The Equation That Tells You Where It Starts

Luminary Quantum Institute, LLC (dba Mitra Institute of Education)

SIRPY · Problem 5 — A Nonlinear Volterra Integral Equation

An ongoing series. We are building a public library of problems SIRPY has solved — each entry is one problem, the exact input we handed the solver, the exact output it returned, and an honest read of what happened. No slides. No hand-waving. Numbers you can check.

A different kind of equation

Every problem in this library so far has been a differential equation — the unknown appears through its derivatives. This one is different in kind.

$$y(t) \ = \ t(1-t) \ + \ \int_0^t (t-s)\, y(s)^2\, ds \tag{1}$$

This is a **nonlinear Volterra integral equation of the second kind**. The unknown function does not appear differentiated; it appears *under an integral sign.* And notice the upper limit of that integral: it is t itself — the same variable the equation is being evaluated at.

So the value of y at any point depends on an accumulated history of y over everything that came before it, and that history enters squared, which makes the problem nonlinear. There is no formula to rearrange, no derivative to isolate.

Volterra equations of this form arise wherever the present state of a system depends on its own past: viscoelastic materials with memory, population models with delayed recruitment, heat conduction with history-dependent flux, and renewal processes in reliability engineering.

What we handed the solver

```
#| label: volterra-solve
from sirpy import solve_volterra, plot_verification

r_v = solve_volterra(phi='t*(1-t)',       # the forcing term
                     kernel='(t-s)',       # the memory kernel K(t,s)
                     f='y**2',             # the nonlinearity under the integral
                     a=0, t1=1,
                     iterations=20, verbose=True)

plot_verification(r_v, save="verify_v.png", show=True)
```

The equation is described by its three parts, each named for what it is: the forcing term $\varphi(t)$, the kernel $K(t,s)$, and the nonlinear function $f(y)$ appearing inside the integral. The interval is $[0,1]$.

No initial conditions are supplied. That turns out not to be an oversight.

What came back

The equation supplies its own starting point

This is the part worth pausing on:

```
INFO: the equation supplies its own initial data:
      y(0) = phi(0) = 0,  y'(0) = phi'(0) + lam*K(0,0)*f = 1
```

A differential equation is incomplete without initial conditions — you must supply them. A Volterra equation is not. Set $t = 0$ in Equation 1 and the integral collapses to zero, leaving $y(0) = \varphi(0)$ directly. The starting value is **determined by the equation itself**, not by the user.

SIRPY read that off the structure, reported the values it obtained, and showed the relations it used to obtain them. We supplied no initial data because there was none to supply.

The solution is analytic across the whole interval

```
Piecewise power-series solution (2 segments of degree 20 covering [0, 1])
  segment 1: [0, 0.525115]  centered at 0
  segment 2: [0.525115, 1]  centered at 0.525115

INFO: the interval [0, 1] exceeds a single series' reliable range;
      the solution was continued analytically over 2 segments
      (piecewise power series - still fully analytic on every piece).
```

A single power series has a finite radius in which it can be trusted, and $[0, 1]$ exceeded it here. Rather than pushing a series beyond where it is reliable — or quietly switching to a numerical grid — SIRPY continued the solution analytically onto a second segment and reported exactly where the handover occurred.

The result remains an explicit analytic object on every piece: readable, differentiable, and evaluable at any point.

Verified

```
INFO: verification by substitution - VERIFIED: the returned series
      satisfies the Volterra equation to 3.27e-13 (relative 3.21e-13).
```

The returned solution was substituted back into Equation 1 and the defect measured: 3.27×10^{-13} absolute, 3.21×10^{-13} relative.

Table 1: Solver output and verification

Quantity	Value
$y(0)$, determined by the equation	0
$y'(0)$, determined by the equation	1
Segments covering $[0, 1]$	2 (analytic continuation at $t = 0.525115$)
Series degree per segment	20
Residual by substitution	3.27×10^{-13} (relative 3.21×10^{-13})

The figures

What this problem shows

- **Integral equations, posed as written.** The forcing term, kernel, and nonlinearity are each named for what they are. No conversion, no reformulation on the user's part.

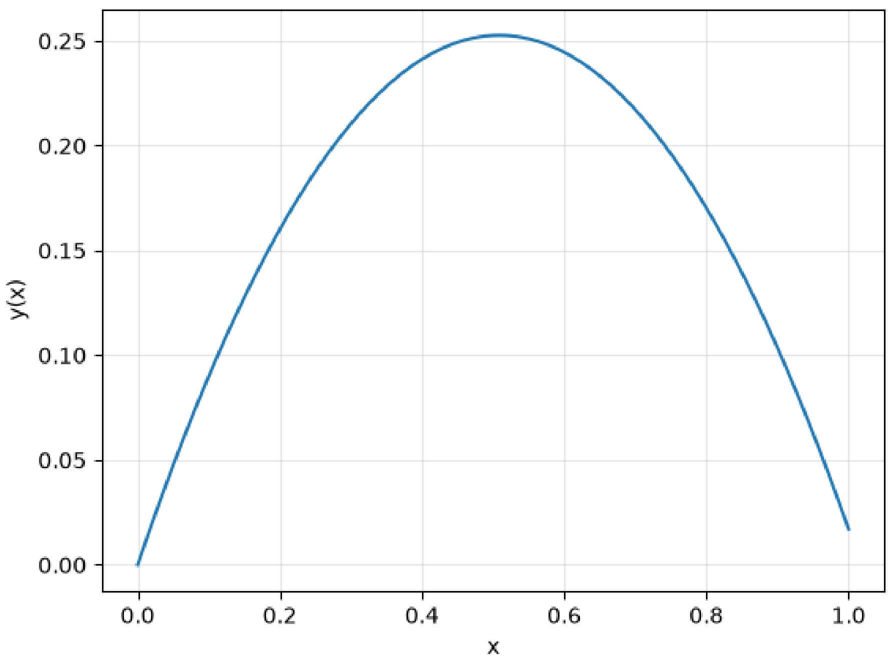

Figure 1: The solution on $[0, 1]$, assembled from two analytic segments.

