
CS/MATH 3414 — Numerical Methods

Lecture 7 Solving Systems of Non-Linear Equations

Virginia Tech Department of Computer Science, Fall 2026

Instructor: Sumeet Khatri

Date: October 1, 2026

1	The problem: several equations, several unknowns	2
1.1	Real-world motivation: robots and electrical networks	2
1.2	Writing the general system	4
1.3	What changes when the equations are nonlinear?	4
2	Non-linear systems and Newton's method	5
2.1	From tangent lines to tangent linear systems	5
2.2	A polynomial example: a circle and a parabola	6
2.3	A non-polynomial example	8
3	Why Newton's method converges near a simple root	9
3.1	Two estimates behind the convergence proof	9
3.2	A local quadratic convergence theorem	10
3.3	Residual size versus error size	12
4	More worked examples of Newton's method	13
5	Gröbner bases: illustrating the idea	19
5.1	A very simple example	19
5.2	The basic operation: multiply, subtract, simplify	20
5.3	No roots, nonreal roots, and infinitely many roots	29
5.4	A missing elimination equation does not allow every value	30

Learning Objectives

- Carry out Newton iterations and check the original equations.
- Illustrate the idea of a Gröbner basis through a simple polynomial system.
- Use elimination to find polynomial solutions, distinguishing real and complex roots.

From linear systems and regression to nonlinear equations

Earlier lectures solved $A\mathbf{x} = \mathbf{b}$ and minimized $\|A\mathbf{x} - \mathbf{b}\|_2$. Here the equations themselves can contain products, powers, exponentials, or other nonlinear functions of the unknowns. Newton's method repeatedly solves a *linearized* system. For polynomial equations, exact algebra can sometimes replace the original equations by simpler ones with the same solutions. We use this observation to illustrate the idea behind Gröbner bases. The two viewpoints address finding one nearby solution and understanding a polynomial system's complete solution set.

Primary References

- [1] T. A. Driscoll and R. J. Braun, *Fundamentals of numerical computation, Newton for nonlinear systems*, Online edition, Section 4.5: the Jacobian, linear models, and Newton iteration for systems, <https://fncbook.com/newtonsys/> (visited on 09/29/2026).
- [2] C. T. Kelley, *Iterative methods for linear and nonlinear equations*, Frontiers in Applied Mathematics 16, Chapters 5 and 8: local Newton convergence and globalization by line searches (Society for Industrial and Applied Mathematics, Philadelphia, 1995).
- [3] J. E. Dennis Jr. and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Classics in Applied Mathematics 16, Reprint of the 1983 edition. Chapters 5–7: Newton’s method, global modifications, stopping, and scaling (Society for Industrial and Applied Mathematics, Philadelphia, 1996).
- [4] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., Springer Series in Operations Research and Financial Engineering, Chapters 3 and 11: line searches and nonlinear equations (Springer, New York, 2006).
- [5] D. A. Cox, J. Little, and D. O’Shea, *Ideals, varieties, and algorithms, An introduction to computational algebraic geometry and commutative algebra*, 4th ed., Undergraduate Texts in Mathematics, Chapters 2 and 3: Gröbner bases, Buchberger’s algorithm, and elimination theory (Springer, Cham, 2015).
- [6] B. Buchberger, “Bruno Buchberger’s PhD Thesis 1965: an algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal”, *Journal of Symbolic Computation* **41**, English translation by Michael P. Abramson of the 1965 doctoral thesis introducing the algorithmic theory of Gröbner bases, 475 (2006).
- [7] K. M. Lynch and F. C. Park, *Modern robotics, Mechanics, planning, and control*, Section 6.2: Numerical inverse kinematics (Cambridge University Press, 2017).
- [8] N. Horigome, A. Terui, and M. Mikawa, “A design and an implementation of an inverse kinematics computation in robotics using Gröbner bases”, in *Mathematical software—icms 2020*, Vol. 12097, Lecture Notes in Computer Science (2020), pp. 3–13.
- [9] R. D. Zimmerman and C. E. Murillo-Sánchez, *MATPOWER user’s manual*, version 8.1, Section 4.1: AC power flow, Power Systems Engineering Research Center (July 12, 2025).

1 The problem: several equations, several unknowns

Where should a robot place its elbow to put its hand at a prescribed target? What voltages will satisfy the power balances in an electrical network? These are practical reasons to solve several nonlinear equations at once. We will follow a simple robot-positioning problem from its physical model to a numerical Newton solution and then to an exact description of both possible configurations.

1.1 Real-world motivation: robots and electrical networks

Robot positioning. In *inverse kinematics*, a robot’s desired hand position is known, and we seek a configuration that reaches it. The geometric constraints are nonlinear. Newton’s method and Jacobian-based variants are used to compute configurations numerically [7, Section 6.2]. Gröbner-basis methods have also been used to solve polynomial formulations of inverse kinematics; for example, Horigome, Terui, and Mikawa

implemented such a calculation for a robot manipulator with three degrees of freedom [8]. Our two-link model below is a simplified teaching example of the same kind of geometric question.

Electrical power networks. Given specified loads and generation, an AC power-flow calculation determines network voltage magnitudes and phase angles from nonlinear power-balance equations. The MATPOWER software includes a Newton solver that forms and factorizes a Jacobian at each iteration [9, Section 4.1]. Here, too, a physical operating condition becomes a system $F(\mathbf{x}) = \mathbf{0}$.

Example 1 (A robot arm that must reach a target). An ideal planar robot has two rigid links, each of length 2 m. Its base is fixed at $O = (0, 0)$, and its hand must reach $T = (2, 2)$. We use numerical coordinates measured in metres. The unknown elbow position is $E = (x, y)$. The first link joins O to E , and the second joins E to T . Both links must have the prescribed length, so

$$\underbrace{x^2 + y^2}_{\text{squared distance from } O} = 4, \quad \underbrace{(x - 2)^2 + (y - 2)^2}_{\text{squared distance from } T} = 4. \quad (1.1)$$

These are two circle equations. A valid elbow must lie on both circles. The unknowns describe a physical configuration, rather than an arbitrary intersection of curves.

For a first guess, take $E_0 = (3/2, 1/2)$. The two resulting link lengths are

$$\begin{aligned} \ell_{1,0} &= \sqrt{(3/2)^2 + (1/2)^2} = \sqrt{5/2} \approx 1.581 \text{ m}, \\ \ell_{2,0} &= \sqrt{(3/2 - 2)^2 + (1/2 - 2)^2} = \sqrt{5/2} \approx 1.581 \text{ m}. \end{aligned} \quad (1.2)$$

Both are too short. How should we correct the elbow position, and how many configurations reach the target? Example 7 will compute Newton corrections from this guess. Example 13 will use exact cancellation to find every configuration of this ideal model.

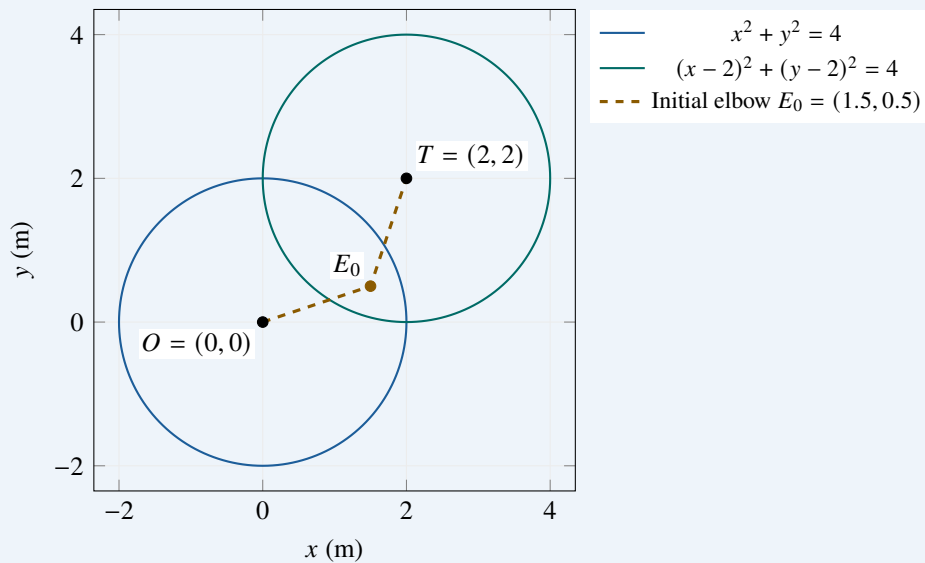


FIGURE 1: The blue circle enforces the first link length; the teal circle enforces the second. The dashed segments through the initial elbow guess are shorter than 2 m. A valid configuration requires an elbow at an intersection of the two circles.

Real robots also impose joint limits and collision constraints. The equations here isolate the geometric length constraints so that the numerical and algebraic calculations can be followed in full.

1.2 Writing the general system

For m equations in n unknowns, collect the residuals into a vector-valued function:

$$F(\mathbf{x}) = \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{pmatrix}, \quad F(\mathbf{x}) = \mathbf{0}. \quad (1.3)$$

A *solution* or *root* makes every component zero at the same point. For example, the two equations $x^2 + y^2 = 3$ and $x = y^2 - 1$ define a circle and a sideways parabola. Solving the system means finding their intersections, not solving each equation with unrelated values of x and y .

Example 2 (A running example: a circle and a parabola). We will study

$$f_1(x, y) = x^2 + y^2 - 3 = 0, \quad f_2(x, y) = x - y^2 + 1 = 0. \quad (1.4)$$

The points $(1, \sqrt{2})$ and $(1, -\sqrt{2})$ are solutions: $1 + 2 - 3 = 0$ and $1 - 2 + 1 = 0$. How would we find them without being told the answers? How could we know whether other solutions exist? Newton iteration addresses the first question near an initial guess; polynomial elimination will address both.

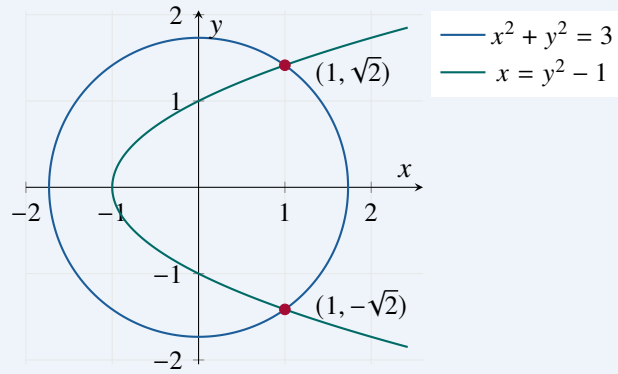


FIGURE 2: The real zero curves intersect at $(1, \sqrt{2})$ and $(1, -\sqrt{2})$. Polynomial elimination later proves that these are the only real intersections.

The drawing suggests two real intersections. A drawing alone does not prove that the list is complete, and it cannot show nonreal solutions.

1.3 What changes when the equations are nonlinear?

A square linear system with a nonsingular matrix has exactly one solution. A square nonlinear system can have no solutions, finitely many solutions, or infinitely many solutions. For instance, the systems

$$(x^2 + 1, y) = \mathbf{0}, \quad (x^2 - 1, y) = \mathbf{0}, \quad (x^2 + y^2 - 1, 2x^2 + 2y^2 - 2) = \mathbf{0} \quad (1.5)$$

have, respectively, zero, two, and infinitely many real solutions. The last system repeats one constraint, so the number of equations alone tells us little.

