Jurnal MIPA dan Pembelajarannya
e-issn: 2798-0634, email: jmipap.journal@um.ac.id

| VOL | ISS | YEAR | DOI |
|---|---|---|---|
| 6 | 7 | 2026 | 10.17977/um067.v6.i7.2026.2 |


# A PICARD–INTEGRATING FACTOR ITERATIVE SCHEME FOR NONLINEAR FRACTIONAL DIFFERENTIAL EQUATIONS WITH ERROR AND CONVERGENCE ANALYSIS

Muayyad Mahmood Khalil

Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq
*Corresponding author, email: medomath80@tu.edu.iq

**Keywords**
Picard Iteration
Integrating Factor
Convergence Analysis
Absolute Difference

**Abstract**
In this paper, an iterative algorithm, namely Picard–integrating factor, is developed and analyzed for a class of nonlinear fractional differential equations. Fractional models exhibit a nonlocal and memory-dependent structure, which typically does not have an exact closed-form solution, motivating their study. This proposed scheme is based on the Riemann–Liouville fractional integral and the use of an integrating factor in the equivalent fixed-point formulation, together with Picard-type recursive construction. The method is formulated in the presence of the Lipschitz condition on the nonlinear term, and a convergence result is presented in a weighted supremum norm to make the assumptions under which the iteration is a contraction clear. Three nonlinear fractional initial value problems are analyzed: one Riccati-type model, one Bernoulli-type model, and a fourth model, which is considered with a known closed-form solution, to see the true error directly. It presents the numerical results for two cases of $\alpha = 1$ and $\alpha = 0.5$. In all examples, the absolute value of the difference between the two Picard approximations and the Picard – integrating factor approximation is tabulated and plotted to visualize their accuracy; residual diagnostics are calculated for Riccati and Bernoulli-type examples, and the true absolute errors are reported for all the examples for which the solution is known. On a quantitative level, for the $\alpha = 1$ case, the mean absolute deviation of the respective approximations is minimized when using the Picard–integrating factor approximation, being about 54.4% lower than that of the Picard approximation, while the residual diagnostics present lower endpoint residuals for the nonlinear Riccati and Bernoulli-type tests. The results suggest that the formulation with the integration factor gives a semi-analytical approximation method that is straightforward, structured, and convenient, without requiring the calculation of Adomian polynomials or correction functionals.

## 1. Introduction

Fractional differential equations provide an effective modelling framework for systems with memory, hereditary behaviour, and nonlocal temporal dependence (Lazarevic et al., 2014; Ray et al., 2014). In contrast with classical integer-order differential equations, fractional models can capture inherited effects through nonlocal operators, which makes them suitable for many nonlinear physical and applied problems (Acharya et al., 2023; Koning et al., 2015). From a physical viewpoint, the fractional memory kernel represents the accumulated influence of the previous states of the system; therefore, the present response in viscoelastic, diffusion-type, or hereditary dynamical processes is affected not only by the local value at $x$ but also by the history accumulated over the interval $[0, x]$. This accumulated memory may amplify nonlinear deviations during successive approximation, particularly near the terminal part of the computational interval.

Many nonlinear fractional differential equations do not admit exact analytical solutions. Consequently, semi-analytical and numerical approaches such as the Adomian decomposition method, homotopy-based methods, variational iteration techniques, Haar wavelet operational matrices, neural-network methods, Picard iteration, and Taylor-basis approximations have been developed to approximate fractional models (Ali & Ziada, 2021; Chandel et al., 2017; Daraghmeh et

al., 2020; Das & Gupta, 2011; Elsaid, 2011; Jafari & Daftardar-Gejji, 2006; Kammanee, 2021; Odibat et al., 2010; Qu & Liu, 2015). These methods are especially important when nonlinear terms prevent direct closed-form integration (Acharya et al., 2023; Ilejimi et al., 2019).

Even though a lot of works have been devoted to develop the nonlinear fractional differential equations decomposition and homotopy methods, the practical applications of both of these methods to the nonlinear fractional differential equations still need to build out the nonlinear correction terms, Adomian polynomials, auxiliary homotopy parameters, or repeated integral components. When more approximation terms are retained, this algebraic workload will increase and when evaluating the solution repeatedly for measurement points defined on a numerical grid it may decrease the computational transparency (Ali & Ziada, 2021; Das & Gupta, 2011; Elsaid, 2011; Jafari & Daftardar-Gejji, 2006; Odibat et al., 2010). The classical Picard iteration is easier but when the memory is fractional, successive approximations can have larger deviations towards the end of the interval due to nonlinear and fractional-memory effects (Lyons et al., 2017). This research gap can therefore not only be understood as formulating a new iteration scheme that has an organized form as a Picard iteration, but also it positions a demand for just reducing the computational complexity related to decomposition and homotopy procedures, and also for maintaining the recursive simplicity of Picard iteration. The linear contribution is incorporated into a bounded weight structure as a formal integrating factor of the fixed-point map and the nonlinear term is directly semi-analytically updated recursively. In this formulation, the role of μ(x) is to smooth the propagation of the linear memory contribution in the iteration, and to be able to decrease the actual Lipschitz load of the recursive map when the weight is well conditioned, which can then have a beneficial effect on the stability of endpoint residual behaviour. This will yield an interpretation of the scheme that is mathematically clear and can be evaluated using absolute-difference analysis, residual analysis and ADM analyses (Youssef et al., 2011).

The paper has the following main contributions: The setting for the fractional operator is explicitly mentioned, and the use of the Caputo derivative, Riemann–Liouville integral and functional assumptions are used equiprobably. Secondly, the Picard – integrating factor iteration is written as a mapping of themselves which is based on a Picard type iteration but takes into account the effect of the linear coefficient multiplying the Picard iteration as an integrating factor. Thirdly it is stressed that, as well as being an estimator of the fractional derivative exponent, μ(x) is a bounded device to be used within the fixed-point operator, rather than a proposition that the classical integer-order product rule will hold for fractional order derivatives. This distinction is important in providing an analytical basis for the proposed framework, and allows the proposed weighting mechanism to be distinct from formal product-rule assumptions. Third, a convergence statement which is presented in a contraction form is done under a Lipschitz condition. Fifth, the proposed scheme is compared with the classical Picard iteration, and the Adomian decomposition method is reported, both in terms of absolute-difference diagnostics and residual diagnostics, to check the accuracy and behaviour of the scheme proposed.

In the remainder of this paper, we discuss the following: The fractional-operator framework along with the Caputo derivative, Riemann–Liouville integral and functional assumptions that are needed for the proposed formulation are introduced in Section 2. In Sec. 3 the Picard – integrating factor scheme is developed, its algorithmic structure is mentioned and the measures of convergence and comparison. The numerical experiments and comparison of the Picard iteration, the Adomian decomposition method, the absolute-difference diagnostics, the "residual analysis" and the graphical interpretation are reported in section 4. The major results that are obtained and the importance of computing the proposed methodology are summarised in Section 5.

## 2. Fractional operator framework and preliminaries

The analysis is carried out on the finite interval

$$0 < \alpha \leq 1, \qquad x \in [0, T].$$

The unknown function is assumed to satisfy

$$y \in AC[0, T], \qquad y(0) = y_0,$$

where $(\mathrm{AC}[0,T])$ denotes the space of absolutely continuous functions on $([0,T])$. This regularity is required in order to use the standard inverse relation between the Riemann–Liouville fractional integral and the Caputo derivative, namely

$$I^{\alpha}D_{C}^{\alpha}y(x) = y(x) - y(0) = y(x) - y_0, \qquad 0 < \alpha \leq 1.$$

The differential operator in the initial value problem is taken in the Caputo sense, while the Riemann–Liouville fractional integral is used as the inverse-type integral operator in the Picard construction. Thus, the model form is written as

$$D_{C}^{\alpha}y(x) = g\big(x, y(x)\big), \qquad y(0) = y_0.$$

The nonlinear term is assumed to satisfy

$$g \in C([0,T] \times \mathbb{R}, \mathbb{R})$$

and to be Lipschitz continuous with respect to the dependent variable; that is, there exists a constant ($L$>0) such that

$$|g(x,u) - g(x,v)| \leq L|u - v|, \qquad x \in [0,T], \quad u, v \in \mathbb{R}.$$

When a linear coefficient is present in the weighted formulation, it is assumed that

$$p \in C[0,T], \qquad \| p \|_{\infty} < \infty.$$

The Riemann–Liouville fractional integral of order () is defined by (Koning et al., 2015)

$$I^{\alpha}f(x) = \frac{1}{\Gamma(\alpha)}\int_{0}^{x}(x-s)^{\alpha-1}f(s)\,ds, \qquad \alpha > 0.$$

The main computational identity used for powers is (Koning et al., 2015)

$$I^{\alpha}x^{n} = \frac{\Gamma(n+1)}{\Gamma(n+1+\alpha)}x^{n+\alpha}, \qquad n > -1.$$

These assumptions provide the functional setting needed to control the distance between successive Picard approximations and to establish the contraction property of the proposed iteration (Lyons et al., 2017).

