An efficient algorithm for the evaluation of certain convolution integrals with singular kernels

Shidong Jiang

1 Introduction

In this short note, we are concerned with the evaluation of certain convolution integrals with singular kernels. The problem is as follows. Given a kernel function $K(t)$ which is very often singular at the origin and a density function $\sigma(t)$, evaluate $C(t) = \int_0^t K(t-\tau)\sigma(\tau)d\tau$ for $t = \Delta t, 2\Delta t, \cdots, T = N\Delta t$. We will oftern write $C(t_k) = C(k\Delta t) = C_k$ in short, and similarly for σ. Direct computation of C_k requires the storage of all previous densities $\sigma_1, \cdots, \sigma_k$ and $O(k)$ flops at the kth step. Thus it requires on average $O(N)$ storage and flops for each time step and the total amount of flops is $O(N^2)$, which forms a bottleneck for long time simulations.

We present here an efficient algorithm for such problems. Our algorithm reduces the storage requirement from $O(N)$ to $O(\log(N))$ and the overall computational cost from $O(N^2)$ to $O(N\log(N))$. The basic idea is that in the region away from its singular point we may approximate the kernel K accurately by a sum of exponentials with the number of exponentials proportional to $\log(N)$, and the computation of the convolution with the exponential functions as the kernel can be sped up using a very simple recurrence relation.

The note is organized as follows. In Section 2, we present an outline of the algorithm. In Section 3, we present a numerical example with applications on the Havriliak-Negami model for dielectric medias.

2 Outline of the Algorithm

We break the integral into a sum of local part and history part, that is,

$$\begin{aligned} C(t) &= \int_0^t K(t-\tau)\sigma(\tau)d\tau \\ &= \int_{t-\Delta t}^t K(t-\tau)\sigma(\tau)d\tau + \int_0^{t-\Delta t} K(t-\tau)\sigma(\tau)d\tau \\ &:= C_l(t) + C_h(t), \end{aligned} \tag{1}$$

where the last equality defines the local part and the history part respectively.

2.1 Computation of the local part

Very often the convolution kernel K is singular at the origin with known singularity, i.e., $K(t) = \phi(t)S(t) + \psi(t)$ as $t \to 0$, where ϕ and ψ are smooth functions and $S(t)$ is a known singular function. We also observe that the kernel K can only be evaluated *in toto* in most cases.

To compute the local part $C_l(t) = \int_{t-\Delta t}^{t} K(t-\tau)\sigma(\tau)d\tau$, we first make a simple change of variable $s = t - \tau$ to convert the integral as follows:

$$C_l(t) = \int_0^{\Delta t} K(s)\sigma(t-s)ds. \tag{2}$$

Making a further change of variable $s = \Delta t u$, we obtain

$$C_l(t) = \Delta t \int_0^1 K(\Delta t u)\sigma(t - \Delta t u)du. \tag{3}$$

Next, we apply the generalized Gaussian quadrature algorithm developed by Rokhlin et al. (see, for example, [6, 7, 8]) to obtain a two point quadrature which can integrate four functions 1, t, $S(t)$, $S(t)t$ exactly on the interval $[0, 1]$. That is

$$\int_0^1 f_k(x)dx = \sum_{i=1}^{2} w_i f_k(x_i) \quad \text{for } k = 1, \ldots, 4, \tag{4}$$

where $\{f_k\} = \{1, x, S(x), S(x)x\}$. Applying the above quadrature to compute the local part, we have

$$C_l(t) = \Delta t(w_1 K(x_1\Delta t)\sigma(t - x_1\Delta t) + w_2 K(x_2\Delta t)\sigma(t - x_2\Delta t)). \tag{5}$$

The density σ is often specified at $t = k\Delta t$ for $k = 0, 1, \cdots, N$. We may use linear interpolation to obtain that

$$\sigma(t - x_i\Delta t) = x_i\sigma(t - \Delta t) + (1 - x_i)\sigma(t), \quad t = 1, 2. \tag{6}$$

Substituting (6) into (5) and simplifying, we obtain

$$\begin{aligned} C_l(t) = \Delta t\{&[w_1 x_1 K(x_1\Delta t) + w_2 x_2 K(x_2\Delta t)]\sigma(t - \Delta t) \\ &+ [w_1(1 - x_1)K(x_1\Delta t) + w_2(1 - x_2)K(x_2\Delta t)]\sigma(t)\}. \end{aligned} \tag{7}$$

Specifically, for $t = k\Delta t$, we have

$$\begin{aligned} C_l(k\Delta t) = \Delta t\{&[w_1 x_1 K(x_1\Delta t) + w_2 x_2 K(x_2\Delta t)]\sigma_{k-1} \\ &+ [w_1(1 - x_1)K(x_1\Delta t) + w_2(1 - x_2)K(x_2\Delta t)]\sigma_k\}. \end{aligned} \tag{8}$$