The numerical part of this note focuses on *square* systems, $m = n$, with a differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ on an open domain. For $m \neq n$, one can instead solve a linearized least-squares problem, but existence, uniqueness, and convergence require different hypotheses. The algebraic part allows any finite number of polynomial equations. Coefficients will usually lie in $\mathbb{Q}$, while solutions may be sought in $\mathbb{R}$ or $\mathbb{C}$.

2 Non-linear systems and Newton's method

We now turn the intersection problem into a sequence of linear solves. The derivative of a scalar function becomes a matrix of partial derivatives, and a single correction must satisfy the linear approximations to all the equations simultaneously.

Definition 1 (System function, root, and Jacobian). Let $D \subseteq \mathbb{R}^n$ be open. A *system function* is a map

$$F : D \longrightarrow \mathbb{R}^n, \quad F(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix}. \quad (2.1)$$

A *root* is a vector $\mathbf{x}_* \in D$ for which $F(\mathbf{x}_*) = \mathbf{0}$. If F is continuously differentiable, its *Jacobian matrix* is

$$J(\mathbf{x}) = DF(\mathbf{x}) = \left[\frac{\partial f_i}{\partial x_j}(\mathbf{x}) \right]_{i,j=1}^n. \quad (2.2)$$

A root is *simple* (or *regular*) when $J(\mathbf{x}_*)$ is nonsingular; otherwise it is *singular*.

Throughout the numerical discussion, $\|\mathbf{x}\|$ denotes the Euclidean norm, and $\|A\| = \max_{\|\mathbf{u}\|=1} \|A\mathbf{u}\|$ is its subordinate matrix norm. Thus $\|A\mathbf{u}\| \leq \|A\|\|\mathbf{u}\|$ and $\|AB\| \leq \|A\|\|B\|$. We write $\overline{B}(\mathbf{x}_*, r) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_*\| \leq r\}$. The analysis is for *square* systems: the number of equations equals the number of unknowns. Overdetermined systems usually require a least-squares formulation; the Newton equation below cannot simply be assumed to have a unique solution for a rectangular Jacobian.

2.1 From tangent lines to tangent linear systems

At a current approximation $\mathbf{x}_k$, differentiability gives the local model

$$F(\mathbf{x}_k + \mathbf{s}) = F(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{s} + o(\|\mathbf{s}\|). \quad (2.3)$$

Newton's method chooses a correction that makes the *linear model* zero:

$$J(\mathbf{x}_k)\mathbf{s}_k = -F(\mathbf{x}_k), \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k. \quad (2.4)$$

Every row is the tangent approximation to one equation, and the correction must satisfy all of these linear equations at once. This is the direct extension of the scalar formula $x_{k+1} = x_k - f(x_k)/f'(x_k)$; see [1].

Key Idea

A Newton iteration for a non-linear system consists of evaluating a residual, forming a Jacobian, and *solving a linear system*. The notation $\mathbf{x}_{k+1} = \mathbf{x}_k - J(\mathbf{x}_k)^{-1}F(\mathbf{x}_k)$ is useful for proofs. An implementation should solve (2.4), for example using LU factorization with pivoting, instead of explicitly forming an inverse.

Proposition 2 (Affine equations are solved in one step). Suppose $F(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$, where $A \in \mathbb{R}^{n \times n}$ is nonsingular. From every initial vector $\mathbf{x}_0$, one exact Newton step gives the unique root $A^{-1}\mathbf{b}$.

Proof. Here $J(\mathbf{x}) = A$ everywhere. The correction equation is $As_0 = \mathbf{b} - A\mathbf{x}_0$, so $A(\mathbf{x}_0 + s_0) = \mathbf{b}$. Nonsingularity gives the conclusion. There is no Taylor remainder: the local model is the function itself. ■

Algorithm: Basic Newton iteration

Given F, J , an initial vector $\mathbf{x}_0$, a residual tolerance, and a maximum number of iterations:

1. Evaluate $F(\mathbf{x}_k)$. If its appropriately scaled norm is below the chosen tolerance, return $\mathbf{x}_k$ and the residual.
2. Form $J(\mathbf{x}_k)$ and solve $J(\mathbf{x}_k)s_k = -F(\mathbf{x}_k)$. If a reliable correction cannot be computed, report the linear-solve failure.
3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + s_k$, and repeat. If the iteration limit is reached, return a failure status along with the last residual and step size.

Testing the residual *before* solving is useful: a root may have a singular Jacobian, but no further step is needed once a satisfactory root has been found.

2.2 A polynomial example: a circle and a parabola

Example 3 (Two equations, one correction). Consider

$$F(x, y) = \begin{pmatrix} x^2 + y^2 - 3 \\ x - y^2 + 1 \end{pmatrix}, \quad J(x, y) = \begin{pmatrix} 2x & 2y \\ 1 & -2y \end{pmatrix}. \quad (2.5)$$

The first equation is the circle of radius $\sqrt{3}$; the second is the parabola $x = y^2 - 1$. We first seek the root in the first quadrant numerically.

Step 1: evaluate at the initial point. Take $\mathbf{x}_0 = (2, 1)^\top$. Then

$$F(2, 1) = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad J(2, 1) = \begin{pmatrix} 4 & 2 \\ 1 & -2 \end{pmatrix}. \quad (2.6)$$

Writing $s_0 = (u, v)^\top$, the correction equation is

$$4u + 2v = -2, \quad u - 2v = -2. \quad (2.7)$$

Adding the equations gives $5u = -4$, so $u = -4/5$ and $v = 3/5$. Hence

$$\mathbf{x}_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \begin{pmatrix} -4/5 \\ 3/5 \end{pmatrix} = \begin{pmatrix} 6/5 \\ 8/5 \end{pmatrix}. \quad (2.8)$$

This point solves both tangent equations, but the original residual is $F(6/5, 8/5) = (1, -9/25)^\top$, which is not zero.

Step 2: rebuild the model. The next correction uses the *new* Jacobian:

$$\begin{pmatrix} 12/5 & 16/5 \\ 1 & -16/5 \end{pmatrix} \mathbf{s}_1 = \begin{pmatrix} -1 \\ 9/25 \end{pmatrix}. \quad (2.9)$$

This gives $\mathbf{s}_1 \approx (-0.1882352941, -0.1713235294)^\top$ and $\mathbf{x}_2 \approx (1.0117647059, 1.4286764706)^\top$.

Step 3: repeat and check both equations. The exact positive root is $\mathbf{x}_* = (1, \sqrt{2})^\top$. The following values were computed using unrounded iterates; displayed coordinates are rounded.

k	x_k	y_k	$\ F(\mathbf{x}_k)\ $	$\ \mathbf{x}_k - \mathbf{x}_*\ $
0	2.0000000000	1.0000000000	2.8284	1.0824
1	1.2000000000	1.6000000000	1.0628	$2.7298 \cdot 10^{-1}$
2	1.0117647059	1.4286764706	$7.1123 \cdot 10^{-2}$	$1.8644 \cdot 10^{-2}$
3	1.0000457771	1.4143027893	$4.0122 \cdot 10^{-4}$	$1.0028 \cdot 10^{-4}$
4	1.0000000007	1.4142135654	$1.2826 \cdot 10^{-8}$	$3.1402 \cdot 10^{-9}$
5	1.0000000000	1.4142135624	$\approx 6.3 \cdot 10^{-16}$	roundoff level

The rapid final improvement is characteristic of quadratic convergence. The last line reflects finite-precision arithmetic; it is not an exact certificate that the computed vector equals the irrational root.

The determinant is $\det J(x, y) = -2y(2x + 1)$. Thus a Newton step may fail on $y = 0$ or $x = -1/2$, even though the positive root has a nonsingular Jacobian. Starting values matter. Later, algebraic elimination will reveal *all* complex solutions of this same polynomial system, whereas this Newton run has located one real solution.

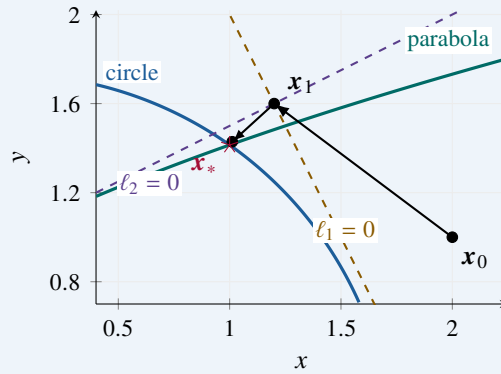


FIGURE 3: The first Newton correction moves from $\mathbf{x}_0 = (2, 1)^\top$ to the intersection $\mathbf{x}_1 = (1.2, 1.6)^\top$ of the zero sets of the two linearized equations (dashed). The next correction approaches the common zero $\mathbf{x}_* = (1, \sqrt{2})^\top$ of the original circle and parabola.

2.3 A non-polynomial example

Example 4 (Set up an exponential equation and a circle). Consider

$$F(x, y) = \begin{pmatrix} e^x + y - 2 \\ x^2 + y^2 - 1 \end{pmatrix}, \quad J(x, y) = \begin{pmatrix} e^x & 1 \\ 2x & 2y \end{pmatrix}. \quad (2.10)$$

Unlike the preceding example, one equation is not polynomial. The Newton construction still applies because the functions are differentiable. For $\mathbf{x}_0 = (0.9, -0.4)^\top$, we obtain

$$F(\mathbf{x}_0) \approx \begin{pmatrix} 0.0596031112 \\ -0.0300000000 \end{pmatrix}, \quad J(\mathbf{x}_0) \approx \begin{pmatrix} 2.4596031112 & 1 \\ 1.8 & -0.8 \end{pmatrix}. \quad (2.11)$$

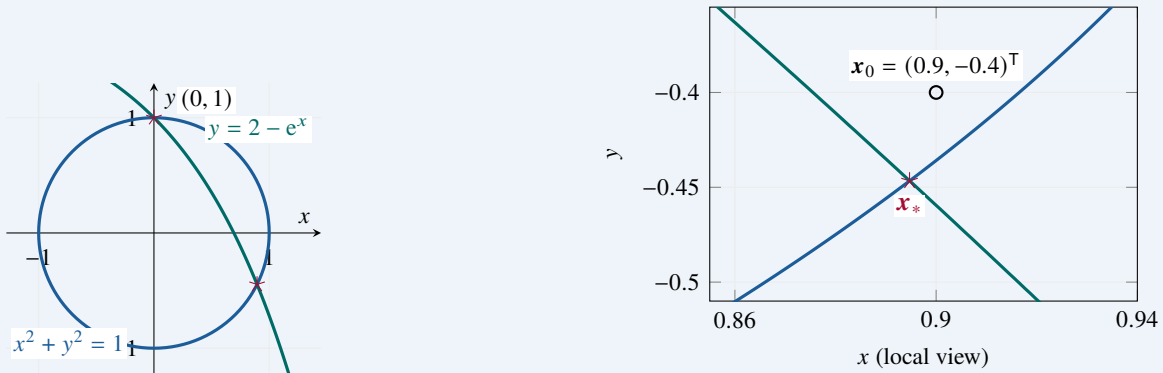


FIGURE 4: The exponential curve and the unit circle have two real intersections, $(0, 1)$ and approximately $(0.894715, -0.446638)$ (left, with equal coordinate scales). The local view on the right separates the initial guess $\mathbf{x}_0 = (0.9, -0.4)^\top$ from the nearby second root $\mathbf{x}_*$.