## 3. Proposed Picard–integrating factor scheme

Consider the fractional initial value problem (Lyons et al., 2017)

$$D^{\alpha}y(x) + p(x)y(x) = g\big(x, y(x)\big), \qquad y(x_0) = y_0.$$

Let

$$\mu(x) = \exp\big(\int p(x)\, d^{\alpha}x\big)$$

be the formal fractional integrating factor associated with the linear part, following the integrating-factor motivation used in Picard fixed-point constructions (Youssef et al., 2011). The Picard–integrating factor scheme is then written in the fixed-point form

In the fractional setting, the integrating factor is not used here to claim a classical product-rule equivalence identical to the integer-order case. Instead, it is used as a weighting device inside the fixed-point map. In this interpretation, the factor is written formally as

$$\mu(x) = \exp\big(I^{\alpha}p(x)\big).$$

and the corresponding weighted fixed-point operator can be expressed as

$$\mathcal{T}y(x) = y_0 + \mu(x)^{-1}I^{\alpha}\big[\mu(s)g\big(s, y(s)\big)\big](x).$$

Consequently, the validity of the procedure is assessed through the contraction property of this operator rather than through an unrestricted classical product rule for fractional derivatives.

## Justification of the weighting factor

The particular choice of the fractional integrating factor is motivated by its ability to absorb the explicitly linear contribution into a bounded weight before the Picard recursion is applied. In the classical Picard formulation, the linear term is treated as part of the right-hand side, so the corresponding fixed-point map can be written as

$$(\mathcal{T}_P y)(x) = y_0 + I^{\alpha}\big[g\big(x, y(x)\big) - p(x)y(x)\big].$$

If the nonlinear part is Lipschitz continuous with constant $L_g$ and p is bounded on $[0, T]$, this formulation leads to the heuristic contraction factor

$$q_P = \frac{\big(L_g + \| p \|_{\infty}\big)T^{\alpha}}{\Gamma(\alpha + 1)}.$$

Thus, the coefficient $p(x)$ contributes directly to the Lipschitz bound of the classical Picard operator. The proposed scheme introduces the weight

$$\mu(x) = \exp\big(I^{\alpha}p(x)\big).$$

and rewrites the iteration in the weighted form

$$\big(\mathcal{T}_{\mu} y\big)(x) = \mu^{-1}(x)\big[y_0\mu(0) + I^{\alpha}(\mu(x)g\big(x, y(x)\big))\big].$$

This construction removes the explicit linear contribution $p(x)y$ from the nonlinear Picard kernel and transfers its effect to the bounded multiplier $\mu$. Under the boundedness assumptions imposed above, the weighted operator satisfies the estimate

$$\| \mathcal{T}_{\mu}u - \mathcal{T}_{\mu}v \|_{\infty} \leq \frac{\| \mu^{-1} \|_{\infty} \| \mu \|_{\infty} L_g T^{\alpha}}{\Gamma(\alpha + 1)} \| u - v \|_{\infty}.$$

where

$$q_{\mu} = \frac{\kappa_{\mu} L_g T^{\alpha}}{\Gamma(\alpha + 1)}, \qquad \kappa_{\mu} = \| \mu^{-1} \|_{\infty} \| \mu \|_{\infty}.$$

Consequently, the weighting is beneficial whenever the condition number of the weight remains moderate and the inequality

$$\kappa_{\mu} L_g < L_g + \| p \|_{\infty}.$$

holds. In this situation, the weighted Picard operator has a smaller effective contraction factor than the classical Picard map. Until now, this has been the main reason for the adoption of this approach to using.$\mu(x) = \exp\big(I^{\alpha}p(x)\big)$: the factor does not replace the fractional product rule, but it reorganizes the linear contribution so that the nonlinear kernel driving the recursion has a reduced effective Lipschitz load. Computationally, this explains why the proposed formulation can remain competitive with ADM while avoiding the repeated construction of Adomian polynomials.

$$y_{n+1}(x) = y_0 K\mu^{-1}(x) + \mu^{-1}(x) I^{\alpha}\big[\mu(s)g\big(s, y_n(s)\big)\big],$$

where $\mathrm{K} = \mu(\mathrm{x}_0)$. In the computations below, the iteration starts from the initial value $y_0$ and generates successive approximations $y_1, y_2, \ldots$ .

## 3.1 Algorithmic form

## Algorithm 1. Picard–integrating factor iteration

Implementation inputs are explicitly defined as: fractional order, interval, initial value, functions, maximum iteration number, and tolerance.

$$\|y_{n+1} - y_n\|_\infty < \varepsilon, \qquad n = 0,1, \dots, N_{\max} - 1.$$

Specify the fractional order $\alpha$, the interval $[0, T]$, the initial value $y(x_0) = y_0$, and the functions $p(x)$ and $g(x, y)$.

Compute the integrating factor $\mu(x)$.

Choose the initial approximation $y_0(x) = y_0$.

For $n = 0,1,2, \dots, N - 1$, compute $y_{n+1}(x)$ from the fixed-point formula.

Evaluate the approximate solution at the selected grid points.

Compute the absolute-difference measure between the classical Picard approximation and the Picard–integrating factor approximation.

Numerical evaluation of the fractional integral. In the computational implementation, the fractional integral appearing in Step 4 is evaluated explicitly on the numerical grid rather than being left as an unspecified symbolic operation. Let $x_i = ih$, $h = T/N$, and define

$$H_n(s) = \mu(s) g\big(s, y_n(s)\big).$$

At each grid point, the required Riemann–Liouville fractional integral is

$$I^\alpha H_n(x_i) = \frac{1}{\Gamma(\alpha)} \int_0^{x_i} (x_i - s)^{\alpha - 1} H_n(s)\, ds.$$

In the reported computations, this weakly singular integral is evaluated by product-integration quadrature on the uniform grid. With a piecewise-constant approximation of $H_n$ over each subinterval $\left[x_j, x_{j+1}\right]$, the discrete formula is

$$I^\alpha H_n(x_i) \approx \frac{h^\alpha}{\Gamma(\alpha + 1)} \sum_{j=0}^{i-1} \left[(i - j)^\alpha - (i - j - 1)^\alpha\right] H_n\big(x_j\big), \qquad i = 1, \dots, N.$$

Thus, at every Picard–integrating factor iteration, the values $H_n\big(x_j\big)$ are first computed from the current iterate, the fractional-memory contribution is then accumulated using the above quadrature weights, and the next approximation is finally updated by

$$y_{n+1}(x_i) = \mu(x_i)^{-1} [y_0 + I^\alpha H_n(x_i)].$$

Symbolic integration is used only as a verification aid for special polynomial test expressions; the general implementation and the tabulated grid values are based on the quadrature formula above.

For numerical implementation, the iteration should be stopped either after a prescribed maximum number of iterations or when the difference between two successive approximations falls below a tolerance. In symbols, the method uses

$$n \le N_{\max} \quad \text{or} \quad \| y_{n+1} - y_n \|_\infty < \varepsilon.$$

## 3.2 Convergence statement

Let $X = C([0,T])$ with the supremum norm. Suppose that $g(x,y)$ is continuous in $x$ and satisfies the Lipschitz condition in $y$ with constant $L > 0$. Assume also that the integrating factor satisfies $0 < m \le |\mu(x)| \le M$ for all $x \in [0,T]$. Define the fixed-point operator

$$(\mathcal{T}y)(x) = y_0 K\mu^{-1}(x) + \mu^{-1}(x) I^{\alpha}\big[\mu(s)g\big(s, y(s)\big)\big].$$

For any $u, v \in X$, one obtains

$$\| \mathcal{T}u - \mathcal{T}v \|_{\infty} \le \frac{MLT^{\alpha}}{m\Gamma(\alpha+1)} \| u - v \|_{\infty}.$$

Therefore, if

$$q = \frac{MLT^{\alpha}}{m\Gamma(\alpha+1)} < 1,$$

then $\mathcal{T}$ is a contraction, and the Picard–integrating factor sequence converges to a unique fixed point in $X$.