2.2 Computation of the history part

We first approximate the kernel K on the interval $[\Delta t, T]$ by a sum of exponentials, i.e.,

$$|K(t) - \sum_{i=1}^{L} w_i e^{-s_i t}| \le \epsilon, \quad t \in [\Delta t, T], \tag{9}$$

where ϵ is a prescribed precision. For many kernels of practical interest, it has been shown that the number of exponentials needed is $O(\log N \log \epsilon)$, where $N = T/\Delta t$ is the number of time steps in the simulation (see, for example, [9, 10]). The kernel K is a causal function, which requires the sum-of-exponentials approximation have $\mathrm{Re} s_i > 0$, that is, all exponentials are decaying for $t > 0$.

There are several ways to obtain such approximations numerically for various circumstances. First, one may apply the algorithm for finding sum-of-exponentials approximation developed by Beylkin et al. (see, for example, [9, 10]) if the kernel is available directly. Second, if the kernel can be expressed as a contour integral involving exponential functions, then one may utilize the generalized Gaussian quadrature algorithm developed by Rokhlin et al. (see, for example, [6, 7, 8]). Third, in some cases, the kernel is known indirectly through its Laplace transform, one may find a sum-of-poles approximation for the Laplace transform of kernel using the bootstrapping method developed by Jiang et al. (see, for example, [5]), and the sum-of-exponentials approximation for the kernel is easily obtained due to the fact that the inverse Laplace transform of a pole function is an exponential function.

Once such approximation for K is obtained, we can then approximate the history part as follows:

$$\begin{aligned} C_h(t) &= \int_0^{t-\Delta t} K(t-\tau)\sigma(\tau)d\tau \\ &\approx \sum_{i=1}^{L} w_i \int_)^{t-\Delta t} e^{-s_i(t-\tau)}\sigma(\tau)d\tau \\ &:= \sum_{i=1}^{L} w_i C_{hi}(t) \end{aligned} \tag{10}$$

To evaluate $C_{hi}(k\Delta t)$ for $k = 1, 2, \cdots, N$, we observe the following simple recurrence relation:

$$C_{hi}(k\Delta t) = e^{-s_j \Delta t} C_{hi}((k-1)\Delta t) + \int_{(k-2)\Delta t}^{(k-1)\Delta t} e^{-s_i(k\Delta t - \tau)} \sigma(\tau) d\tau. \quad (11)$$

At each time step, we only need $O(1)$ work to compute $C_{hi}(k\Delta t)$ since $C_{hi}((k-1)\Delta t)$ is known at that point. Thus, the total work is reduced from $O(N^2)$ to $O(NL) = O(N \log N)$, and the total memory requirement is reduced from $O(N)$ to $O(L) = O(\log N)$.

One may compute the integral on the right hand side of (11) by interpolating σ via a linear function and then evaluating the resulting approximation analytically. We have

$$\begin{aligned} \int_{(k-2)\Delta t}^{(k-1)\Delta t} e^{-s_i(k\Delta t - \tau)} \sigma(\tau) d\tau = \frac{e^{-s_i \Delta t}}{s_i^2 \Delta t} \big[& (e^{-s_i \Delta t} - 1 + s_i \Delta t) \sigma_{k-1} \\ & + (1 - e^{-s_i \Delta t} - e^{-s_i \Delta t} s_i \Delta t) \sigma_{k-2} \big] \end{aligned} \quad (12)$$

We note that the right hand side of (12) is subject to significant cancellation error when $s_i \Delta t$ is small. In that case, we can compute the weights by a Taylor expansion of exponentials with a sufficient number of terms.

Finally, another popular fast method for computing the convolution with exponential functions is to solve an initial value problem for an ordinary differential equation. We would like to point out that in our case this may force one to choose a very small time step Δt for the overall scheme. This is because s_i $(i = 1, 2, \cdots, L)$ usually varies in orders of different magnitudes and the resulting ODE system will be very stiff. Thus we prefer to evaluate the convolution via the simple recurrence relation (11).

3 Application to the Havriliak-Negami model for dielectric media

For the Havriliak-Negame model, the convolution kernel is given by the formula

$$K(t) = \mathcal{L}^{-1}\{F(s)\}, \quad (13)$$

where

$$F(s) = \frac{1}{(1 + s^{\alpha})^{\beta}}. \quad (14)$$

The asymptotic analysis shows that (see, for example, [1])

$$K(t) \sim \frac{1}{t^{1-\alpha\beta}}, \quad t \to 0^+ \tag{15}$$

and

$$K(t) \sim \frac{1}{t^{1+\alpha}}, \quad t \to \infty. \tag{16}$$

Applying Mellin's inversion formula and choosing the Bromwich contour along the negative real axis, we obtain