Example 5 (Solve the exponential-system correction and iterate). Using the preceding residual and Jacobian, the second row of the correction equation is $1.8u - 0.8v = 0.03$, so $v = 2.25u - 0.0375$. Substitution into the first row gives

$$(2.4596031112 + 2.25)u = -0.0596031112 + 0.0375. \quad (2.12)$$

Consequently $u \approx -0.0046932004$, $v \approx -0.0480597009$, and $\mathbf{x}_1 \approx (0.8953067996, -0.4480597009)^\top$.

k	x_k	y_k	$\ F(\mathbf{x}_k)\ $
0	0.9000000000	-0.4000000000	$6.6727 \cdot 10^{-2}$
1	0.8953067996	-0.4480597009	$2.3319 \cdot 10^{-3}$
2	0.8947154930	-0.4466391766	$2.4059 \cdot 10^{-6}$
3	0.8947148013	-0.4466379119	$2.1588 \cdot 10^{-12}$
4	0.8947148013	-0.4466379119	roundoff level

There is also the exact root $(0, 1)^T$. The iteration table neither lists all roots nor establishes uniqueness. It demonstrates successful local refinement near one root. A polynomial Gröbner basis method does not directly apply to this system because e^x is not a polynomial.

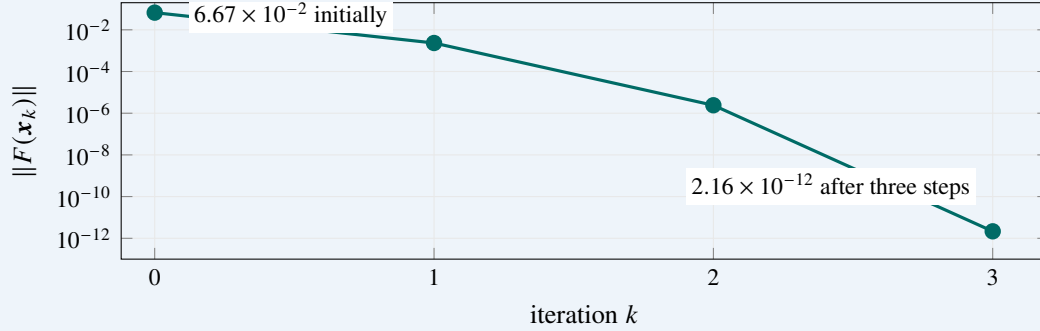


FIGURE 5: Residual norms for the exponential system on a logarithmic vertical scale. The increasingly steep decrease shows the rapid local improvement. Values are evaluated from unrounded iterates; the final row at floating-point roundoff is omitted.

3 Why Newton's method converges near a simple root

The essential mechanism has two parts. The error in the linear model is quadratic in the distance from the root, and solving with the nearby Jacobian magnifies this error by only a bounded factor. We prove both claims and then combine them. This is a local theorem: the existence of a simple root is an assumption, and the initial approximation must be sufficiently close to it. For broader treatments of non-linear equation solving, see [2, 3].

3.1 Two estimates behind the convergence proof

Lemma 3 (Taylor remainder with a Lipschitz Jacobian). Suppose F is continuously differentiable on an open set containing a convex set C , and there is $L \geq 0$ such that

$$\|J(\mathbf{x}) - J(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\| \quad \text{for all } \mathbf{x}, \mathbf{y} \in C. \quad (3.1)$$

Then, for every $\mathbf{x}, \mathbf{y} \in C$,

$$\|F(\mathbf{y}) - F(\mathbf{x}) - J(\mathbf{x})(\mathbf{y} - \mathbf{x})\| \leq \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|^2. \quad (3.2)$$

Proof. Set $\mathbf{h} = \mathbf{y} - \mathbf{x}$. The segment $\mathbf{x} + t\mathbf{h}$, $0 \leq t \leq 1$, is in C . Applying the fundamental theorem of calculus to each component gives

$$F(\mathbf{y}) - F(\mathbf{x}) = \int_0^1 J(\mathbf{x} + t\mathbf{h})\mathbf{h} \, dt. \quad (3.3)$$

Subtracting $J(\mathbf{x})\mathbf{h}$ and taking norms yields

$$\begin{aligned} \|F(\mathbf{y}) - F(\mathbf{x}) - J(\mathbf{x})\mathbf{h}\| &\leq \int_0^1 \|J(\mathbf{x} + t\mathbf{h}) - J(\mathbf{x})\| \|\mathbf{h}\| \, dt \\ &\leq \int_0^1 Lt \|\mathbf{h}\|^2 \, dt = \frac{L}{2} \|\mathbf{h}\|^2. \end{aligned} \quad (3.4)$$

■

Lemma 4 (Inverse perturbation bound). Let A be nonsingular and suppose E is a matrix such that $\theta = \|A^{-1}E\| < 1$. Then $A + E$ is nonsingular and

$$\|(A + E)^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \theta}. \quad (3.5)$$

Proof. Put $H = A^{-1}E$. The matrix series $S = \sum_{j=0}^{\infty} (-H)^j$ converges, since the norms of its terms are bounded by the summable geometric series $\sum_{j=0}^{\infty} \theta^j$. Its partial sums satisfy

$$(\mathbb{1} + H) \sum_{j=0}^m (-H)^j = \mathbb{1} - (-H)^{m+1} = \left(\sum_{j=0}^m (-H)^j \right) (\mathbb{1} + H). \quad (3.6)$$

Taking the limit shows $(\mathbb{1} + H)S = S(\mathbb{1} + H) = \mathbb{1}$, and $\|S\| \leq (1 - \theta)^{-1}$. Since $A + E = A(\mathbb{1} + H)$,

$$(A + E)^{-1} = SA^{-1}, \quad \|(A + E)^{-1}\| \leq \|S\| \|A^{-1}\|, \quad (3.7)$$

which proves the estimate. ■

3.2 A local quadratic convergence theorem

Definition 5 (Quadratic convergence). A sequence $\mathbf{x}_k$ converges *at least quadratically* to $\mathbf{x}_*$ if it converges to $\mathbf{x}_*$ and there are $C > 0$ and k_0 such that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq C \|\mathbf{x}_k - \mathbf{x}_*\|^2 \quad \text{for every } k \geq k_0. \quad (3.8)$$

The constant and the neighborhood matter: squaring an error is useful only once that error is sufficiently small relative to C .

Theorem 1 (Local quadratic convergence and local uniqueness). Let F be continuously differentiable on an open set containing $\overline{B}(\mathbf{x}_*, R)$, and suppose

- (i) $F(\mathbf{x}_*) = \mathbf{0}$ and $A = J(\mathbf{x}_*)$ is nonsingular;
- (ii) $\|J(\mathbf{x}) - J(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$ on this ball, with $L > 0$.

Let $\beta = \|A^{-1}\|$, and choose

$$0 < \rho \leq \min \left\{ R, \frac{1}{2\beta L} \right\}. \quad (3.9)$$

For every $\mathbf{x}_0 \in \overline{B}(\mathbf{x}_*, \rho)$, the exact Newton iteration is well defined, stays in that ball, and converges to $\mathbf{x}_*$. More precisely,

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq \beta L \|\mathbf{x}_k - \mathbf{x}_*\|^2 \leq \frac{1}{2} \|\mathbf{x}_k - \mathbf{x}_*\|. \quad (3.10)$$

The root $\mathbf{x}_*$ is the only root in $\overline{B}(\mathbf{x}_*, \rho)$.

Proof. 1. Nearby Jacobians remain invertible. For $\mathbf{x} \in \bar{B}(\mathbf{x}_*, \rho)$, write $J(\mathbf{x}) = A + E$, where $E = J(\mathbf{x}) - A$. The assumptions give

$$\|A^{-1}E\| \leq \beta L \|\mathbf{x} - \mathbf{x}_*\| \leq \beta L \rho \leq \frac{1}{2}. \quad (3.11)$$

Lemma 4 therefore applies and implies

$$\|J(\mathbf{x})^{-1}\| \leq 2\beta. \quad (3.12)$$

Thus a unique Newton correction exists at every point in this ball.

2. One step squares the error. Let $\mathbf{e} = \mathbf{x} - \mathbf{x}_*$, and set $\mathbf{x}^+ = \mathbf{x} - J(\mathbf{x})^{-1}F(\mathbf{x})$. Apply Lemma 3 with base point $\mathbf{x}$ and endpoint $\mathbf{x}_*$:

$$\mathbf{0} = F(\mathbf{x}) - J(\mathbf{x})\mathbf{e} + \mathbf{r}, \quad \|\mathbf{r}\| \leq \frac{L}{2}\|\mathbf{e}\|^2. \quad (3.13)$$

Consequently

$$\mathbf{x}^+ - \mathbf{x}_* = J(\mathbf{x})^{-1}(J(\mathbf{x})\mathbf{e} - F(\mathbf{x})) = J(\mathbf{x})^{-1}\mathbf{r}. \quad (3.14)$$

Using (3.12) gives $\|\mathbf{x}^+ - \mathbf{x}_*\| \leq \beta L \|\mathbf{e}\|^2 \leq \|\mathbf{e}\|/2$.

3. The iteration stays in the ball and converges. The last estimate shows that one Newton step maps the ball into itself. Induction therefore proves that every step is defined and that

$$\|\mathbf{x}_k - \mathbf{x}_*\| \leq 2^{-k} \|\mathbf{x}_0 - \mathbf{x}_*\| \longrightarrow 0. \quad (3.15)$$

The stronger first inequality in (3.10) proves quadratic convergence. In fact, with $C = \beta L$,

$$C\|\mathbf{x}_k - \mathbf{x}_*\| \leq (C\|\mathbf{x}_0 - \mathbf{x}_*\|)^{2^k}. \quad (3.16)$$

This bound follows by repeatedly squaring the one-step inequality.

4. No other root is in this ball. Suppose $F(\mathbf{y}) = \mathbf{0}$ and $\|\mathbf{y} - \mathbf{x}_*\| \leq \rho$. Taylor expansion about $\mathbf{x}_*$ gives $\mathbf{0} = A(\mathbf{y} - \mathbf{x}_*) + \mathbf{r}$, with $\|\mathbf{r}\| \leq (L/2)\|\mathbf{y} - \mathbf{x}_*\|^2$. Multiplying by A^{-1} yields

$$\|\mathbf{y} - \mathbf{x}_*\| \leq \frac{\beta L}{2} \|\mathbf{y} - \mathbf{x}_*\|^2 \leq \frac{1}{4} \|\mathbf{y} - \mathbf{x}_*\|. \quad (3.17)$$

This forces $\mathbf{y} = \mathbf{x}_*$. If an iterate already equals the root, the correction is zero and all following iterates equal the root as well. ■

Remark 2. The case $L = 0$ means that the Jacobian is constant on the ball; the local problem is affine and the same argument gives an exact step to the root. Any known upper bound $B \geq \|A^{-1}\|$ can replace β in the theorem, with the corresponding conservative radius $1/(2BL)$. The theorem is a sufficient local condition, not a description of the entire region of convergence and not a proof that a root exists for an arbitrary system.

