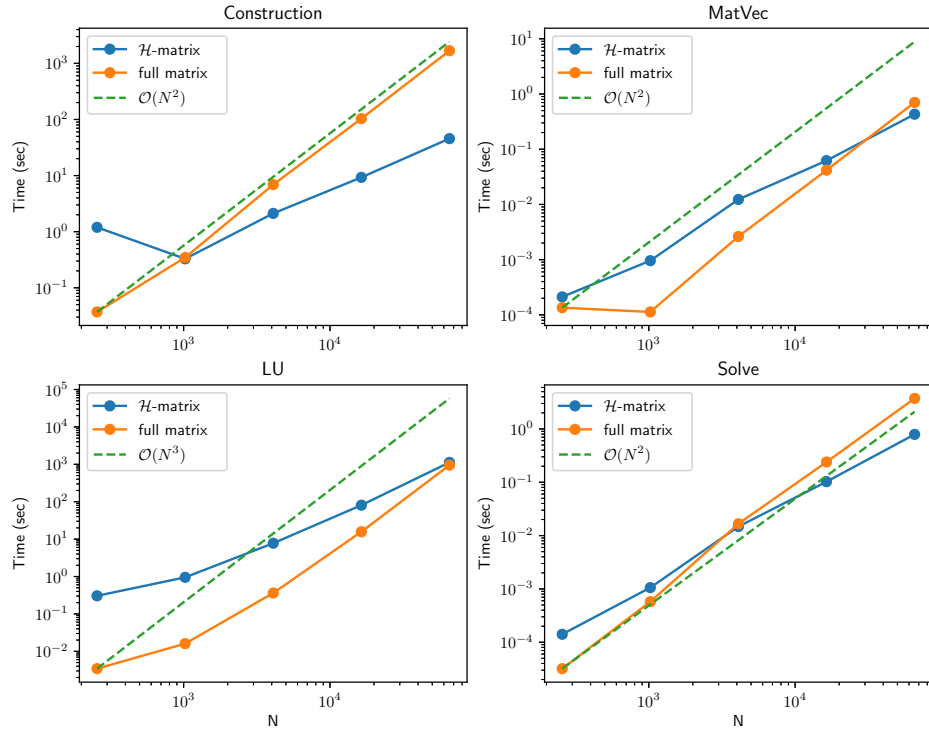
Figure 9: Comparison of construction time, matrix vector multiplication time, LU decomposition, and solving for both dense matrices as well as $\mathcal{H}$-matrices. The $\mathcal{H}$-matrix technique has better asymptotic rate than that of the dense matrices in terms of time consumption. The green dashed line shows the theoretical complexity asymptotic rate for dense matrices.

7.3 Convection Diffusion Equation Driven by the Lévy Process

In this section, we consider solving the model problem using $\mathcal{H}$-matrix and full matrix. We use a large rank 5 for the kernel function and therefore the $\mathcal{H}$ matrix as well as its LU decomposition are very accurate. Thus, we do not need iterative solvers. Instead we can directly do triangular solve given the LU decomposition at each iteration. Another choice is to use the LU decomposed matrix as preconditioners, which we will not pursue here.

Figure 10 shows the computational cost as well as the storage cost for both $\mathcal{H}$-matrix and the full matrix. We see that both in terms of time and storage complexity $\mathcal{H}$-matrix has advantage over full matrices.
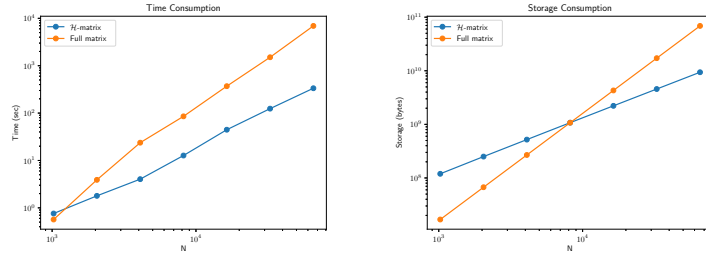

Figure 10: Time and storage complexity for $\mathcal{H}$-matrix and full matrix. Both in terms of time and storage complexity $\mathcal{H}$-matrix has advantage over full matrices for large scale problems.

