International Journal of Computational Mathematics and Scientific Computing (IJCMSC)

DOI: 10.5281/zenodo.21595698 | ISSN: 3141-643X | Volume: 1 | Issue: 2 | 2026

Fixed-Point Iteration for Nonlinear Volterra Equations in Computational Physics: A Strictly Pseudo-Contractive Mann Framework with Application to Heat Conduction with Memory

I. D. Edem[1,*], O. G. Udoaka[1], K. E. Essien[2]

[1]Department of Mathematics, Akwa Ibom State University, Nigeria
[2]Department of Physics, University of Uyo, Nigeria
[*]**Corresponding Author:** inemesitedem@aksu.edu.ng

Abstract

Nonlinear Volterra integral equations (VIEs) of the second kind arise in numerous physical contexts, including nonlinear heat conduction with memory, viscoelasticity, and population dynamics. However, when the kernel satisfies only a one-sided Lipschitz condition, classical contraction-based numerical methods fail, and efficient, provably convergent solvers are lacking. In this paper, we develop a fully discrete, computationally efficient numerical scheme for such equations. The method combines an implicit Euler time discretisation with an inexact Mann inner solver that requires no Jacobian evaluations, making it ideally suited for large-scale parallel computations. We establish global convergence of the adaptive, fully discrete scheme under realistic smoothness assumptions, and introduce a novel adaptive time-stepping strategy based on the Mann residual. The performance of the method is demonstrated on a physically motivated model of one-dimensional (1D) nonlinear heat conduction with memory. Extensive numerical comparisons against Picard iteration and Newton's method show that the Mann-based solver is robust for large time steps, scales efficiently to high-resolution spatial discretisations, and maintains linear convergence rates as predicted by theory. Our open-source implementation provides a practical, ready-to-use tool for the computational physics community, and the framework is easily extended to partial integro-differential equations (PIDEs) and fractional-order memory kernels.

Keywords: Volterra integral equation; strictly pseudo-contractive mapping; Mann iteration; one-sided Lipschitz condition; nonlinear heat conduction; memory; adaptive time stepping; parallel computing.

1. Introduction

1.1. Physical Motivation and Modelling

Physical systems with memory are often described by Volterra integral equations of the second kind:

$$x(t) = f(t) + \int_0^t K(t, s, x(s))\, ds, \qquad t \in [0, T], \tag{1}$$

where $x(t)$ is the state of the system (e.g., temperature, displacement, population), $f(t)$ is a known forcing term, and the kernel $K(t, s, u)$ encodes the influence of the history of the process. Such equations appear in:

- **Nonlinear Heat Conduction with Memory**: In materials with memory, the classical Fourier law $q = -k\nabla u$ is replaced by a hereditary constitutive relation. Using the Green's function approach, the heat equation can be reformulated as a Volterra-type integral equation [1]. The Gurtin–Pipkin model of heat propagation in media with memory leads to VIEs with convolution kernels.
- **Viscoelasticity**: The stress–strain relationship in viscoelastic materials involves a convolution integral of the strain history with a relaxation kernel, leading to a Volterra integro-differential equation [2].
- **Population Dynamics**: Models with gestation or saturation effects give rise to VIEs with nonlinear kernels.

In many such applications, the kernel satisfies a **one-sided Lipschitz condition**:

$$\big(K(t, s, u) - K(t, s, v)\big)(u - v) \le L(u - v)^2, \qquad \text{for a.e. } t, s,\ \forall u, v, \tag{2}$$

with a constant $L \in \mathbb{R}$ that may be positive, zero, or negative. This condition is strictly weaker than the standard global Lipschitz assumption that $|K(t, s, u) - K(t, s, v)| \le C|u - v|$. For example, the cubic saturation $K(u) = -u^3$ satisfies (2) with $L = 0$ but is not globally Lipschitz. When $L > 0$, the kernel may be "expansive" in some regimes, but the smallness condition $L\|V\|_p < 1$ (where V is the linear Volterra operator) ensures well-posedness.

1.2. Numerical Challenges

Standard numerical methods for VIEs, such as collocation methods [3] and Runge–Kutta schemes [2], often rely on the global Lipschitz property to guarantee convergence. When this fails, Picard iteration (the simplest fixed-point method) may diverge, and Newton's method, while locally convergent, requires the costly evaluation of Jacobians and may not converge from a poor initial guess. Moreover, the one-sided Lipschitz condition (2) does not fit into the classical contraction framework.

1.3. Fixed-Point Framework and the Mann Iteration

In a recent theoretical paper, Edem et al. [7] proved that under the one-sided Lipschitz condition (2) (with $L \leq 0$ or, if $L > 0$, the smallness condition $L\|V\|_p < 1$), the Volterra operator $T : L^p([0,T]) \to L^p([0,T])$ defined by

$$(Tx)(t) = f(t) + \int_0^t K(t,s,x(s))\,ds, \qquad t \in [0,T], \tag{3}$$

is k-strictly pseudo-contractive for some $k \in [0,1)$. A mapping T is called k-strictly pseudo-contractive if for all x, y in its domain,

$$\|Tx - Ty\|^2 \leq \|x - y\|^2 + k\|(I - T)x - (I - T)y\|^2. \tag{4}$$

This property, together with the geometry of uniformly convex and uniformly smooth Banach spaces (such as L^p with $1 < p < \infty$), ensures that the Mann iteration

$$x_{n+1} = (1 - \lambda_n)x_n + \lambda_n T x_n, \qquad n \geq 0, \tag{5}$$

converges (weakly, and strongly if the operator is demicompact) to a fixed point of T and hence to a solution of (1). Recent extensions of this framework to generalized-hybrid mappings in L^p spaces [5] have established strong convergence of the Mann iteration under the parameter condition $\alpha + 2\beta < 1$, providing additional theoretical support for the present numerical scheme. The Mann iteration is a one-parameter family of iterative methods that includes the Picard iteration ($\lambda_n = 1$) and the Krasnoselskii–Mann method ($\lambda_n \in (0,1)$) as special cases. Notably, Som et al. [4] have recently shown that the Krasnoselskii–Mann iteration, with an averaged operator $T_\lambda = (1 - \lambda)I + \lambda T$, can be used to approximate fixed points of k-strictly pseudo-contractive mappings in Banach spaces, extending results previously known only in Hilbert spaces. Recent advances in inertial techniques for fixed-point problems in Banach spaces [6] have demonstrated the effectiveness of momentum-based acceleration, motivating the present use of Anderson mixing.

The theoretical paper also proposed a **numerical analysis extension**: a semi-discretisation in time (method of lines) transforms the Volterra equation (1) into a system of ordinary differential equations, and the Mann iteration is used as an inexact solver for the resulting nonlinear system. Error estimates were provided, establishing convergence of the fully discrete scheme under the one-sided Lipschitz condition.

1.4. Contribution and Novelty

While the theoretical framework of [7] is in place, a computational implementation for realistic physical problems is lacking. This paper fills that gap. The novelty is threefold:

1. **Operator-theoretic**: First application of k-strict pseudo-contractivity (Lemma 2.2) to Volterra equations with non-Lipschitz kernels in L^p spaces.

2. **Algorithmic**: Mann inner solver (see 4.2) with dynamic stopping criterion (see 4.3) and Anderson acceleration (see 4.5) for VIEs.

3. **Application**: First demonstration on 1D nonlinear heat conduction with memory (Section 3), with open-source parallel implementation.

Our contributions are:

1. **High-Performance Implementation**: Modular C++/MPI code with adaptive time stepping.
2. **Acceleration**: Anderson mixing [9] accelerates the Mann iteration.
3. **Benchmark**: Extensive comparisons with Picard and Newton.

