# Graphical Abstract

**On Trigonometric Interpolation and Its Applications II**

Xiaorong Zou

# Highlights

**On Trigonometric Interpolation and Its Applications II**

Xiaorong Zou

- Introduce a new trigonometric interpolation algorithm that can be carried out by Fast Fourier Transform for optional performance; establish relevant properties, especially showing that it has desired convergence rate for both interpolant and high order derivatives of interpolant, which provides theoretic support for its applications. (Part I);
- Extend the algorithm so it can be used to non-periodic function (Part II);
- Study numerical performance of the algorithm (Part II);
- Study applications of the algorithm, including to estimate integrals and solve non-linear ordinary differential equation (ODE). Test results show that it outperforms Trapezoid/Simpson method to cope with integral and standard Runge-Kutta algorithm in handling ODE. (Part II)

# On Trigonometric Interpolation and Its Applications II

Xiaorong Zou

*Global Market Risk Analytics, Bank of America, 1114 6th Ave, 7th floor, New York City, 10110, New York, USA*

**Abstract**

This is the second part of our study on trigonometric Interpolation and its applications with a focus on algorithm enhancement and related applications. We enhance the algorithm established in the first part so it can be applied to non-periodic functions defined on bounded intervals. For its applications, we demonstrate how it can be applied to estimate integrals and solve non-linear ordinary differential equations (ODEs). The numerical experiments show that it outperforms Trapezoid/Simpson rules to cope with integral and standard Runge-Kutta algorithm in handling ODE. In addition, we show some numerical evidences that the estimation error of the algorithm likely exhibits "local property", i.e. an estimation error at a grid point does not propagate and produce significant compounding error outside its neighborhood, a remarkable advantage compared to polynomial-based approximations.

*Keywords:* Trigonometric Interpolation, Fast Fourier Transformation (FFT), Ordinary Differential Equation, Runge-Kutta method, Trapezoid rule, Simpson's rule.
*2008 MSC:* Primary 42A10, 65T40; Secondary 65L05

## 5. Trigonometric Estimation of General Functions

This section is used to develop a trigonometric interpolation algorithm that can be applied to a non-periodic function $f$ over a bounded interval $[s, e]$. As such, $f(x)$ shall denote a function whose $K+1$-th derivative $f^{(K+1)}(x)$ exists and is bounded for some $K \geq 1$. Note that $f(x)$ need not be periodic in this paper.

One can shift $f(x)$ by $s$ and then evenly extend it to $[s-e, e-s]$. Such direct extension deteriorates the smoothness at $0, \pm(e-s)$, and leads to a

poor convergence performance as showed in Section 6. To seek for a smooth periodic extension, we assume that $f$ can be extended smoothly such that $f^{(K+1)}$ exists and is bounded over $[s-\delta, e+\delta]$ for certain $\delta > 0$, and then multiply $f$ by a cut-off smooth function $h(x)$ with following property:

$$h(x) = \begin{cases} 1 & x \in [s, e], \\ 0 & x < s-\delta \text{ or } x > e+\delta. \end{cases}$$

Such cut-off functions can be constructed in different ways and we shall adopt one with a closed-form analytic expression as follows:

$$h(x; r, s, e, \delta) = B(\frac{x-(s-\delta)}{\delta}, r) \times B(\frac{e+\delta-x}{\delta}, r), \tag{1}$$

where

$$B(x; r) = \frac{G(x; r)}{G(x; r) + G(1-x; r)}, \quad G(x; r) = \begin{cases} e^{-\frac{r}{x^2}} & x > 0, \\ 0 & x \leq 0. \end{cases}$$

To see the effect of parameter $r$, Figure 1 plots left wing of the cut-off functions with three scenarios $r \in \{0.1, 0.5, 1\}$ for $(s, e, \delta) = (-1, 1, 1)$. As $r$ increases from 0.1 to 1, $h$ takes more space in $x - axis$ to increase around 0 to near 1 and therefore change is less rapidly over the process, which is preferable. On the other hand, smaller $r$ provides more rooms for $h(x)$ converges to 0 and 1, which is also preferable for the performance of $h$ around $s$ and $s-\delta$.

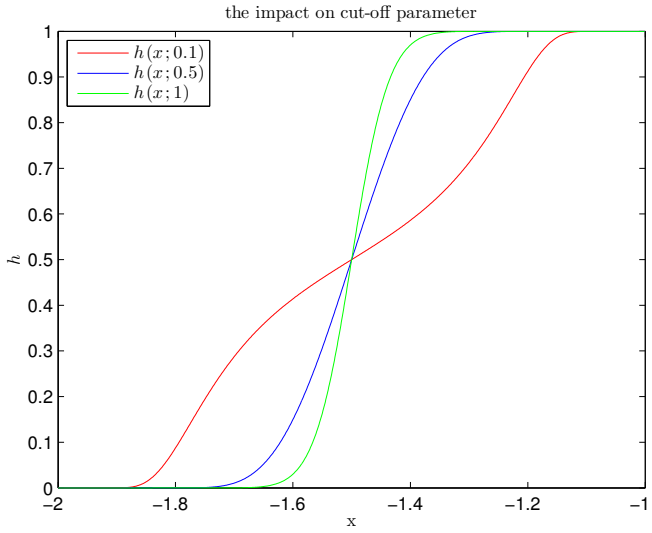


Figure 1: The graphs of $h_M$ over $[s-\delta, s]$ with $(s, e, \delta) = (-1, 1, 1)$ and $M = 2^6$ for three cases: $r = 0.1$ (red), $r = 0.5$ (blue) and $r = 1$ (green).

The trigonometric interpolation of $h$ can be useful in applications, and Table 1 shows max error of the interpolation in 18 scenarios with the combination of 4 parameters $s = -1, e = 1, \delta \in \{0.5, 1\}, q \in \{6, 7, 8\}, r \in \{0.1, 0.5, 1\}$. Max Error is negligible in the case of $\delta = 1$, and becomes visibly away from 0 with $\delta = 0.5$, which is expected since a larger $\delta$ provides more distance for $h(x)$ to move from 1 to 0. The test result suggests that $r = 0.5$ is a good option and will be used on all numerical experiments conducted in rest of the paper.

