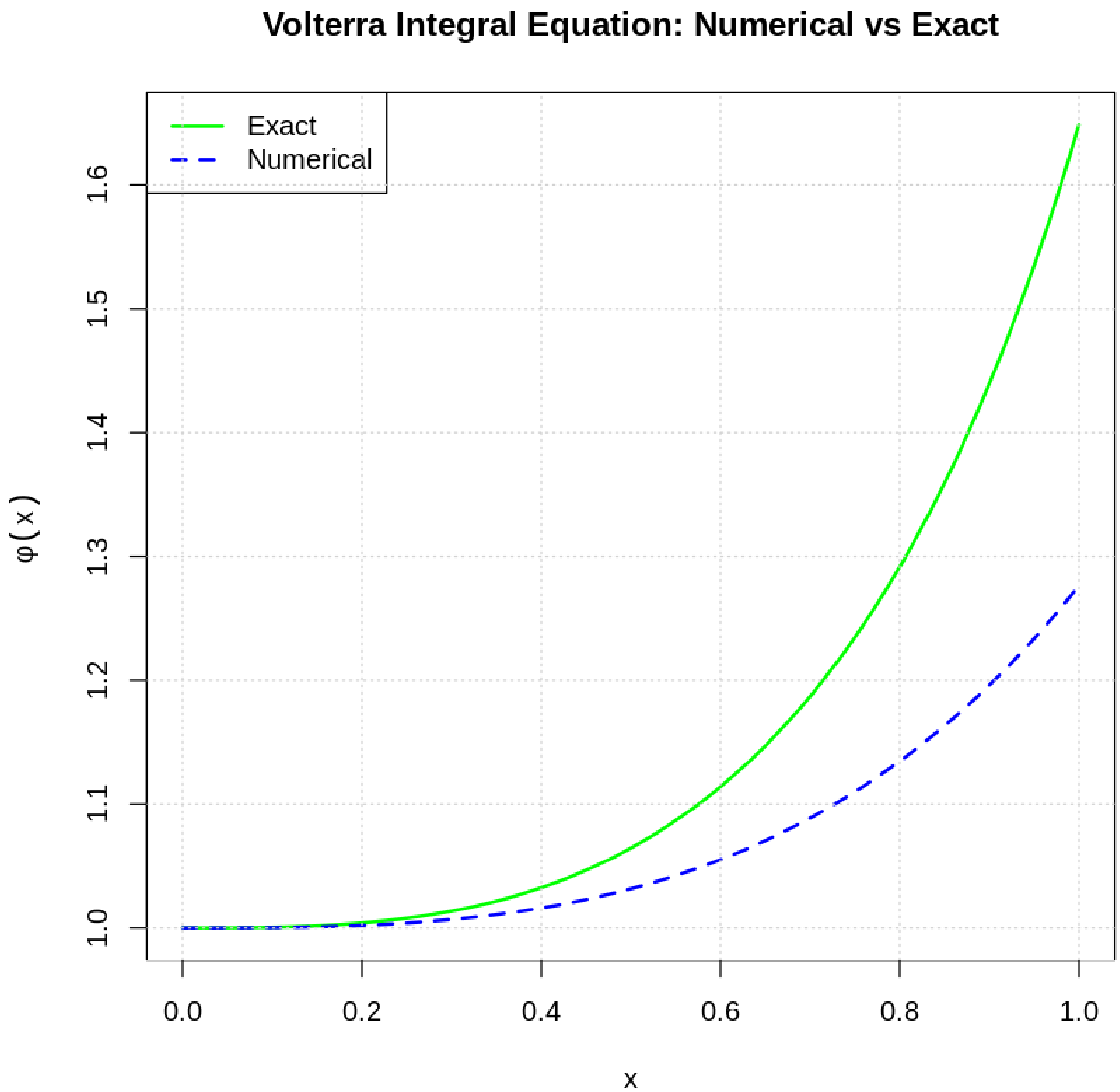
```
# Volterra Integral Equation Solver using Successive Approximation
# φ(x) = f(x) + λ ∫[a,x] K(x, t) φ(t) dt

# Define problem parameters
a <- 0              # Lower limit
lambda <- 0.5       # λ value
n <- 500            # Number of grid points
max_iter <- 10      # Max iterations
tol <- 1e-6         # Convergence tolerance

# Discretize the interval [a, endpoint]
x_end <- 1
x <- seq(a, x_end, length.out = n)
h <- (x_end - a) / (n - 1)

# Define f(x)
f <- function(x) {
  return(1)  # Example: f(x) = 1
}

# Define kernel K(x, t)
K <- function(x, t) {
  return(x * t)  # Example: K(x, t) = x * t
}

# Initial guess φ₀(x)
phi_old <- rep(0, n)

# Successive approximations
for (iter in 1:max_iter) {
  phi_new <- numeric(n)

  for (i in 1:n) {
    xi <- x[i]
    t_vals <- x[1:i]              # Integration from a to x[i]
    phi_vals <- phi_old[1:i]
    kernel_vals <- K(xi, t_vals)
    integrand <- kernel_vals * phi_vals

    if (i == 1) {
      integral <- 0
    } else {
      integral <- h * (0.5 * integrand[1] + sum(integrand[2:(i - 1)]) + 0.5 * integrand[i])
```

```r
    }

    phi_new[i] <- f(xi) + lambda * integral
  }

  # Check convergence
  if (max(abs(phi_new - phi_old)) < tol) {
    cat("Converged in", iter, "iterations.\n")
    break
  }

  phi_old <- phi_new
}

# Exact solution using the known series
phi_exact <- function(x, lambda, terms = 10) {
  y <- 0
  for (n in 0:terms) {
    y <- y + (x^(3 * n)) / (factorial(n) * (2^n))
  }
  return(y)
}

phi_exact_vals <- sapply(x, phi_exact, lambda = lambda)
error_vals <- abs(phi_new - phi_exact_vals)

# Display comparison
comparison_table <- data.frame(
  x = round(x, 4),
  phi_numerical = round(phi_new, 6),
  phi_exact = round(phi_exact_vals, 6),
  abs_error = round(error_vals, 6)
)
print(head(comparison_table, 10))

# Plot numerical vs exact solution
plot(x, phi_exact_vals, type = "l", col = "green", lwd = 2,
     ylim = range(c(phi_new, phi_exact_vals)),
     main = "Volterra Integral Equation: Numerical vs Exact",
     xlab = "x", ylab = expression(phi(x)))
lines(x, phi_new, col = "blue", lwd = 2, lty = 2)
legend("topleft", legend = c("Exact", "Numerical"),
       col = c("green", "blue"), lty = c(1, 2), lwd = 2)
grid()
```

Output:
Converged in 7 iterations.

	x	phi_numerical	phi_exact	abs_error
1	0.000	1.000000	1.000000	0e+00
2	0.002	1.000000	1.000000	0e+00
3	0.004	1.000000	1.000000	0e+00
4	0.006	1.000000	1.000000	0e+00
5	0.008	1.000000	1.000000	0e+00
6	0.010	1.000000	1.000001	0e+00
7	0.012	1.000000	1.000001	0e+00
8	0.014	1.000001	1.000001	1e-06
9	0.016	1.000001	1.000002	1e-06
10	0.018	1.000001	1.000003	1e-06



Volterra Integral Equation: Numerical vs Exact
Exact
Numerical
φ(x)
1.0
1.1
1.2
1.3
1.4
1.5
1.6
0.0
0.2
0.4
0.6
0.8
1.0
x