$$\begin{aligned} K(t) &= \frac{1}{2\pi i}\int_{)}^{\infty} e^{-rt}\left\{\frac{1}{(1+r^\alpha e^{-i\alpha\pi})^\beta} - \frac{1}{(1+r^\alpha e^{i\alpha\pi})^\beta}\right\} dr \\ &= \int_0^\infty \frac{\sin\left(\beta \arctan \frac{r^\alpha \sin(\pi\alpha)}{1+r^\alpha\cos(\pi\alpha)}\right)}{\pi(1+r^{2\alpha}+2r^\alpha\cos(\pi\alpha))^{\beta/2}} e^{-rt} dr \\ &= \int_0^{\frac{\pi}{2}} \frac{\sin\left(\beta \arctan \frac{(\tan x)^\alpha \sin(\pi\alpha)}{1+(\tan x)^\alpha\cos(\pi\alpha)}\right)}{\pi(1+(\tan x)^{2\alpha}+2(\tan x)^\alpha\cos(\pi\alpha))^{\beta/2}} e^{-(\tan x)t}(1+\tan^2 x)dx, \end{aligned} \tag{17}$$

where the last equality follows from a simple change of variable $r = \tan x$. The above integral representation represents the kernel K as a continuous sum (i.e., an integral) of exponential functions of t. To obtain an efficient sum-of-exponentials approximation for K which is accurate for all t at a specified interval $[\Delta t, T]$, we apply the generalized Gaussian quadrature for the last integral in (17).

We will use the particular case $\alpha = 0.7$, $\beta = 1.0$ to illustrate our method.

3.1 The local part

From (15), we know that

$$K(t) \sim \frac{1}{t^{1-\alpha\beta}} = \frac{1}{t^{0.3}}, \quad t \to 0^+ \tag{18}$$

Applying the algorithm in [8], we found that

$$\int_0^1 f_i(x)dx = w_1 f_i(x_1) + w_2 f_i(x_2), \tag{19}$$

for $f_1(x) = 1,\ f_2(x) = x,\ f_3(x) = 1/x^{0.3},\ f_4(x) = x^{0.7}$. The weights and nodes are as follows:

$$x_1 = 0.62805956324192793727D{-}01, \qquad x_2 = 0.64564523226253778265D{+}00,$$
$$w_1 = 0.24988918605057447442D{+}00, \qquad w_2 = 0.75011081394943734946D{+}00.$$

Next, we compute $K(x_1\Delta t)$ and $K(x_2\Delta t)$ using a high-order adaptive integrator for the last integral in (17) (note the integrand is a bounded continuous function there). For example, for $\Delta t = 5D - 4$, we have

$$K(x_1\Delta t) = 17.2650549595034, \qquad K(x_2\Delta t) = 8.54538021524574$$

Substituting the above values into (8), we obtain

$$\begin{aligned} C_l(k\Delta t) &= \int_{(k-1)\Delta t}^{k\Delta t} K(k\Delta t - \tau)\sigma(\tau)d\tau \\ &\approx \Delta t\{[w_1 x_1 K(x_1\Delta t) + w_2 x_2 K(x_2\Delta t)]\sigma_{k-1} \\ &+ [w_1(1-x_1)K(x_1\Delta t) + w_2(1-x_2)K(x_2\Delta t)]\sigma_k\} \\ &\approx c_1\sigma((k-1)\Delta t) + c_2\sigma(k\Delta t), \end{aligned} \tag{20}$$

where $c_1 = 2.204770649213286D - 3$ and $c2 = 3.157395670643779D - 3$. In general the quadrature of the above type is accurate to $O(\Delta t^{2+\alpha\beta})$.

3.2 The history part

We apply the algorithm in [8] to find an accurate sum-of-exponentials approximation for K. For example, for t from $\Delta t = 5D - 4$ to $T = 300$, we find that we need $L = 43$ exponentials to achieve 9-digit accuracy, i.e.,

$$|K(t) - \sum_{i=1}^{43} w_i e^{-s_i t}| < 1D - 9, \quad t \in [5D - 4, 300]. \tag{21}$$

