Cubic-Convergent Newton-Type Methods for Nonlinear Differential Equations: Applications to Navier–Stokes via Finite Elements

Malik Mohammed Abdelbagi[1], **Aymen Laadhari**[1]

[1]Mathematics Department, College of Computing and Mathematical Sciences, Khalifa University, UAE

E-mail: `100059981@ku.ac.ae, aymen.laadhari@ku.ac.ae`

Abstract. We investigate cubic-convergent Newton-type methods for nonlinear differential problems, focusing on the variants of Kou, Homeier, and Weerakoon and their multidimensional formulations. The schemes are validated on scalar roots, a system of nonlinear equations, an initial value problem, and then applied to the steady incompressible Navier–Stokes problem. Across all tests, cubic methods reach the target tolerance in fewer iterations than classical Newton, with measured orders confirming third- versus second-order behavior. Notably, only Kou's method reuses a single Jacobian per outer iteration, reducing assembly and linear-solve cost while retaining cubic convergence; the other cubic variants require additional Jacobian evaluations or factorizations. Overall, the results highlight the practical efficiency of high-order Newton-type strategies, with Kou's method delivering the best performance in our experiments.

1. Introduction

The process of locating equation zeros constitutes a fundamental domain within computational analysis, providing essential techniques for determining solutions to $\psi(\xi) = 0$. These problems naturally emerge in diverse applications including fluid mechanics, thermodynamics, chemical kinetics, population dynamics, and financial modeling. Efficient and reliable algorithms for locating the roots of nonlinear equations are therefore crucial for both theoretical investigations and applied simulations.

Among the numerous available iterative techniques, the Newton's method is known for its elegant formulation, rapid (quadratic) convergence, and ease of implementation. Historically introduced by Isaac Newton and Joseph Raphson in the 17th century, the iterative process reads

$$\xi_{n+1} = \xi_n - \frac{\psi(\xi_n)}{\psi'(\xi_n)}, \tag{1}$$

and converges quadratically toward the simple root ξ^* whenever ψ is sufficiently smooth and ξ_0 is chosen sufficiently ξ^*. Specifically, if ψ is continuously differentiable and $\psi'(\xi^*) \neq 0$, then there exists a radius $\beta > 0$ such that any ξ_0 with $|\xi_0 - \xi^*| < \beta$ produces an iterate sequence $\{\xi_n\}$ that tends to ξ^*, with error ratio $|e_{n+1}| \approx C|e_n|^2$ (for some $C > 0$) as n grows.

Despite its efficiency, the Newton method may become computationally expensive or even divergent for highly nonlinear systems or poor initial guesses. This has motivated extensive research on constructing higher-order variants that preserve the simplicity of Newton's structure while improving its convergence order. Such developments include the cubic modifications proposed by [3], [2], and [1], which achieve third-order convergence through additional correction terms without requiring higher derivatives.

Over the past two decades, many authors have applied these high-order Newton-type methods to a wide range of nonlinear PDE models. For example, recent studies have addressed reaction-diffusion equations, population growth dynamics [7], and epidemiological models through fractional or hybrid iterative techniques [4, 5, 6]. In computational physics, [7, 8] designed families of high-order iterative algorithms applicable to nonlinear systems and heat transfer problems. In fluid dynamics, several works have demonstrated the potential of cubic and enhanced Newton methods within fully implicit finite element formulations, covering free-surface flows [9], bubble dynamics [10, 11], biomembrane deformation [12], and two-phase fluid interactions [14]. More recently, [13, 15] have investigated higher-order schemes, confirming the efficiency of such methods in CFD computations.

In this context, the present work investigates and applies in particular cubic-convergent Newton-type algorithms to nonlinear differential systems, focusing on two benchmark applications: an initial-value problem with exact solution, and the steady incompressible Navier–Stokes equations. The proposed framework is implemented using finite element discretization and tested across various problem settings to assess its features.

In what follows, Section 2 presents the theoretical formulation of several Newton-type modifications, including Kou's method and its generalization to nonlinear systems. Section 3 presents numerical experiments and comparative performance analyses on representative problems. Finally, concluding remarks are drawn in Section 4, highlighting the perspectives for future work.

2. Mathematical Setting

In this section, we examine three modifications of Newton's method, emphasizing their construction principles and (local) convergence orders.

2.1. Newton's Method: Kou's Modification

A third–order variant of the classical scheme (1) was proposed by J. Kou [1]. The derivation starts from the definite integral identity

$$\psi(\xi_{n+1}) = \psi(\lambda_n) + \int_{\lambda_n}^{\xi_{n+1}} \psi'(t)\,\mathrm{d}t, \tag{2}$$

with the auxiliary point

$$\lambda_n = \xi_n + \frac{\psi(\xi_n)}{\psi'(\xi_n)}. \tag{3}$$

Approximating the integral in (2) by the midpoint rule yields

$$\int_{\lambda_n}^{\xi_{n+1}} \psi'(t)\,\mathrm{d}t \;\approx\; \left(\xi_{n+1} - \lambda_n\right) f'\!\left(\frac{\xi_{n+1} + \lambda_n}{2}\right),$$

and, after setting $\psi(\xi_{n+1}) = 0$ and isolating ξ_{n+1}, one obtains

$$\xi_{n+1} = \lambda_n - \frac{\psi(\lambda_n)}{f'\!\left(\frac{\xi_{n+1} + \lambda_n}{2}\right)}. \tag{4}$$

A practical one–step implementable formula follows by replacing the midpoint derivative in (4) with the first–order proxy $\psi'(\xi_n)$ (using (1) to justify the approximation near a simple root), which leads to

$$\xi_{n+1} = \lambda_n - \frac{\psi(\lambda_n)}{\psi'(\xi_n)}, \tag{5}$$

the Kou modification [1].