With a cut-off function $h(x)$, $h(x)f(x)$ can be smoothly extended to $[2s - e - 3\delta, s - \delta]$ symmetrically with respect to vertical line $x = s - \delta$. The idea is demonstrated in Figure 2, where $f(x) = (x - 2.5)^2$ for $x \in [2, 3]$ with $(s, e, \delta) = 2, 3, 1$.

Table 1: Max differences between $h_M$ and $h$ where $h_M$ is the trigonometric estimation of $h$ with $M = 2^q$ grid points. The 18 test scenarios include the combination of 4 parameters $s = -1, e = 1, \delta \in \{0.5, 1\}, q \in \{6, 7, 8\}, r \in \{0.1, 0.5, 1\}$.

| $\delta$ | r | q | Max Error | $\delta$ | r | q | Max Error |
|---|---|---|---|---|---|---|---|
| 0.5 | 0.1 | 6 | 3.2E-03 | 1 | 0.1 | 6 | 8.9E-16 |
| 0.5 | 0.5 | 6 | 4.6E-04 | 1 | 0.5 | 6 | 4.4E-16 |
| 0.5 | 1 | 6 | 8.3E-03 | 1 | 1 | 6 | 5.6E-16 |
| 0.5 | 0.1 | 7 | 1.4E-04 | 1 | 0.1 | 7 | 8.9E-16 |
| 0.5 | 0.5 | 7 | 1.6E-06 | 1 | 0.5 | 7 | 1.3E-15 |
| 0.5 | 1 | 7 | 2.2E-04 | 1 | 1 | 7 | 7.8E-16 |
| 0.5 | 0.1 | 8 | 1.0E-06 | 1 | 0.1 | 8 | 1.3E-15 |
| 0.5 | 0.5 | 8 | 2.6E-10 | 1 | 0.5 | 8 | 1.1E-15 |
| 0.5 | 1 | 8 | 1.7E-07 | 1 | 1 | 8 | 1.1E-15 |

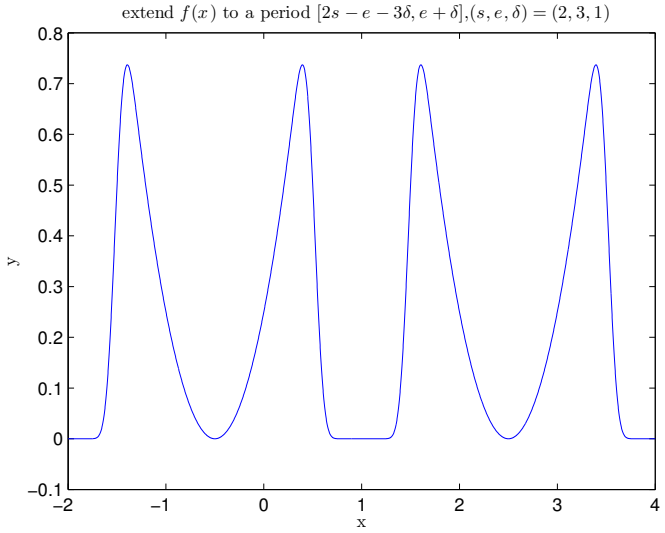


Figure 2: The graphs of the $fh$'s extension $(hf)_{ext}$ over a period $[2s - e - 3\delta, e + \delta]$ with $(s, e, \delta) = (2, 3, 1)$. The function is even after parallel shift left by $s - \delta = 1$. Note that $fh \equiv f$ over $[s, e]$ as expected.

The periodic extension can be summarized as follows.

**Algorithm 1.** *Let $f(x)$ be defined over $[s, e]$.*

1. *Select two integers $0 < p < q$ such that $f(x)$ can be smoothly extended*

*to* $[s-\delta, e+\delta]$*, where*

$$\begin{aligned} n &= 2^p, \quad M = 2^q, \quad \lambda = \frac{e-s}{n}, \\ m &= \frac{M-n}{2}, \qquad \delta = m\lambda. \end{aligned}$$

2. *Construct the cut-off function* $h(x)$ *with parameter* $(s, e, \delta, r = 0.5)$.
3. *Let*
$$o = s - \delta, \quad b = e + \delta - o, \tag{2}$$
*and define* $F(x) := h(x+o)f(x+o)$ *for* $x \in [0, b]$.
4. *Extend* $F(x)$ *evenly by* $F(x) = F(-x)$ *for* $x \in [-b, 0]$ [1]*. It is clear that* $F(x)$ *can be treated as an periodic even function.*
5. *Define grid points by*
$$x_j = -b + j\lambda, \quad j = 0, 1, \cdots N-1, \quad N = 2M,$$
*and apply them to construct trigonometric expansion* $F_M$ *by Theorem 2* [2]
$$F_M(x) = \sum_{0 \le j < M} a_j \cos \frac{j\pi x}{b}.$$
6. *The target periodic interpolation of* $f$ *can be formulated by*
$$\hat{f}_M(x) = F_M(x-o) = \sum_{0 \le j < M} a_j \cos \frac{j\pi(x-o)}{b}. \tag{3}$$

$\hat{f}$ will be used to denote the extended periodic function by Algorithm 1 and $\hat{f}_M$ be the interpolant of $\hat{f}$ by Algorithm 1 in the rest of this paper. Clearly, $\hat{f}$ inherits same smoothness as $f(x)$ does. We shall discuss the performance of Algorithm 1 and its applications in Section 6 and 7 respectively.

Figure 3 compares $f = (x-2.5)^2$ and $\hat{f}_M$ over $[s-\delta, e+\delta]$ with $[s, e] = [2, 3]$. Note that $\hat{f}_M$ recovers $f$ over $[s, e]$ and approaches to 0 around boundary points $s-\delta, e+\delta$ as expected.

[1] Alternatively, extend $F(x)$ oddly by $F(x) = -F(-x)$ for $x \in [-b, 0]$ if odd trigonometric estimation is desired.

[2] Alternatively, $F(x) = \sum_{1 \le j < M} a_j \sin \frac{j\pi x}{b}$ if odd trigonometric interpolant is desired.