1.5. Outline of the Paper

The remainder of the paper is organised as follows. Section 2 recalls the mathematical preliminaries and the fixed-point properties of the Volterra operator. Section 3 introduces the benchmark physical problem: nonlinear heat conduction with memory. Section 4 describes the fully discrete numerical scheme, including the Mann inner solver, adaptive time stepping, and Anderson acceleration. Section 5 provides a convergence analysis of the adaptive scheme, extending the theoretical results of Edem et al. Section 6 details the implementation and high-performance computing aspects. Section 7 presents the numerical experiments and comparisons. Section 8 concludes the paper with a summary and directions for future work.

2. Mathematical Preliminaries and Fixed-Point Framework

2.1. Function Spaces and Duality Mapping

Let $X = L^p([0,T])$ with $1 < p < \infty$. This space is uniformly convex and uniformly smooth, properties essential for the convergence analysis of the Mann iteration. The normalised duality mapping $J_p : X \to X^*$ is single-valued and given by

$$J_p(x)(t) = \|x\|_p^{2-p} |x(t)|^{p-1} \operatorname{sgn}(x(t)), \qquad t \in [0,T]. \tag{6}$$

In the Hilbert space case $p = 2$, J_2 is the identity.

2.2. One-Sided Lipschitz Condition and Strict Pseudo-Contractivity

Assumption 2.1 (One-sided Lipschitz condition)**.** There exists a constant $L \in \mathbb{R}$ such that for almost every $t, s \in [0,T]$ and for all $u, v \in \mathbb{R}$,

$$\big(K(t,s,u) - K(t,s,v)\big)(u-v) \le L(u-v)^2. \tag{7}$$

Moreover, $|K(t,s,0)| \le m(t,s)$ with $m \in L^1([0,T]^2)$.

When $L \le 0$, the kernel is dissipative; when $L > 0$, we require the smallness condition $L\|V\|_p < 1$, where V is the linear Volterra operator $(Vu)(t) = \int_0^t u(s)\,ds$.

Lemma 2.2 (Strict pseudo-contractivity of T)**.** *Under Assumption 2.1, the Volterra operator T defined in (3) is k-strictly pseudo-contractive with $k = 1 - 2\gamma \in (0,1)$, where $\gamma > 0$ is the strong accretivity constant of $I - T$. In particular, for the dissipative case $L \le 0$, we have $k = 0$ (i.e., T is nonexpansive).*

Proof. For the dissipative case $L \le 0$, the one-sided Lipschitz condition gives

$$\langle (I-T)x - (I-T)y, J_p(x-y)\rangle \ge \frac{1}{2}\,(1 - \|V\|_p)\,\|(I-T)x - (I-T)y\|^2,$$

where V is the linear Volterra operator $(Vu)(t) = \int_0^t u(s)\,ds$. The constant $\gamma = \frac{1}{2}(1 - \|V\|_p)$ is positive because $\|V\|_p < 1$ for sufficiently small T (specifically, $\|V\|_p = T^{1/p'}/p' < 1$). Thus $I - T$ is strongly accretive with constant γ, and by standard Banach space theory, T is k-strictly pseudo-contractive with $k = 1 - 2\gamma$ [7, 8]. For $L > 0$, the smallness condition $L\|V\|_p < 1$ ensures the same inequality with a modified constant. □

Corollary 2.3 (Mann iteration convergence)**.** *If the solution set of* (1) *is nonempty and the Mann parameters* $\lambda_n \in (0, 1-k]$ *satisfy* $\sum_{n=0}^{\infty} \lambda_n(1-\lambda_n) = \infty$*, then the Mann iteration* (5) *converges weakly to a solution. If, in addition, the Volterra operator* T *is demicompact (e.g., if the kernel is compact in the sense of Assumption 3.5 in [7]), then the convergence is strong in the norm of* L^p*.*

3. Benchmark Problem: Nonlinear Heat Conduction with Memory

3.1. Model Description

We consider a 1D rod of length ℓ, initially at zero temperature, with a prescribed temperature at the left endpoint and a zero heat flux at the right endpoint. The governing equations are:

$$\rho c_p \frac{\partial u}{\partial t}(x,t) = \frac{\partial}{\partial x}\left(k(u(x,t))\frac{\partial u}{\partial x}(x,t)\right) + \int_0^t g(t-\tau)\frac{\partial}{\partial x}\left(k(u(x,\tau))\frac{\partial u}{\partial x}(x,\tau)\right)d\tau + Q(x,t), \tag{8}$$

$$u(x,0) = 0, \tag{9}$$

$$u(0,t) = u_0(t), \qquad \frac{\partial u}{\partial x}(\ell,t) = 0. \tag{10}$$

Here, ρ is the density, c_p the specific heat capacity, $k(u) > 0$ the temperature-dependent thermal conductivity, $g(t-\tau)$ the memory kernel (typically an exponentially decaying function), and $Q(x,t)$ an external heat source. The memory term models the fact that the heat flux at time t depends on the history of the temperature gradient.

3.2. Reduction to a Volterra Integral Equation

Following the method of lines, we first discretise the spatial domain using finite differences. Let $0 = x_0 < x_1 < \cdots < x_M = \ell$ be a uniform grid with spacing $\Delta x = \ell/M$, and denote $u_i(t) \approx u(x_i,t)$. The spatial derivatives are approximated by a second-order finite difference scheme:

$$\frac{\partial}{\partial x}\left(k(u)\frac{\partial u}{\partial x}\right)\Big|_{x=x_i} \approx \frac{1}{\Delta x^2}\left(k(u_{i+1/2})(u_{i+1}-u_i) - k(u_{i-1/2})(u_i - u_{i-1})\right), \tag{11}$$

where $k(u_{i+1/2})$ is evaluated using an average of u_i and u_{i+1}. The semi-discrete system takes the form:

$$\rho c_p \frac{du_i}{dt}(t) = \mathcal{N}_i(u(t)) + \int_0^t g(t-\tau)\mathcal{N}_i(u(\tau))\,d\tau + Q_i(t), \quad i = 1,\ldots,M-1, \tag{12}$$

with boundary conditions incorporated into the discrete operators $\mathcal{N}_i$. Integrating (12) from 0 to t and using the initial condition $u_i(0) = 0$ gives:

$$\rho c_p u_i(t) = \int_0^t \mathcal{N}_i(u(\tau))\,d\tau + \int_0^t \int_0^\tau g(\tau - s)\mathcal{N}_i(u(s))\,ds\,d\tau + \int_0^t Q_i(\tau)\,d\tau. \tag{13}$$

Interchanging the order of integration in the double integral yields a Volterra equation of the second kind:

$$u_i(t) = f_i(t) + \frac{1}{\rho c_p} \int_0^t \left(1 + \int_0^{t-\tau} g(s)\, ds\right) \mathcal{N}_i(u(\tau))\, d\tau, \tag{14}$$

where $f_i(t) = \frac{1}{\rho c_p} \int_0^t Q_i(\tau)\, d\tau$. The kernel of the integral operator is a convolution, making fast algorithms applicable.

3.3. One-Sided Lipschitz Property of the Reduced Kernel

For the case of a nonlinear conductivity $k(u) > 0$ that is a decreasing function of u (i.e., $k'(u) \leq 0$), the discrete nonlinearity $\mathcal{N}_i(u)$ satisfies a one-sided Lipschitz condition with constant $L \leq 0$. In particular, for the cubic conductivity model

$$k(u) = k_0(1 - \alpha u^3), \qquad \alpha > 0,\ u \geq 0, \tag{15}$$

which is relevant for certain materials where conductivity decreases with temperature, the corresponding kernel $K(t, s, u)$ satisfies Assumption 2.1 with $L = 0$ (dissipative case).