Theorem (Kou [1]). If $\psi \in C^3[a,b]$ and ψ possesses a single zero ξ^*, the iteration scheme (5) converges cubically to ξ^*, provided the starting estimate ξ_0 lies sufficiently near that point.

2.2. Newton's Method: Homeier's Modification

A further cubic refinement of Newton's scheme was proposed by H. Homeier [2]. The iteration is defined by

$$F(\xi) = \xi - \frac{\psi(\xi)}{\psi'(\xi + \phi(\xi)\psi(\xi))},$$

with ϕ a sufficiently smooth scalar function. Several admissible choices of ϕ yield third-order convergence; the selection used here is

$$\phi(\xi) = -\frac{1}{2\psi'(\xi)}.$$

With this choice, the practical update becomes

$$\xi_{n+1} = \xi_n - \frac{\psi(\xi_n)}{\psi'(\lambda_n)}, \tag{6}$$

where

$$\lambda_n = \xi_n - \frac{\psi(\xi_n)}{2\psi'(\xi_n)}.$$

Theorem (Homeier [2]). Let $\psi \in C^3[a,b]$ and let ξ^* be a simple root. If the third derivative of the iteration function F is bounded in a neighborhood of ξ^*, then the scheme (6) converges cubically to ξ^* for initial guesses sufficiently close to ξ^*.

2.3. Newton's Method: Weerakoon's Modification

A third–order variant due to S. Weerakoon and T. Fernando [3] starts from the identity

$$\psi(\xi_{n+1}) = \psi(\xi_n) + \int_{\xi_n}^{\xi_{n+1}} \psi'(t)\,dt, \tag{7}$$

and applies the trapezoidal rule to the integral, whichafter algebraic rearrangementgives the implicit update

$$\xi_{n+1} = \xi_n - \frac{2\psi(\xi_n)}{\psi'(\xi_n) + \psi'(\xi_{n+1})}. \tag{8}$$

Introducing the Newton predictor

$$\lambda_n = \xi_n - \frac{\psi(\xi_n)}{\psi'(\xi_n)}, \tag{9}$$

and substituting (9) into (8) yields the explicit scheme

$$\xi_{n+1} = \xi_n - \frac{2\psi(\xi_n)}{\psi'(\xi_n) + \psi'(\lambda_n)}, \tag{10}$$

known as the Weerakoon–Fernando modification.

Theorem (Weerakoon–Fernando [3]). Assume $\psi \in C^3[a,b]$ and ξ^* is simple zero. Then, the iteration (10) converges cubically for initial guesses sufficiently close to ξ^*.

Numerical Convergence. Let (ξ_n) be the iterates produced by a root-finding method for a (simple) root ξ^* of ψ, and define the errors $e_n := \xi_n - \xi^*$. The method converges with (asymptotic) order $\alpha > 0$ and error constant $\kappa \in (0, \infty)$ if

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = \kappa,$$

which quantifies the rate at which the sequence approaches the exact solution. In computations, α is commonly estimated from three successive errors via the logarithmic quotient:

$$\widehat{\alpha}_n \approx \frac{\ln|\xi_{n+1} - \xi^*|/|\xi_n - \xi^*|}{\ln|\xi_n - \xi^*|/|\xi_{n-1} - \xi^*|}, \qquad \widehat{\kappa}_n \approx |e_{n+1}|\,|e_n|^{-\widehat{\alpha}_n},$$

providing an empirical measure used to assess and compare the efficiency of iterative algorithms (linear when $\alpha = 1$, quadratic when $\alpha = 2$, etc.). When the exact root ξ^* is unavailable to form e_n, one may replace errors by iterate differences $\Delta\xi_n := \xi_{n+1} - \xi_n$ and use the apparent convergence order:

$$\widehat{\alpha}_n^{(A)} \approx \frac{\ln|\Delta\xi_{n+1}/\Delta\xi_n|}{\ln|\Delta\xi_n|/|\Delta\xi_{n-1}|}, \qquad \widehat{\kappa}_n^{(A)} \approx |\Delta\xi_{n+1}|\,|\Delta\xi_n|^{-\widehat{\alpha}_n^{(A)}},$$

which asymptotically matches α and κ under standard regularity. For the Newton's approach, if $\psi \in C^2$ near the single root ξ^*, the error satisfies

$$e_{n+1} = \frac{f''(\xi^*)}{2\psi'(\xi^*)}\, e_n^2 + \mathcal{O}(e_n^3),$$

so the order is $\alpha = 2$ (quadratic), while $\kappa = \left|f''(\xi^*)/(2\psi'(\xi^*))\right|$; these estimators are reliable in the asymptotic regime where the errors are sufficiently small and nonzero.

2.4. Newton's Method for Nonlinear Systems

We consider systems $\boldsymbol{\Psi}(\Xi) = \mathbf{0}$ with Jacobian $\boldsymbol{K}(\Xi) = D\boldsymbol{\Psi}(\Xi)$. Local convergence is ensured when the initial guess satisfies $\|\Xi_0 - \Xi^*\| < \beta$ for some $\beta > 0$. The order $\alpha > 0$ is defined by

$$\alpha \approx \frac{\log\|\Xi_{n+1} - \Xi^*\| - \log\|\Xi_n - \Xi^*\|}{\log\|\Xi_n - \Xi^*\| - \log\|\Xi_{n-1} - \Xi^*\|}. \tag{11}$$

Denote $\boldsymbol{e}_n = \Xi_n - \Xi^*$. We next introduce, in a compact form, the multivariate Newton iteration and its cubic variants used later.