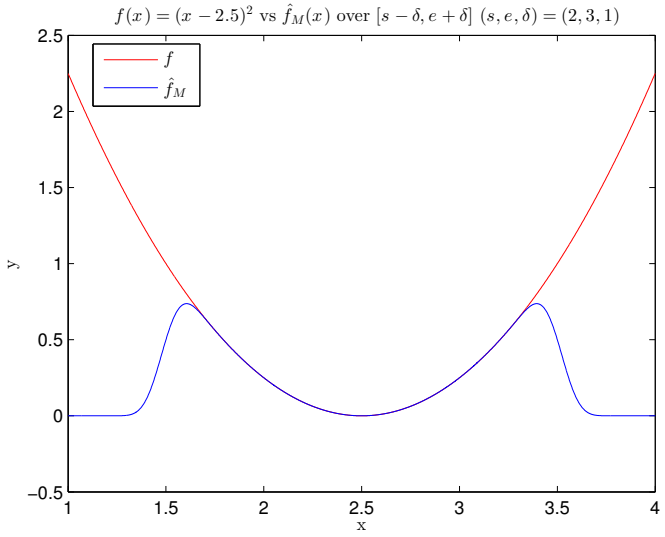


Figure 3: The graphs of $f(x)$ vs $\hat{f}_M(x)$ over $[e - \delta, s + \delta]$ with $(s, e, p, q, \delta) = (2, 3, 7, 8, 1)$. The figure is plotted by $2^{12}$ sample points.

## 6. Numerical Performance

This section provides some numerical results to test the performance of $\hat{f}_M(x)$ defined By Eq. (3). First, we test the convergence performance of Algorithm 1, and show that it is sensitive to the smoothness of the underlying function $f$ as expected. Secondly, we apply the algorithm to two sets of functions whose values can be highly oscillated and changed rapidly, and show that the algorithm exhibits stable and accurate performance.

### *6.1. Numerical Performance on Periodic Functions*

Let $f$ be the even periodic function with period $2\pi$ and defined as follows over $[-\pi, \pi]$

$$f(x; d)|_{[-\pi,\pi]} = (1 - (\frac{x}{\pi})^2)^d,$$

where $d$ can be $1, 2$. It is clear that $f(x; 1)$ is not differentiable at $\pm\pi$ and $f(x; 2)$ is 2-th continuous differentiable. We expect that the interpolant of $f_M(x; 2)$ has significant better performance than $f_M(x; 1)$'s.

Table 2 provides max errors under various settings on grid points. It confirms that performance is sensitive to the number of grid points and especially degree of $f$'s smoothness as expected.

Table 2: The max errors $|f_M(x) - f(x)|$ and $|f'_M(x) - f'(x)|$

| $d$ | $M$ | $\max(\lvert f - f_M \rvert)$ | $\max(\lvert f' - f'_M \rvert)$ |
|---|---|---|---|
| 1 | 64 | 6.0E-03 | 6.4E-01 |
| 1 | 256 | 1.5E-03 | 6.4E-01 |
| 1 | 1024 | 3.6E-04 | 6.4E-01 |
| 2 | 64 | 5.6E-07 | 3.5E-05 |
| 2 | 256 | 8.3E-09 | 2.2E-06 |
| 2 | 1024 | 1.3E-10 | 1.4E-07 |

*6.2. Numerical Performance on General Functions*

Let $\hat{f}$ be the smooth extension described in Section 5 and $\hat{f}_M$ be the trigonometric estimation of $\hat{f}$ by Algorithm 1.

We apply Algorithm 1 to the following functions to test its convergence performance.

$$\begin{aligned} y &= \cos\theta x, \quad \theta = 1, 10, 100, \\ y &= x^n, \qquad n = 4, 8, 10. \end{aligned}$$

For any estimation $\tilde{w}(x)$ of a function $w(x)$, define max error in log space over a subset $S$ of the domain of $w$ by

$$E(w) = \max_{x \in S} \{log_{10} |\tilde{w}(x) - w(x)|\}.$$

Let $f_M$ be the trigonometric interpolant of the direct periodic extension of $f$ without using cut-off function. Table 3 shows the max errors up to second derivatives of both $f_M$ and $\hat{f}_M$ for the two functions with the parameters $(s, e, p, q, \delta) = (-1, 1, 7, 8, 1)$, and the subset $S = \{s + k\frac{e-s}{2^{12}},\ 0 \le k \le 2^{12}\}$. Note that we deliberately make $S$ larger than the set of grid points used in Theorem 2 to test that $\hat{f}_M$ converges at non-grid points. The performance of $\hat{f}_M$ is stable across all the test scenarios and max estimation errors are negligible while $f_M(x)$ generates significant errors, especially on derivatives, which confirms the impact of smoothness on the convergence performance of Algorithm 1 as expected.

Table 3: Max error in log space (EL) with parameters $(s, e, p, q, \delta) = (-1, 1, 7, 8, 1)$.

| *para* | $EL(f_M)$ | $EL(\hat{f}_M)$ | $EL(f'_M)$ | $EL(\hat{f}'_M)$ | $EL(f''_M)$ | $EL(\hat{f}''_M)$ |
|---|---|---|---|---|---|---|
| | | | $f = \cos\theta x$ | | | |
| $\theta = 1$ | -3.2 | -14.7 | 0.0 | -13.1 | 2.8 | -10.7 |
| $\theta = 10$ | -2.4 | -14.8 | -0.3 | -14.2 | 1.6 | -11.8 |
| $\theta = 100$ | -1.4 | -14.0 | -0.3 | -14.0 | 0.6 | -11.9 |
| | | | $f = x^n$ | | | |
| $n = 4$ | 0.4 | -14.8 | 0.0 | -13.6 | 2.4 | -11.1 |
| $n = 8$ | 0.3 | -14.3 | 0.0 | -13.1 | 2.0 | -10.6 |
| $n = 10$ | 0.3 | -14.0 | 0.0 | -12.9 | 1.9 | -10.4 |

*6.3. Discussion on Error behavior of $\hat{f}_M(x)$*

The error analysis in Section 3 is based upon two identities Eqs. (23-24) of [21], which suggests certain feature of trigonometric polynomials, i.e. its estimation error tends to be small due to certain cancellation when target function is smooth. If further assuming that $f$ bears some symmetry (even or odd) and the set of grid points is selected by equispaced grid points, then performance with high accuracy rate is expected since cancellation effect becomes more propounded as evidenced by subsequently derived identities Eqs. (25-26) and (29-30) of [21], which plays essential roles in the error analysis.