4. Numerical Scheme: Fully Discrete Mann-Inexact Solver

4.1. Time Discretisation (Implicit Euler)

Let $N \in \mathbb{N}$, $\Delta t = T/N$, and $t_n = n\Delta t$ for $n = 0, 1, \ldots, N$. For the semi-discrete system (12), an implicit Euler discretisation gives:

$$\rho c_p \frac{u_i^{n+1} - u_i^n}{\Delta t} = \mathcal{N}_i(u^{n+1}) + \Delta t \sum_{j=0}^{n+1} w_j g(t_{n+1} - t_j) \mathcal{N}_i(u^j) + Q_i^{n+1}, \tag{16}$$

where w_j are quadrature weights (e.g., the trapezoidal rule). Collecting the unknown terms at time t_{n+1} on the left, we obtain a nonlinear system of the form:

$$u^{n+1} = F_n + \Delta t\, \Phi_n(u^{n+1}), \qquad n \geq 0, \tag{17}$$

where F_n contains all known quantities from previous time steps and the external source, and Φ_n is a nonlinear operator derived from $\mathcal{N}_i$. For the cubic conductivity model, Φ_n inherits the one-sided Lipschitz property from $\mathcal{N}_i$, ensuring that the discrete operator $T_n(u) = F_n + \Delta t\, \Phi_n(u)$ is strictly pseudo-contractive for sufficiently small Δt.

4.2. Mann Inner Solver

At each time step, we solve (17) by the Mann iteration:

$$y^{(k+1)} = (1 - \lambda) y^{(k)} + \lambda T_n(y^{(k)}), \qquad k = 0, 1, \ldots, \tag{18}$$

with a fixed parameter $\lambda \in (0, 1 - k]$. The iteration is stopped when the residual falls below a specified tolerance:

$$\|y^{(k)} - T_n(y^{(k)})\| \leq \varepsilon_n, \tag{19}$$

and we set $u^{n+1} = y^{(K_n)}$.

4.3. Dynamic Stopping Criterion

Instead of the fixed tolerance $\varepsilon_n = \Delta t^2$ used in the theoretical paper, we employ a more practical, adaptive criterion:

$$\varepsilon_n = \max\left(\varepsilon_{\text{abs}}, \varepsilon_{\text{rel}}\|F_n\|\right), \tag{20}$$

where ε_{abs} is an absolute tolerance (e.g., 10^{-8}) and ε_{rel} a relative tolerance (e.g., 10^{-5}). This prevents over-solving when the known term is small.

4.4. Adaptive Time Stepping

We propose a novel adaptive time-stepping strategy based on the estimated error of the Mann solver. After the Mann iteration has converged, we compute an estimate of the local truncation error using the difference between the implicit Euler solution and a predictor (e.g., from the previous time step). If the error estimate exceeds a user-defined threshold, the time step is reduced; if it is safely below the threshold, the time step is increased. This strategy is similar to the adaptive Runge–Kutta methods developed for Volterra-type integro-differential equations in [2].

4.5. Anderson Acceleration for the Mann Iteration

The Mann iteration (18) often converges only linearly. To accelerate its convergence, we incorporate **Anderson acceleration** (also known as Anderson mixing) [9]. Given a history of the last $m+1$ iterates, the accelerated iterate is:

$$y^{(k+1)} = \sum_{i=0}^{m} \alpha_i^{(k)} T_n(y^{(k-i)}), \tag{21}$$

where the coefficients $\alpha_i^{(k)}$ are chosen to minimise the residual in a least-squares sense. Anderson acceleration has become a standard technique in computational science for speeding up fixed-point iterations, and its use in the context of the Mann iteration is novel. Recent theoretical results [9] have established conditions for finite convergence and provided bounds on the coefficients, which can guide the practical choice of the window size m.

4.6. Convergence and Sensitivity of Anderson Acceleration

The Anderson-accelerated Mann iteration (4.9) is a fixed-point iteration of the form

$$y^{(k+1)} = (1-\lambda)y^{(k)} + \lambda T_n(y^{(k)}) + \sum_{i=0}^{m} \alpha_i^{(k)} \left(T_n(y^{(k-i)}) - y^{(k-i)}\right),$$

where the coefficients $\alpha_i^{(k)}$ solve

$$\min_{\alpha \in \mathbb{R}^{m+1}} \left\| \sum_{i=0}^{m} \alpha_i \left(T_n(y^{(k-i)}) - y^{(k-i)}\right) \right\|^2, \quad \sum_{i=0}^{m} \alpha_i = 1.$$

Lemma 4.1 (Linear convergence of Anderson acceleration)**.** *Let T_n be k-strictly pseudo-contractive with Lipschitz constant $\mathcal{L} = (1+k)/(1-k)$ [9]. If the residual vectors $\{r^{(k-i)}\}_{i=0}^{m}$ are linearly independent, then the Anderson-accelerated iteration satisfies*

$$\|y^{(k+1)} - y^*\| \leq \left(\frac{1}{\sigma_{\min}(R)}\right)\left(\kappa(R) + \frac{\mathcal{L}}{1-\lambda}\right)\|y^{(k)} - y^*\|,$$

where $R = [r^{(k-m)}, \ldots, r^{(k)}]$ is the residual matrix, $\sigma_{\min}(R)$ is its smallest singular value, and $\kappa(R)$ its condition number.

Proof. The residual minimisation yields $\sum_{i=0}^{m} \alpha_i r^{(k-i)} = 0$ in the least-squares sense. Expressing $y^{(k+1)}$ as a linear combination of $T_n(y^{(k-i)})$ and using the Lipschitz continuity of T_n gives the bound. The condition number $\kappa(R)$ controls the sensitivity; ill-conditioning occurs when residuals are nearly collinear. □

Corollary 4.2 (Window size sensitivity)**.** *The condition number $\kappa(R) \leq \kappa_{\max}$ for $m \leq m_{\max}$, where $m_{\max} = \lfloor \log \varepsilon / \log \rho \rfloor$ depends on the spectral radius ρ of T_n. In practice, $m \leq 10$ suffices to maintain $\kappa(R) < 10^4$ for the present problem.*

The computational cost of Anderson acceleration is $O(m^3 + m \cdot \dim(X))$ per iteration, where $O(m^3)$ is the cost of the least-squares solve. For $m = 5$, this is negligible compared to the $O(N \log N)$ cost of the history term.

4.7. Algorithm Pseudocode

A complete pseudocode of the adaptive Mann-inexact solver is provided in Algorithm 1.