Classical scheme. The extension to the multidimensional case reads

$$\Xi_{n+1} = \Xi_n - \boldsymbol{K}(\Xi_n)^{-1} \cdot \boldsymbol{\Psi}(\Xi_n). \tag{12}$$

Kou's modification. The Kou-type extension of (5) is

$$\begin{aligned}\Xi_{n+1} &= \boldsymbol{\chi}_n - \boldsymbol{K}(\Xi_n)^{-1}\cdot\boldsymbol{\Psi}(\boldsymbol{\chi}_n), \qquad \text{with}\\ \boldsymbol{\chi}_n &= \Xi_n + \boldsymbol{K}(\Xi_n)^{-1}\cdot\boldsymbol{\Psi}(\Xi_n).\end{aligned} \tag{13}$$

Homeier's modification. The vector form of (6) is

$$\begin{aligned}\Xi_{n+1} &= \Xi_n - \boldsymbol{K}(\boldsymbol{\chi}_n)^{-1}\cdot\boldsymbol{\Psi}(\Xi_n), \qquad \text{with}\\ \boldsymbol{\chi}_n &= \Xi_n - \frac{1}{2}\cdot\boldsymbol{K}(\Xi_n)^{-1}\cdot\boldsymbol{\Psi}(\Xi_n).\end{aligned} \tag{14}$$

Weerakoon–Fernando modification. The multidimensional extension of (10) is

$$\begin{aligned}\Xi_{n+1} &= \Xi_n - 2\cdot\left(\boldsymbol{K}(\Xi_n)+\boldsymbol{K}(\boldsymbol{\chi}_n)\right)^{-1}\cdot\boldsymbol{\Psi}(\Xi_n), \qquad \text{with}\\ \boldsymbol{\chi}_n &= \Xi_n - \boldsymbol{K}(\Xi_n)^{-1}\cdot\boldsymbol{\Psi}(\Xi_n).\end{aligned} \tag{15}$$

Remark that the notation $\boldsymbol{K}(\cdot)^{-1}$ indicates the operation of obtaining a solution to the associated linear equations defined by $\boldsymbol{K}(\cdot)$; explicit matrix inversion is not required.

2.5. High–order solvers for the steady Navier–Stokes

This problem is a cornerstone of Computational Fluid Dynamics. In a non-dimensional setting with Reynolds number $\mathrm{Re} > 0$, the steady problem on a domain Ω seeks the velocity $\boldsymbol{v}$ and pressure p satisfying

$$\mathrm{Re}\,(\boldsymbol{v}\cdot\nabla)\,\boldsymbol{v} - \nabla\cdot\big(2\,\mathbf{D}(\boldsymbol{v})\big) + \nabla p = \mathbf{0} \quad \text{in } \Omega, \tag{16a}$$

$$\nabla\cdot\boldsymbol{v} = 0 \quad \text{in } \Omega, \tag{16b}$$

$$\boldsymbol{v} = \boldsymbol{v}_b \quad \text{on } \Sigma_D = \partial\Omega, \tag{16c}$$

with $2\mathbf{D}(\boldsymbol{v}) := \nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T$. In the canonical driven-cavity configuration, Ω is the unit square (2D); the top lid moves tangentially and the remaining walls are stationary, yielding the no-slip condition $\boldsymbol{v} = \boldsymbol{v}_b$ on all of $\partial\Omega$.

For the functional setting, we introduce the velocity and pressure spaces

$$\mathbb{X}(\boldsymbol{v}_b) = \big\{\boldsymbol{\zeta}\in(H^1(\Omega))^d:\ \boldsymbol{\zeta}=\boldsymbol{v}_b \text{ on } \Sigma_D\big\}, \qquad \mathbb{P} = L^2_0(\Omega) = \Big\{\varsigma\in L^2(\Omega):\ \int_\Omega \varsigma\,\mathrm{d}x = 0\Big\}.$$

For $\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2,\boldsymbol{\zeta}_3\in(H^1(\Omega))^d$ and $\varsigma\in L^2(\Omega)$, define

$$\begin{aligned}\mathcal{D}(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2) &= \int_\Omega 2\,\mathbf{D}(\boldsymbol{\zeta}_1):\mathbf{D}(\boldsymbol{\zeta}_2)\,\mathrm{d}x, \qquad \mathcal{B}(\boldsymbol{\zeta}_1,\varsigma) = -\int_\Omega \varsigma\,\nabla\cdot\boldsymbol{\zeta}_1\,\mathrm{d}x,\\ \mathcal{C}(\boldsymbol{\zeta}_1,\boldsymbol{\zeta}_2;\boldsymbol{\zeta}_3) &= \int_\Omega \big((\boldsymbol{\zeta}_1\cdot\nabla)\boldsymbol{\zeta}_3 + (\boldsymbol{\zeta}_3\cdot\nabla)\boldsymbol{\zeta}_1\big)\cdot\boldsymbol{\zeta}_2\,\mathrm{d}x,\end{aligned}$$

where c is a symmetrized convective form. The weak form corresponding to the steady problem is given by: Search for $(\boldsymbol{v},p)\in\mathbb{X}(\boldsymbol{v}_b)\times\mathbb{P}$ s.t. for all test functions $(\boldsymbol{\zeta},\varsigma)\in\mathbb{X}(\mathbf{0})\times\mathbb{P}$,

$$\frac{\mathrm{Re}}{2}\,\mathcal{C}(\boldsymbol{v},\boldsymbol{\zeta};\boldsymbol{v}) + \mathcal{D}(\boldsymbol{v},\boldsymbol{\zeta}) + \mathcal{B}(\boldsymbol{\zeta},p) = 0, \tag{17}$$