It is hard, if possible, to fully exploit this feature in theoretic analysis. We provide some supplementary numerical evidence by looking into error behaviors of $\hat{f}_M(x)$ in this subsection. Intuitively, as mentioned above, the errors should exhibit "local property", i.e. error at a point should not propagate and cause large compounding error outside its neighborhood. This is not the case when polynomial based method is used as shown in Section 7.2. We conduct relevant tests on four basic types of functions including power, exponential, sin and cos functions. Figure 6.3 shows the normalized differences of consecutive errors defined by

$$\frac{1}{\max|f(x)|}\{\hat{f}_M(x_i) - f(x_i) - (\hat{f}_M(x_{i-1}) - f(x_{i-1}))\}.$$

A clear sawtooth pattern is shown in Figure 6.3 for all test cases. The magnitude of errors keep reasonable stable, but their values move in alternative directions, a strong sign that error is not accumulating, but canceled with each other. Same phenomenon has been observed on the estimation error

of the numerical solution for a non-linear ODE explained in Remark 2. As such, the real performance of the trigonometric interpolant $\hat{f}(x)$ is likely better than what is proved in Section 2.

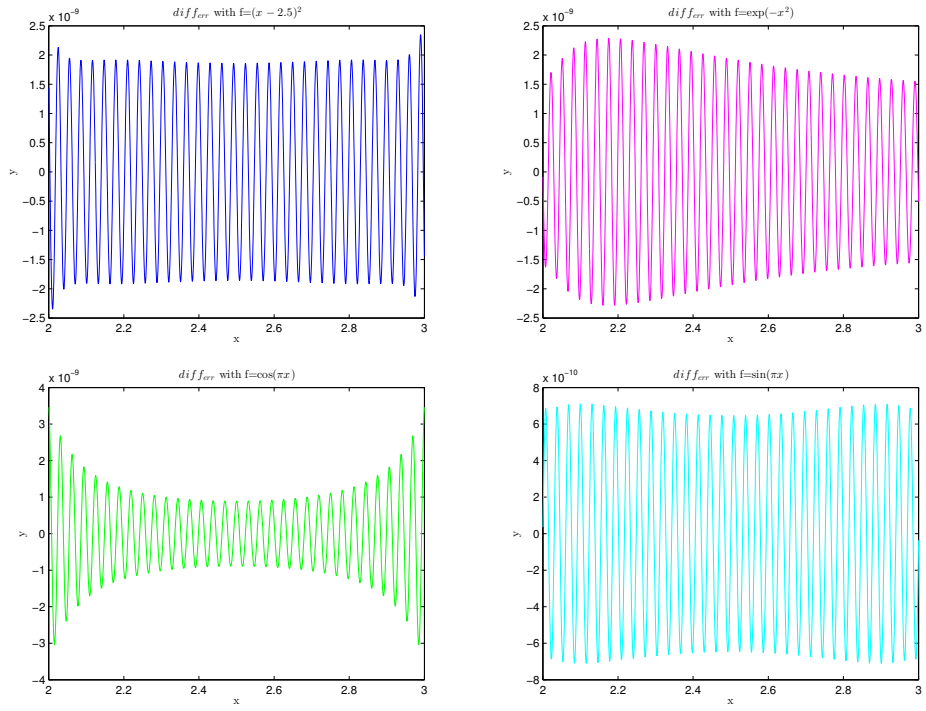


Figure 4: Plots of the normalized difference of consecutive error at grid points with $\lambda = 1/2^6$ with $[s, e] = [2, 3]$

## 7. Applications

The trigonometric estimation $\hat{f}_M(x)$ of a general function $f(x)$ by Algorithm 1 can be used to develop some numerical computational algorithms in various subjects that we shall discuss in subsequent papers mentioned in Section 1. In this section, we provide two examples. First, $\hat{f}_M$ is used to solve an integral with a general integrand. The test results show that the new method exhibits stable performance and outperforms some popular algorithms such as Trapezoid and Simpson methods, especially when the integrands are highly oscillated. Secondly, we demonstrate how the algorithm can be effectively used to solve a non-linear ODE of first order.

The notations in Section 5 will be adapted in this Section.

*7.1. Numerical Solution of a General Integral*

Let $f$ be a $K+1$ differentiable function over $[s-\delta, e+\delta]$ with $\delta > 0$. We aim to solve an integral $\int_s^e f(u)du$. Applying the approximation $\hat{f}$ described in Algorithm 1, the integral can be estimated by

$$\int_s^e f(u)du \approx a_0(e-s) + \sum_{1 \le j < M} \frac{ba_j}{j\pi}(\sin\frac{j\pi(e-o)}{b} - \sin\frac{j\pi(s-o)}{b}), \quad (4)$$

where $o, b$ are defined in Eq. (2).

To see the performance, we conduct two sets of tests:

$$\int_{-1}^{1} x^n dx, \qquad \int_{-1}^{1} \cos\theta x dx.$$

For each test, we use the max error defined in Section 6.2 to compare the performance of three methods: Estimation (4), Trapezoid as well as Simpson, and display the results in Table 4 and 5. One can see that Simpson

Table 4: The max error in log space with the integrand $f(x) = x^n$ associated to three methods: Estimation Eq (4), Trapezoid, and Simpson. $(s, e, p, q, \delta) = (-1, 1, 7, 8, 1)$.

| n | Approx. (4) | Trapezoid | Simpson |
|---|---|---|---|
| 4 | -15.5 | -5.0 | -10.2 |
| 8 | -14.3 | -4.7 | -9.1 |
| 10 | -14.3 | -4.6 | -8.7 |

Table 5: Similar comparison as shown in Table 4 with $f(x) = \cos\theta x$.

| $\theta$ | Approx. (4) | Trapezoid | Simpson |
|---|---|---|---|
| 1 | -15.4 | -5.7 | -11.7 |
| 10 | -16.4 | -4.9 | -8.9 |
| 100 | -16.8 | -3.9 | -5.9 |

outperforms significantly Trapezoid as expected. When $n$ and $\theta$ increases, the performance of Simpson and Trapezoid deteriorates as expected since the integrals change more dramatically, especially for $\cos\theta x$. Estimation (4) turns out to be more robust to handle rapid changes and high oscillation of function values as long as the integrand is sufficient smooth.