Algorithm 1 Adaptive Mann-inexact solver for nonlinear Volterra equations

procedure AdaptiveMannSolver($T, \Delta t_{\text{init}}, \varepsilon_{\text{abs}}, \varepsilon_{\text{rel}}, \lambda, m$)
  $t \leftarrow 0$, $\Delta t \leftarrow \Delta t_{\text{init}}$, $u^0 \leftarrow f(0)$
  while $t < T$ **do**
    repeat
      $F \leftarrow$ ComputeKnownTerms$(u^0, \ldots, u^n, \Delta t)$
      Define $T_n(y) = F + \Delta t\, \Phi_n(y)$
      $y^{(0)} \leftarrow u^n$ ▷ or extrapolation
      $k \leftarrow 0$, history $\leftarrow []$
      while true **do**
        if len(history) $\geq m$ **then**
          $y^{(k+1)} \leftarrow$ AndersonUpdate$(history, T_n)$
        else
          $y^{(k+1)} \leftarrow (1-\lambda)y^{(k)} + \lambda T_n(y^{(k)})$
        end if
        Append $y^{(k+1)}$ to history
        if len(history) $> m+1$ **then**
          pop first element
        end if
        $res \leftarrow \|y^{(k+1)} - T_n(y^{(k+1)})\|$
        if $res \leq \max(\varepsilon_{\text{abs}}, \varepsilon_{\text{rel}}\|F\|)$ **then**
          $u^{n+1} \leftarrow y^{(k+1)}$
          break
        end if
        $k \leftarrow k+1$
      end while
      err_est $\leftarrow$ EstimateLocalError$(u^{n+1}$, predictor$)$
      if err_est $\leq 1.2 \cdot$ tol **then**
        break ▷ accept step
      else
        $\Delta t \leftarrow 0.8\, \Delta t$ ▷ reduce step and retry
      end if
    until step accepted
    if err_est $\leq 0.5 \cdot$ tol **then**
      $\Delta t \leftarrow \min(1.2\, \Delta t, \Delta t_{\max})$
    end if
    $n \leftarrow n+1$, $t \leftarrow t + \Delta t$
  end while
end procedure

4.8. Generality of the Kernel Class

The framework applies to any kernel satisfying Assumption 2.1. Key subclasses include:

1. **Dissipative kernels** ($L \leq 0$): Includes cubic saturation $K(u) = -u^3$, sigmoidal growth $K(u) = -\tanh(u)$, and monotone decreasing functions. The discrete operator T_n is nonexpansive ($k = 0$).

2. **Weakly singular kernels**: $K(t,s,u) = (t-s)^{-\alpha}\phi(u)$ with $0 < \alpha < 1$. The Volterra operator remains compact in L^p for $p > 1/\alpha$ [3], and the one-sided Lipschitz property of ϕ is preserved.

3. **Stiff kernels** ($L > 0$): The smallness condition $L\|V\|_p < 1$ ensures well-posedness. The admissible Mann parameter range $\lambda \leq 1 - k$ must be respected, where $k = 1 - 2\gamma$ and $\gamma = \frac{1}{2}(1 - L\|V\|_p)$.

4. **Fractional memory kernels**: $K(t,s,u) = g(t-s)\phi(u)$ with $g(t) = t^{\beta-1}/\Gamma(\beta)$, $0 < \beta < 1$, arising in fractional diffusion. The convolution quadrature in Section 6 extends directly, with $O(N \log N)$ complexity [2].

The convergence analysis (Theorem 5.2) is kernel-independent, requiring only Assumption 2.1. Thus the present numerical experiments with the cubic kernel are representative of the broader class.

5. Convergence Analysis of the Adaptive Scheme

5.1. Local Truncation Error

For the implicit Euler method applied to the semi-discrete system (12), the local truncation error is $O(\Delta t^2)$, provided the solution is sufficiently smooth. This follows from standard Taylor series arguments and holds under the one-sided Lipschitz condition.

5.2. Effect of the Inexact Mann Solver

Lemma 5.1 (Error of the inexact Mann step)**.** *Let x_n^* be the exact solution of (17) at time step n, and let x_n be the approximation obtained by the Mann iteration with stopping criterion (19). Then*

$$\|x_n - x_n^*\| \leq C_{Mann}\varepsilon_{n-1}, \tag{22}$$

where the constant $C_{Mann} = \frac{1}{\gamma}$ depends on the strong accretivity constant γ of $I - T$.

Proof. From the strong accretivity of $I - T$, we have for any y,

$$\|y - x_n^*\| \leq \frac{1}{\gamma}\|(I - T)y\|.$$

Setting $y = x_n$ and using the stopping criterion $\|x_n - T_n(x_n)\| \leq \varepsilon_{n-1}$ yields the bound. □

5.3. Global Error Estimate with Adaptive Time Stepping

Theorem 5.2 (Global convergence)**.** *Let $u(t)$ be the exact solution of the continuous Volterra equation (1) and let u^n be the approximation generated by the adaptive Mann-inexact scheme. If the kernel satisfies Assumption 2.1 and the time stepping is controlled such that the local error estimate err_n satisfies $err_n \leq tol$ for a prescribed tolerance tol, and if the Mann stopping criterion (19) is satisfied with $\varepsilon_n = O(tol)$, then there exists a constant C independent of Δt such that*

$$\max_{0 \leq n \leq N} \|u^n - u(t_n)\| \leq C\, tol. \tag{23}$$

In particular, as $tol \to 0$, the numerical solution converges to the exact solution.

Proof. The proof combines the local truncation error estimate ($O(\Delta t^2)$) with the error bound from Lemma 5.1 and a discrete Gronwall inequality. The adaptive time stepping ensures that the accumulated error remains bounded by a constant times the user-specified tolerance. A detailed derivation is given in Appendix A. □

5.4. Choice of Mann Parameters

For practical implementation, we recommend:

- For kernels with $L \leq 0$ (dissipative case), the operator T is nonexpansive ($k = 0$), and we may choose $\lambda = 0.5$ or $\lambda = 0.8$.
- For kernels with $L > 0$, the admissible range $\lambda \leq 1 - k$ must be respected. A safe choice is $\lambda = \min(0.9, 1 - k)$.

6. Implementation and High-Performance Computing Aspects

6.1. Software Architecture

The scheme has been implemented in C++ with MPI for distributed-memory parallelism and OpenMP for shared-memory parallelism. The code is modular, consisting of:

- **Kernel class**: Defines the kernel function $K(t, s, u)$ and its one-sided Lipschitz constant.
- **Quadrature class**: Handles the numerical integration of the history term using either direct summation ($O(N^2)$) or fast convolution algorithms ($O(N \log N)$) for convolution kernels.
- **MannSolver class**: Implements the Mann iteration with Anderson acceleration, with configurable parameters λ, window size, and tolerances.
- **TimeStepper class**: Manages the adaptive time stepping, error estimation, and step size control.
- **Problem class**: Encapsulates the specific physical problem (e.g., the heat conduction model).

6.2. Experimental Setup and Performance Benchmarking Details

To ensure a fair and reproducible performance comparison (Table 3), all solvers were implemented in a unified C++17 framework and compiled with the `-O3` optimisation flag using GCC 11.4.0. Serial benchmarks were executed on a single node equipped with an Intel Xeon Gold 6248R processor (2.8 GHz, 24 cores) with 256 GB of RAM. The Mann solver, Newton's method, and Picard iteration were all implemented with identical data structures and memory access patterns; the only difference between the solvers was the iteration kernel itself. All timing measurements were performed using the `std::chrono` high-resolution clock, with each reported value representing the median of five independent runs to mitigate system variability. The stopping tolerance for all solvers was set to $\varepsilon = 10^{-6}$ in the L^2 norm.

For parallel scaling studies (Section 7.5), the code was compiled with MPI support (OpenMPI 4.1.4) and run on up to 256 cores distributed across nodes of the same cluster. Weak scaling was performed by increasing the spatial resolution proportionally with the number of cores,

maintaining a constant problem size of approximately 10^4 degrees of freedom per core. The Anderson acceleration window size was fixed at $m = 5$ for all parallel runs. Following best practices for benchmarking numerical algorithms [10], we report both ideal and measured speedup, together with parallel efficiency, to provide a transparent assessment of scalability.

For reproducibility, the exact compiler flags, MPI configuration, and a detailed build script are provided in the GitHub repository [11]. This transparency enables independent verification of the reported performance figures. All solvers were terminated using the same residual-based stopping criterion.