Example 6 (A numerical convergence guarantee for the circle system). For (2.5) at $\mathbf{x}_* = (1, \sqrt{2})^\top$,

$$A = \begin{pmatrix} 2 & 2\sqrt{2} \\ 1 & -2\sqrt{2} \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 1/3 & 1/3 \\ 1/(6\sqrt{2}) & -1/(3\sqrt{2}) \end{pmatrix}. \quad (3.18)$$

The Frobenius norm $\|M\|_F = (\sum_{i,j} |m_{ij}|^2)^{1/2}$ bounds the subordinate Euclidean norm from above. Thus

$$\beta = \|A^{-1}\| \leq \|A^{-1}\|_F = \sqrt{\frac{7}{24}} =: B. \quad (3.19)$$

For $\mathbf{x} = (x, y)^\top$ and $\zeta = (u, v)^\top$,

$$J(\mathbf{x}) - J(\zeta) = \begin{pmatrix} 2(x-u) & 2(y-v) \\ 0 & -2(y-v) \end{pmatrix}. \quad (3.20)$$

Its Frobenius norm squared is $4(x-u)^2 + 8(y-v)^2 \leq 8\|\mathbf{x} - \zeta\|^2$, so $L = 2\sqrt{2}$ is valid on all of $\mathbb{R}^2$. We may take

$$\rho = \frac{1}{2BL} = \frac{1}{2} \sqrt{\frac{3}{7}} \approx 0.32733, \quad BL = \sqrt{\frac{7}{3}} \approx 1.52753. \quad (3.21)$$

The initial point $(2, 1)^\top$ is outside this guaranteed ball, but the first iterate $(1.2, 1.6)^\top$ is inside it because its error is approximately 0.27298. The theorem therefore certifies all subsequent exact iterates and gives the explicit bound

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq \sqrt{\frac{7}{3}} \|\mathbf{x}_k - \mathbf{x}_*\|^2, \quad k \geq 1. \quad (3.22)$$

A sufficient convergence neighborhood can be smaller than the set of starting points that actually succeed.

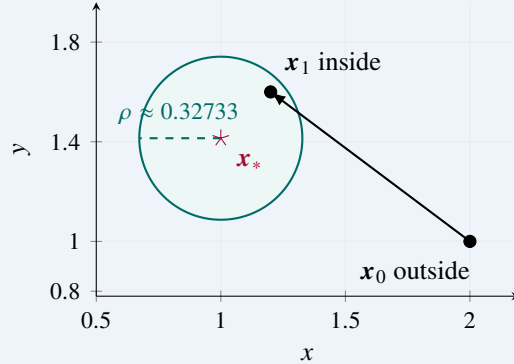


FIGURE 6: The guaranteed convergence ball has radius $\rho = \frac{1}{2}\sqrt{3/7} \approx 0.32733$ about $(1, \sqrt{2})^\top$. The first iterate has error $0.27298 < \rho$, although the initial error is $1.08239 > \rho$. The axes use equal coordinate scales, so the displayed ball represents the Euclidean norm.

3.3 Residual size versus error size

A residual is computable even when the root is unknown; the error usually is not. The following estimate explains when a small residual is informative.

Theorem 3 (Local residual–error comparison). Under the assumptions of Theorem 1, let $\mathbf{x} \in \bar{B}(\mathbf{x}_*, \rho)$ and $\mathbf{e} = \mathbf{x} - \mathbf{x}_*$. Then

$$\left(\frac{1}{\beta} - \frac{L\rho}{2}\right) \|\mathbf{e}\| \leq \|F(\mathbf{x})\| \leq \left(\|A\| + \frac{L\rho}{2}\right) \|\mathbf{e}\|. \quad (3.23)$$

In particular,

$$\|\mathbf{x} - \mathbf{x}_*\| \leq \frac{4\beta}{3} \|F(\mathbf{x})\|. \quad (3.24)$$

Proof. Taylor expansion about the root gives $F(\mathbf{x}) = A\mathbf{e} + \mathbf{r}$ with $\|\mathbf{r}\| \leq (L/2)\|\mathbf{e}\|^2$. Since $\mathbf{e} = A^{-1}A\mathbf{e}$, we have $\|A\mathbf{e}\| \geq \|\mathbf{e}\|/\beta$. The reverse triangle inequality therefore gives

$$\|F(\mathbf{x})\| \geq \frac{\|\mathbf{e}\|}{\beta} - \frac{L}{2}\|\mathbf{e}\|^2 \geq \left(\frac{1}{\beta} - \frac{L\rho}{2}\right) \|\mathbf{e}\|. \quad (3.25)$$

The ordinary triangle inequality gives the upper bound. Finally, $\beta L\rho \leq 1/2$ implies $1/\beta - L\rho/2 \geq 3/(4\beta)$, proving the last claim. ■

The factor $\beta = \|J(\mathbf{x}_*)^{-1}\|$ measures sensitivity of the root to small residual perturbations in these units. The comparison is *local*: a residual tolerance alone does not establish that an arbitrary point lies in the neighborhood where the theorem applies.

4 More worked examples of Newton's method

These examples emphasize three useful habits: solve the correction equations together, exploit any exact linear constraints, and substitute the computed point into every original equation. All residual norms below are Euclidean norms evaluated from unrounded iterates; displayed coordinates are rounded.

Example 7 (Newton's method positions the robot elbow). Return to Example 1: two links of length 2 m connect the base $O = (0, 0)$ to the target $T = (2, 2)$ through an unknown elbow. With coordinates measured in metres, the system function and Jacobian are

$$F(x, y) = \begin{pmatrix} x^2 + y^2 - 4 \\ (x - 2)^2 + (y - 2)^2 - 4 \end{pmatrix}, \quad J(x, y) = \begin{pmatrix} 2x & 2y \\ 2(x - 2) & 2(y - 2) \end{pmatrix}. \quad (4.1)$$

Each component of F measures a squared-length discrepancy. Thus the physical units of $\|F\|$ are m^2 ; this norm is not itself the distance from the trial elbow to a correct elbow.

Step 1: compute the first correction. At the initial guess $\mathbf{x}_0 = (3/2, 1/2)^\top$ from the opening,

$$F(\mathbf{x}_0) = \begin{pmatrix} -3/2 \\ -3/2 \end{pmatrix}, \quad J(\mathbf{x}_0) = \begin{pmatrix} 3 & 1 \\ -1 & -3 \end{pmatrix}. \quad (4.2)$$

Writing the correction as $(u, v)^\top$, solve

$$3u + v = \frac{3}{2}, \quad -u - 3v = \frac{3}{2}. \quad (4.3)$$

Subtracting the second equation from the first gives $4u + 4v = 0$, so $v = -u$. The first equation then gives $2u = 3/2$. Hence

$$(u, v) = \left(\frac{3}{4}, -\frac{3}{4}\right), \quad \mathbf{x}_1 = \begin{pmatrix} 9/4 \\ -1/4 \end{pmatrix} = \begin{pmatrix} 2.25 \\ -0.25 \end{pmatrix}. \quad (4.4)$$

Both squared link lengths at this point are $41/8 = 5.125$, so $F(\mathbf{x}_1) = (9/8, 9/8)^\top$. The correction has crossed the required elbow position; satisfying the linearized equations does not yet satisfy the original length constraints.

Step 2: rebuild the Jacobian and correct again. At the new point, the correction equations are

$$\begin{pmatrix} 9/2 & -1/2 \\ 1/2 & -9/2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -9/8 \\ -9/8 \end{pmatrix}. \quad (4.5)$$

Subtracting the second row from the first again gives $u + v = 0$. The first row then gives $5u = -9/8$. Thus

$$(u, v) = (-9/40, 9/40), \quad \mathbf{x}_2 = (81/40, -1/40)^\top = (2.025, -0.025)^\top. \quad (4.6)$$

Continuing in the same way gives the following values. Calculations used unrounded iterates and high precision; the displayed coordinates are rounded.

k	x_k (m)	y_k (m)	$\ F(\mathbf{x}_k)\ $ (m ²)
0	1.500000000	0.500000000	2.1213
1	2.250000000	-0.250000000	1.5910
2	2.025000000	-0.025000000	$1.4319 \cdot 10^{-1}$
3	2.000304878	-0.000304878	$1.7249 \cdot 10^{-3}$
4	2.000000046	$-4.6461 \cdot 10^{-8}$	$2.6282 \cdot 10^{-7}$
5	2.000000000	$-1.0793 \cdot 10^{-15}$	$6.1056 \cdot 10^{-15}$

Step 3: interpret and check the result. The iterates approach $(2, 0)$. This elbow gives the link vectors $(2, 0)$ and $(0, 2)$, each of length 2 m, and the hand reaches $(2, 2)$. The arm extends right and then bends upward. Interchanging the initial coordinates to $(0.5, 1.5)$ interchanges every iterate and produces the configuration with elbow $(0, 2)$: up, then right. Example 13 will establish that these are all the configurations allowed by the two length constraints.

The line followed by the iterates has an algebraic explanation. Subtracting the two Newton correction equations at a general point gives $u + v = 2 - x - y$. Hence every updated point satisfies $x^+ + y^+ = 2$ exactly. On this line, write $x = 1 + t$ and $y = 1 - t$; both components of F become $2(t^2 - 1)$. Newton's formula therefore reduces to

$$t^+ = \frac{1}{2} \left(t + \frac{1}{t} \right), \quad t \neq 0. \quad (4.7)$$

Here $t_0 = 1/2$, $t_1 = 5/4$, and $t_2 = 41/40$, followed by convergence to 1.

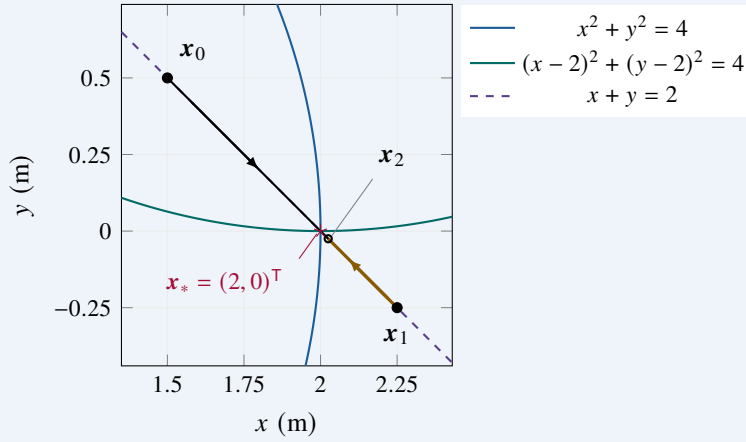


FIGURE 7: Newton corrections for the elbow coordinates. The first step crosses the root; the following steps refine it. The iterates remain on $x + y = 2$, and converge to the circle intersection $(2, 0)$.

Finally, $\det J = 8(y - x)$, which is -16 at $(2, 0)$ and 16 at $(0, 2)$. Both roots are simple. The midpoint $(1, 1)$ has a singular Jacobian: its correction equations demand both $2u + 2v = 2$ and $-2u - 2v = 2$, which is impossible. Thus a symmetric initial guess can be a poor choice. The iterates are trial coordinates used in a calculation; they are not instructions for a physical robot to move through those positions.

Example 8 (A hyperbola, a line, and two different limits). Consider

$$F(x, y) = \begin{pmatrix} xy - 1 \\ x + y - 3 \end{pmatrix}, \quad J(x, y) = \begin{pmatrix} y & x \\ 1 & 1 \end{pmatrix}. \quad (4.8)$$

At $\mathbf{x}_0 = (2, 1)^\top$, the residual is $(1, 0)^\top$. Writing the correction as $(u, v)^\top$, we solve

$$\begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \implies v = -1, \quad u = 1, \quad \mathbf{x}_1 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}. \quad (4.9)$$

The first step does not reduce the residual norm: $F(3, 0) = (-1, 0)^\top$ still has norm 1. It does, however, preserve the linear equation. Indeed, the second correction equation always gives $u + v = 3 - x - y$, so every updated point satisfies $x^+ + y^+ = 3$ exactly.