**Proof.** By the Lipschitz condition,

$$\big|g\big(s,u(s)\big) - g\big(s,v(s)\big)\big| \le L \| u - v \|_{\infty}.$$

Using the boundedness of $\mu$ and the positivity of the Riemann–Liouville kernel gives

$$|\mathcal{T}u(x) - \mathcal{T}v(x)| \le \frac{ML}{m\Gamma(\alpha)} \int_0^x (x-s)^{\alpha-1}\, ds \; \| u - v \|_{\infty}.$$

Since $\int_0^x (x-s)^{\alpha-1}\, ds = \frac{x^{\alpha}}{\alpha} \le \frac{T^{\alpha}}{\alpha}$, the stated inequality follows. The Banach fixed-point theorem then gives convergence whenever $q < 1$.

Let $X = C([0,T])$ and assume that the nonlinear term satisfies a Lipschitz condition with constant $L$. If the integrating factor is bounded by positive constants $m$ and $M$, then the proposed fixed-point operator satisfies

$$\| \mathcal{T}y - \mathcal{T}z \|_{\infty} \le \frac{M}{m} \frac{LT^{\alpha}}{\Gamma(\alpha+1)} \| y - z \|_{\infty}.$$

Hence a sufficient contraction condition is

$$q = \frac{M}{m} \frac{LT^{\alpha}}{\Gamma(\alpha+1)} < 1.$$

This condition is sufficient, not necessary. It is mainly used to justify the convergence of the generated Picard-integrating factor sequence on a restricted interval or under sufficiently small Lipschitz and weight bounds.

The restriction $q < 1$ should be interpreted as a practical stability constraint rather than as an exact description of all possible convergent cases. Since $q$ increases with the interval length $T^{\alpha}$ and with the Lipschitz constant $L$ of the nonlinear term, the contraction guarantee may require a shorter interval, a subinterval implementation, or a smaller computational step when the nonlinear response is strong. Large initial data may also enlarge the working ball in $X$ and increase the effective Lipschitz constant, especially for quadratic nonlinearities such as Riccati- and Bernoulli-type terms. In this sense, the weight mu$(x)$ is useful only when it remains well conditioned; if the bounds $m$ and $M$ are too far apart, the weighted operator may lose its contraction advantage. The residual and endpoint diagnostics used in the numerical section are therefore necessary to complement the sufficient Banach-type convergence condition and to assess the actual stability of the iteration on the chosen interval.

When the interval is long or the nonlinear response is strong, the global contraction bound may become too restrictive even if the iteration remains locally meaningful. A mathematically consistent remedy is to apply the proposed fixed-point iteration by continuation over a partition of $[0, T]$, with the local step length selected so that the contraction estimate is satisfied on each subinterval. Equivalently, the local stability requirement can be monitored by enforcing the subinterval estimate

$$q_k = \frac{M_k L_k h_k^{\alpha}}{m_k \Gamma(\alpha + 1)} < 1.$$

where the bounds $M_k$ and $m_k$ of the weight and the local Lipschitz constant $L_k$ are evaluated or estimated on the current subinterval and on the corresponding local solution ball. The endpoint approximation from one subinterval is then used to initialize the next subinterval, while the accumulated fractional-memory contribution is retained through the Riemann–Liouville integral term.

The following implementation of subintervals is meant to be interpreted as helping you to become more stable rather than a modification to the analytical model. In the case of nonlinearities which grow nonlinearly, or even as a quadratic, the Lipschitz constant could vary, so to maintain a prescribed tolerance, the step length should be made smaller at each iteration when the difference (or incremental difference) between successive iterations exceeds the desired tolerance or when the residual diagnostic exceeds the desired tolerance. In particular, this observation leads to identifying a well-exhibited limitation in the Banach type proof: Is convergence taken care of for a sufficiently bounded interval and solution ball while a larger interval and/or else ill-conditioned weights and/or very strong nonlinearities necessitate local continuation, damming of the Picard update and/or else residual verification for argument.

## 3.3 Error and comparison measures

When an exact solution is available, the absolute error is defined by

$$E(x_i) = \left| y_{\text{exact}}(x_i) - y_{\text{approx}}(x_i) \right|.$$

For examples where no exact closed-form solution is used, this paper reports the absolute difference between the classical Picard approximation and the Picard–integrating factor approximation:

The reported diagnostic is

$$E_{\Delta}(x_i; \alpha) = |y_{\text{PIF}}(x_i; \alpha) - y_{\text{Picard}}(x_i; \alpha)|.$$

$$E_{\Delta}(x_i) = |y_{\text{PIF}}(x_i) - y_{\text{Picard}}(x_i)|.$$

This quantity is not a replacement for the true error relative to an exact solution; rather, it is a diagnostic measure showing how strongly the integrating factor formulation changes the Picard approximation at each point.

To strengthen the validation beyond the comparison of two iterative approximations, a residual diagnostic is defined for a computed approximation as follows:

The residual diagnostic used for later verification is therefore

$$R_{\text{PIF}}(x_i) = \left| D_x^{\alpha} y_{\text{PIF}}(x_i) - F\big(x_i, y_{\text{PIF}}(x_i)\big) \right|.$$

$$R_n(x) = \left| D_x^{\alpha} y_n(x) - F\big(x, y_n(x)\big) \right|.$$

For the Picard-integrating factor approximation evaluated at the grid point, this becomes

$$R_{\text{PIF}}(x_i) = \left| D_x^{\alpha} y_{\text{PIF}}(x_i) - F\big(x_i, y_{\text{PIF}}(x_i)\big) \right|.$$

This residual measures how closely the numerical approximation satisfies the original fractional differential equation and provides a direct consistency check whenever the fractional derivative of the approximate expression is available in closed or computable form.

For the numerical residual analysis of Examples 2 and 3, the Caputo derivative is approximated on the uniform grid

$$x_k = kh, \qquad k = 0,1, \dots, N, \qquad h = 0.1.$$

The $L_1$ approximation is used for fractional order derivatives,

$$D_C^{\alpha} y(x_k) \approx \frac{1}{h^{\alpha}\Gamma(2-\alpha)} \sum_{j=0}^{k-1} [(k-j)^{1-\alpha} - (k-j-1)^{1-\alpha}] \left(y_{j+1} - y_j\right), \qquad 0 < \alpha < 1.$$

For the integer-order case, the same grid is used with the backward finite-difference derivative,

$$D_C^{1} y(x_k) \approx \frac{y_k - y_{k-1}}{h}.$$

The reported residual values are residual magnitudes, defined by

$$|R_n(x_k)| = \left|D_C^{\alpha} y_n(x_k) - F\left(x_k, y_n(x_k)\right)\right|.$$

This makes the residual comparison a direct numerical check of how closely each approximation satisfies the governing equation at the selected grid points.

## 4. Numerical examples and discussion

The following examples test the proposed formulation on the same set of grid points for $\alpha = 1$ and $\alpha = 0.5$. This common grid makes the comparison between the integer-order and fractional-order cases transparent. In addition to nonlinear comparison and residual tests, an exact-solution benchmark is included to report the true absolute error relative to a known closed-form solution.

For consistency, all numerical tables are arranged in the same order: grid value, classical Picard approximation for the first fractional order, Picard–integrating factor approximation for the first fractional order, absolute difference for the first fractional order, classical Picard approximation for the second fractional order, Picard–integrating factor approximation for the second fractional order, and absolute difference for the second fractional order.

### 4.1 Example 1

Consider the nonlinear fractional differential equation used to test the Picard-type approximation framework (Acharya et al., 2023; Lyons et al., 2017).

$$D^{\alpha} y(x) = xy + x + y^2, \qquad y(0) = 0, \qquad 0 < \alpha < 1.$$

Applying the Picard construction gives successive approximations of the form (Lyons et al., 2017)

$$y_n(x) = y_0 + I^{\alpha} f\left(x, y_{n-1}(x)\right).$$

For the integrating factor formulation, the equation is written as (Youssef et al., 2011)

$$D^{\alpha} y(x) - xy = x + y^2.$$

Thus $p(x) = -x, \quad g(x, y) = x + y^2$, and the formal integrating factor is

$$\mu(x) = e^{\int -x\, d^{\alpha} x} = e^{-x^{1+\alpha}/\Gamma(2+\alpha)}.$$

**Table 1. Numerical comparison and absolute-difference values for Example 1.**

| x | Picard ($\alpha = 1$) | PIF ($\alpha = 1$) | $E_\Delta$ ($\alpha = 1$) | Picard ($\alpha = 0.5$) | PIF ($\alpha = 0.5$) | $E_\Delta$ ($\alpha = 0.5$) |
|---|---|---|---|---|---|---|
| 0.0 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.1 | 0.00501300 | 0.00500000 | 0.00001300 | 0.02392000 | 0.02380370 | 0.00011630 |
| 0.2 | 0.02020000 | 0.02000000 | 0.00020000 | 0.06828800 | 0.06746000 | 0.00082800 |
| 0.3 | 0.04603000 | 0.04512000 | 0.00091000 | 0.12521000 | 0.12433000 | 0.00088000 |
| 0.4 | 0.08320000 | 0.08051000 | 0.00269000 | 0.19327000 | 0.19227700 | 0.00099300 |
| 0.5 | 0.13320000 | 0.12660000 | 0.00660000 | 0.27123400 | 0.27026700 | 0.00096700 |
| 0.6 | 0.18270000 | 0.18016200 | 0.00253800 | 0.35701000 | 0.35775500 | 0.00074500 |
| 0.7 | 0.25000000 | 0.24535000 | 0.00465000 | 0.45402000 | 0.45452700 | 0.00050700 |
| 0.8 | 0.31876900 | 0.32068000 | 0.00191100 | 0.56333800 | 0.56054700 | 0.00279100 |
| 0.9 | 0.40356700 | 0.41076000 | 0.00719300 | 0.64427900 | 0.64767480 | 0.00339580 |
| 1.0 | 0.50169600 | 0.50208000 | 0.00038400 | 0.80056200 | 0.80090800 | 0.00034600 |