$$\mathcal{B}(\boldsymbol{v},\varsigma) = 0. \tag{18}$$

Let $\boldsymbol{u} = (\boldsymbol{v}, p)^T$ denote the unknown pair and set $\boldsymbol{W} = (\boldsymbol{\zeta}, \varsigma)^T$. The residual $\boldsymbol{\mathcal{F}}(\boldsymbol{u})$ associated with (17) reads

$$\big\langle \boldsymbol{\mathcal{F}}(\boldsymbol{u}), \boldsymbol{W} \big\rangle = \Big(\underbrace{\tfrac{\mathrm{Re}}{2}\,\mathcal{C}(\boldsymbol{v},\boldsymbol{\zeta};\boldsymbol{v}) + \mathcal{D}(\boldsymbol{v},\boldsymbol{\zeta}) + \mathcal{B}(\boldsymbol{\zeta},p)}_{\langle \boldsymbol{\mathcal{F}}_{\boldsymbol{v}}(\boldsymbol{u}),\boldsymbol{\zeta}\rangle},\ \underbrace{\mathcal{B}(\boldsymbol{v},\varsigma)}_{\langle \boldsymbol{\mathcal{F}}_p(\boldsymbol{u}),\varsigma\rangle} \Big)^T = \boldsymbol{0}, \quad \forall\, \boldsymbol{W} \in \mathbb{X}(\boldsymbol{0}) \times \mathbb{P}.$$

Given an iterate $\boldsymbol{u}^{(l)} = (\boldsymbol{v}^{(l)}, p^{(l)})$, Newton's method (and its higher-order variants) compute an update $\delta\boldsymbol{u}^{(l+1)} = (\delta\boldsymbol{v}^{(l+1)}, \delta p^{(l+1)})$ by solving the tangent system obtained from the Gateaux derivative of $\boldsymbol{\mathcal{F}}$:

$$\begin{aligned} \frac{\mathrm{Re}}{2}\,\mathcal{C}\big(\delta\boldsymbol{v}^{(l+1)},\boldsymbol{\zeta};\boldsymbol{v}^{(l)}\big) + \frac{\mathrm{Re}}{2}\,\mathcal{C}\big(\boldsymbol{v}^{(l)},\boldsymbol{\zeta};\delta\boldsymbol{v}^{(l+1)}\big) + \mathcal{A}\big(\delta\boldsymbol{v}^{(l+1)},\boldsymbol{\zeta}\big) \\ + \mathcal{B}\big(\boldsymbol{\zeta},\delta p^{(l+1)}\big) = -\big\langle \boldsymbol{\mathcal{F}}_{\boldsymbol{v}}(\boldsymbol{u}^{(l)}),\boldsymbol{\zeta}\big\rangle, \qquad (19) \\ \mathcal{B}\big(\delta\boldsymbol{v}^{(l+1)},\varsigma\big) = -\big\langle \boldsymbol{\mathcal{F}}_p(\boldsymbol{u}^{(l)}),\varsigma\big\rangle, \qquad (20) \end{aligned}$$

for all $(\boldsymbol{\zeta},\varsigma) \in \mathbb{X}(\boldsymbol{0}) \times \mathbb{P}$. Hence, $\boldsymbol{u}^{(l)}$ is updated by adding the correction. For high–order nonlinear variants, additional terms are appended to the previous tangent problem, which is adapted as detailed in subsections 13, 14 and 15.

We thereafter provide the entire Algorithm 1 when using Kou's method 13. Remark that this algorithm can be easily adapted for the other variants. Let $\boldsymbol{K}(\boldsymbol{u}) = D_{\boldsymbol{u}}\boldsymbol{\mathcal{F}}(\boldsymbol{u})$ denote the Jacobian at the current iterate. Kou's modification uses a frozen Jacobian evaluated at $\boldsymbol{u}^{(l)}$ for both substeps.

Algorithm 1 Cubic-order Kou strategy for Navier–Stokes

```
Input: Initial guess $\boldsymbol{u}^{(0)} = (\boldsymbol{v}^{(0)}, p^{(0)})$, tolerance $\varepsilon_{\mathrm{res}} > 0$, max iterations $l_{\max}$.
for $l = 0, 1, 2, \ldots, l_{\max}$ do
    Residual: form $\langle \boldsymbol{\mathcal{F}}(\boldsymbol{u}^{(l)}), \boldsymbol{W} \rangle$ for all $\boldsymbol{W} = (\boldsymbol{\zeta},\varsigma) \in \mathbb{X}(\boldsymbol{0}) \times \mathbb{P}$.
    if $\|\boldsymbol{\mathcal{F}}(\boldsymbol{u}^{(l)})\| \le \varepsilon_{\mathrm{res}}$ then stop
    end if
    Frozen Jacobian: $\boldsymbol{K}^{(l)} := D_{\boldsymbol{u}}\boldsymbol{\mathcal{F}}(\boldsymbol{u}^{(l)})$.
    Stage 1 (predictor, corrected sign): find $\delta\boldsymbol{u}^{(l+\frac{1}{2})}$ from
        $\big\langle \boldsymbol{K}^{(l)}\,\delta\boldsymbol{u}^{(l+\frac{1}{2})}, \boldsymbol{W} \big\rangle = \big\langle \boldsymbol{\mathcal{F}}(\boldsymbol{u}^{(l)}), \boldsymbol{W} \big\rangle, \quad \forall\, \boldsymbol{W}.$
    Intermediate state: $\mathbf{Y}^{(l)} := \boldsymbol{u}^{(l)} + \delta\boldsymbol{u}^{(l+\frac{1}{2})}$.
    Residual at $\mathbf{Y}^{(l)}$: form $\langle \boldsymbol{\mathcal{F}}(\mathbf{Y}^{(l)}), \boldsymbol{\Phi} \rangle$ for all $\boldsymbol{\Phi} = (\boldsymbol{\zeta},\varsigma) \in \mathbb{X}(\boldsymbol{0}) \times \mathbb{P}$.
    Stage 2 (corrector with frozen Jacobian): find $\delta\boldsymbol{u}^{(l+1)}$ from
        $\big\langle \boldsymbol{K}^{(l)}\,\delta\boldsymbol{u}^{(l+1)}, \boldsymbol{\Phi} \big\rangle = -\big\langle \boldsymbol{\mathcal{F}}(\mathbf{Y}^{(l)}), \boldsymbol{\Phi} \big\rangle, \quad \forall\, \boldsymbol{\Phi}.$
    Update: $\boldsymbol{u}^{(l+1)} := \mathbf{Y}^{(l)} + \delta\boldsymbol{u}^{(l+1)}$.
    if $\|\boldsymbol{\mathcal{F}}(\boldsymbol{u}^{(l+1)})\| \le \varepsilon_{\mathrm{res}}$ then stop
    end if
end for
Output: $(\boldsymbol{v}, p) = \boldsymbol{u}^{(l+1)}$.
```