On this line, $xy = 1$ becomes $x^2 - 3x + 1 = 0$. For $x + y = 3$, the first correction equation reduces to $(3 - 2x)u = 1 - 3x + x^2$, giving

$$x^+ = x - \frac{x^2 - 3x + 1}{2x - 3}, \quad y^+ = 3 - x^+. \quad (4.10)$$

Thus the two-variable iteration agrees with scalar Newton iteration on the constraint line. The next step from $(3, 0)^\top$ gives $(8/3, 1/3)^\top$.

k	x_k	y_k	$\ F(\mathbf{x}_k)\ $
0	2.000000000	1.000000000	1
1	3.000000000	0.000000000	1
2	2.666666667	0.333333333	$1.1111 \cdot 10^{-1}$
3	2.619047619	0.380952381	$2.2676 \cdot 10^{-3}$
4	2.618034448	0.381965552	$1.0265 \cdot 10^{-6}$
5	2.618033989	0.381966011	$2.1061 \cdot 10^{-13}$

The two exact solutions are

$$\begin{pmatrix} (3 + \sqrt{5})/2 \\ (3 - \sqrt{5})/2 \end{pmatrix}, \quad \begin{pmatrix} (3 - \sqrt{5})/2 \\ (3 + \sqrt{5})/2 \end{pmatrix}. \quad (4.11)$$

Their coordinate sums are 3, and their products are $(9 - 5)/4 = 1$, checking both equations. The table approaches the first solution. Interchanging x and y interchanges every Newton iterate, so starting at $(1, 2)^\top$ approaches the second solution. Here $\det J = y - x$; the midpoint $(3/2, 3/2)^\top$ on the same line has a singular Jacobian and cannot be treated by this correction formula.

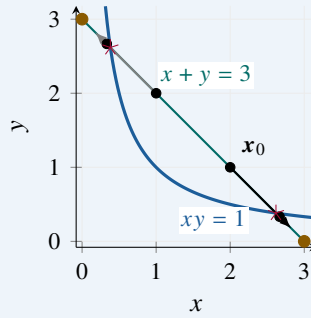


FIGURE 8: The two roots are marked by red stars. Arrows show the first corrections from $(2, 1)$ and its reflection $(1, 2)$; gold markers show their endpoints. Subsequent iterates remain on the line and approach different intersections.

Example 9 (A trigonometric system and a unique real root). Let

$$F(x, y) = \begin{pmatrix} \sin x + y - 1 \\ x + \cos y - 1 \end{pmatrix}, \quad J(x, y) = \begin{pmatrix} \cos x & 1 \\ 1 & -\sin y \end{pmatrix}. \quad (4.12)$$

Angles are measured in radians. At $\mathbf{x}_0 = (0.5, 0.5)^\top$, the Newton correction satisfies

$$\begin{pmatrix} 0.8775825619 & 1 \\ 1 & -0.4794255386 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \approx \begin{pmatrix} 0.0205744614 \\ -0.3775825619 \end{pmatrix}. \quad (4.13)$$

The first row gives $v \approx 0.0205744614 - 0.8775825619u$. Substituting into the second row yields $u \approx -0.2588227306$ and $v \approx 0.2477127764$, so

$$\mathbf{x}_1 \approx (0.2411772694, 0.7477127764)^\top. \quad (4.14)$$

Re-evaluating both the trigonometric functions and the Jacobian at every new point gives:

k	x_k	y_k	$\ F(\mathbf{x}_k)\ $
0	0.500000000	0.500000000	$3.7814 \cdot 10^{-1}$
1	0.241177269	0.747712776	$2.8894 \cdot 10^{-2}$
2	0.262087110	0.740849355	$5.6412 \cdot 10^{-5}$
3	0.262119525	0.740871739	$2.2957 \cdot 10^{-10}$
4	0.262119525	0.740871739	roundoff level

Substitution gives $\sin(0.262119525067) \approx 0.259128260689$ and $\cos(0.740871739311) \approx 0.737880474933$, checking both equations to the displayed precision. The determinant $-1 - \cos x \sin y$ is nonzero at this root.

We can also establish uniqueness independently of the iteration table. Every real solution has $x = 1 - \cos y \in [0, 2]$; hence $\sin x \geq 0$ and $y = 1 - \sin x \in [0, 1]$. Substitution reduces the problem to

$$\begin{aligned} h(y) &= \sin(1 - \cos y) + y - 1 = 0, & 0 \leq y \leq 1, \\ h'(y) &= 1 + \cos(1 - \cos y) \sin y \geq 1. \end{aligned} \quad (4.15)$$

Indeed, $0 \leq 1 - \cos y \leq 1 - \cos 1 < \pi/2$. Since $h(0) = -1$ and $h(1) = \sin(1 - \cos 1) > 0$, continuity gives a root, and strict monotonicity gives exactly one. Newton's method supplies its numerical coordinates; the separate argument establishes existence and uniqueness.

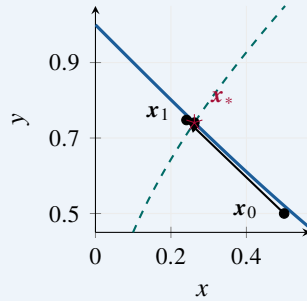


FIGURE 9: The solid curve is $y = 1 - \sin x$; the dashed curve is $x = 1 - \cos y$. The first correction moves toward their intersection, and the next iterate is already very close to it. The curves and Jacobian evaluations use radians.

Example 10 (Three unknowns with two exact linear constraints). Solve a sphere–plane intersection system:

$$F(x, y, z) = \begin{pmatrix} x^2 + y^2 + z^2 - 9 \\ x + y - 2 \\ y + z - 3 \end{pmatrix}, \quad J(x, y, z) = \begin{pmatrix} 2x & 2y & 2z \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \quad (4.16)$$

Starting at $\mathbf{x}_0 = (1, 1, 2)^\top$, the residual is $(-3, 0, 0)^\top$. For a correction $(u, v, w)^\top$, the linear system is

$$2u + 2v + 4w = 3, \quad u + v = 0, \quad v + w = 0. \quad (4.17)$$

The last two equations give $u = w = -v$. The first then gives $-4v = 3$, so

$$(u, v, w) = (3/4, -3/4, 3/4), \quad \mathbf{x}_1 = (1.75, 0.25, 2.75)^\top. \quad (4.18)$$

Every Newton update satisfies both affine constraints exactly. Parameterize their intersection line by $(x, y, z) = (2 - t, t, 3 - t)$. The remaining equation and its Newton step become

$$q(t) = (2 - t)^2 + t^2 + (3 - t)^2 - 9 = 3t^2 - 10t + 4, \quad (4.19)$$

$$t^+ = t - \frac{q(t)}{6t - 10}.$$

This reduction explains the three-dimensional computation; one scalar t determines all three coordinates. From $t_1 = 1/4$, for example, $q(t_1) = 27/16$ and $q'(t_1) = -17/2$, giving $t_2 = 61/136 \approx 0.448529412$.

k	x_k	y_k	z_k	$\ F(\mathbf{x}_k)\ $
0	1.000000000	1.000000000	2.000000000	3
1	1.750000000	0.250000000	2.750000000	1.6875
2	1.551470588	0.448529412	2.551470588	$1.1824 \cdot 10^{-1}$
3	1.535292638	0.464707362	2.535292638	$7.8518 \cdot 10^{-4}$
4	1.535183763	0.464816237	2.535183763	$3.5561 \cdot 10^{-8}$
5	1.535183758	0.464816242	2.535183758	roundoff level

The two exact solutions are $(2 - t_-, t_-, 3 - t_-)^\top$ and $(2 - t_+, t_+, 3 - t_+)^\top$, where

$$t_{\pm} = \frac{5 \pm \sqrt{13}}{3}. \quad (4.20)$$

Both satisfy the two linear equations by construction, and $q(t_{\pm}) = 0$ checks the sphere equation. The table converges to the t_- solution. There are exactly two solutions because every solution lies on the parameterized line and q has exactly these two roots. Moreover, $\det J = 2(x - y + z) = 10 - 6t$ on the line, which equals $\mp 2\sqrt{13}$ at $t_{\pm}$; both roots are simple. Once the affine constraints hold, the three-dimensional error is $\sqrt{3}|t - t_{\pm}|$.

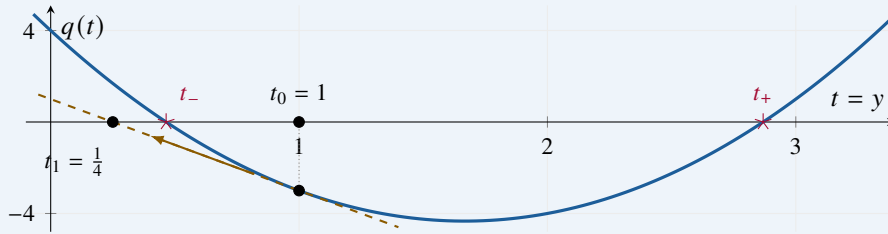


FIGURE 10: Restricting the sphere equation to the line of affine constraints gives the scalar parabola $q(t)$. Its tangent at $t_0 = 1$ meets the axis at $t_1 = 1/4$. The two scalar roots correspond to the two sphere–line intersections in three dimensions.

5 Gröbner bases: illustrating the idea

A useful idea in solving polynomial equations is to replace a complicated system by simpler equations obtained through exact algebra. Gröbner bases provide a systematic version of this idea. We begin with a very simple example of a Gröbner basis, then explore cancellation and back-substitution through calculations that can be checked by hand. Our aim is to illustrate the idea; a formal definition and a general construction algorithm are outside the scope of this lecture. For the general theory, see [5, Chapters 2–3]; the foundational algorithm is due to Buchberger [6].

5.1 A very simple example

Example 11 (A nonlinear system with a simple Gröbner basis). Consider the two equations

$$f_1 = x + y - 2 = 0, \quad f_2 = x^2 + xy - 2 = 0. \quad (5.1)$$

The second equation is nonlinear, but its first two terms are $x(x + y)$. This suggests using the first equation to simplify it.

Step 1: cancel the quadratic terms. Multiply f_1 by x and subtract it from f_2 :

$$f_2 - xf_1 = (x^2 + xy - 2) - (x^2 + xy - 2x) = 2x - 2. \quad (5.2)$$

Every common solution must therefore have $x = 1$. Substitution into $x + y = 2$ then gives $y = 1$.