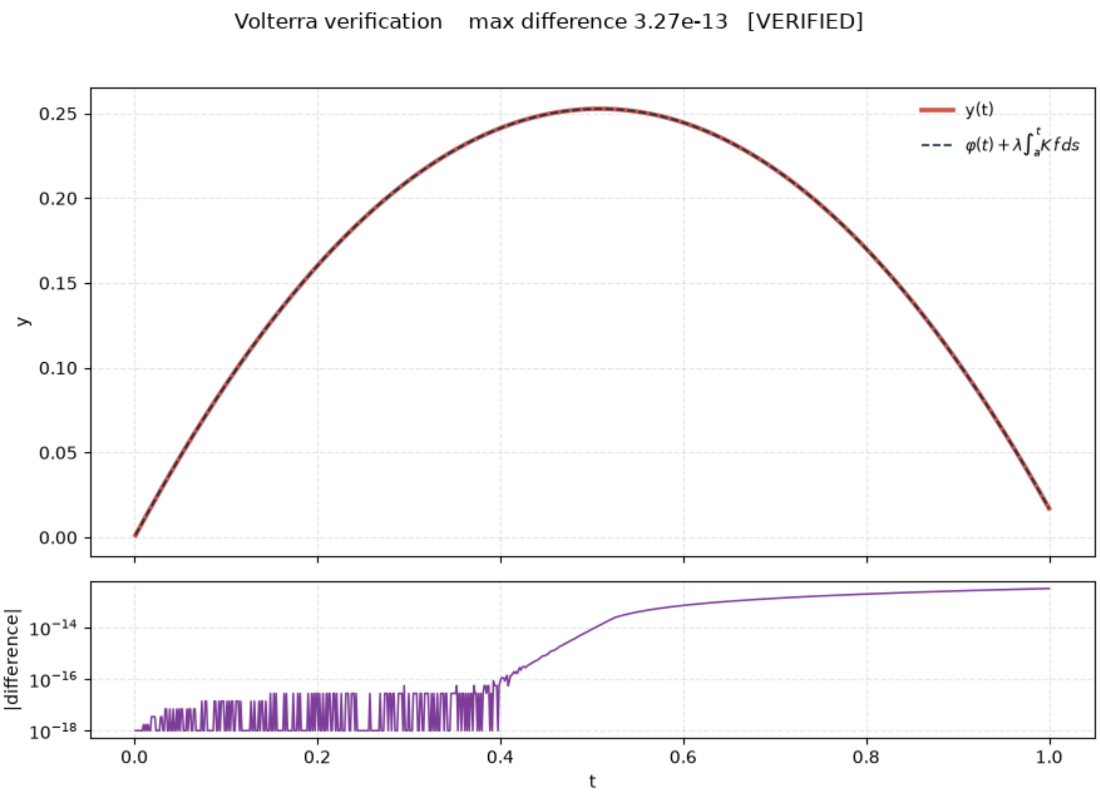

Figure 2: Verification: the returned solution substituted back into the integral equation, with the defect measured across the interval.

- **The equation's own initial data is recognised and reported.** SIRPY does not ask for conditions the equation already contains — it extracts them and shows the relations used.
- **Analytic continuation, disclosed.** When one series was not enough to cover the interval reliably, the solution was continued onto a second segment and the handover point reported. Still analytic everywhere, and stated plainly rather than hidden.
- **Verified by substitution** to 3×10^{-13}, with both absolute and relative residuals reported.

A note on scope

The residual here, 3×10^{-13}, is larger than the machine-precision figures reported elsewhere in this library. That is the honest floor for this class of check, and we report it as measured rather than rounding it toward a more flattering number. What is claimed is exactly what was verified: the returned solution satisfies Equation 1 to 3.27×10^{-13} across $[0, 1]$.

Next in the series

A Volterra equation determines its own solution completely — which raises an awkward question. What happens if you try to impose an *additional* condition on it? That problem is next.

The problem library

This entry is part of the SIRPY Problem Library — an ongoing, public catalog of mathematical systems solved and verified by SIRPY.

- **GitHub** — github.com/Mitra-Institute-of-Education-MIE/sirpy-problem-library
- **Web** — mitrainstituteofeducation.org/sirpy

Submit a problem

Have a differential or integral equation you believe is challenging, unusual, or simply beautiful? We would like to see it. We will run it through SIRPY and report honestly what happened — if it solves it, how; if it struggles, why; if it reaches a current limit, we will say so.

Contact: info@mitrainstituteofeducation.org · mitrainstituteofeducation.org/contact

Please only send problems you are free to share publicly.

SIRPY is in final validation prior to its first public release. Every line quoted above is verbatim from a single run of the solver; nothing is reproduced from memory.