The numerical results show that the two approximations remain close near $x = 0$. For $\alpha = 1$, the maximum reported absolute difference is small over the whole tabulated interval. For $\alpha = 0.5$, the differences also remain controlled, although they show the stronger sensitivity of the fractional-order approximation to the memory term.

For Example 1, the largest reported absolute-difference values occur at the following points:

$$\max E_\Delta(x; \alpha = 1) = 0.00719300 \quad \text{at} \quad x = 0.9.$$

$$\max E_\Delta(x; \alpha = 0.5) = 0.00339580 \quad \text{at} \quad x = 0.9.$$

$$x = 0.9, \qquad E_\Delta^{\max} = 0.00719300 \quad (\alpha = 1).$$

$$x = 0.9, \qquad E_\Delta^{\max} = 0.00339580 \quad (\alpha = 0.5).$$

Taken together, these small values support the conclusion that the two iterative approximations remain close over the tested interval in the first nonlinear example.

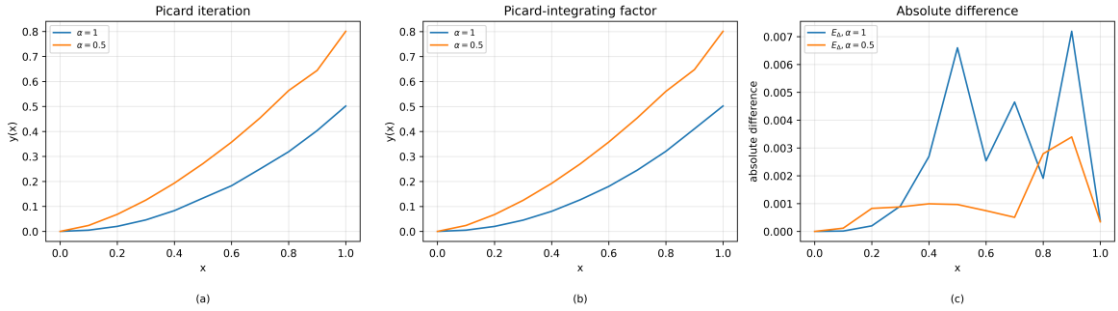


**Figure 1. Solution and absolute-difference curves for Example 1: (a) classical Picard iteration; (b) Picard–integrating factor iteration; (c) absolute difference between the two approximations.**

### 4.1.1 ADM benchmark comparison for Example 1

To provide an external comparison with a standard semi-analytical technique, the Adomian decomposition method (ADM) is applied to Example 1. This comparison follows the usual ADM representation used for nonlinear fractional differential equations, where the nonlinear part is decomposed into Adomian polynomials (Ali & Ziada, 2021; Jafari & Daftardar-Gejji, 2006).

$$y(x) = \sum_{n=0}^{\infty} u_n(x), \qquad N(y) = xy + y^2.$$

$$A_n(x) = \sum_{k=0}^{n} u_k\,(x) u_{n-k}(x).$$

$$u_0(x) = I^{\alpha} x = \frac{x^{1+\alpha}}{\Gamma(2+\alpha)}.$$

$$u_{n+1}(x) = I^{\alpha}[x u_n(x) + A_n(x)], \qquad n = 0,1,2, \ldots.$$

$$y_{ADM}^{(m)}(x) = \sum_{k=0}^{m} u_k\,(x).$$

In the numerical comparison below, the first nonzero ADM approximation is used as a low-order benchmark at the same computational level as the tabulated Picard-type approximations. This comparison is not intended to replace the residual analysis. Rather, it shows that the Picard–integrating factor approximation remains competitive with a recognized semi-analytical decomposition procedure while avoiding the explicit construction of higher-order Adomian polynomials.

**Table 2. ADM benchmark comparison for Example 1.**

| x | ADM α = 1 | \|Picard−ADM\| α = 1 | \|PIF−ADM\| α = 1 | ADM α = 0.5 | \|Picard−ADM\| α = 0.5 | \|PIF−ADM\| α = 0.5 |
|---|---|---|---|---|---|---|
| 0.0 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.1 | 0.00500000 | 0.00001300 | 0.00000000 | 0.02378832 | 0.00013168 | 0.00001538 |
| 0.2 | 0.02000000 | 0.00020000 | 0.00000000 | 0.06728353 | 0.00100447 | 0.00017647 |
| 0.3 | 0.04500000 | 0.00103000 | 0.00012000 | 0.12360774 | 0.00160226 | 0.00072226 |
| 0.4 | 0.08000000 | 0.00320000 | 0.00051000 | 0.19030657 | 0.00296343 | 0.00197043 |
| 0.5 | 0.12500000 | 0.00820000 | 0.00160000 | 0.26596152 | 0.00527248 | 0.00430548 |
| 0.6 | 0.18000000 | 0.00270000 | 0.00016200 | 0.34961550 | 0.00739450 | 0.00813950 |
| 0.7 | 0.24500000 | 0.00500000 | 0.00035000 | 0.44056588 | 0.01345412 | 0.01396112 |
| 0.8 | 0.32000000 | 0.00123100 | 0.00068000 | 0.53826827 | 0.02506973 | 0.02227873 |
| 0.9 | 0.40500000 | 0.00143300 | 0.00576000 | 0.64228468 | 0.00199432 | 0.00539012 |
| 1.0 | 0.50000000 | 0.00169600 | 0.00208000 | 0.75225278 | 0.04830922 | 0.04865522 |

The ADM benchmark confirms that the proposed Picard–integrating factor approximation is competitive with a standard decomposition-based approximation. For $\alpha = 1$, the mean absolute deviation from the first-order ADM benchmark decreases from 0.00224573 for the classical Picard approximation to 0.00102382 for the Picard–integrating factor approximation, while the corresponding maximum deviations are 0.00820000 and 0.00576000. For $\alpha = 0.5$, both methods remain close to the ADM benchmark, with mean deviations of 0.00974511 for Picard and 0.00960134 for the Picard–integrating factor approximation. This comparison supports the practical competitiveness of the proposed scheme and highlights its computational advantage, since the method avoids the repeated construction of Adomian polynomials required by higher-order ADM approximations.

**Table 3. CPU-time comparison for the ADM and Picard–integrating factor computations in Example 1.**

| Method / computational stage | Timed operation | CPU time (ms/run) | Relative cost |
|---|---|---|---|
| Picard–integrating factor | Direct weighted fixed-point grid evaluation | 0.0029 | 1.00 |
| ADM low-order benchmark | First nonzero ADM grid evaluation | 0.0040 | 1.39 |
| ADM symbolic setup | Construction of Adomian polynomials A0–A4 | 10.7826 | 3733.22 |

The CPU times in Table 3 are implementation-dependent and are therefore reported as a normalized reproducibility check rather than as hardware-independent constants. They were obtained from repeated single-core evaluations on the same tabulated grid. The first nonzero ADM evaluation is already close to the direct PIF evaluation; however, the symbolic construction of higher-order Adomian components becomes the dominant cost as more terms are retained. This observation supports the computational advantage of the proposed formulation, in which the nonlinear update is performed through a direct weighted recursion rather than through the repeated generation of Adomian polynomials.