*7.2. Numerical Solution of First-Order Ordinary Differential Equation*

In this subsection, we aim to develop an algorithm to solve a non-linear ODE of first order

$$\begin{aligned} y'(x) &= f(x,y), \quad x \in [s,e] && (5)\\ y(s) &= \xi, && (6) \end{aligned}$$

where $f(x,y)$ is continuously differentiable on the range $[s-\delta, e+\delta] \times R^1$. It is well-known that there is a unique solution of Eq. (5-6) [20] under certain assumptions.

Replacing $f(x,y)$ by $f(x+o,y)$ if needed, we assume that $o := s-\delta = 0$ and denote $h(x)$ as the cut-off function in Section 5 such that

$$h(x) = \begin{cases} 1 & s \le x \le e, \\ 0 & x \le s-\delta \text{ or } x \ge e+\delta. \end{cases}$$

Apply $h$ to extend $f(x,y)$ as follows [3]:

$$F(x,u) = \begin{cases} f(x,u)h(x) & \text{if } x \in [0,b], \\ -f(-x,u)h(-x) & \text{if } x \in [-b,0]. \end{cases} \quad (7)$$

We shall search the solution for the extended ODE:

$$\begin{aligned} u'(x) &= F(x,u), \quad x \in [-b,b], && (8)\\ u(s) &= \xi. && (9) \end{aligned}$$

Since $(u(x)-u(-x))' \equiv 0$, $u(x)$ is even and its derivative $z(x) := u'(x)$ is odd. It is clear that $u$ can be smoothly extended to even periodic function with period $2b$ and $u(x)|_{[s,e]}$ solves Eqs. (5-6). Let $\{(x_k, z_k)\}_{0 \le k < N}$ be a grid set of $z(x)$:

$$\begin{aligned} x_k &= -b + \frac{2b}{N}k, \qquad k = 0, 1, \cdots, N-1, && (10)\\ z_0 &= 0, \quad z_{N-k} = -z_k, \qquad 1 \le k < M, && (11) \end{aligned}$$

and

$$z_M(x) = \sum_{0 \le j < M} b_j \sin \frac{j\pi x}{b} \quad (12)$$

[3] We use $u$ to denote the periodic extension of $y$.

be the interpolant of $z(x)$ with

$$b_j = \frac{2}{N}\sum_{k=0}^{N-1}(-1)^j z_k \sin\frac{2\pi jk}{N} = \frac{4}{N}\sum_{k=0}^{M-1}(-1)^j z_k \sin\frac{2\pi jk}{N}, \quad 0 \le j < M. \quad (13)$$

It is clear

$$\frac{\partial b_j}{\partial z_k} = \frac{4}{N}(-1)^j \sin\frac{2\pi jk}{N}, \quad 0 \le j,k < M.$$

$u$ can be approximated based on Eq. (12) by

$$\tilde{u}_M(x) = \sum_{0\le j<M} a_j \cos\frac{j\pi x}{b}, \quad a_j = -\frac{bb_j}{j\pi}, \quad 1 \le j < M, \qquad (14)$$

and $a_0$ can be solved by the initial condition $u(-s) = u(x_{m+n}) = \xi$

$$a_0 = \xi - \sum_{1\le j<M}(-1)^j a_j \cos\frac{2\pi j(m+n)}{N}.$$

Let $0_M$ be the $M$-dim zero vector and define $\frac{1}{0} := 0$. The following notations will be adopted in the rest of this subsection.

$$\begin{aligned}
u_k &= \tilde{u}_M(x_k), \quad F_k = F(x_k,u_k), \quad DF_k = \frac{\partial F}{\partial u}(x_k,u_k),\\
Z &= \{z_k\}_{0\le k<M}, \quad U = \{u_k\}_{0\le k<M},\\
F &= \{F_k\}_{0\le k<M}, \quad DF = \{DF_k\}_{0\le k<M},\\
J &= [0,1,\frac{1}{2},\ldots,\frac{1}{M-1},0_M],\\
\Phi &= [\frac{1}{j}\cos\frac{2\pi j(m+n)}{N}]_{j=0}^{M-1}, \quad \Phi_N = [\Phi, 0_M],\\
\Psi &= \{(z_j - F_j)DF_j\}_{j=0}^{M-1}, \quad \Psi_N = [\Psi, 0_M], \qquad I = sum(\Psi).
\end{aligned}$$

ODE (8)-(9) can be solved by minimizing the following error function:

$$\phi(z_0, z_1, ..., z_{M-1}) = \frac{1}{2M}\sum_{0\le k<M}(z_k - F_k)^2. \qquad (15)$$

We need an effective way to calculate its gradient $\frac{\partial\phi}{\partial Z}$ when $M$, the number of variables of $\phi$, is not small.

$$M\frac{\partial\phi}{\partial z_t} = (z_t - F_t) - \sum_{0\le k<M}(z_k - F_k)DF_k\frac{\partial u_k}{\partial z_t}. \qquad (16)$$

To cope with $\frac{\partial U}{\partial Z}$ in Eq. (16), we need express $U$ in term of $Z$. By Eq. (13) and $z_0 = z_M = 0$, we obtain for $0 \le k < N$

$$\begin{aligned} u_k &= a_0 - \sum_{0\le j<M} (-1)^j \frac{bb_j}{j\pi} \cos\frac{2\pi jk}{N} \\ &= a_0 - \frac{2b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sum_{0\le l<N} z_l \sin\frac{2\pi jl}{N} \qquad (17) \\ &= a_0 - \frac{2b}{\pi N} \sum_{0\le l<N} z_l \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sin\frac{2\pi jl}{N} \\ &= a_0 - \frac{4b}{\pi N} \sum_{0\le l<M} z_l \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sin\frac{2\pi jl}{N}. \qquad (18) \end{aligned}$$