Step 2: record the simpler polynomials. The calculation produces

$$g_1 = \frac{1}{2}(f_2 - xf_1) = x - 1, \quad g_2 = f_1 - g_1 = y - 1. \quad (5.3)$$

Conversely, the identities

$$f_1 = g_1 + g_2, \quad f_2 = (x + 2)g_1 + xg_2 \quad (5.4)$$

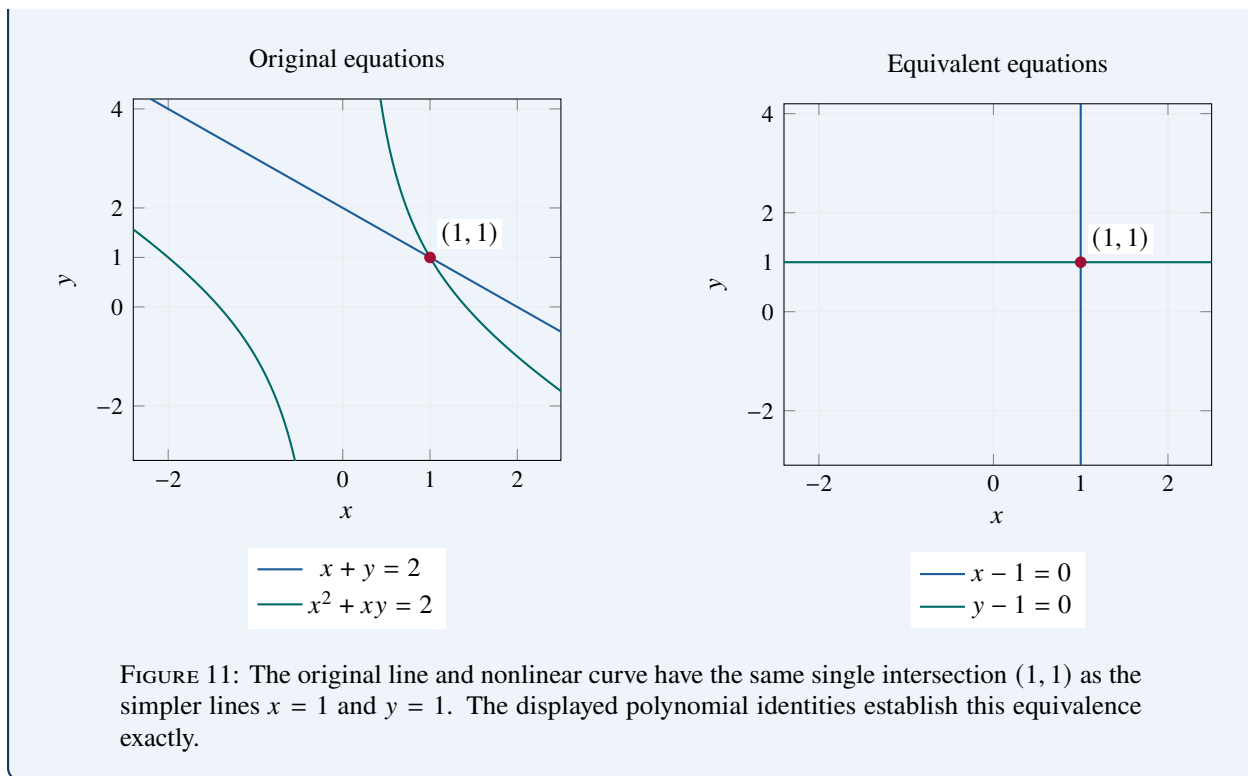
recover both original polynomials. Thus the old and new equations imply each other. Moreover, every polynomial combination of one pair can be rewritten as a polynomial combination of the other pair.

Step 3: identify the basis and read the solution. For this system, the set

$$G = \{x - 1, y - 1\} \quad (5.5)$$

is a *Gröbner basis*. Its equations separate the variables completely: $x - 1 = 0$ and $y - 1 = 0$. Hence $(x, y) = (1, 1)$ is the only solution, over either $\mathbb{R}$ or $\mathbb{C}$. Checking the original equations gives $1 + 1 = 2$ and $1^2 + 1 \cdot 1 = 2$.

This example shows the intended benefit: the simpler polynomial list retains the original algebraic information while making the solution immediate. With $x - 1$ and $y - 1$ available, simplifying any polynomial amounts to substituting $x = 1$ and $y = 1$. We use this concrete example to illustrate the name, without introducing the general test for a Gröbner basis.



How hard is it to find a Gröbner basis?

Example 11 was deliberately chosen to be easy: $f_2 - xf_1 = 2(x - 1)$ immediately removes the quadratic terms, and one more subtraction gives $y - 1$. We can discover the basis in a few lines by hand. For general polynomial systems with exact rational coefficients, algorithms such as Buchberger's provide a systematic construction that terminates after finitely many exact arithmetic steps [5, Chapter 2].

Finite termination does not guarantee a short calculation. Intermediate polynomials may have much higher degrees, many more terms, and coefficients whose numerators and denominators require many digits. Storing and manipulating these expressions can consume substantial time and memory, even when the original equations look simple. The number of variables, the degrees, and the algebraic structure all affect the difficulty.

Many small or suitably structured systems are manageable with computer algebra software; other systems are prohibitively expensive. Our short examples illustrate the idea, rather than typical running times. Furthermore, finding a basis and solving its equations are separate tasks: polynomial factorization and root-finding may still be needed afterward. When only one nearby approximate root is wanted, Newton's method can be a more practical approach, provided a suitable starting point is available.

5.2 The basic operation: multiply, subtract, simplify

In Gaussian elimination we add multiples of equations to cancel variables. For polynomial equations, the multipliers may themselves be polynomials. If $f_1 = \dots = f_m = 0$, then

$$a_1 f_1 + \dots + a_m f_m = 0 \quad (5.6)$$

for any polynomial multipliers $a_1, \dots, a_m$. This produces a consequence that every original solution must satisfy. To *replace* the old equations by a shorter or simpler list, we also check the reverse direction: every solution of the new list must satisfy all of the old equations.

Key Idea

Look for combinations that remove a variable. Solve the simpler equations when possible, then substitute back into the original system. Adding a polynomial consequence preserves every solution; division by a variable requires separate attention to the case where that variable is zero.

Example 12 (Discover a useful equation by cancellation). Consider

$$f_1 = xy - 1 = 0, \quad f_2 = y^2 - 1 = 0. \quad (5.7)$$

We want an equation that makes the relation between x and y explicit.

Step 1: arrange a cancellation. Multiply the first polynomial by y and the second by x :

$$yf_1 = xy^2 - y, \quad xf_2 = xy^2 - x. \quad (5.8)$$

Their xy^2 terms match, so subtracting gives

$$h = yf_1 - xf_2 = (xy^2 - y) - (xy^2 - x) = x - y. \quad (5.9)$$

At any original solution, both yf_1 and xf_2 are zero, so $h = 0$. We have discovered the simple relation $x = y$.

Step 2: check the simpler system in both directions. Keep the equations

$$h = x - y = 0, \quad f_2 = y^2 - 1 = 0. \quad (5.10)$$

The original pair implies this pair by the cancellation above. Conversely,

$$yh + f_2 = y(x - y) + (y^2 - 1) = xy - 1 = f_1. \quad (5.11)$$

Thus $h = f_2 = 0$ also gives $f_1 = 0$. The two systems have exactly the same solutions. We checked this by an elementary identity, without dividing by x or y .

Step 3: solve and retain both branches. Factor $y^2 - 1 = (y - 1)(y + 1)$. If $y = 1$, then $x = y = 1$; if $y = -1$, then $x = y = -1$. The complete solution list is

$$(x, y) = (1, 1), \quad (x, y) = (-1, -1). \quad (5.12)$$

Both points satisfy $xy = 1$ and $y^2 = 1$. There are no further complex solutions, because the factorization lists every possible value of y .

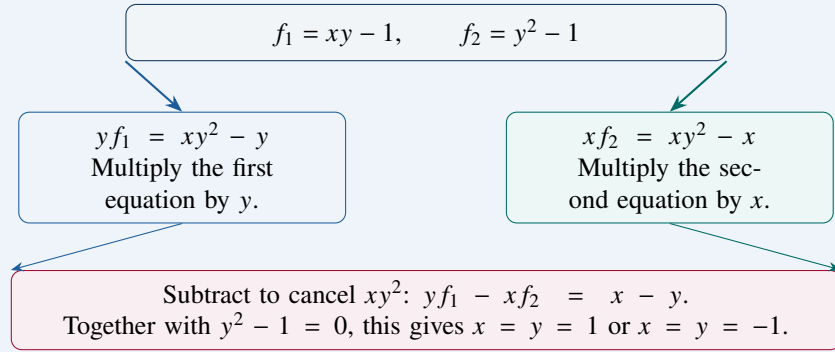


FIGURE 12: Polynomial multipliers create matching xy^2 terms, which cancel on subtraction. The new equation $x - y = 0$ makes back-substitution immediate; keeping $y^2 - 1 = 0$ preserves both branches.

The next examples continue with elementary polynomial elimination. In each case, we show the combinations that produce simpler equations, check the reverse implication, and solve the resulting equations by back-substitution. These calculations illustrate the motivation for Gröbner bases; they do not require a general Gröbner-basis procedure.

Example 13 (The robot arm revisited: Recover both configurations of the two-link robot). Return to Examples 1 and 7: the robot has base $(0, 0)$, target $(2, 2)$, and two links of length 2 m. If the elbow is at (x, y) , the squared-distance constraints are

$$f_1 = x^2 + y^2 - 4 = 0, \quad f_2 = (x - 2)^2 + (y - 2)^2 - 4 = 0. \quad (5.13)$$

The first equation keeps the elbow two metres from the base; the second keeps it two metres from the target. We will find every possible elbow position by exact elimination. This illustrates the algebraic elimination behind Gröbner-basis computations using combinations we can check by hand.

Step 1: subtract to cancel the squared terms. Expand the second polynomial:

$$f_2 = x^2 + y^2 - 4x - 4y + 4. \quad (5.14)$$

The x^2 and y^2 terms are identical in the two polynomials, so subtraction gives

$$f_1 - f_2 = 4x + 4y - 8 = 4(x + y - 2). \quad (5.15)$$

Every solution must therefore satisfy the simpler equation

$$g_1 = x + y - 2 = 0, \quad x = 2 - y. \quad (5.16)$$

Geometrically, the elbow lies on the perpendicular bisector of the segment joining the base to the target: it is equally far from both endpoints.

Step 2: substitute to obtain an equation in one variable. Insert $x = 2 - y$ into $f_1 = 0$:

$$\begin{aligned} 0 &= (2 - y)^2 + y^2 - 4 \\ &= 4 - 4y + y^2 + y^2 - 4 \\ &= 2y^2 - 4y = 2y(y - 2). \end{aligned} \quad (5.17)$$

Dividing by the nonzero constant 2 gives

$$g_2 = y^2 - 2y = y(y - 2) = 0. \quad (5.18)$$

We have reached a triangular system: first solve $g_2 = 0$ for y , then use $g_1 = 0$ to recover x .

Step 3: check that the simpler system preserves both directions. The cancellation and substitution can be recorded by polynomial identities:

$$\begin{aligned} g_1 &= \frac{f_1 - f_2}{4}, \\ g_2 &= \frac{f_1 - (x - y + 2)g_1}{2}, \\ f_1 &= (x - y + 2)g_1 + 2g_2, \\ f_2 &= f_1 - 4g_1. \end{aligned} \quad (5.19)$$

The first two lines show that $f_1 = f_2 = 0$ implies $g_1 = g_2 = 0$; the last two show the converse. Thus the original distance constraints and the triangular system have exactly the same solutions. We divided only by nonzero constants, so no case was discarded.