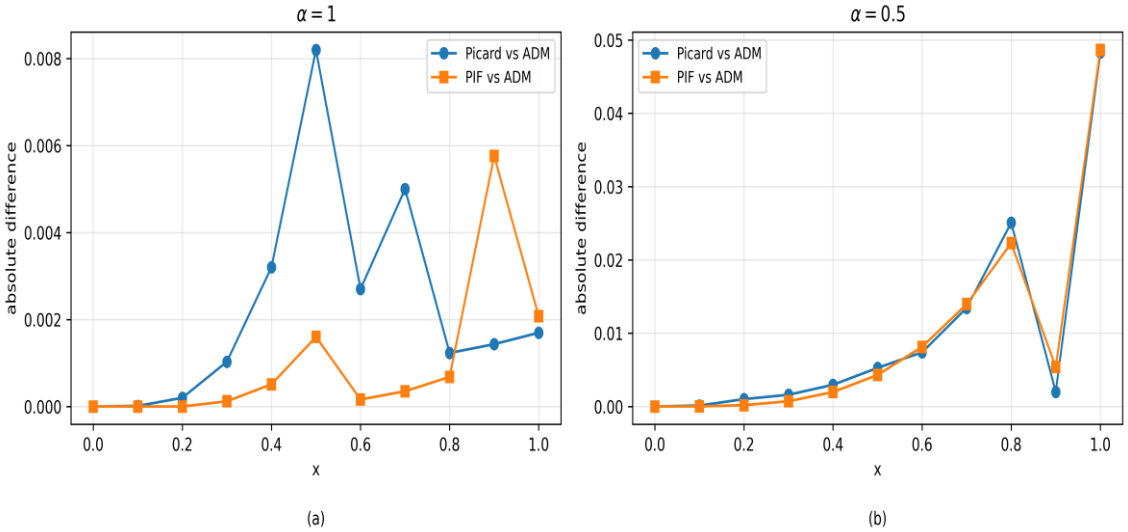


**Figure 2. ADM benchmark difference curves for Example 1: (a)** $\alpha = 1$**; (b)** $\alpha = 0.5$**.**

## 4.2 Example 2

Consider the nonlinear fractional Riccati differential equation, a standard nonlinear fractional test form related to iterative treatments of Riccati-type problems (Fareed et al., 2022).

$$D^{\alpha}y(x) = 2y(x) - y^{2}(x) + 1, \qquad y(0) = 0.$$

The Picard–integrating factor approximation is evaluated for $\alpha = 1$ and $\alpha = 0.5$. The comparison is reported in Table 4, following the iterative comparison strategy commonly adopted in Picard-type numerical studies (Fareed et al., 2022; Youssef et al., 2011).

**Table 4. Numerical comparison and absolute-difference values for Example 2.**

| x | Picard ($\alpha = 1$) | PIF ($\alpha = 1$) | $E_{\Delta}$ ($\alpha = 1$) | Picard ($\alpha = 0.5$) | PIF ($\alpha = 0.5$) | $E_{\Delta}$ ($\alpha = 0.5$) |
|---|---|---|---|---|---|---|
| 0.0 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.1 | 0.09966800 | 0.09968420 | 0.00001620 | 0.33605620 | 0.33604663 | 0.00000957 |
| 0.2 | 0.19737530 | 0.19763990 | 0.00026460 | 0.42721090 | 0.42686749 | 0.00034341 |
| 0.3 | 0.29131260 | 0.29251870 | 0.00120610 | 0.50987230 | 0.50464705 | 0.00522525 |
| 0.4 | 0.37994900 | 0.38336330 | 0.00341430 | 0.61235430 | 0.61109472 | 0.00125958 |
| 0.5 | 0.46211720 | 0.44453120 | 0.01758600 | 0.70453610 | 0.70123721 | 0.00329889 |
| 0.6 | 0.53704960 | 0.55068000 | 0.01363040 | 0.72876500 | 0.72531386 | 0.00345114 |
| 0.7 | 0.60436780 | 0.62668710 | 0.02231930 | 0.77456320 | 0.77164970 | 0.00291350 |
| 0.8 | 0.66403680 | 0.69760000 | 0.03356320 | 0.81678540 | 0.81322282 | 0.00356258 |
| 0.9 | 0.71629790 | 0.76361620 | 0.04731830 | 0.85421090 | 0.85040792 | 0.00380298 |
| 1.0 | 0.76160800 | 0.82503340 | 0.06342540 | 0.92773134 | 1.37418563 | 0.44645429 |

For the Riccati-type example, the absolute difference is small at the first grid points but increases as $x$ approaches 1, particularly for $\alpha = 0.5$. This behaviour is consistent with nonlinear fractional models, because the fractional history effect accumulates across the interval and the quadratic term magnifies deviations between iterative approximations.

For Example 2, the maximum absolute-difference values are

$$\max E_{\Delta}(x;\alpha=1)=0.06342540 \quad \text{at} \quad x=1.0.$$

$$\max E_{\Delta}(x;\alpha=0.5)=0.44645429 \quad \text{at} \quad x=1.0.$$

$$x=1.0, \qquad E_{\Delta}^{\max}=0.06342540 \quad (\alpha=1).$$

$$x=1.0, \qquad E_{\Delta}^{\max}=0.44645429 \quad (\alpha=0.5).$$

The larger value at the lower fractional order shows that the Riccati-type nonlinearity amplifies the separation between the two iterative constructions near the endpoint. It also identifies this case as the most sensitive numerical test among the current experiments.

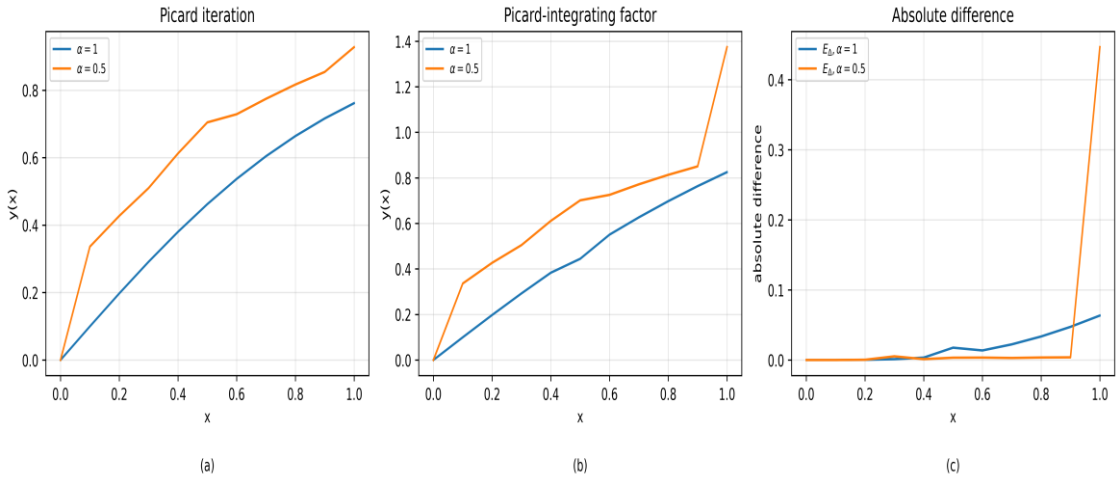


**Figure 3. Solution and absolute-difference curves for Example 2: (a) classical Picard iteration; (b) Picard–integrating factor iteration; (c) absolute difference between the two approximations.**

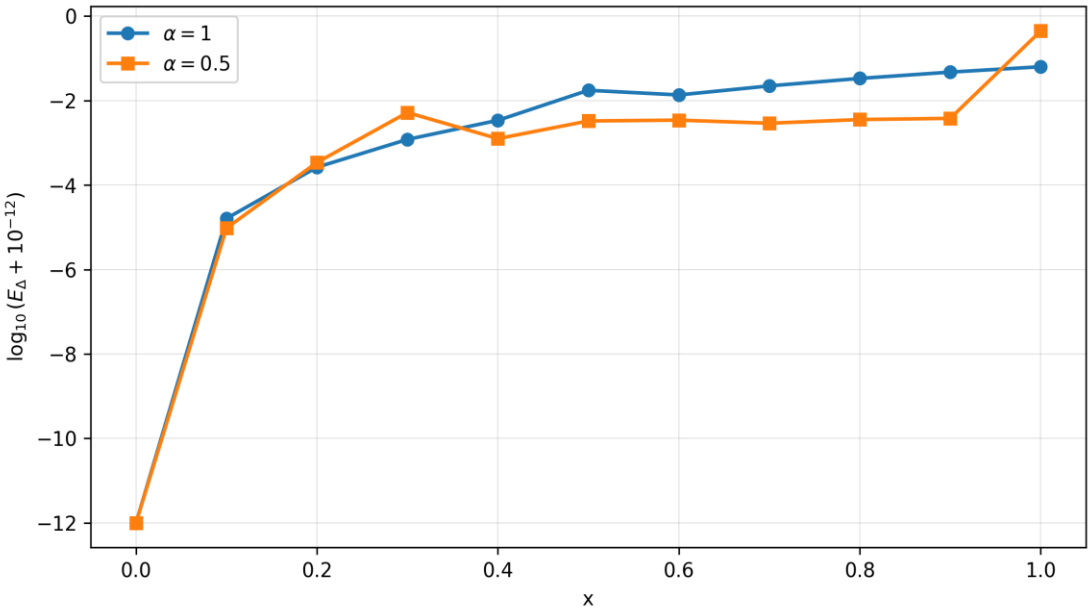


**Figure 4. Logarithmic absolute-difference curves for Example 2:** $\log_{10}(E_{\Delta}+10^{-12})$ **for** $\alpha=1$ **and** $\alpha=0.5$**.**