The last step is due to $z_l \sin\frac{2\pi jl}{N} = z_{N-1} \sin\frac{2\pi j(N-l)}{N}$. One can rewrite (17) as follows

$$U = a_0 - \frac{2bN}{\pi} Re\{ifft(J \circ Im\{ifft(Z)\})\}, \qquad (19)$$

where $\circ$ denotes the Hadamard product, which applies the element-wise multiplication to two matrices of same dimension. $a_0$ in (18) can be further interpreted by $Z$ as follows

$$\begin{aligned} a_0 &= \xi - \sum_{1\le j<M} (-1)^j a_j \cos\frac{2\pi j(m+n)}{N} \\ &= \xi + \frac{b}{\pi} \sum_{1\le j<M} (-1)^j \frac{b_j}{j} \cos\frac{2\pi j(m+n)}{N} \\ &= \xi + \frac{2b}{\pi N} \sum_{1\le j<M} \frac{1}{j} \cos\frac{2\pi j(m+n)}{N} \sum_{k=0}^{N-1} z_k \sin\frac{2\pi jk}{N}, \qquad (20) \end{aligned}$$

which implies

$$\frac{\partial a_0}{\partial z_k} = \frac{4b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi j(m+n)\pi}{N} \sin\frac{2\pi jk}{N}, \quad 0 \le k < M. \qquad (21)$$

Combining (18) and (21), we obtain

$$\begin{aligned} \frac{\partial u_k}{\partial z_t} &= \frac{4b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi j(m+n)}{N} \sin\frac{2\pi jt}{N} \\ &- \frac{4b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi jk}{N} \sin\frac{2\pi jt}{N}. \qquad (22) \end{aligned}$$

We are ready to attack the non-trivial term in Eq. (16). Define

$$w_t := \sum_{0\le k<M} (z_k - F_k) DF_k \frac{\partial u_k}{\partial z_t}.$$

By (22),

$$\begin{aligned} w_t &= \frac{4b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \cos\frac{2\pi j(m+n)\pi}{N} \sin\frac{2\pi jt}{N} \sum_{0\le k<M} (z_k - F_k) DF_k \\ &- \frac{4b}{\pi N} \sum_{0\le j<M} \frac{1}{j} \sin\frac{2\pi jt}{N} \sum_{0\le k<M} (z_k - F_k) DF_k \cos\frac{2\pi jk}{N} \\ &= \frac{4b}{\pi N} \sum_{0\le j<M} \Phi_j \sin\frac{2\pi jt}{N} \sum_{0\le k<M} (z_k - F_k) DF_k \\ &- \frac{4b}{\pi N} \sum_{0\le j<M} J_j \sin\frac{2\pi jt}{N} \sum_{0\le k<M} (z_k - F_k) DF_k \cos\frac{2\pi jk}{N}. \end{aligned}$$

The gradient vector (16) can be formulated by FFT as follows

$$W = \frac{4\pi I}{b} Im(ifft(\Phi_N)) - \frac{4\pi N}{b} Im\{ifft(J \circ Re[ifft(\Psi_N)])\}, \quad (23)$$

$$\frac{\partial \phi}{\partial Z} = \frac{1}{M}(Z - F - W[0 : M-1]). \quad (24)$$

One can implement the algorithm by following steps.

**Algorithm 2.** *For a given ODE (5-6),*

1. *Select $(p, q, \delta)$ as in Algorithm 1 such that $f(x, y)$ can be smoothly extended to $[s - \delta, e + \delta] \times R$.*
2. *Construct the cut-off function $h(x)$ with parameter $(s, e, \delta, r = 0.5)$ by Eq. (1).*
3. *Construct $F(x, u)$ by Eq. (7).*
4. *Construct initial values $\{u_i\}_{0\le i<N}$ at $\{x_i\}_{0\le i<N}$ defined by Eq. (10) and calculate initial value $z(x_i) = F(x_i, u_i)$ in Eq. (11) that is required by an optimization function.*
5. *Apply an optimization function with the gradient function $\frac{\partial \phi}{\partial Z}$ formulated by Eq. (24).*

6. *Apply the opt values of Z returned by the optimization function in previous step to calculate required U by Eqs. (19-20) and return* $U|_{[s,e]}$.

**REMARK 1.** 1. *Algorithm 2 is expected to be efficient since the gradient of the target function can be carried out by* $O(N \ln_2 N)$ *operations.*

2. *A standard difference method can be used to construct initial values* $\{u_i\}_M^{N-1}$ *in Step 4 of Algorithm 2, which should be smoothed by the cut-off function and then be extend evenly at the grid points over* $[-b, 0]$.

In the rest of this subsection, we study the performance of Algorithm 2, labeled *tibo*, by solving the following ODE

$$y'(x) = f(x, y) + xy + y^2, \qquad y(1) = 0, \tag{25}$$

where

$$f(x, y) = \cos\theta x - \theta x \sin\theta x - xy - y^2, \theta \in \{\frac{\pi}{2}, \frac{3\pi}{2}\}.$$

The analytic solution is available as follows:

$$y(x) = x \cos\theta x.$$

We shall compare *tibo* to the classic Runge–Kutta method, labeled $rk4$, outlined in Eqs. (26-28) ([5]).

$$y_{n+1} = y_n + \frac{\lambda}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{26}$$

$$k_1 = f(x_n, y_n), \quad k_2 = f(x_n + \frac{\lambda}{2}, y_n + \frac{\lambda}{2}k_1), \tag{27}$$

$$k_2 = f(x_n + \frac{\lambda}{2}, y_n + \frac{\lambda}{2}k_2), \quad k_2 = f(x_n + \lambda, y_n + \lambda k_3). \tag{28}$$

It is well-known that local truncation error of $rk4$ is on the order of $O(\lambda^5)$ and hence the total accumulated error is supposed to be $O(\lambda^4)$.

In addition, we implement a benchmark method, labeled as *benc*, by adjusting $rk4$ in a "cheating" way such that $y_{n+1}$ is estimated by the true value $Y_n$ at $x_n$. Mathematically, all $y_n$ in Eq (26)-(28) is replaced by $Y_n$. In this way, *benc* prevents accumulating error and is supposed to be on the order of $O(\lambda^5)$.