3. Numerical Testing

We assess the schemes on representative examples to assess the convergence order and practical efficiency under the tolerance $\epsilon = 10^{-12}$. The diagnostic quantity α (11) is reported to verify the observed order of convergence.

Consider $\psi(\xi) = x^3 + 4x^2 - 15$ with $\xi_0 = -0.9$. We solve $\psi(\xi) = 0$ using Newton (N2) and the cubic schemes (called K3, H3, and W3). The table reports the converged iterate ξ_n, the residual $|\psi(\xi_n)|$, the iteration count n to reach the tolerance, and NFE, the total number of scalar evaluations of ψ and ψ'.

Table 1. Performance comparison of the different variants. Here, n is the number of iterations; NFE counts scalar ψ and ψ' evaluations.

	n	NFE	ξ_n	$\|\psi(\xi_n)\|$
N2	25	50	1.63198080556606672786	0.00000000000006750156
K3	4	12	1.63198080556606339719	0.00000000000000355271
H3	9	27	1.63198080556606339719	0.00000000000000355271
W3	6	18	1.63198080556606339719	0.00000000000000355271

To complement the table, Figure 1 plots log $|\psi(\xi_n)|$ versus the iteration index n for all schemes.

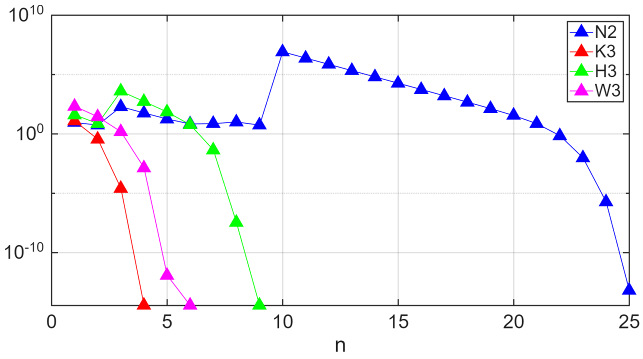

Figure 1. Error decay for N2 and its cubic variants applied to $\psi(\xi) = 0$.

Table 1 and Figure 1 show the expected behavior: the cubic schemes reduce the residual much faster than the quadratic Newton iteration. Among the cubic variants, Kou's method (K3) typically attains the tolerance in the fewest steps in this test. Importantly, K3 reuses the Jacobian at ξ_n for both substeps (see Subsection 13), so only one Jacobian assembly/linear solve per outer iteration is required, which is advantageous for the PDE examples below.

We further test a non-polynomial case $g(\xi) = \xi e^{\xi^2} - \sin^2(\xi) + 3\cos(\xi) + 5$ from the initial guess $\xi_0 = -2$. Here we solve $g(\xi) = 0$ with the same set of methods; to verify the convergence order, we report the estimated α per iteration in Table 2.

Table 2. Estimated numerical convergence order.

n	N2	K3	H3	W3
2	1.51	2.53	2.25	1.98
3	1.68	3.27	2.83	2.52
4	1.86	3.06	3.00	2.92
5	1.96	–	–	3.00
6	1.99	–	–	–
7	2.00	–	–	–

As the iterates approach the root, the measured α stabilizes near the theoretical orders: α approaches 2 for Newton (N2) and α approaches 3 for the cubic variants (K3, H3, W3) when the convergence is established. Additionally, the number of iterations is another performance measure, for which Kou's method generally requires the fewest steps in our tests.

3.1. Nonlinear Algebraic System

We now focus on a two-variable system $\boldsymbol{\Psi}(\boldsymbol{\Xi}) = \mathbf{0}$ and apply all schemes until the stopping criterion $\|\boldsymbol{\Psi}(\boldsymbol{\Xi}_n)\| \le \epsilon$ is met. Set $\boldsymbol{\Xi}_0 = (1,1)^T$. We report the converged iterate $\boldsymbol{\Xi}_n$, the residual norm $\|\boldsymbol{\Psi}(\boldsymbol{\Xi}_n)\|$, the iteration count n, and the measured CPU time t_{CPU}.

$$\boldsymbol{\Psi}(\boldsymbol{\Xi}) = \begin{pmatrix} \sin(\xi_1\xi_2) + \xi_2^3 - 4 \\ e^{\xi_1} + \xi_1\cos(\xi_2) - 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

with the Jacobian matrix given by

$$\boldsymbol{K}(\boldsymbol{\Xi}) = \begin{pmatrix} \xi_2\cos(\xi_1\xi_2) & \xi_1\cos(\xi_1\xi_2) + 3\xi_2^2 \\ e^{\xi_1} + \cos(\xi_2) & -\xi_1\sin(\xi_2) \end{pmatrix}.$$