Step 4: retain and interpret both branches. Factoring g_2 gives two possibilities:

$$\begin{aligned} y = 0 &\implies x = 2, & (x, y) &= (2, 0), \\ y = 2 &\implies x = 0, & (x, y) &= (0, 2). \end{aligned} \quad (5.20)$$

For the first configuration, the links run from $(0, 0)$ to $(2, 0)$ and then to $(2, 2)$: right, then up. For the second, they run from $(0, 0)$ to $(0, 2)$ and then to $(2, 2)$: up, then right. Each segment has length 2 m. Equivalently, the original equations give $4 - 4 = 0$ for both squared distances at each elbow position. These are all solutions: every solution must satisfy $y(y - 2) = 0$, and each of its two roots determines exactly one x . Both configurations are real and satisfy the stated link-length and target constraints. Newton's method can approximate one configuration from a suitable starting point; the factorization here reveals both configurations and establishes that the list is complete.

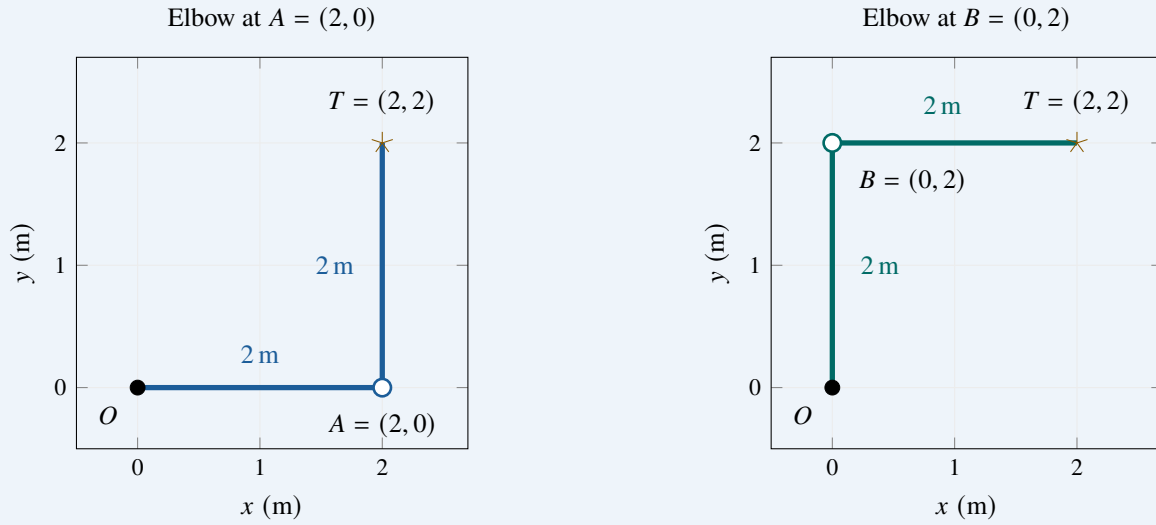


FIGURE 13: The two elbow positions found by exact elimination. Each configuration has two links of length 2 m and reaches the same target. Newton's method with the original starting guess finds the configuration on the left; reflection of that guess finds the one on the right.

Example 14 (The circle and parabola revisited: From the original equations to a simpler polynomial system). Return to (1.4) and write

$$f_1 = x^2 + y^2 - 3 = 0, \quad f_2 = x - y^2 + 1 = 0. \quad (5.21)$$

We will combine these polynomials to remove x and obtain an equation in y .

Step 1: cancel x^2 . The polynomial $x f_2$ contains the same x^2 term as f_1 . Subtract it:

$$\begin{aligned} h &= f_1 - x f_2 \\ &= x^2 + y^2 - 3 - x(x - y^2 + 1) \\ &= x y^2 - x + y^2 - 3. \end{aligned} \quad (5.22)$$

Every original solution satisfies $h = 0$, since it satisfies $f_1 = f_2 = 0$.

Step 2: cancel the remaining terms that contain x . Multiplying f_2 by y^2 creates the term $x y^2$ that appears in h . Subtract this multiple:

$$h - y^2 f_2 = (x y^2 - x + y^2 - 3) - y^2(x - y^2 + 1) = -x + y^4 - 3. \quad (5.23)$$

There is now just one term containing x . Add f_2 to cancel it:

$$(-x + y^4 - 3) + (x - y^2 + 1) = y^4 - y^2 - 2 =: g_2. \quad (5.24)$$

The equation $g_2 = 0$ contains only y . Keep it together with $g_1 = f_2$:

$$g_1 = x - y^2 + 1 = 0, \quad g_2 = y^4 - y^2 - 2 = 0. \quad (5.25)$$

This pair is triangular: the second equation determines possible values of y , and the first gives $x = y^2 - 1$ for each choice.

Step 3: verify that the simpler pair is equivalent to the original pair. The cancellations combine into the identities

$$\begin{aligned} g_2 &= f_1 - (x + y^2 - 1)f_2, \\ f_1 &= (x + y^2 - 1)g_1 + g_2, \quad f_2 = g_1. \end{aligned} \quad (5.26)$$

If $f_1 = f_2 = 0$, then $g_1 = g_2 = 0$. Conversely, if $g_1 = g_2 = 0$, then $f_1 = f_2 = 0$. Hence the two systems have exactly the same solutions, including any possible zero coordinates.

There is a useful substitution check on the arithmetic. From $f_2 = 0$, $x = y^2 - 1$, and therefore

$$f_1 = (y^2 - 1)^2 + y^2 - 3 = y^4 - y^2 - 2. \quad (5.27)$$

This recovers exactly the equation found by cancellation.

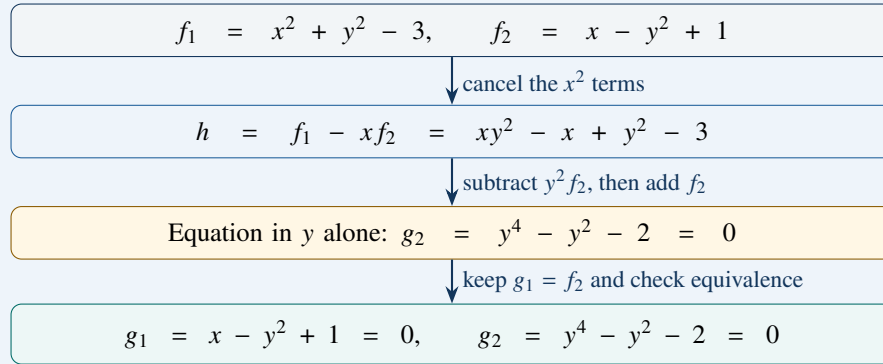


FIGURE 14: Successive polynomial differences remove every occurrence of x . The identity $f_1 = (x + y^2 - 1)g_1 + g_2$, together with $g_1 = f_2$, verifies that the simpler pair has exactly the original solutions.

Example 15 (Recover every solution and check the original equations). Treat $y^4 - y^2 - 2 = 0$ as a quadratic in $t = y^2$:

$$t^2 - t - 2 = (t - 2)(t + 1) = 0. \quad (5.28)$$

There are two cases, $y^2 = 2$ and $y^2 = -1$.

1. If $y^2 = 2$, then $y = \sqrt{2}$ or $y = -\sqrt{2}$. In both cases, $x = y^2 - 1 = 1$, giving $(1, \sqrt{2})$ and $(1, -\sqrt{2})$.
2. If $y^2 = -1$, there is no real y . Over $\mathbb{C}$, however, $y = i$ or $y = -i$, and $x = y^2 - 1 = -2$, giving $(-2, i)$ and $(-2, -i)$.

The complete complex solution set is

$$(x, y) = (1, \sqrt{2}), \quad (1, -\sqrt{2}), \quad (-2, i), \quad (-2, -i), \quad (5.29)$$

with only the first two points real. Rational coefficients do not restrict the roots to rational numbers.

Check the original equations. For the real points, $x = 1$ and $y^2 = 2$, so

$$x^2 + y^2 - 3 = 1 + 2 - 3 = 0, \quad x - y^2 + 1 = 1 - 2 + 1 = 0. \quad (5.30)$$

For the nonreal points, $x = -2$ and $y^2 = -1$, so

$$x^2 + y^2 - 3 = 4 - 1 - 3 = 0, \quad x - y^2 + 1 = -2 - (-1) + 1 = 0. \quad (5.31)$$

Why no solutions are missing. The cancellation identities force every original solution to satisfy $y^4 - y^2 - 2 = 0$. Its four roots exhaust the possibilities for y , and $x = y^2 - 1$ determines one x for each. Every resulting point passes the original equations.

Compare with Newton's method. Starts $(2, 1)$ and $(2, -1)$ approach $(1, \sqrt{2})$ and $(1, -\sqrt{2})$, respectively. A run that keeps all iterates real does not discover $(-2, \pm i)$. Here

$$J_F(x, y) = \begin{pmatrix} 2x & 2y \\ 1 & -2y \end{pmatrix}, \quad \det J_F(x, y) = -2y(2x + 1). \quad (5.32)$$

The determinant is nonzero at all four roots, but zero along $y = 0$ and $x = -1/2$. The real roots admit local Newton refinement; arbitrary initial guesses can still fail.

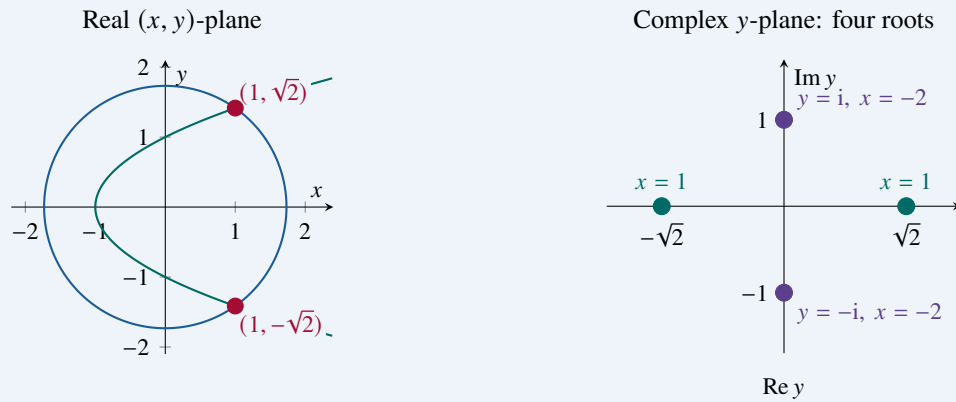


FIGURE 15: Left: the real circle (blue) and parabola (teal) intersect at two points. Right: the elimination polynomial has four roots in the complex y -plane; $x = y^2 - 1$ recovers each corresponding x . The nonreal solutions are represented only in the complex-coordinate panel.

Example 16 (Solving a triangular system: Three numbers from their symmetric sums). Suppose three unknown numbers satisfy

$$x + y + z = 6, \quad xy + xz + yz = 11, \quad xyz = 6. \quad (5.33)$$

Seek simpler equations involving y, z , followed by an equation in z alone.

Step 1: express x using the first equation. Write

$$g_1 = x + y + z - 6 = 0, \quad x = 6 - y - z. \quad (5.34)$$

Substitute into the second equation and expand:

$$\begin{aligned} xy + xz + yz &= (6 - y - z)y + (6 - y - z)z + yz \\ &= 6y - y^2 - yz + 6z - yz - z^2 + yz \\ &= 6y + 6z - y^2 - yz - z^2. \end{aligned} \quad (5.35)$$

Setting this equal to 11 and moving the terms gives

$$g_2 = y^2 + yz + z^2 - 6y - 6z + 11 = 0. \quad (5.36)$$

The variable x has disappeared.