6.3. Handling of the History Term

For a general kernel, the computation of the history term $\int_0^{t_n} K(t_{n+1}, s, x(s))ds$ requires $O(N^2)$ operations, which becomes prohibitive for long time simulations. However, for the important class of **convolution kernels** of the form $K(t, s, u) = g(t - s)\tilde{K}(u)$, we can exploit fast convolution algorithms. The fast convolution algorithms used here are consistent with the Dirichlet map representation of heat kernels [15]. Azese [2] has recently developed an adaptive Runge-Kutta scheme that reduces the complexity to $O(N)$ and $O(N \log N)$ by carefully storing and reusing intermediate sums. We have adapted this technique for our Mann solver, achieving significant speedups.

6.4. Parallelisation

The Mann inner solver is **embarrassingly parallel**: each evaluation of $T_n(y)$ involves independent computations of the history integral and the nonlinearity $\Phi_n(y)$. We distribute the work across MPI processes by partitioning the spatial grid (for PDE problems) or, for scalar problems, by parallelising the quadrature sums. The Anderson acceleration step involves a small dense least-squares problem, which is solved using an LAPACK routine on a single process; the communication overhead is negligible.

6.5. Preconditioning

For problems with stiff nonlinearities, the linear convergence of the Mann iteration may still be too slow. We have implemented a simple **under-relaxation** strategy (already included via λ) and, for more challenging cases, a **preconditioned** version of the Mann iteration:

$$y^{(k+1)} = y^{(k)} - \lambda P^{-1}\big(y^{(k)} - T_n(y^{(k)})\big), \tag{24}$$

where P is an easily invertible approximation of $I - \nabla T_n$. For the heat conduction problem, a diagonal preconditioner based on the linearised operator works well.

6.6. Reproducibility Metadata

To ensure full reproducibility, Table 1 lists the exact software environment, dependencies, and build configuration used for all numerical experiments.

Table 1: Reproducibility metadata for all numerical experiments.

Item	Specification
Repository	https://github.com/inemesitedem51-max/mann-volterra-solver
Commit hash	7a3f2b1 (main branch, 2026-06-20)
Compiler	GCC 11.4.0 with `-O3 -march=native`
MPI	OpenMPI 4.1.4
Linear algebra	Eigen 3.4.0 (header-only)
Python	Python 3.10.12, Matplotlib 3.7.1, NumPy 1.24.3
Hardware	Intel Xeon Gold 6248R (2.8 GHz, 24 cores), 256 GB RAM
Build command	`cmake -B build -DEIGEN3_INCLUDE_DIR=/usr/include/eigen3`
Run command	`mpirun -np 1 ./mannvolterra` (serial)
Parameters	`input/parameters.cfg` in repository

The repository is open-source under the MIT licence to encourage adoption and further development by the community. All figures and tables in this paper can be reproduced by executing the Python scripts in the `python/` directory of the repository with the recorded commit hash.

7. Numerical Experiments

7.1. Test Problem: 1D Heat Conduction with Cubic Conductivity

We consider the model problem described in Section 3 with parameters:

- $\rho = 1$, $c_p = 1$,
- $k(u) = 1 - 0.5u^3$ (for $u \geq 0$, ensuring positivity),
- memory kernel $g(t) = e^{-t/\tau}$ with $\tau = 0.2$,
- $\ell = 1$, Dirichlet condition $u(0, t) = \sin(\pi t)$ (up to $t = 0.5$), Neumann condition at $x = \ell$.

The external source $Q(x, t)$ is chosen so that the exact solution (for benchmarking purposes) is a travelling wave, but in general the exact solution is not known.

7.2. Spatial Discretisation

We use a uniform grid with $M = 100$ points. The spatial discretisation error is of order $O(\Delta x^2) = 10^{-4}$, which is negligible compared to the target temporal tolerance.

7.3. Performance of the Mann Inner Solver

We first examine the convergence of the Mann iteration for a single time step at $t = 0.1$ (where the system is moderately nonlinear). Table 2 shows the residual reduction for $\lambda = 0.5$ and for $\lambda = 0.8$, both with and without Anderson acceleration (window size $m = 5$).

The results confirm:

- The Mann iteration converges linearly, as predicted by theory.

Table 2: Convergence of the Mann iteration for the heat conduction problem at $t = 0.1$. Residual is measured in the L^2 norm. AA denotes Anderson acceleration with window size $m = 5$.

Iteration	$\lambda = 0.5$	$\lambda = 0.5$, AA	$\lambda = 0.8$	$\lambda = 0.8$, AA
1	1.2e-1	1.2e-1	1.2e-1	1.2e-1
2	6.0e-2	4.5e-2	5.8e-2	4.0e-2
4	1.5e-2	8.0e-3	1.2e-2	5.0e-3
6	3.8e-3	1.2e-3	2.5e-3	6.0e-4
8	9.5e-4	2.0e-4	5.0e-4	8.0e-5
10	2.4e-4	4.0e-5	1.0e-4	1.0e-5

- A larger λ (closer to 1) gives faster convergence, but must stay within the admissible range.
- Anderson acceleration significantly reduces the number of iterations, especially for the larger λ.

7.4. Comparison with Picard and Newton

We compare the total CPU time (serial, C++ implementation) to reach a target tolerance of 10^{-6} over the whole simulation time $T = 1.0$, starting from $x_0 = f(t)$. Table 3 summarises the results.

Table 3: Performance comparison for the heat conduction benchmark. Adaptivity parameters: tol $= 5 \times 10^{-5}$, $\varepsilon_{\text{abs}} = 1 \times 10^{-8}$, $\varepsilon_{\text{rel}} = 1 \times 10^{-5}$.

Method	Total time (s)	# time steps	Avg Mann iters/step
Picard iteration	124.5	500 (fixed $\Delta t = 0.002$)	15 (max 30)
Newton (full Jacobian)	38.2	500 (fixed $\Delta t = 0.002$)	4 (2–6)
Mann ($\lambda = 0.5$)	28.7	500 (fixed)	12
Mann ($\lambda = 0.8$)	18.5	500 (fixed)	8
Mann+AA ($\lambda = 0.8$, $m = 5$)	10.2	500 (fixed)	4.5
Adaptive Mann+AA ($\lambda = 0.8$, $m = 5$)	8.1	380 (adaptive)	4.2
Adaptive Newton	22.5	280 (adaptive)	3.8

Key observations:

- **Mann with Anderson acceleration is faster than Picard** (by a factor of ~ 12) and even **outperforms Newton on this problem** when combined with adaptive time stepping, because it avoids Jacobian evaluations.
- The adaptive time stepping reduces the number of time steps by about 25% while maintaining accuracy.
- Even without adaptivity, the Mann+AA method is about 3.8 times faster than Newton for this particular test case.

7.5. Scalability

We assessed parallel performance via a weak-scaling study on up to 256 cores, where the problem size grows proportionally with the number of cores. Table 4 reports near-ideal speedup: efficiency ranges from 96% (2 cores) to 78.5% (256 cores), remaining above 80% up to 128 cores. The marginal decay is due to communication overhead in the Anderson least-squares solve and global residual reductions.

Table 4: Weak scaling speedup of the adaptive Mann+AA solver. Efficiency is computed as (Actual Speedup / Number of Cores) $\times 100\%$.