Verification Method We briefly mention how we verify our numerical solution. We compare the numerical solution to the analytical solution. The analytical solution to the model problem eq. (29) can be found using the Fourier transform when the coefficients are constant. For simplicity, we consider the case where the jump distribution is the Gaussian distribution, i.e.

$$\nu(y) = d \exp(-\varepsilon^2 y^2) \tag{120}$$

Then the corresponding characteristic function of the Lévy measure is

$$\eta(\xi) = -a\xi^2 + b\mathrm{i}\xi + c + \int_{\mathbb{R}} (e^{\mathrm{i}\xi y} - 1)\nu(y)dy \tag{121}$$

The last term can be computed analytically,

$$\int_{\mathbb{R}} (e^{\mathrm{i}\xi y} - 1)\nu(y)dy = d \int_{\mathbb{R}} e^{\mathrm{i}\xi y}\nu(y)dy - d\frac{\sqrt{\pi}}{\varepsilon} \tag{122}$$

$$= \frac{d\sqrt{\pi}}{\varepsilon}\left(e^{-\frac{\xi^2}{4\varepsilon^2}} - 1\right) \tag{123}$$

Therefore, from

$$\hat{u}_t(\xi, t) = \eta(\xi)\hat{u}(\xi, 0) \tag{124}$$

we obtain

$$\hat{u}(\xi, t) = \exp(\eta(\xi)t)\hat{u}(\xi, 0) \tag{125}$$

The values of $u(x,t)$ at time $t = T$ can be evaluated through inverse Fourier transform

$$u(x_i, T) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{u}(\xi, T) e^{i\xi x_i} d\xi = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \exp(\eta(\xi)t)\hat{u}(\xi, 0) e^{i\xi x_i} d\xi \quad (126)$$

which can be evaluated using FFT for uniform grids x_i.

7.4 Singular and Slow Decaying Lévy Measure

Finally, we consider the case where $\nu(y)$ grows to infinity at $y = 0$ and has heavy tail. The case is quite challenging and extensively studied by the community nowadays. For simplicity, we will consider the specific case where $n(\mathbf{x}) = \frac{C_{d,s}}{|\mathbf{x}|^{d+2s}}$, i.e., the fractional Laplacian. Fortunately we can compute the analytical non-local derivative or gradient for some functions.

For the first example, we consider $u(x) = \exp(-x^2)$ in 1D. Then we have

$$(-\Delta)^s u(0) = 2^{2s}\Gamma\left(\frac{1+2s}{2}\right)/\sqrt{\pi} \quad (127)$$

For this example, since $u(x)$ decay to zero exponentially, we can assume that the far-field interaction $f_x^{L_W} = 0$ is zero for sufficiently large L_W. The numerical value is computed using eqs. (101) to (103) and compared with the exact value eq. (127). The parameters are: $L_W = 5.0$, $r = 0.2$. The convergence plot is shown in fig. 11. We can see that the error converges like or better than $\mathcal{O}(h^2)$.
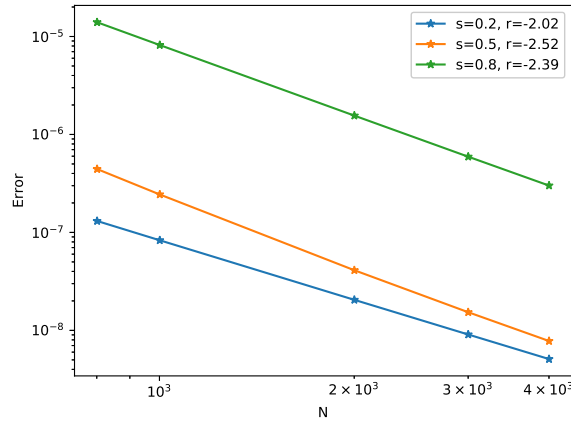

Figure 11: Numerical error for approximating eq. (127). The error converges like or better than $\mathcal{O}(h^2)$.

We also test the scheme on a challenging problem: the fractional Poisson problem. The PDE

$$\begin{cases} (-\Delta)^s u(x) = 1 & x \in [-1, 1] \\ u(x) = 0 & x \notin [-1, 1] \end{cases} \quad (128)$$

has a unique solution

$$u(x) = \frac{2^{2s}\Gamma(1+s)\Gamma\left(\frac{1+2s}{2}\right)}{\Gamma(1/2)}(1-x^2)^s \tag{129}$$

Note that $u(x)$ is not smooth across the boundary. In fact, it only belongs to $C^{0,s}([-1,1])$, the s-order Hölder space. Numerical algorithms usually exhibit reduced convergence. We use $L = 1.0$ and $Lw = 2.0$ so that the support of $u(x)$ is included in the near-field or local interaction. Thus we have $f_x^{Lw} = 0$. Since the current implementation only supports forward computation of the nonlocal operator, i.e., given function values, the nonlocal derivative or gradient is computed, we resort to a conjugate gradient approach for recovering $u(x)$ in $[-L, L]$.