The overall performance is reported in Table 6, where the max magnitude of errors at grid points $\{x_j\}_{j=0}^{N-1}$ are shown under three methods. In

addition, to see the performance of *tibo* at non-grid points, the max error is also reported under $tibo_g$ where the max is taken over the error set obtained by applying identified $\tilde{u}_M$ (see Eq. (14)) to the grid points with step size $\lambda/4$. Table 6 also includes the value of the optimization target function Eq. (15) returned by Matlab function $fmincon$ under Column $opt_{err}$. We also

| $\theta$ | $tibo$ | $rk4$ | $benc$ | $tibo_g$ | $opt_{err}$ |
|---|---|---|---|---|---|
| $\pi/2$ | 3.2E-09 | 7.7E-07 | 3.0E-08 | 3.2E-09 | 3.2E-17 |
| $3\pi/2$ | 4.8E-07 | 2.1E-03 | 1.1E-05 | 4.8E-07 | 1.0E-17 |

Table 6: The max magnitudes of four sets of errors and the optimization error described above. Algorithm 2 is implemented with $p = 6, q = 7, \delta = 1$.

look into changes of consecutive errors for three covered methods similar as we did in Section 6.3. Figure 5 plots such changes defined by

$$\{\hat{f}_M(x_i) - f(x_i) - (\hat{f}_M(x_{i-1}) - f(x_{i-1}))\}.$$

for *tibo* and *benc*. Figure 6 compares same changes among three covered methods. We have the following comments.

**REMARK 2.** 1. *The optimization process successfully converges to a interpolant $z_M(x)$ of the target function $z(x) = u'(x)$ for both test scenarios as shown by $opt_{err}$ values in Table 6.*

2. *tibo is almost same as $tibo_g$, suggesting that $\tilde{u}_M$ uniformly converges the target function u with same accuracy as exhibited at grid points with $\{x_j\}_{0\le j<N}$.*
3. *The performances at the scenario $\theta = \pi/2$ are significantly better than that at the scenario $\theta = 3\pi/2$ as expected since the target function y by (25) with small $\theta$ is less volatile than with large $\theta$.*
4. *tibo outperforms benc significantly not just by the measure on max error in Table 6, but also its error is consistently smaller than benc's as shown in Fig 5.*
5. *$rk4$ has worst performance based on max error, especially for $\theta = 3\pi/2$. Fig. 6 plots the difference of two consecutive errors for each of three methods. As one can see, errors from $rk4$ moves in one direction from certain point $x_j$ and leads to significant aggregated error at the end. On the other hand, the error of tibo moves in sawtooth around 0, an error-correct sign, which makes tibo outperform bech, a method without error*

*propagation. See Section 6.3 for more discussions of error behaviors of* $\hat{f}_M(x)$.

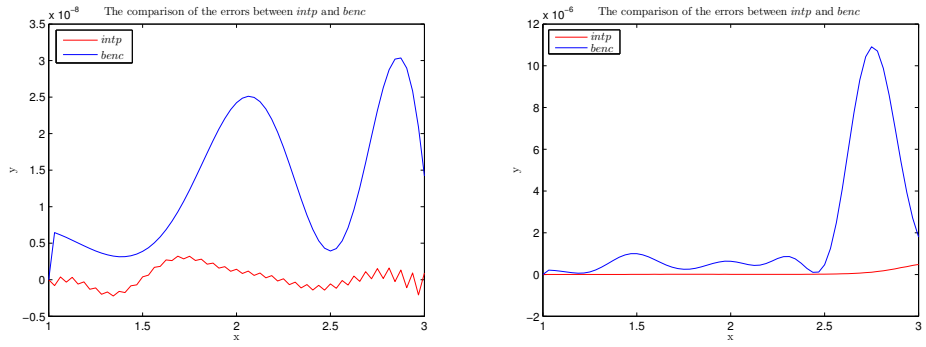


Figure 5: The comparison of consecutive errors between *tibo* and *benc* for $\theta = \pi/2$ on the left and $\theta = 3\pi/2$ on the right.

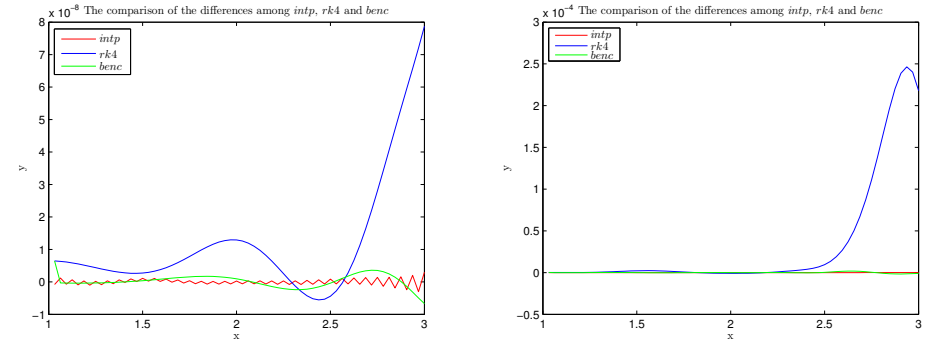


Figure 6: The comparison of changes of consecutive errors among *tibo*, *benc*, $rk4$ for $\theta = \pi/2$ on the left panel and $\theta = 3\pi/2$ on the right panel.

Fig. 7 plots the target $y$ and identified $\tilde{u}_M$ over $[0, b]$. $\tilde{u}_M$ recovers $y$ over the range $[s, e] = [1, 3]$. It becomes flat near boundaries and can be treated as even periodic function.

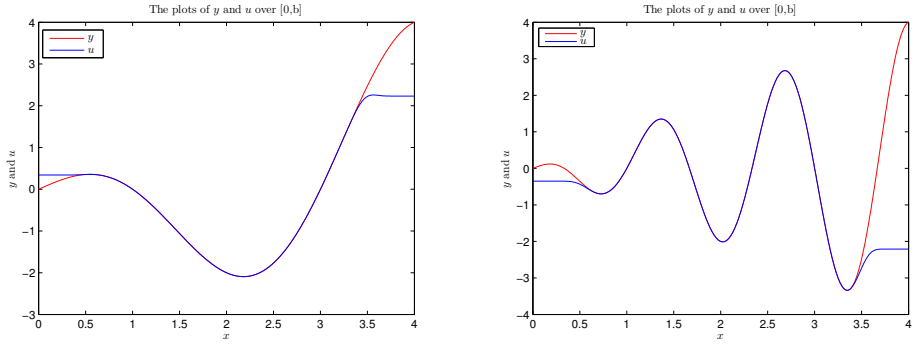


Figure 7: Comparison between the target solution $y$ and trigonometric estimation $\tilde{u}_M$ over the half period $[0, b] = [0, 4]$. Note that $\tilde{u}_M$ is supposed to approximate $y$ over $[s, e] = [2, 3]$.