The weights w_i and the nodes s_i are as follows:

```
c quadrature weights w_i:
        data ws1/
     1  0.14095906806379888691D-04,0.71639740006113443991D-04,
     2  0.20177616828336396976D-03,0.45218478351158818921D-03,
     3  0.90887890410196842898D-03,0.17316437476012753148D-02,
     4  0.32181506328802116958D-02,0.59116532654059760626D-02,
     5  0.10766304860454649350D-01,0.19350897765653039817D-01,
```

```
     6  0.33838525521340617752D-01,0.56038304322703999527D-01,
     7  0.85055634739979457670D-01,0.11579761659297206633D+00,
     8  0.14186112344124510165D+00,0.16037095396368225564D+00,
     9  0.17270485834058918462D+00,0.18178021582581138649D+00,
     *  0.18996924857098801409D+00,0.19863318098534249456D+00,
     1  0.20840236049096691473D+00,0.21948826574723789840D+00,
     2  0.23180147705241960221D+00,0.24507939511011400668D+00,
     3  0.25914471228278596060D+00,0.27404984713960772735D+00,
     4  0.29009339974692277542D+00,0.30788749214252342234D+00,
     5  0.32832326854495585966D+00,0.35222362332239287586D+00,
     6  0.37992379104411305724D+00,0.41121245062310646246D+00,
     7  0.44568969101465438776D+00,0.48320674785351308556D+00,
     8  0.52410035930693221751D+00,0.56916594214346727387D+00,
     9  0.61966397777602333363D+00,0.67806714497007403164D+00,
     *  0.74989976658543089361D+00,0.84706102435500740810D+00,
     1  0.99706951591060244411D+00,0.12829854136005385268D+01,
     *  0.21440211171375991306D+01/
c quadrature nodes s_i:
        data xs1/
     1  0.21391998879852340033D-02,0.80204982827138324164D-02,
     2  0.17975940090626967016D-01,0.32658729310982348437D-01,
     3  0.53126419542560657239D-01,0.81004264693796207419D-01,
     4  0.11870691779441321090D+00,0.16969577580708028464D+00,
     5  0.23877057966689074764D+00,0.33251119814344876069D+00,
     6  0.45997507398536019441D+00,0.63354832002280392356D+00,
     7  0.87020015051697130470D+00,0.11939003412105173574D+01,
     8  0.16390813911051753671D+01,0.22542088698583353512D+01,
     9  0.31057740510007003643D+01,0.42842196294990406713D+01,
     *  0.59126165534315227035D+01,0.81579257738486976592D+01,
     1  0.11244915990875989920D+02,0.15473372784785480505D+02,
     2  0.21237659525457804222D+02,0.29046139978520521652D+02,
     3  0.39543555163282846365D+02,0.53545184488233154241D+02,
     4  0.72090442712454901653D+02,0.96530260462681695799D+02,
     5  0.12867220939356602116D+03,0.17098908507052237837D+03,
     6  0.22686115302565954721D+03,0.30082471286187700343D+03,
     7  0.39885485290103991929D+03,0.52876868485676675391D+03,
     8  0.70085789484127781179D+03,0.92885703535043728607D+03,
     9  0.12313829767469908347D+04,0.16343323222105084369D+04,
     *  0.21757881390405691491D+04,0.29172372647376669192D+04,
     1  0.39720563838309517450D+04,0.55977350387814594797D+04,
```

```
     *  0.86822734669851506624D+04/
```

Finally, we would like to remark that the quadrature for the local part depends on α, β, the weights and nodes of the sum-of-exponentials approximation for the history part depend on α, β, Δt and T. However, this is really a precomputation step and we only need to do it once for each pair of α, β.

References

[1] P. Petropolous, *A Numerical Scheme to Simulate Electromagnetic Pulses Propagating in Cole-Cole Dielectrics*,

[2] S. Jiang and L. Greengard, *Efficient representation of nonreflecting boundary conditions for the time-dependent Schrdinger equation in two dimensions.* Comm. Pure Appl. Math. **61** (2008), no. 2, 261–288.

[3] S. Jiang and L. Greengard, *Fast evaluation of nonreflecting boundary conditions for the Schrdinger equation in one dimension.* Comput. Math. Appl. **47** (2004), no. 6-7, 955–966.

[4] S. Jiang, *Fast evaluation of the nonreflecting boundary conditions for the Schrodinger equation*, Ph.D. Thesis, New York University, New York, 2001.

[5] B. Alpert, S. Jiang, K. Xu, *A bootstrap method for rational approximations*, in preparation.

[6] H. Cheng, V. Rokhlin, and N. Yarvin, *Nonlinear optimization, quadrature, and interpolation.* SIAM J. Optim. **9** (1999), no. 4, 901–923.

[7] N. Yarvin, V. Rokhlin, *Generalized Gaussian quadratures and singular value decompositions of integral operators.* SIAM J. Sci. Comput. **20** (1998), no. 2, 699–718.

[8] J. Ma, V. Rokhlin, and S. Wandzura, *Generalized Gaussian quadrature rules for systems of arbitrary functions.* SIAM J. Numer. Anal. **33** (1996), no. 3, 971–996.

[9] G. Beylkin and L. Monzn, *On approximation of functions by exponential sums.* Appl. Comput. Harmon. Anal. **19** (2005), no. 1, 17–48.

[10] G. Beylkin and L. Monzn, *On generalized Gaussian quadratures for exponentials and their applications*. Appl. Comput. Harmon. Anal. **12** (2002), no. 3, 332–373.