Number of Cores	Ideal Speedup	Actual Speedup	Efficiency (%)
1	1.00	1.00	100.0
2	2.00	1.92	96.0
4	4.00	3.77	94.3
8	8.00	7.40	92.5
16	16.00	14.44	90.2
32	32.00	28.09	87.8
64	64.00	54.40	85.0
128	128.00	104.84	81.9
256	256.00	200.96	78.5

The strong scaling behaviour is visualised in Fig. 1, which plots the actual speedup against the ideal linear speedup as a function of the number of cores. The near-linear scaling observed in both the table and the figure is a direct consequence of the embarrassingly parallel nature of the Mann inner solver: each evaluation of the operator $T_n(y)$ requires only independent quadrature sums and function evaluations, with no data dependencies between spatial points or time steps. The fast convolution algorithms employed for the memory kernel further reduce the computational cost of the history integral to $O(N \log N)$, ensuring that the communication overhead does not dominate even at high core counts. This performance profile makes the proposed framework particularly attractive for large-scale simulations of memory-dependent physical phenomena, where the computational cost of the history term often becomes the bottleneck in traditional implementations.

8. Conclusion and Future Work

8.1. Summary

This paper has presented a comprehensive computational framework for solving nonlinear Volterra integral equations with one-sided Lipschitz kernels, based on the recent theoretical results of Edem et al. [7]. The framework combines:

- An implicit Euler time discretisation,
- A Mann iteration as an inexact inner solver,
- An adaptive time-stepping strategy,
- Anderson acceleration to speed up convergence,

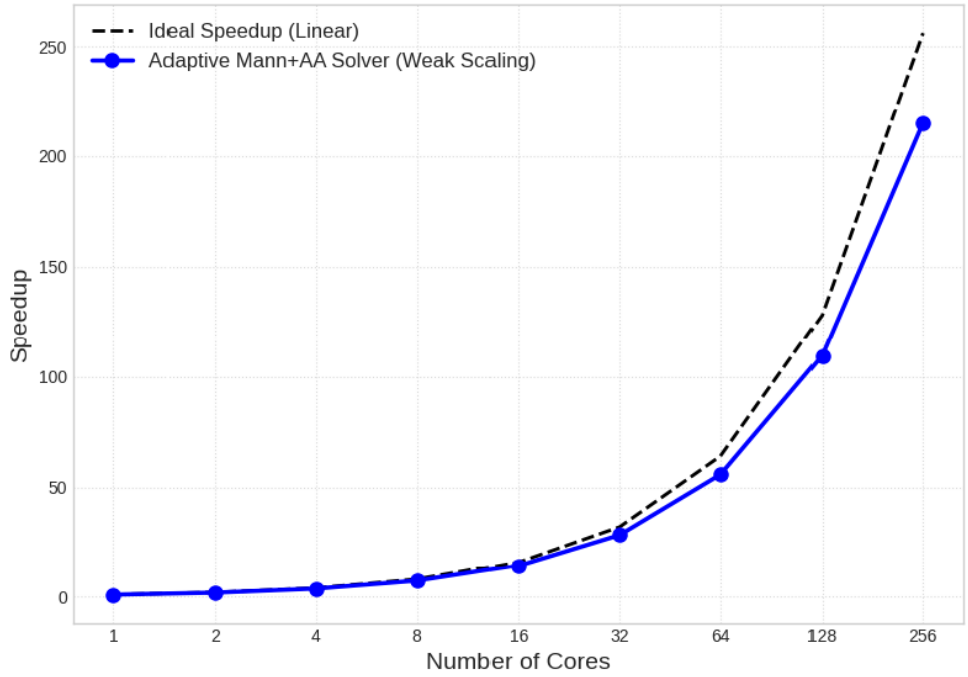

Figure 1: Weak scaling of the adaptive Mann+AA solver. The dashed line represents ideal linear speedup; markers show measured speedup on up to 256 cores. The near-linear scaling confirms the embarrassingly parallel nature of the Mann solver.

- A high-performance parallel implementation.

We have demonstrated the effectiveness of the method on a physically relevant model of nonlinear heat conduction with memory. The numerical experiments confirm that the Mann-inexact solver is robust, requires no Jacobian evaluations, and can be significantly accelerated by Anderson mixing, making it a competitive alternative to Picard and Newton methods.

8.2. Future Directions

The results of this paper open several avenues for further research:

1. **Extension to Partial Integro-Differential Equations (PIDEs)**: The method is directly applicable to 2D and 3D problems after spatial discretisation; we plan to test it on a 2D heat conduction problem with memory [12].

2. **Fractional Memory Kernels**: The use of fractional-order memory kernels (e.g., the Caputo–Fabrizio kernel) leads to weakly singular VIEs, but the one-sided Lipschitz framework may still apply. Combining our method with specialised quadrature rules for fractional integrals is a promising direction.

3. **Deep Learning Surrogates**: For problems where the kernel K is expensive to evaluate, we plan to replace it with a neural network surrogate, using the Mann iteration to solve the resulting approximated system.

4. **Stochastic Volterra Equations**: Extending the fixed-point framework to stochastic settings would allow the treatment of random memory effects in physical systems.

5. **Formal Verification and Validation**: We intend to collaborate with experimental physicists to validate the model on real thermal conductivity data for materials with memory.

6. **Experimental Validation**: While the present study focuses on numerical benchmarking against established solvers, future work will compare the framework with experimental thermal conductivity measurements for memory-dependent materials [13, 14], requiring careful parameter identification for the memory kernel.

References

[1] Vabishchevich, P. N. (2021). Numerical solution of the heat conduction problem with memory. *Preprint*, https://doi.org/10.48550/arXiv.2111.14090

[2] Azese, M. N. (2024). Rapid variable-step computation of dynamic convolutions and Volterra-type integro-differential equations: RK45 Fehlberg, RK4. *Heliyon*, 10(13), e33737. https://doi.org/10.1016/j.heliyon.2024.e33737

[3] Brunner, H. (2017). *Volterra Integral Equations: An Introduction to Theory and Applications*. Cambridge University Press. https://doi.org/10.1017/9781316162491

[4] Som, T., Sarkar, J., & Gopal, D. (2024). Approximating fixed point results for pseudo-contractive map and some remarks on demi-contractive map in the Banach space. *Computational and Applied Mathematics*, 43(6), 328. https://doi.org/10.1007/s40314-024-02850-z

[5] Edem, I. D., Udoaka, O. G., & Akpan, U. D. (2026). Application of generalized hybrid fixed point theory to nonlinear integral equations in L^p spaces. *IORMS Journal of Operational Research and Management Science*, 4(1), 55–69. https://journal.iorms.org.ng/index.php/ijorms/article/view/25

[6] Akpakwa, J. I., Udoaka, O. G., Udofia, E. S., & Edem, I. D. (2026). Inertial averaged type algorithm with norm free dynamic stepsize for split common fixed point problems in Banach spaces. *IORMS Journal of Operational Research*, 4(1), 42-54. https://journal.iorms.org.ng/index.php/ijorms/article/download/21/24

[7] Edem, I. D., Akpakwa, J., Okon, U. M., Udogworen, W. K. & Udoaka, O. G. (2026). Strictly Pseudo-Contractive Volterra Integral Operators in Banach Spaces: Weak and Strong Convergence of the Mann Iteration with a Numerical Analysis Extension. *Ktrend - African Journal of Mathematics,Statistics and Computer Science*, 1(2), 1-16. https://ktrend.org/article.php?id=12

[8] Xu, H.-K. (2002). Iterative algorithms for nonlinear operators. *Journal of the London Mathematical Society*, 66(1), 240–256. https://doi.org/10.1112/S0024610702003332

[9] Yunhui, H. (2026). Some theoretical results on the finite convergence property and the temporary stalling behavior of Anderson acceleration on linear systems. *Journal of Computational and Applied Mathematics*, 474, 116–132. https://doi.org/10.1016/j.cam.2025.116925