Step 2: use the product equation to remove y as well. Substitution in the third equation gives

$$q = yz(6 - y - z) - 6 = 6yz - y^2z - yz^2 - 6 = 0. \quad (5.37)$$

The terms y^2z and yz^2 can be canceled using g_2 . Multiply it by z :

$$zg_2 = y^2z + yz^2 + z^3 - 6yz - 6z^2 + 11z. \quad (5.38)$$

Adding this to q cancels every term containing y :

$$\begin{aligned} q + zg_2 &= (6yz - y^2z - yz^2 - 6) \\ &\quad + (y^2z + yz^2 + z^3 - 6yz - 6z^2 + 11z) \\ &= z^3 - 6z^2 + 11z - 6 =: g_3. \end{aligned} \quad (5.39)$$

Thus the original system implies the triangular system

$$\begin{aligned} g_1 &= x + y + z - 6 = 0, \\ g_2 &= y^2 + yz + z^2 - 6y - 6z + 11 = 0, \\ g_3 &= z^3 - 6z^2 + 11z - 6 = 0. \end{aligned} \quad (5.40)$$

Its structure is useful: solve $g_3 = 0$ for z , then $g_2 = 0$ for y , and finally $g_1 = 0$ for x .

Step 3: check that the substitutions work in reverse. Suppose $g_1 = g_2 = g_3 = 0$. The first equation gives $x = 6 - y - z$. The calculation in Step 1 then makes $xy + xz + yz = 11$. Also, $g_3 = q + zg_2$ gives $q = g_3 - zg_2 = 0$, so $xyz = 6$. Thus the triangular and original systems have exactly the same solutions. The next example works from g_3 upward and keeps every branch.

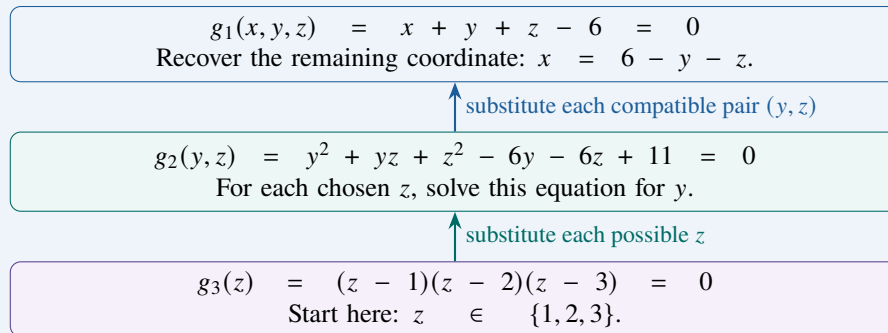


FIGURE 16: The new equations are triangular in variable dependence: g_3 uses only z , g_2 uses y, z , and g_1 uses x, y, z . Read the diagram from bottom to top to solve by back-substitution.

Example 17 (Enumerate all branches of the symmetric system). Begin with the last equation of (5.40):

$$g_3 = z^3 - 6z^2 + 11z - 6 = (z - 1)(z - 2)(z - 3) = 0. \quad (5.41)$$

For each possible value $z = 1, 2, 3$, solve the remaining quadratic for y .

1. **Branch $z = 1$.** Substitute into $g_2 = 0$:

$$y^2 + y + 1 - 6y - 6 + 11 = y^2 - 5y + 6 = (y - 2)(y - 3) = 0. \quad (5.42)$$

Using $x = 6 - y - z$, the choices $y = 2, 3$ give $(3, 2, 1)$ and $(2, 3, 1)$.

2. **Branch $z = 2$.** This time,

$$y^2 + 2y + 4 - 6y - 12 + 11 = y^2 - 4y + 3 = (y - 1)(y - 3) = 0. \quad (5.43)$$

The choices $y = 1, 3$ give $(3, 1, 2)$ and $(1, 3, 2)$.

3. **Branch $z = 3$.** Finally,

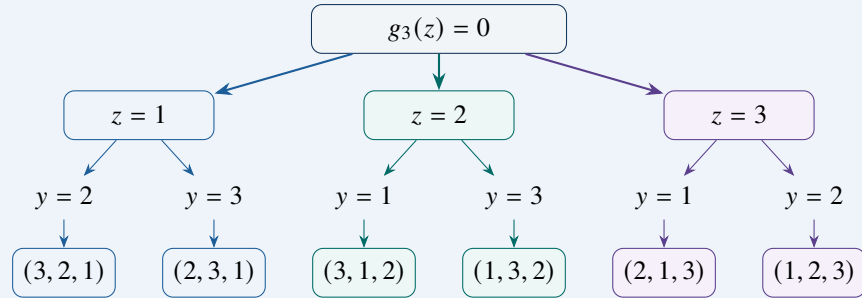
$$y^2 + 3y + 9 - 6y - 18 + 11 = y^2 - 3y + 2 = (y - 1)(y - 2) = 0. \quad (5.44)$$

The choices $y = 1, 2$ give $(2, 1, 3)$ and $(1, 2, 3)$.

The six solutions are exactly the six permutations of $(1, 2, 3)$. For every permutation, the original quantities are

$$\begin{aligned} x + y + z &= 1 + 2 + 3 = 6, \\ xy + xz + yz &= 2 + 3 + 6 = 11, \\ xyz &= 1 \cdot 2 \cdot 3 = 6. \end{aligned} \quad (5.45)$$

The three z branches and their two y choices exhaust the possibilities, even over $\mathbb{C}$. All six solutions are real. Stopping after one successful branch would miss valid solutions.



Each leaf is an ordered triple (x, y, z) , with $x = 6 - y - z$.

FIGURE 17: All branches must be retained. Each of the three choices of z allows two choices of y ; back-substitution gives exactly the six permutations of $(1, 2, 3)$.

5.3 No roots, nonreal roots, and infinitely many roots

Example 18 (Three outcomes that should not be confused). **An inconsistent system.** Consider

$$x + y - 1 = 0, \quad x + y - 2 = 0. \quad (5.46)$$

Subtracting the second left-hand side from the first gives

$$(x + y - 1) - (x + y - 2) = 1. \quad (5.47)$$

At a common root, the left side would be $0 - 0 = 0$. Thus the calculation produces the contradiction $1 = 0$, and there are no real or complex solutions. The constant polynomial 1 records the contradiction directly. The real lines are parallel and distinct.

A system with complex roots but no real roots. Now consider

$$x^2 + 1 = 0, \quad y = 0. \quad (5.48)$$

The equations are already separated. Over $\mathbb{R}$, the first is impossible because $x^2 \geq 0$. Over $\mathbb{C}$,

$$x^2 + 1 = (x - i)(x + i) = 0, \quad (5.49)$$

so the complete solution set is $(i, 0), (-i, 0)$. An algebraically consistent system need not have a real solution; the desired solution field must be specified.

A system with infinitely many roots. For the single equation $xy = 0$, one factor must be zero. There are two families:

$$(x, y) = (0, t) \quad \text{or} \quad (x, y) = (t, 0), \quad t \in \mathbb{R}. \quad (5.50)$$

One coordinate is free on each axis. Replacing $\mathbb{R}$ by $\mathbb{C}$ gives the complex solution set. Even this one-equation system has infinitely many solutions. Every y occurs with $x = 0$, so elimination cannot produce a finite list of possible y values.

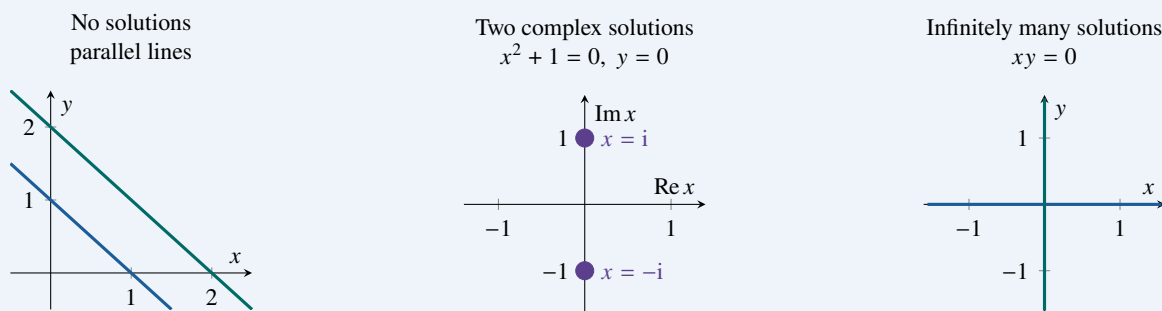


FIGURE 18: Three distinct outcomes: the real lines $x + y = 1$ and $x + y = 2$ never meet; the complex x -plane displays the two solutions $(\pm i, 0)$ of the middle system, which has no real solution; and the two real coordinate axes give infinitely many solutions of $xy = 0$.

Checkpoint

Why is division by a variable dangerous during elimination? In $xy = 0$, dividing by x and retaining only $y = 0$ loses every point $(0, y)$ with $y \neq 0$. Such division is valid only on the branch where $x \neq 0$; the branch $x = 0$ must be handled separately. Keeping both branches avoids this silent loss of solutions.

5.4 A missing elimination equation does not allow every value

The circle–parabola and symmetric-system examples produced an equation in one variable and a finite list of branches. The next system behaves differently.

Example 19 (Check a candidate by back-substitution). Consider the single polynomial equation

$$xy - 1 = 0. \quad (5.51)$$

There is no nonzero polynomial equation in y alone that follows from it. Indeed, every nonzero real or complex y occurs in a solution, $(x, y) = (1/y, y)$. A polynomial in y that vanishes at all these infinitely many values must be the zero polynomial. Thus polynomial elimination cannot supply a nonzero equation in y alone for this system.

Test the apparently unrestricted value $y = 0$. Substitution into the original equation gives $-1 = 0$, so this value is impossible. For every $y \neq 0$, division is allowed and gives exactly one corresponding value $x = 1/y$. The solution set is

$$\{(1/t, t) : t \neq 0\}, \quad (5.52)$$

with real or complex t according to the problem. It is an infinite family.

Even when elimination supplies no restriction on a coordinate, check that a proposed value extends to a solution of all the original equations. In particular, inspect the zero case before dividing by an expression.

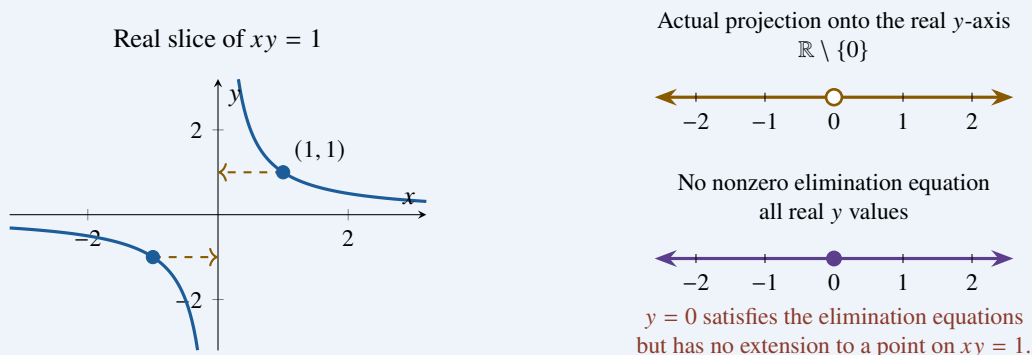


FIGURE 19: The real hyperbola allows every nonzero y , but not $y = 0$. With no nonzero equation in y alone, elimination leaves that coordinate unrestricted; back-substitution still rejects zero. The same distinction holds over $\mathbb{C}$.