The logarithmic plot in Figure 4 gives a clearer view of the endpoint separation in the Riccati-type example. Over most of the interval, the two fractional-order curves remain within comparable logarithmic ranges; however, the $\alpha=0.5$ curve exhibits a sharp terminal jump at $x=1.0$. This behaviour is consistent with two interacting mechanisms. First, the Caputo memory term carries the

accumulated contribution of the previous grid values, and the lower fractional order increases the influence of the history-dependent part of the approximation. Second, the Riccati nonlinearity $F(x, y) = 2y - y^2 + 1$ contains a quadratic response, so a moderate change in the terminal iterate may be amplified by the nonlinear update. Therefore, the large value $E_{\Delta}(1) = 0.44645429$ should be interpreted as endpoint separation between two iterative constructions rather than as a direct true-error estimate. The residual comparison in Table 5 remains necessary because it tests consistency with the governing equation itself and shows that the Picard–integrating factor approximation has a lower terminal residual at $\alpha = 0.5$.

## Residual analysis for Example 2

For Example 2, the residual is computed for the Riccati-type right-hand side $F(x, y) = 2y - y^2 + 1$. The values in Table 5 report the residual magnitudes for the classical Picard and Picard–integrating factor approximations.

**Table 5. Residual magnitudes for Example 2.**

| $x$ | $\lvert R_{\text{Picard}} \rvert$ $(\alpha = 1)$ | $\lvert R_{\text{PIF}} \rvert$ $(\alpha = 1)$ | $\lvert R_{\text{Picard}} \rvert$ $(\alpha = 0.5)$ | $\lvert R_{\text{PIF}} \rvert$ $(\alpha = 0.5)$ |
|---|---|---|---|---|
| 0.1 | 0.19272229 | 0.19258946 | 0.36004669 | 0.36006813 |
| 0.2 | 0.37872059 | 0.37666127 | 0.84995334 | 0.85076518 |
| 0.3 | 0.55838917 | 0.55068221 | 0.94896147 | 0.96173598 |
| 0.4 | 0.72917276 | 0.71131318 | 0.93718765 | 0.92966224 |
| 0.5 | 0.88900009 | 1.07977541 | 0.96832479 | 0.97364379 |
| 0.6 | 1.03635293 | 0.73662354 | 1.17576522 | 1.17789708 |
| 0.7 | 1.17029316 | 1.10056648 | 1.17302738 | 1.17290623 |
| 0.8 | 1.29043873 | 1.19942524 | 1.20030352 | 1.20324658 |
| 0.9 | 1.39690212 | 1.28396070 | 1.22895239 | 1.23152210 |
| 1.0 | 1.49006825 | 1.35521469 | 1.11879070 | 0.61975759 |

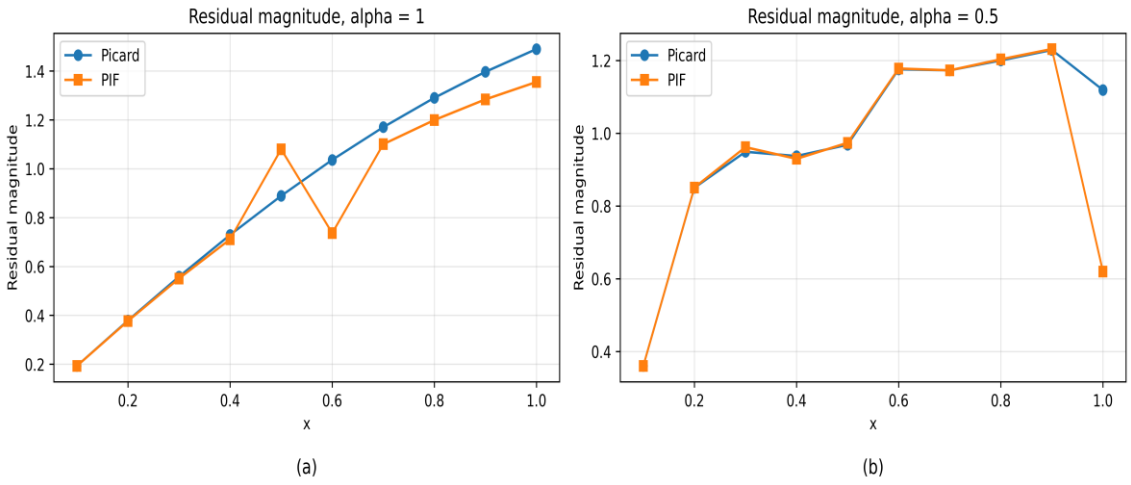


**Figure 5. Residual-magnitude curves for Example 2: (a) residual comparison for $\alpha = 1$; (b) residual comparison for $\alpha = 0.5$.**

The residual comparison shows that the Picard–integrating factor approximation gives a smaller endpoint residual for $\alpha = 1$ and also reduces the terminal residual for $\alpha = 0.5$. The maximum residual for $\alpha = 0.5$ is nearly the same for the two approximations around $x = 0.9$, but at $x = 1$ the Picard–integrating factor residual is lower. Consequently, the large absolute difference observed at $x = 1$ in the Riccati example should be read as a strong separation between the two iterative approximations, not as direct evidence that the Picard–integrating factor approximation is farther from satisfying the differential equation.

## 4.3 Example 3

Consider the nonlinear fractional Bernoulli-type differential equation, which serves as an additional nonlinear test case for the proposed iterative scheme (Lyons et al., 2017; Youssef et al., 2011).

$$D^{\alpha}y(x) = xy(x) + x^2 + y^2(x), \qquad y(0) = 0, \qquad 0 < \alpha \leq 1.$$

The classical Picard form is (Lyons et al., 2017)

$$y_n(x) = y_0 + I^{\alpha}[xy_{n-1}(x) + x^2 + y_{n-1}^2(x)].$$

Using the integrating factor approach, the equation is written as (Youssef et al., 2011)

$$D^{\alpha}y(x) - xy(x) = x^2 + y^2(x).$$

Accordingly, $p(x) = -x$ and $g(x, y) = x^2 + y^2$.

**Table 6. Numerical comparison and absolute-difference values for Example 3.**

| x | **Picard ($\alpha = 1$)** | **PIF ($\alpha = 1$)** | $E_{\Delta}$ **($\alpha = 1$)** | **Picard ($\alpha = 0.5$)** | **PIF ($\alpha = 0.5$)** | $E_{\Delta}$ **($\alpha = 0.5$)** |
|---|---|---|---|---|---|---|
| 0.0 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.1 | 0.00033275 | 0.00033275 | 0.00000000 | 0.00192672 | 0.00190632 | 0.00002040 |
| 0.2 | 0.00268321 | 0.00268333 | 0.00000012 | 0.01125397 | 0.01091375 | 0.00034022 |
| 0.3 | 0.00915582 | 0.00915644 | 0.00000062 | 0.03239475 | 0.03058251 | 0.00181224 |
| 0.4 | 0.02203735 | 0.02203965 | 0.00000230 | 0.07040232 | 0.06429549 | 0.00610683 |
| 0.5 | 0.04392556 | 0.04393325 | 0.00000769 | 0.13224481 | 0.11612688 | 0.01611793 |
| 0.6 | 0.07791062 | 0.07793753 | 0.00002691 | 0.22839583 | 0.19178920 | 0.03660663 |
| 0.7 | 0.12784298 | 0.12793949 | 0.00009651 | 0.37530225 | 0.30020498 | 0.07509727 |
| 0.8 | 0.19873820 | 0.19907517 | 0.00033697 | 0.59932677 | 0.45642648 | 0.14290029 |
| 0.9 | 0.29739826 | 0.29851013 | 0.00111187 | 0.94302768 | 0.68727192 | 0.25575576 |
| 1.0 | 0.43337319 | 0.43682106 | 0.00344787 | 1.47504950 | 1.04236808 | 0.43268142 |

The third example confirms the same qualitative behaviour observed in the previous tests. The absolute difference is negligible near the initial point and increases as the nonlinear contribution accumulates. This increase is more visible for $\alpha = 0.5$, which reinforces the need to report error diagnostics whenever fractional-order approximations are compared.