Table 3. Performance comparison of the different methods when applied to the nonlinear algebraic system.

	n	t_{CPU}	$\boldsymbol{\Xi}_n$	$\|\|\boldsymbol{\Psi}(\boldsymbol{\Xi}_n)\|\|$
N2	5	0.01562500000000000000	$\begin{pmatrix} 0.65936106092230228892 \\ 1.46985549775510748738 \end{pmatrix}$	0.00000000000000088818
K3	4	0.01562500000000000000	$\begin{pmatrix} 0.65936106092230239994 \\ 1.46985549775510770942 \end{pmatrix}$	0.00000000000000000000
H3	3	0.01562500000000000000	$\begin{pmatrix} 0.65936106092229718190 \\ 1.46985549775506685322 \end{pmatrix}$	0.00000000000028466638
W3	4	0.01562500000000000000	$\begin{pmatrix} 0.65936106092230239994 \\ 1.46985549775510770942 \end{pmatrix}$	0.00000000000000000000

Figure 2 complements the table by displaying the decay of the residual norm along the iterations, showing a better convergence of K3, H3 and W3.

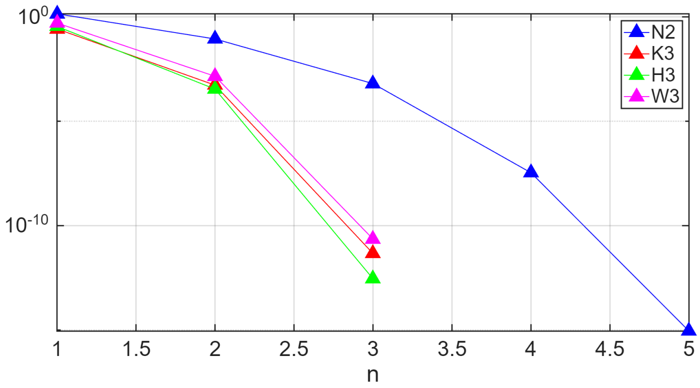

Figure 2. Error decay for N2 and the modified schemes.

Table 4. Nonlinear algebraic system: convergence rates of N2 and modified schemes.

n	N2	K3	H3	W3
2	2.79	2.99	2.91	2.92
3	1.74	2.99	–	3.04
4	2.00	–	–	–

As shown in Table 3, the cubic variants reach the tolerance in at most three iterations, whereas the classical Newton method needs five. CPU times are essentially indistinguishable here due to the small problem size and coarse timing resolution, but the iteration counts still show the small advantage of higher order. Table 4 confirms the expected orders ($\alpha \approx 3$ for K3/H3/W3 and $\alpha \approx 2$ for N2), and Figure 2 shows the steeper residual decay for the cubic methods.

3.2. System of Ordinary Differential Equations with Exact Solution

We consider the IVP:

$$\begin{cases} \dfrac{dy_1}{dt} = -y_1 + y_1 y_2, \\ \dfrac{dy_2}{dt} = -y_2, \end{cases} \qquad \text{for } t \in [0,5], \quad y_1(0) = 2,\ y_2(0) = 2.5.$$

The exact solution is

$$y_1(t) = 2e^{-t} e^{2.5(1-e^{-t})}, \qquad y_2(t) = 2.5e^{-t}.$$

Let t_m be the discretized times ($m = 0, \ldots, M$) and d_t the step size. Let $\gamma_{i,m} \approx y_i(t_m)$ and discretize by backward Euler:

$$\begin{aligned} \gamma_{1,m+1} &= \gamma_{1,m} + d_t\big(-\gamma_{1,m+1} + \gamma_{1,m+1}\gamma_{2,m+1}\big), \\ \gamma_{2,m+1} &= \gamma_{2,m} + d_t(-\gamma_{2,m+1}), \end{aligned}$$

with $\gamma_{1,0} = 2$, $\gamma_{2,0} = 2.5$. We take $d_t = 0.01$ on $[0,5]$, i.e., $t_m = 0.01\,m$ for $m = 0, \ldots, 500$. At each t_{m+1}, the unknown vector $\boldsymbol{W}_{m+1} = (\gamma_{1,m+1}, \gamma_{2,m+1})^\top$ solves the nonlinear system

$$\boldsymbol{\Psi}(\boldsymbol{W}_{m+1}) = \begin{pmatrix} \gamma_{1,m+1} - \gamma_{1,m} + d_t\big(\gamma_{1,m+1} - \gamma_{1,m+1}\gamma_{2,m+1}\big) \\ \gamma_{2,m+1} - \gamma_{2,m} + d_t\, \gamma_{2,m+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

with Jacobian

$$\boldsymbol{K}(\boldsymbol{W}_{m+1}) = \begin{pmatrix} 1 + d_t(1 - \gamma_{2,m+1}) & -d_t\, \gamma_{1,m+1} \\ 0 & 1 + d_t \end{pmatrix}.$$

We solve this 2×2 system at each time step with the different Newton-type schemes; the iteration counts in Table 5 are accumulated over all steps.

