Jacobi Iterative Method

Lecture 18 | Iterative methods for linear systems

Core idea	Start with an approximate solution and repeatedly update all components.
Key distinction	Every new component uses only values from the previous iteration.
Learning focus	Iteration formulas, convergence, numerical examples and Python.

Learning outcomes

By the end of this lecture, students should be able to:

Derive	Rearrange Ax = b into a simultaneous fixed-point iteration.
Compute	Generate successive approximations and check stopping criteria.
Evaluate	Identify convergence conditions and compare residuals and errors.

Direct versus iterative methods

Recall Gaussian elimination and LU decomposition.

Direct methods	Use factorization or elimination; obtain a solution in finitely many steps in exact arithmetic.
Iterative methods	Start with $x^{(0)}$ and generate $x^{(1)}$, $x^{(2)}$, … until a tolerance is satisfied.
Why Jacobi?	Simple updates, parallel computation and usefulness for sparse systems.

Derivation of the update

Assume each diagonal entry a_{ii} is nonzero.

Row i $a_{i1}x_1 + \dots + a_{ii}x_i + \dots + a_{in}x_n = b_i$

Isolate x_i $x_i = (b_i - \Sigma\, a_{ij}x_j) / a_{ii}$, summed over $j \neq i$.

Jacobi formula $x_i^{(k+1)} = (b_i - \Sigma\, a_{ij}x_j^{(k)}) / a_{ii}$, $j \neq i$.

Matrix formulation

Split A = D + L + U; L and U contain the actual off-diagonal entries.

Diagonal part	$D = \mathrm{diag}(a_{11}, \ldots, a_{nn})$. Then $Dx = b - (L + U)x$.
Iteration	$x^{(k+1)} = D^{-1}b - D^{-1}(L + U)x^{(k)}$.
Iteration matrix	$B = -D^{-1}(L + U)$; $c = D^{-1}b$. Thus $x^{(k+1)} = Bx^{(k)} + c$.

Algorithm and simultaneous updates

Use two vectors so that old values are never overwritten prematurely.

Initialize	Choose $x^{(0)}$, tolerance ε and maximum iteration count. Check nonzero diagonal.
Update	For each i, compute new[i] using only old[j]. Then evaluate residual and step size.
Repeat or stop	If tolerance is met, return new; otherwise copy new to old and repeat. Report failure at limit.

Example 1 — two variables

Solve $4x + y = 9$ and $x + 3y = 7$, starting at $(0, 0)$.

Rearrange $x^{(k+1)} = (9 - y^{(k)})/4$; $y^{(k+1)} = (7 - x^{(k)})/3$.

First iteration $x^{(1)} = 9/4 = 2.25$; $y^{(1)} = 7/3 = 2.333333$.

Second iteration $x^{(2)} = (9 - 2.333333)/4 = 1.666667$; $y^{(2)} = (7 - 2.25)/3 = 1.583333$.

Example 1 — iteration table

Keep full precision internally; display rounded values.

k = 0, 1, 2	(0, 0) → (2.250000, 2.333333) → (1.666667, 1.583333)
k = 3, 4, 5	(1.854167, 1.777778) → (1.805556, 1.715278) → (1.821181, 1.731481)
Exact solution	x = 20/11 ≈ 1.818182; y = 19/11 ≈ 1.727273. The iterates approach this pair.

Example 2 — three variables

Start from $x^{(0)} = (0, 0, 0)$.

System	$10x - y + 2z = 6;\ -x + 11y - z = 25;\ 2x - y + 10z = -11.$
Update equations	$x' = (6 + y - 2z)/10;\ y' = (25 + x + z)/11;\ z' = (-11 - 2x + y)/10.$
First iteration	$x^{(1)} = 0.600000;\ y^{(1)} = 2.272727;\ z^{(1)} = -1.100000.$

Example 2 — second iteration

Substitute only values from iteration 1.

x update	$(6 + 2.272727 - 2(-1.1))/10 = 1.047273$
y update	$(25 + 0.6 - 1.1)/11 = 2.227273$
z update	$(-11 - 2(0.6) + 2.272727)/10 = -0.992727$

Example 2 — subsequent iterations

Columns: k | x | y | z | residual infinity norm.

k = 3, 4	3 | 1.021273 | 2.277686 | −1.086727 | 0.238413 4 | 1.045114 | 2.266777 | −1.076486 | 0.058592
k = 5, 6	5 | 1.041975 | 2.269875 | −1.082345 | 0.014817 6 | 1.043457 | 2.269057 | −1.081407 | 0.003781
Limit	$x \approx 1.043269$; $y \approx 2.269231$; $z \approx -1.081731$.

Convergence: diagonal dominance

A useful sufficient condition, not a necessary condition.

Strict dominance	For every row i: $\lvert a_{ii}\rvert > \Sigma \lvert a_{ij}\rvert$, $j \neq i$.
Example 2	10 > 3; 11 > 2; 10 > 3. Therefore Jacobi converges from any starting vector.
General criterion	Jacobi converges for every initial vector iff spectral radius $\rho(B) < 1$.

Error and stopping criteria

A small change is not by itself proof of an accurate solution.

Step size	$s_k = \|x^{(k+1)} - x^{(k)}\|_\infty$; use a scaled tolerance if variables differ in magnitude.
Residual	$r_k = b - Ax^{(k)}$. Monitor $\|r_k\|_\infty$ alongside the step size.
Error bound	If $q = \|B\|_\infty < 1$, then $\|x^* - x^{(k+1)}\|_\infty \le q\, s_k/(1-q)$.

Divergence and limitations

A nonsingular system does not guarantee Jacobi convergence.

Counterexample	$x + 2y = 3$; $2x + y = 3$. Start at $(0,0)$.
Iterates	$(0,0) \to (3,3) \to (-3,-3) \to (9,9)$. The exact solution is $(1,1)$.
Diagnosis	$B = [[0,-2],[-2,0]]$ has spectral radius 2. Reordering rows may help in some systems.

Computational cost

Consider both arithmetic per sweep and the number of sweeps.

Dense matrix	One iteration requires $O(n^2)$ arithmetic; sparse matrices require $O(\text{nnz}(A))$.
Memory and parallelism	Two solution vectors; each new component can be computed independently.
Limitations	May converge slowly or diverge; convergence depends on A, not just the initial guess.

Python implementation

NumPy example: simultaneous updates with residual monitoring.

Initialize

```
D = A.diagonal(); R = A - np.diag(D)
x = np.zeros(len(b))  # A, b are NumPy arrays
```

Loop

```
for k in range(1000):
    new = (b - R @ x) / D
    step = np.max(abs(new - x)); x = new
```

Stop inside loop

```
    if step < 1e-8 and np.max(abs(b-A@x)) < 1e-8: break
else: raise RuntimeError("No convergence")
```

Practice questions

Attempt the exercises before viewing the solutions.

Q1	For 5x + y = 11 and x + 4y = 9, compute two Jacobi iterations from (0,0).
Q2	Does A = [[4,1],[2,3]] satisfy strict diagonal dominance?
Q3	Why must old and new values be stored separately during a Jacobi sweep?

Solutions and recap

Rearrange → update simultaneously → check convergence.

Q1	(2.2, 2.25) → (1.75, 1.7). Exact solution: (35/19, 34/19).
Q2 / Q3	Yes: 4 > 1 and 3 > 2. Separate vectors ensure updates use only the previous iteration.
Next lecture	NC_19 — Gauss-Seidel Iterative Method.