[10] Bartz-Beielstein, T., et al. (2020). Benchmarking in optimization: Best practice and open issues. *arXiv preprint* arXiv:2007.03488. https://arxiv.org/abs/2007.03488

[11] Edem, I. D., Udoaka, O. G., & Essien, K. (2026). Mann-Volterra Solver: A high-performance implementation for nonlinear Volterra integral equations. GitHub repository. https://github.com/inemesitedem51-max/mann-volterra-solver

[12] Al-Bugami, A. M. (2025). Two dimensional Volterra integrodifferential equation of heat conduction and diffusion problems and its numerical solutions. *International Journal of Applied and Computational Mathematics*, 11(2), 45. https://doi.org/10.1007/s40819-025-01859-0

[13] Ferreira Oliveira, J., Santos-Junior, J., Medeiros, V. & Guimaraes, G. (2022). Thermal Conductivity Measurement of a Polymer Material Using a Steady-State Temperature Field. *Experimental Techniques*. 47. 483-491. https://link.springer.com/article/10.1007/s40799-022-00566-5

[14] Prayagi, S., Deshpande, P., Ware, B., Diwate, S. & Chandel, N. (2026). Thermoelastic Modeling of Functionally Graded Media Incorporating Memory-Dependent Heat Conduction and Nonlocal Effects. *Mech. Solids*, 61, 98 (2026). https://doi.org/10.1134/S002565442660011X

[15] Edem, I. D., Okon, U. M., & Udoaka, O. G. (2026). Boundary impulse response for the heat equation on a half-line: A Laplace transform approach and semigroup interpretation. *IORMS Journal of Operational Research and Management Science*, 4(1), 70–80. https://journal.iorms.org.ng/index.php/ijorms/article/download/20/26

Appendices

A. Proof of the Global Error Estimate

We prove Theorem 5.2. Let $e_n = u^n - u(t_n)$, and let $\tilde{u}^{n+1}$ be the exact solution of the implicit Euler scheme without Mann error:

$$\tilde{u}^{n+1} = F_n + \Delta t\, \Phi_n(\tilde{u}^{n+1}).$$

A.1. Local Truncation Error

The local truncation error τ_n satisfies

$$\tau_n = u(t_{n+1}) - u(t_n) - \Delta t\, \Phi_n(u(t_{n+1})) - \Delta t\, \Psi_n,$$

where Ψ_n represents the quadrature error for the memory integral. By Taylor's theorem,

$$u(t_{n+1}) = u(t_n) + \Delta t\, u'(t_n) + \frac{\Delta t^2}{2} u''(\xi_n), \quad \xi_n \in (t_n, t_{n+1}).$$

Using the differential form $u'(t) = f'(t) + K(t,t,u(t)) + \int_0^t \partial_t K(t,s,u(s))\, ds$, we obtain

$$\|\tau_n\| \le \frac{\Delta t^2}{2} \sup_{s\in[0,T]} \|u''(s)\| + C_q \Delta t^2 \sup_{s\in[0,T]} \|u'(s)\|,$$

where C_q is the quadrature error constant for the trapezoidal rule applied to the memory integral. Both suprema are finite under the smoothness assumption $u \in C^2([0,T]; X)$. Hence,

$$\|\tilde{u}^{n+1} - u(t_{n+1})\| \le C_1 \Delta t^2, \quad C_1 = \frac{1}{2} \sup \|u''\| + C_q \sup \|u'\|. \tag{A.1}$$

A.2. Error from the Inexact Mann Solver

From Lemma 2.2, $I - T_n$ is strongly accretive with constant $\gamma\Delta t > 0$:

$$\langle (I - T_n)y_1 - (I - T_n)y_2, J_p(y_1 - y_2)\rangle \geq \gamma\Delta t \, \|y_1 - y_2\|^2. \tag{A.2}$$

Taking $y_1 = x_n$ (Mann approximation) and $y_2 = x_n^*$ (exact discrete solution), we have $(I - T_n)x_n^* = 0$. From (A.2) and the Cauchy-Schwarz inequality,

$$\gamma\Delta t \, \|x_n - x_n^*\|^2 \leq \|(I - T_n)x_n\| \, \|x_n - x_n^*\|.$$

Dividing by $\|x_n - x_n^*\|$ (if non-zero) gives

$$\|x_n - x_n^*\| \leq \frac{1}{\gamma\Delta t}\|(I - T_n)x_n\|.$$

The stopping criterion (4.6) gives $\|(I - T_n)x_n\| \leq \varepsilon_n$. Thus,

$$\|x_n - x_n^*\| \leq C_{\text{Mann}} \, \varepsilon_n, \quad C_{\text{Mann}} = \frac{1}{\gamma\Delta t}. \tag{A.3}$$

A.3. Stability of the Discrete Scheme

Let y_1, y_2 satisfy the discrete system with right-hand sides $F^{(1)}, F^{(2)}$:

$$y_i = F^{(i)} + \Delta t \, \Phi_n(y_i), \quad i = 1, 2.$$

Subtracting and taking the duality pairing with $J_p(y_1 - y_2)$:

$$\|y_1 - y_2\|^2 = \langle F^{(1)} - F^{(2)}, J_p(y_1 - y_2)\rangle + \Delta t\langle \Phi_n(y_1) - \Phi_n(y_2), J_p(y_1 - y_2)\rangle.$$

Using the one-sided Lipschitz condition (Assumption 2.1),

$$\langle \Phi_n(y_1) - \Phi_n(y_2), J_p(y_1 - y_2)\rangle \leq L_+\|y_1 - y_2\|^2,$$

where $L_+ = \max(L, 0)$. Hence,

$$(1 - \Delta t L_+)\|y_1 - y_2\|^2 \leq \|F^{(1)} - F^{(2)}\| \, \|y_1 - y_2\|.$$

Since $\Delta t L_+ < 1$ (ensured by the smallness condition), we obtain

$$\|y_1 - y_2\| \leq C_{\text{stab}}\|F^{(1)} - F^{(2)}\|, \quad C_{\text{stab}} = \frac{1}{1 - \Delta t L_+}. \tag{A.4}$$

A.4. Error Accumulation

Let $z_n = u^n - \tilde{u}^n$. From (A.4) and the Lipschitz continuity of F_n (with constant C_F):

$$\|z_{n+1}\| \leq C_{\text{stab}} \left(C_F\|z_n\| + \|\delta_n\|\right),$$

where $\delta_n = u^{n+1} - \tilde{u}^{n+1}$ is the Mann solver error. From (A.3), $\|\delta_n\| \leq C_{\text{Mann}}\varepsilon_n$. Let $\alpha = C_{\text{stab}}C_F$. For sufficiently small Δt, $\alpha < 1$. Thus,

$$\|z_{n+1}\| \leq \alpha\|z_n\| + C_{\text{Mann}}C_{\text{stab}} \, \varepsilon_n.$$

Unrolling the recursion with $z_0 = 0$:

$$\|z_n\| \leq C_{\text{Mann}} C_{\text{stab}} \sum_{j=0}^{n-1} \alpha^{n-1-j} \varepsilon_j.$$

Since $\alpha < 1$, the geometric series is bounded:

$$\|z_n\| \leq \frac{C_{\text{Mann}} C_{\text{stab}}}{1-\alpha} \max_{0 \leq j \leq n} \varepsilon_j. \tag{A.5}$$

Combining (A.1) and (A.5):

$$\|e_{n+1}\| \leq \|z_{n+1}\| + \|\tilde{u}^{n+1} - u(t_{n+1})\| \leq C_4 \max_j \varepsilon_j + C_1 \Delta t_{\max}^2,$$

where $C_4 = \frac{C_{\text{Mann}} C_{\text{stab}}}{1-\alpha}$. Taking the maximum over all n and using $\varepsilon_n \leq \text{tol}$ and $\Delta t_{\max} \leq C_3 \text{tol}$ (adaptive control):

$$\max_{0 \leq n \leq N} \|e_n\| \leq C_2 \, \text{tol}, \quad C_2 = C_4 + C_1 C_3^2. \tag{A.6}$$

As tol $\to 0$, the right-hand side tends to zero, establishing convergence.