Table 5. Backward Euler with Newton-type solvers for the IVP. Reported n: cumulative nonlinear iterations over all m; t_{CPU}: wall-clock time; $\|\boldsymbol{\Psi}(\boldsymbol{W}_M)\|$: final residual norm.

	n	t_{CPU}	$\|\boldsymbol{\Psi}(\boldsymbol{W}_M)\|$
N2	1000	1.5156250000000000000	0.0000000000003656881
K3	813	0.5781250000000000000	0.0000000003568973799
H3	788	0.6250000000000000000	0.0000000003580280390
W3	788	0.6250000000000000000	0.0000000003580283165

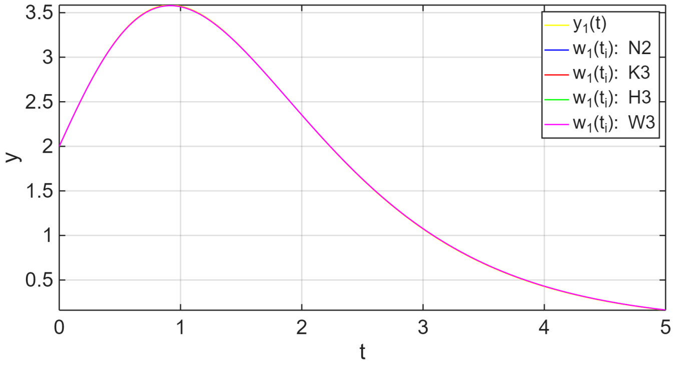

Figure 3. Exact $y_1(t_j)$ versus numerical $\gamma_{1,m}$ for the different solvers.

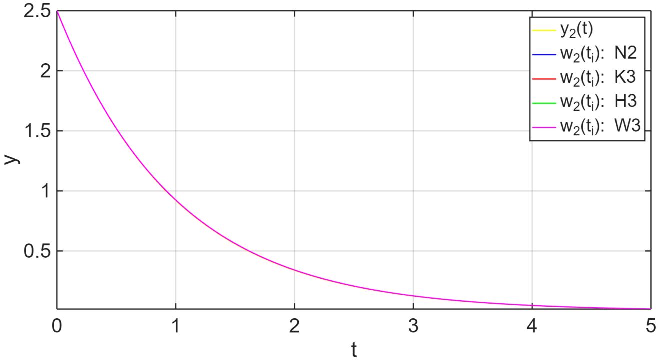

Figure 4. Exact $y_2(t_j)$ versus numerical $\gamma_{2,m}$ for the different solvers.

Figure 3 and Figure 4 confirm that all methods track the analytical solution closely on $[0, 5]$. As seen in Table 5, the cubic schemes reduce the total number of nonlinear iterations by roughly 20–25% (and the CPU time accordingly) compared with classical Newton, reflecting the benefit of higher order when many implicit solves are required.

3.3. Steady-state Navier–Stokes problem

We consider the classical lid-driven cavity in two dimensions with domain $\Omega = (0, 1)^2 \subset \mathbb{R}^2$. A unit tangential lid velocity is prescribed on the top boundary, while the remaining walls are motionless:

$$\boldsymbol{v}_b = (1, 0)^\top \text{ on the top lid}, \qquad \boldsymbol{v}_b = \boldsymbol{0} \text{ on the other three sides.}$$

For flow visualization, we use the vorticity ω and the streamfunction ψ. In 2D, the scalar vorticity measures local rotation and is the out-of-plane curl of the velocity, whereas the streamfunction satisfies

$$-\Delta\psi = \nabla \times \boldsymbol{v} \quad \text{in } \Omega,$$

together with homogeneous Dirichlet conditions on $\partial\Omega$. We examine three Reynolds regimes, Re $= 150$, Re $= 400$ and Re $= 1300$. In the numerical experiments, we use a time step size of $h = 10^{-2}$ and employ piecewise continuous $\mathbb{P}_2/\mathbb{P}_1$ Lagrange elements for discrete $\boldsymbol{v}$ and p. Figure 5 displays streamfunction isolines. At Re $= 150$, the main vortex spans most of the cavity's interior, with weak corner eddies; at Re $= 1300$, the principal eddy shifts further downstream and secondary vortices intensify near the bottom corners, consistent with the canonical flow structure.

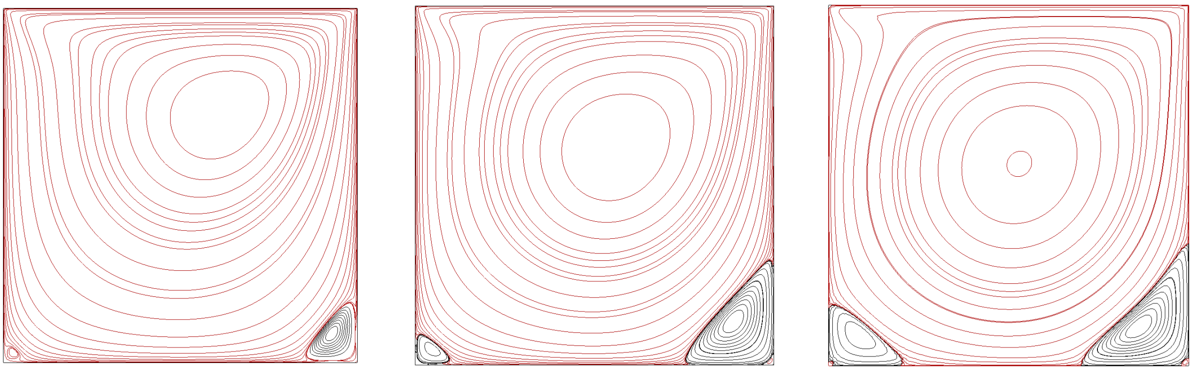

Figure 5. Steady solutions for several Re values: 150 (left), 400 (middle) and 1300 (right). Streamfunction isocontours showing the primary and corner eddies, with negative isovalues in red and positive ones in black.