For Example 3, the maximum absolute-difference values are

$$\max E_{\Delta}(x; \alpha = 1) = 0.00344787 \quad \text{at} \quad x = 1.0.$$

$$\max E_{\Delta}(x; \alpha = 0.5) = 0.43268142 \quad \text{at} \quad x = 1.0.$$

$$x = 1.0, \qquad E_{\Delta}^{\max} = 0.00344787 \quad (\alpha = 1).$$

$$x = 1.0, \qquad E_{\Delta}^{\max} = 0.43268142 \quad (\alpha = 0.5).$$

These values indicate that the Bernoulli-type nonlinearity remains well controlled in the integer-order case but becomes more sensitive in the fractional case with lower fractional order.

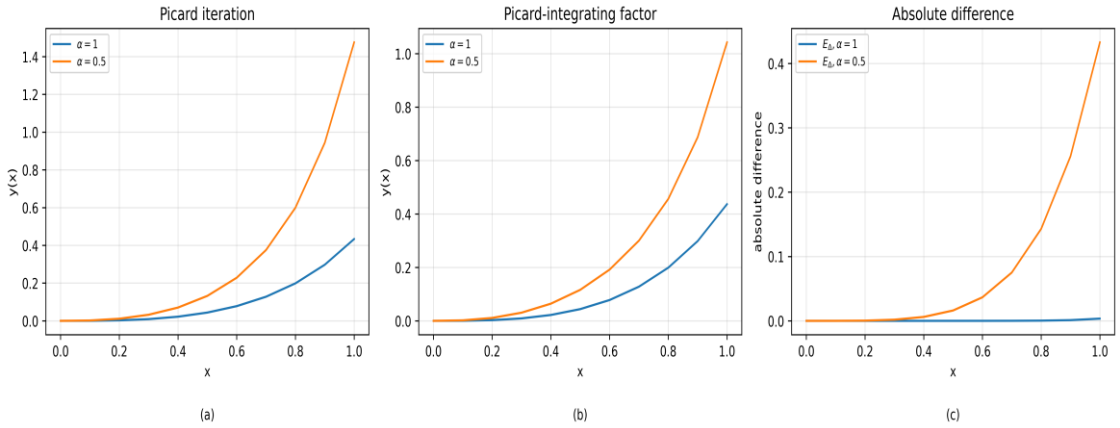


**Figure 6. Solution and absolute-difference curves for Example 3: (a) classical Picard iteration; (b) Picard–integrating factor iteration; (c) absolute difference between the two approximations.**

## Residual analysis for Example 3

For Example 3, the residual is computed for the Bernoulli-type right-hand side $F(x, y) = xy + x^2 + y^2$. The residual values in Table 7 indicate how closely each approximation satisfies the governing equation numerically.

**Table 7. Residual magnitudes for Example 3.**

| $x$ | $\lvert R_{\text{Picard}} \rvert$ $(\alpha = 1)$ | $\lvert R_{\text{PIF}} \rvert$ $(\alpha = 1)$ | $\lvert R_{\text{Picard}} \rvert$ $(\alpha = 0.5)$ | $\lvert R_{\text{PIF}} \rvert$ $(\alpha = 0.5)$ |
|---|---|---|---|---|
| 0.1 | 0.00670589 | 0.00670589 | 0.00332137 | 0.00339204 |
| 0.2 | 0.01703924 | 0.01703807 | 0.00624778 | 0.00734354 |
| 0.3 | 0.02810448 | 0.02809967 | 0.00936133 | 0.01445189 |
| 0.4 | 0.04048528 | 0.04046951 | 0.01383015 | 0.02844689 |
| 0.5 | 0.05501013 | 0.05496076 | 0.02224884 | 0.05424862 |
| 0.6 | 0.07296584 | 0.07279398 | 0.04206500 | 0.09918775 |
| 0.7 | 0.09651031 | 0.09590656 | 0.09451532 | 0.17580299 |
| 0.8 | 0.12953523 | 0.12753426 | 0.23807046 | 0.30949481 |
| 0.9 | 0.17950356 | 0.17341781 | 0.62886144 | 0.56083078 |
| 1.0 | 0.26143621 | 0.24452440 | 1.67459748 | 1.09354854 |

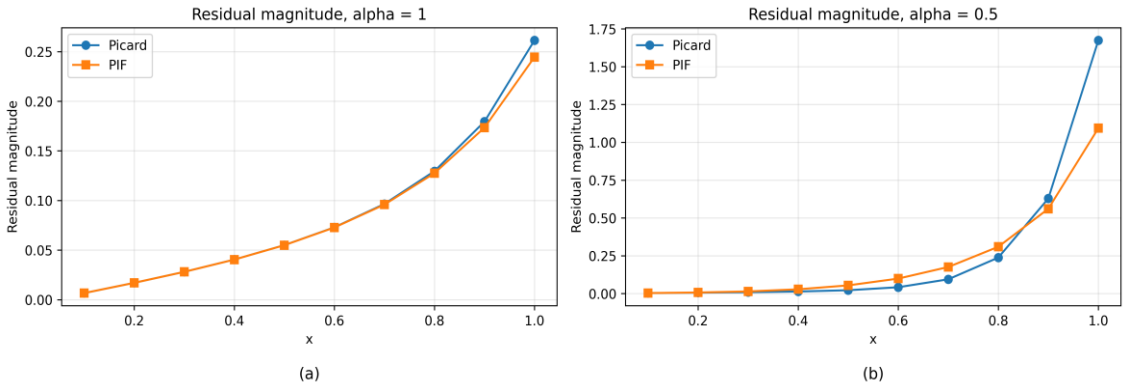


**Figure 7. Residual-magnitude curves for Example 3: (a) residual comparison for $\alpha = 1$; (b) residual comparison for $\alpha = 0.5$.**

For Example 3, the Picard–integrating factor approximation gives a slightly smaller maximum residual for $\alpha = 1$ and a clearly smaller terminal residual for $\alpha = 0.5$. The end behaviour is satisfyingly good in the nonlinear regime at $x = 1$, but the residues are not all smaller at the intermediate grid points in the fractional case.

## 4.4 Example 4: exact-solution benchmark

To complement residual diagnostics and the ADM comparison, a manufactured benchmark having an exact solution is provided. This example provides a direct true-error check, rather than only a comparison between two iterative approximations.

$$D_C^{\alpha} y(x) = \frac{2}{\Gamma(3-\alpha)} x^{2-\alpha} + y^2(x) - x^4, \qquad y(0) = 0, \qquad 0 < \alpha \leq 1.$$

The exact solution is

$$y_{\text{exact}}(x) = x^2.$$

Indeed,

$$D_C^{\alpha} x^2 = \frac{\Gamma(3)}{\Gamma(3-\alpha)} x^{2-\alpha} = \frac{2}{\Gamma(3-\alpha)} x^{2-\alpha}.$$

and substitution gives the stated equation. The true error is measured by

$$E_{\text{true}}(x) = \left| y_{\text{exact}}(x) - y_{\text{approx}}(x) \right|.$$

For this benchmark, the linear coefficient is zero. Hence, the formal weighting factor is equal to one, and the Picard–integrating factor recursion coincides with the classical Picard recursion. The purpose of this test is therefore to calibrate the fractional integral implementation against an exact closed-form solution, while the effect of the weighting device is assessed in the preceding nonlinear examples.