## 8. Conclusions

In this study ([21]-[22]), we have proposed a new trigonometric interpolation method and established convergent properties. The method improves an existing trigonometric interpolation algorithm in the literature such that it can better leverage FFT to enhance efficiency. The interpolant can be formulated in such way that the cancellation effects can be profoundly leveraged for error analysis, which enables us not only to improve the convergent rate of interpolant, but establish similar uniform convergence for the derivatives of interpolants. We have further enhanced the method such that it can be applied to non-periodic functions defined on bounded interval. Numerical testing results confirm accurate performance of the algorithm. In addition, we show numerical evidences that estimation error of the algorithm likely exhibits "local property", i.e. error at a point tends not to propagate and result in significant compounding error at some other place, suggesting its performance is likely better than what has be theoretically approved in this paper.

Considering the accurate performance and analytic attractiveness of the new trigonometric interpolation algorithm, especially it can be applied to non-periodic functions, we expect that the new algorithm developed in this study can be used in a wide spectrum. We demonstrate how it can be applied to estimate integrals and solve non-linear ODE. The test results show that it outperforms Trapezoid/Simpson method to cope with integrals and standard Runge-Kutta algorithm to handle ODE. In subsequent papers [23]-[26], it has

been used to develop algorithms to solve ODEs and IDEs under more general settings with various advantages as explained in Section 1.

**Acknowledgments**

This paper is dedicated to my wife Jian Jiao for her unwavering support to the family in last thirty plus years.

**References**

[1] A. P. Austin, *On Trigonometric Interpolation In an Even Number of Points*, Electronic Transactions on Numerical Analysis, 58 (2023), 271–288.

[2] G. B. Wright, M. Javed, H. Montanelli, and L. N. Trefethen *Extension of Chebfun to Periodic Functions*, SIAM Journal on Matrix Analysis and Applications, 37(5) (2015), pp. 554-573.

[3] James W. Cooley and John W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, SIAM Journal on Matrix Analysis and Applications, 19(4) (1990) 297–301.

[4] V. S. Ryaben'kii and S. V. Tsynkov, *A Theoretical Introduction to Numerical Analysis*, Chapman & Hall/CRC, 2007.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes*, Cambridge University Press, 2007.

[6] R. Kress, *Numerical Analysis, volume 181 of Graduate Texts in Mathematics*, Springer-Verlag, 1998.

[7] G. Elefante, *A barycentric trigonometric Hermite Interpolant via an iterative approach*, Journal of Computational and Applied Mathematics, 439 (2024) pp. 1-13

[8] C. Schneider and W. Werner *Hermite interpolation: The barycentric approach*, Computing, 46 (1) (1991), pp. 35-51

[9] K. Jing, Y. Liu, N. Kang, and G. Zhu *A convergent family of linear Hermite barycentric rational interpolants*, J. Math. Res. Appl., 40 (6) (2020), pp. 628-646

[10] E. Cirillo and K Hormann *An iterative approach to barycentric rational Hermite interpolation*, Numer. Math., 140 (4) (2018), pp. 939-962

[11] F. Dell'Accio, F. Di Tommaso, O. Nouisser, N. Siar *Rational Hermite interpolation on six-tuples and scattered data*, Appl. Math. Comput., 386 (2020), p. 125452, 11

[12] J.-P. Berrut, *Rational functions for guaranteed and experimentally well-conditioned global interpolation*,Comput. Math. Applic. 15 (1) (1988) pp.1-16

[13] P.J. Baddoo *The AAAtrig algorithm for rational approximation of periodic functions*,SIAM J. Sci. Comput., 43 (5) (2021), pp. A3372-A3392

[14] R. Baltensperger, *Some Results on Linear Rational Trigonometric Interpolation*,Computers and Mathematics with Applications 43 (2002) pp. 737-746

[15] M. Dehghan and A. Saadatmandi, *Chebyshev finite difference method for Fredholm integro-differential equation*, Int. J. Comput. Math., 85 (1) (2008), 123–130.

[16] G. Yuksel, S. Yuzbas, and M. Sezer, *A Chebyshev method for a class of high-order linear Fredholm integro-differential equations*, SIAM Journal on Matrix Analysis and Applications, 37 (5) (2015), 554–573.

[17] S. Bushnaq, S. Momani, and Y. Zhou, *A reproducing kernel Hilbert space method for solving integro-differential equations of fractional order*, J. Optim. Theory Appl. 2014 (1) (2013) 96–105.

[18] Z.H. Zhao, Y.Z. Lin, J. Niu, *Convergence order of the reproducing kernel method for solving boundary value problems*, Math. Model. Anal. 21 (4) (2016) 466–477.

[19] Y. Zheng, Y. Lin, Y. Shen *A new multiscale algorithm for solving second order boundary value problems*, Applied Numerical Mathematics 156 (2020), pp. 528–541

[20] W.E. Boyce and R.C. Diprima, *Elementary Differential Equations and Boundary Value Problems*, John Wiley and Sons, 1986.

[21] X. Zou, *On Trigonometric Approximation and Its Applications I*, to appear

[22] X. Zou, *On Trigonometric Approximation and Its Applications II*, to be appear

[23] X. Zou *Trigonometric Interpolation Based Optimization for Solving Second Order Non-Linear ODE with Linear Boundary Conditions*, https://arxiv.org/pdf/2504.19280

[24] X. Zou, *Trigonometric Interpolation Based Approach for Second Order ODE with Mixed Boundary Conditions*, https://arxiv.org/pdf/2505.07183, May, 2025

[25] X. Zou, *Trigonometric Interpolation Based Approach for Second Order Fredholm Integro-Differential Equations*, https://arxiv.org/pdf/2508.09413, Aug. 2025

[26] X. Zou, *Trigonometric Interpolation Based Approach for Second Order Volterra integro-differential equations*, to be appear

[27] K. Liu, B. Wang, X. Wu and X. Zou, *On Application of Trigonometric Interpolation in Solving Non-Linear ODE System with Constraints*, to be appear

[28] Avik Pal, Alan Edelman, Chris Rackauckas, *Semi-Explicit Neural DAEs: Learning Long-Horizon Dynamical Systems with Algebraic Constraints*, https://arxiv.org/pdf/2505.20515v1, May 2025

[29] A. Zygmund, *Trigonometric Series*, Cambridge University Press, 2015.