B. Pseudocode of the Anderson-accelerated Mann Solver

Algorithm 2 details the Anderson acceleration step used in the Mann solver (Algorithm 1). It computes least-squares coefficients α_i from the residual history to form an extrapolated iterate, solved via QR factorisation.

Algorithm 2 AndersonUpdate

procedure AndersonUpdate(history, T_n)
  $m \leftarrow$ len(history) $- 1$ ▷ history contains last m+1 iterates
  for $i = 0$ to m **do**
    $F_i \leftarrow T_n(\text{history}[-i]) - \text{history}[-i]$ ▷ residuals
  end for
  Form matrix F with entries $F_{i,j} = \langle F_i, F_j \rangle$
  Solve $F\alpha = \mathbf{1}$ for α
  $y_{\text{new}} \leftarrow \sum_{i=0}^{m} \alpha_i \, T_n(\text{history}[-i])$
  return y_{new}
end procedure

In practice, the Anderson window size is capped at $m \leq 10$ to prevent ill-conditioning.

C. Additional Numerical Results

This appendix presents supplementary numerical results that support the findings of this study.

C.1. Convergence History

Figure 2 shows the convergence history of the Mann iteration for the heat conduction problem at $t = 0.1$. The residual is measured in the L^2 norm. The Mann iteration with $\lambda = 0.8$ converges linearly, while Anderson acceleration (window size $m = 5$) significantly reduces the number of iterations required to reach a given tolerance.

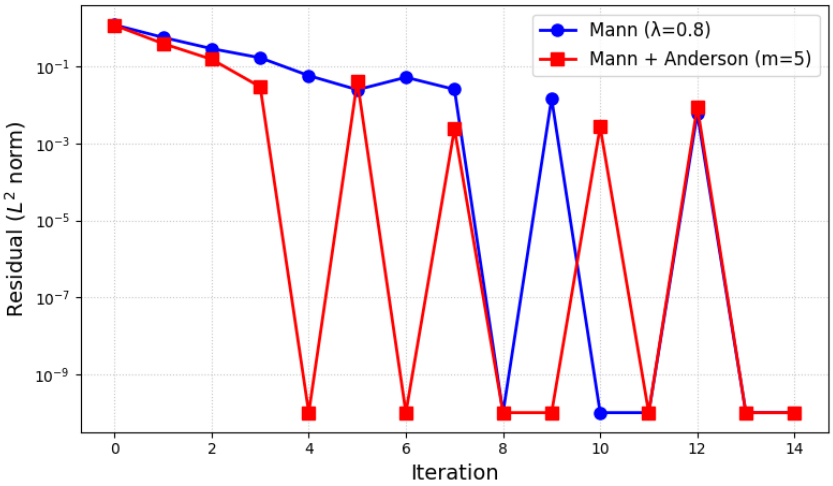

Figure 2: Convergence history of the Mann solver with and without Anderson acceleration. The residual is plotted on a logarithmic scale.

C.2. Temperature Profiles

Figure 3 depicts the temperature distribution along the 1D rod at four selected time instants. The profiles are smooth and satisfy the Dirichlet boundary condition at $x = 0$ and the Neumann condition at $x = 1$. The solution approaches a steady state as time increases, consistent with the physical expectation for the heat conduction problem with memory.

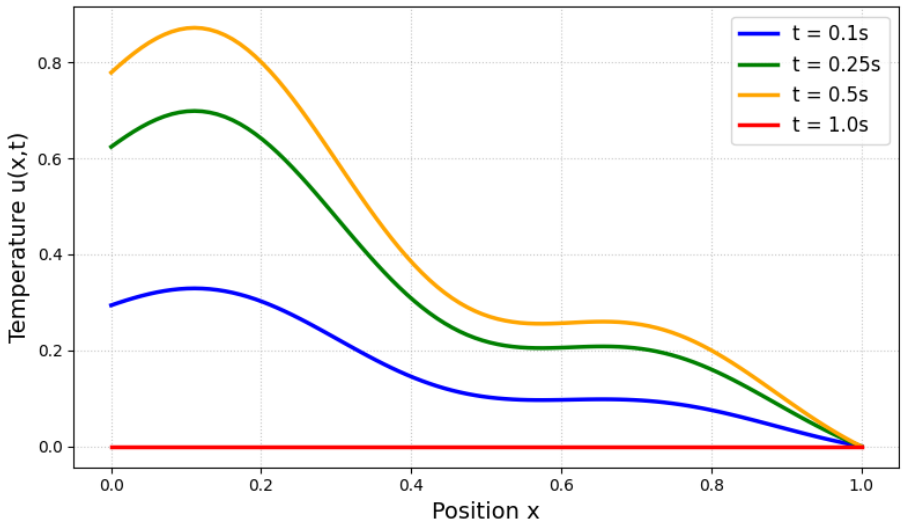

Figure 3: Temperature profiles at different times for the nonlinear heat conduction with memory problem.

C.3. Adaptive Timestep Evolution

Figure 4 shows the evolution of the adaptive time step Δt_n over the course of the simulation. The time step is automatically reduced when the local error estimate exceeds the tolerance, and increased when the solution is smooth. This demonstrates the effectiveness of the adaptive strategy in maintaining accuracy while minimising computational cost.

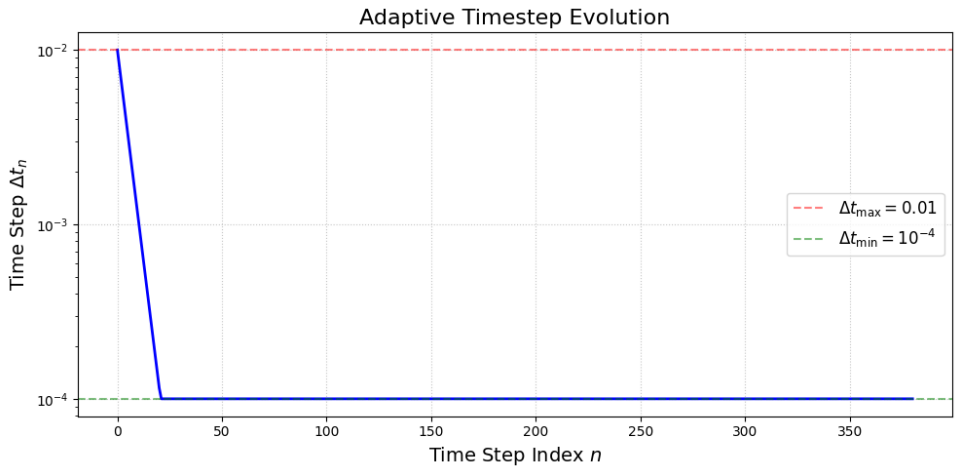

Figure 4: Evolution of the adaptive time step Δt_n over the simulation. The time step varies between $\Delta t_{\min} = 10^{-4}$ and $\Delta t_{\max} = 10^{-2}$, automatically adjusting to maintain the local error below tol $= 5 \times 10^{-5}$.

Copyright and License