Figure 12 presents the finite difference result obtained from our discretization. We can see that the convergence order is 1.0 or less, much worse than the Poisson problem where $\mathcal{O}(h^2)$ convergence rate is typical. We need to emphasize this is a universal problem faced by many fractional Laplacian models if a simple truncation trick is used.
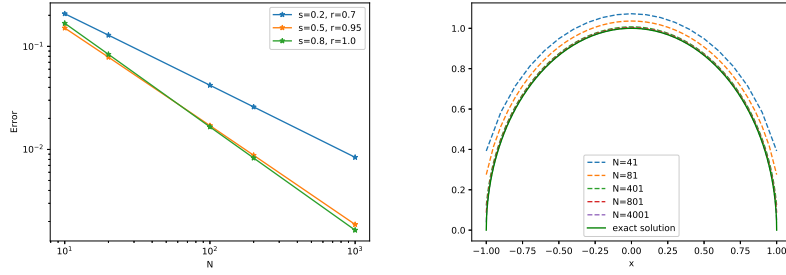

Figure 12: The finite difference result obtained from our discretization. The convergence order is 1.0 or less, much worse than the Poisson problem where $\mathcal{O}(h^2)$ convergence rate is typical

Finally, we also consider the computation of $(-\Delta)^u(\mathbf{x})$ in 2D, where

$$u(\mathbf{x}) = \frac{1}{2^{2s}\Gamma(1+s)^2}((1-|\mathbf{x}|^2)_+^s \tag{130}$$

The analytically result is known for $|\mathbf{x}| \leq 1$, which is

$$(-\Delta)^s u(\mathbf{x}) = 1 \quad |\mathbf{x}| \leq 1 \tag{131}$$

The numerical result is shown in fig. 13. Near the boundary, due to the nonsmoothness of $u(\mathbf{x})$, the algorithm is having a hard time computing the nonlocal gradient and therefore we see the oscillatory behavior. However, the computation for the region near the center is good, which does not suffer much from the far-away contribution from nonsmooth boundaries. In the center the error is only 4%. We used $L_W = 2.0$ and $L = 1.0$ in this case.
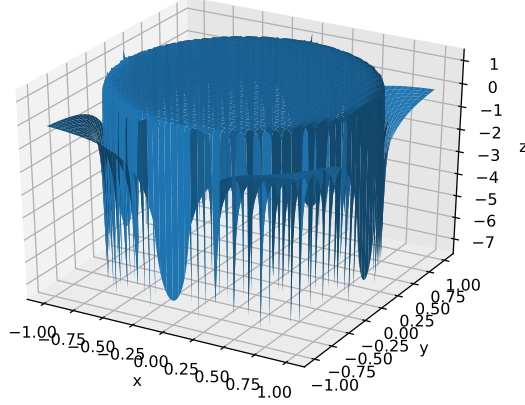
Figure 13: Numerical evaluation of eq. (131). Near the boundary, due to the nonsmoothness of $u(\mathbf{x})$, the algorithm is having a hard time computing the nonlocal gradient and therefore we see the oscillatory behavior. However, the computation for the region near the center is good, which does not suffer much from the far-away contribution from nonsmooth boundaries. In the center the error is only 4%.

These numerical examples demonstrates that the numerical scheme also works for $\nu(y)$ which has heavy tails.

8 Conclusion

In this paper, we presented the $\mathcal{H}$-matrix solver for the convection diffusion equation driven by the Lévy process. We consider both semi-heavy Lévy measure $\nu(y) < \infty$ as well as the challenging case $\nu(y) \to 0$, $y \to 0$, and $\nu(y)$ decays only algebraically. Particularly, when $\nu(y) = \frac{c_{1,s}}{|y|^{1+2s}}$, we recover the so-called fractional Laplacian operator $(-\Delta)^s u(x) = \text{r.v.} \int_{\mathbb{R}} (u(x+y) - u(x)) \nu(y) dy$. In the case $\nu(y)$ is smooth for large y, the corresponding coefficient matrices in the explicit or implicit scheme can be efficiently represented by $\mathcal{H}$-matrix. We implemented $\mathcal{H}$-LU and use it as a preconditioner or a direct solver for the convection diffusion equation. Numerical methods demonstrate that the $\mathcal{H}$-matrix is highly efficient compared to the dense matrices for these tasks.

The algorithms proposed in this paper can also be easily generalized to higher dimensions. To demonstrate, we also present the two dimensional cases in this paper, which also shows advantage over direct methods, especially for large scale problems.

The convection diffusion equation, or other counterparts driven by Lévy process is challenging due to the non-locality of the jump diffusion. This will lead to dense coefficients matrices which makes computation prohibitable for large scale problems. However, the main finding in this paper shows that by adopting

the well-established $\mathcal{H}$-matrix technique, large scale simulation becomes possible and efficient. It could be a first-hand method under researcher's consideration.