Reynolds:	50		150		300		400		500	
k	$e^{(l)}$	α	$e^{(l)}$	ROC	$e^{(l)}$	ROC	$e^{(l)}$	ROC	$e^{(l)}$	ROC
1	9.252E-01	0.13	1.339E+01	0.07	5.197E+01	0.04	9.588E+01	0.03	1.591E+02	0.02
2	6.784E-04	2.06	9.963E-01	1.35	1.571E+01	0.95	3.538E+01	1.06	6.740E+01	1.31
3	5.490E-10	1.94	2.398E-03	2.32	4.850E+00	0.98	3.069E+01	0.14	7.426E+01	0.11
4			3.955E-08	1.83	1.444E-02	4.95	4.297E-01	30.00	4.085E+00	29.92
5			1.836E-11	0.70	5.932E-07	1.74	2.248E-03	1.23	1.536E-01	1.13
6					1.954E-11	1.02	7.960E-08	1.95	2.360E-04	1.97
7							2.007E-11	0.81	1.456E-10	2.21

Table 6. Error and convergence rates throughout the iterations for the Newton's method.

Reynolds:	50		150		300		400		500	
k	$e^{(l)}$	α	$e^{(l)}$	ROC	$e^{(l)}$	ROC	$e^{(l)}$	ROC	$e^{(l)}$	ROC
1	3.709E-04	0.25	1.552E-02	0.13	1.765E+02	0.37	2.347E+02	0.33	2.925E+02	0.30
2	1.812E-11	2.94	2.988E-11	6.44	2.387E+01	0.67	4.784E+01	0.59	7.707E+01	0.54
3					6.665E-02	2.94	4.844E-01	2.89	1.916E+00	2.77
4					5.703E-09	2.77	2.862E-07	3.12	1.980E-05	3.11
5							1.968E-11	0.67	1.966E-11	1.20

Table 7. Error and convergence study for the Kou's cubic scheme using several values of Re.

We then compare the classical Newton iteration (quadratic) against the preferred cubic variant (Kou's method) for both Reynolds numbers. In line with the scalar and small-system tests, Kou's method generally attains the steady-state tolerance in fewer outer iterations while reusing a single Jacobian per iteration, see 13. The corresponding convergence histories (Table 6 for Newton and Table 7 for Kou's method) corroborate the expected third-order behavior and the reduced iteration counts relative to the quadratic standard method.

4. Conclusion

We examined classical Newton's method (quadratic) and three cubic variants due to Kou, Homeier, and Weerakoon. Across scalar roots, nonlinear systems, and an IVP, the cubic schemes consistently met the tolerance in fewer iterations than Newton, with measured orders matching the theory ($\alpha \approx 3$ vs. $\alpha \approx 2$). Among them, Kou's method was especially attractive in practice because it reuses a single Jacobian per iteration while retaining third-order behavior. The steady Navier–Stokes tests confirmed an optimal convergence in a PDE setting.

Overall, high-order Newton-type methods offer tangible efficiency gains for nonlinear solves, particularly when repeated implicit steps are required. Future directions include adaptive initialization strategies and broader PDE applications (e.g., coupled multiphysics and biophysics [16]), where the iteration savings can translate directly to reduced compute time.

References

[1] Kou J, Li Y and Wang X 2006 *Appl. Math. Comput.* **181** 1106–11. DOI: 10.1016/j.amc.2006.01.076

[2] Homeier H 2003 *J. Comput. Appl. Math.* **157** 227–30. DOI: 10.1016/S0377-0427(03)00391-1

[3] Weerakoon S and Fernando T 2000 *Appl. Math. Lett.* **13** 87–93. DOI: 10.1016/S0893-9659(00)00100-2

[4] Suparatulatorn R and Suantai S 2021 *Results Nonlinear Anal.* **4** 159–68. DOI: 10.53006/rna.950067

[5] Tuan N H and Mohammadi H S 2020 *Chaos Solitons Fractals* **140** 110107. DOI: 10.1016/j.chaos.2020.110107

[6] Tuan N H and Zhou Y 2020 *J. Comput. Appl. Math.* **375** 112811. DOI: 10.1016/j.cam.2020.112811

[7] Cordero A, Jordan C, Sanabria-Codesal E and Torregrosa J R 2018 *J. Comput. Appl. Math.* **330** 748–58. DOI: 10.1016/j.cam.2017.02.032

[8] Cordero A, Gomez E and Torregrosa J R 2017 *Complexity* **2017** 3708461. DOI: 10.1155/2017/6457532

[9] Laadhari A and Szekely G 2017 *Int. J. Numer. Methods Eng.* **111** 1047–74. DOI: 10.1002/nme.5493

[10] Laadhari A 2017 *Appl. Math. Lett.* **69** 138–45. DOI: 10.1016/j.aml.2017.01.012

[11] Laadhari A, Saramito P, Misbah C and Szekely G 2017 *J. Comput. Phys.* **343** 271–99. DOI: 10.1016/j.jcp.2017.04.019

[12] Laadhari A 2018 *Appl. Math. Lett.* **81** 35–43. DOI: 10.1016/j.aml.2018.01.001

[13] Laadhari A and Temimi H 2025 *Appl. Math. Comput.* **486** 129058. DOI: 10.1016/j.amc.2024.129058

[14] Laadhari A 2018 *Appl. Math. Comput.* **333** 376–400. DOI: 10.1016/j.amc.2018.03.074

[15] Laadhari A 2025 *Mathematical Modelling and Numerical Simulation with Applications* **5**(3) 644–680. DOI: 10.53391/2791-8564.1007

[16] Laadhari A, Barral Y and Szekely G 2023 *Math. Methods Appl. Sci.* **46**(8) 8855–8876. DOI: 10.1002/mma.9021