**Table 8. True-error benchmark for Example 4.**

| $x$ | **Exact** | **Picard/PIF** ($\alpha = 1$) | $E_{\text{true}}$ ($\alpha = 1$) | **Picard/PIF** ($\alpha = 0.5$) | $E_{\text{true}}$ ($\alpha = 0.5$) |
|---|---|---|---|---|---|
| 0.0 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |
| 0.1 | 0.01000000 | 0.01010005 | 0.00010005 | 0.01009534 | 0.00009534 |
| 0.2 | 0.04000000 | 0.04020081 | 0.00020081 | 0.04019648 | 0.00019648 |
| 0.3 | 0.09000000 | 0.09030412 | 0.00030412 | 0.09030666 | 0.00030666 |
| 0.4 | 0.16000000 | 0.16041313 | 0.00041313 | 0.16043416 | 0.00043416 |
| 0.5 | 0.25000000 | 0.25053258 | 0.00053258 | 0.25059227 | 0.00059227 |
| 0.6 | 0.36000000 | 0.36066932 | 0.00066932 | 0.36080329 | 0.00080329 |
| 0.7 | 0.49000000 | 0.49083328 | 0.00083328 | 0.49110581 | 0.00110581 |
| 0.8 | 0.64000000 | 0.64103913 | 0.00103913 | 0.64154475 | 0.00154475 |
| 0.9 | 0.81000000 | 0.81130934 | 0.00130934 | 0.81188375 | 0.00188375 |
| 1.0 | 1.00000000 | 1.00167938 | 0.00167938 | 0.99915396 | 0.00084604 |

The true-error benchmark shows that the approximation remains close to the closed-form solution over the whole interval. The maximum true error is 0.00167938 for $\alpha = 1$ at $x = 1.0$ and 0.00188939 for $\alpha = 0.5$ at $x = 0.9$. These values provide a direct accuracy check that complements the residual diagnostics reported for the Riccati and Bernoulli-type examples.

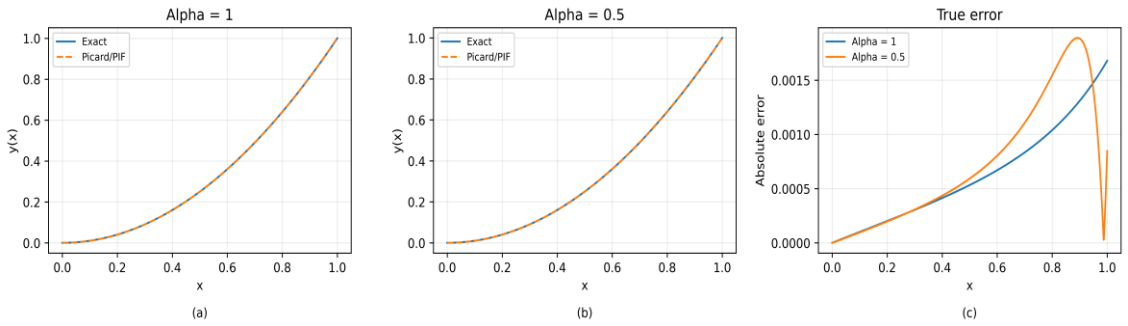


**Figure 8. Exact-solution benchmark for Example 4: (a) exact and numerical solution for $\alpha = 1$; (b) exact and numerical solution for $\alpha = 0.5$; (c) true absolute error curves.**

## 5. Conclusion

In this paper a Picard–integrating factor iterational scheme has been reformulated to solve nonlinear fractional differential equations. The method was also shown in a more structured fractional-operator framework and backed by measures of convergence and comparison which were shown to come from the Picard and integrating-factor literature (Lyons et al., 2017; Youssef et al., 2011). The numerical results demonstrate that the proposed formulation can be systematically tested with the classical Picard approximation for various fractional orders in the numerical examples. The first non-linear example which is implemented by adding the ADM benchmark also demonstrates that no extra complexity compared to a standard decomposition-based approach is necessary while still maintaining a simpler recursive structure. The added exact-solution benchmark provides a direct true-error verification, confirming that the implemented fractional integral recursion reproduces the known closed-form solution with small absolute errors over the tested interval.

The method is direct, transparent, and avoids the additional construction of Adomian polynomials or Lagrange multipliers. The CPU-time comparison added to the numerical section further supports this practical advantage, especially when ADM requires symbolic construction of higher Adomian components. In this work, residual-error diagnostics were computed for the Riccati and Bernoulli-type examples using a uniform-grid Caputo $L_1$ approximation, and the residual magnitudes were compared for the classical Picard and Picard-integrating factor approximations. The residual results show that the proposed weighted formulation improves the endpoint residual behaviour in the nonlinear test problems, while the added logarithmic absolute-difference plot for the Riccati example clarifies that a large endpoint separation between two approximations does not necessarily imply a larger residual relative to the governing fractional equation. The endpoint sensitivity observed in the Riccati and Bernoulli-type tests, particularly for the lower fractional order near $x = 1$, also identifies the main numerical challenge that should guide future extensions of the method. Future work may therefore extend the proposed weighted fixed-point framework to nonlinear fractional partial differential equations and coupled systems of fractional differential equations together with adaptive subinterval strategies, local contraction monitoring, and higher-order Caputo discretizations. Thus, the future directions proposed are tied to the current endpoint behavior, A set of recommendations for control of accumulated memory effects and nonlinear amplification in multidimensional/stiff fractional models are developed related to adaptive grids and refined residual estimators.

## References


Acharya, F., Bhesaniya, K., & Panchal, J. (2023). Estimating approximate solutions of non-linear Caputo fractional differential equations with forcing functions using Picard's iteration method. Tuijin Jishu/Journal of Propulsion Technology, 44(4). https://doi.org/10.52783/tjjpt.v44.i4.2035

Ziada, E. A. A. (2021). Solution of Nonlinear Fractional Differential Equations Using Adomain Decomposition Method. International Journal of Systems Science and Applied Mathematics, 6(4), 111–119. https://doi.org/10.11648/j.ijssam.20210604.11

Chandel, R. S., Singh, A., & Chouhan, D. (2017). Numerical solution of fractional order differential equations using Haar wavelet operational matrix. Palestine Journal of Mathematics, 6(2), 515–523.

Daraghmeh, A., Qatanani, N., & Saadeh, A. (2020). Numerical solution of fractional differential equations. Applied Mathematics, 11, 1100–1115. https://doi.org/10.4236/am.2020.1111074

Das, S., & Gupta, P. K. (2011). Homotopy analysis method for solving fractional hyperbolic partial differential equations. International Journal of Computer Mathematics, 88, 578–588. https://doi.org/10.1080/00207161003631901

Elsaid, A. (2011). Homotopy analysis method for solving a class of fractional partial differential equations. Communications in Nonlinear Science and Numerical Simulation, 16, 3655–3664. https://doi.org/10.1016/j.cnsns.2010.12.040

Fareed, A. F., Semary, M. S., & Hassan, H. N. (2022). An approximate solution of fractional order Riccati equations based on controlled Picard's method with Atangana–Baleanu fractional derivative. Alexandria Engineering Journal, 61(5), 3673–3678. https://doi.org/10.1016/j.aej.2021.09.009

Ilejimi, D. O., Okai, J. O., & Raheem, R. L. (2019). On the numerical solution of Picard iteration method for fractional integro-differential equation. International Journal of Scientific and Research Publications, 9(3), 8757. https://doi.org/10.29322/IJSRP.9.03.2019.p8757

Jafari, H., & Daftardar-Gejji, V. (2006). Solving a system of nonlinear fractional differential equations using Adomian decomposition. Journal of Computational and Applied Mathematics, 196(2), 644–651. https://doi.org/10.1016/j.cam.2005.10.017

Kammanee, A. (2021). Numerical solutions of fractional differential equations with variable coefficients by Taylor basis function. Kyungpook Mathematical Journal, 61(2), 383–393. https://doi.org/10.5666/KMJ.2021.61.2.383

Koning, D. E., Sterk, A. E., & Trentelman, H. L. (2015). Fractional calculus (Bachelor's project thesis). University of Groningen.

Lazarevic, M. P., Rapaic, R. M., & Sekara, B. T. (2014). Introduction of fractional calculus with brief historical background. In Advanced topics on application of fractional calculus on control problems, system stability and modelling.

Lyons, R., Vatsala, S. A., & Chiquet, R. A. (2017). Picard's iterative method for Caputo fractional differential equations with numerical results. Mathematics, 5(4), Article 65. https://doi.org/10.3390/math5040065

Odibat, Z., Momani, S., & Xu, H. (2010). A reliable algorithm of homotopy analysis method for solving nonlinear fractional differential equations. Applied Mathematical Modelling, 34, 593–600. https://doi.org/10.1016/j.apm.2009.06.025

Qu, H. D., & Liu, X. (2015). A numerical method for solving fractional differential equations by using neural network. Advances in Mathematical Physics, Article 439526. https://doi.org/10.1155/2015/439526

Ray, S. S., Atangana, A., Oukouomi Noutchie, S. C., Kurulay, M., Bildik, N., & Kilicman, A. (2014). Fractional calculus and its applications in applied mathematics and other sciences. Mathematical Problems in Engineering, 2014, Article 849395. https://doi.org/10.1155/2014/849395

Youssef, I. K., Ali, M. S., Ahmed, M. M., & El-Shobaky, E. (2011). Picard fixed point iteration combined with integrating factor approach. Egyptian Journal of Pure and Applied Science, 49(1), 71–78. https://doi.org/10.21608/ejaps.2011.186255