

The Structure-Preserving Doubling Algorithm

Solve a Riccati equation by joining optimal-control blocks

Ordinary Riccati iteration adds one time step. SDA joins a block to a copy of itself, eliminates their shared boundary, and doubles the effective horizon: 1, 2, 4, 8, ... The acceleration is a consequence of exact composition.

1. The problem: how much does the future cost?

Consider the time-invariant, discrete-time linear-quadratic regulator (LQR):

$$x_{t+1} = Ax_t + Bu_t, \quad J = \frac{1}{2} \sum_{t=0}^{\infty} (x_t^T Q x_t + u_t^T R u_t). \quad (1)$$

Here $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, $Q = Q^T \succeq 0$, and $R = R^T \succ 0$. The optimal infinite-horizon value is $V(x) = \frac{1}{2}x^T P x$. Bellman's principle gives the discrete-time algebraic Riccati equation (DARE):

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A. \quad (2)$$

We assume (A, B) is stabilizable and $(Q^{\frac{1}{2}}, A)$ is detectable. These standard sufficient conditions ensure a unique positive-semidefinite stabilizing solution. “Stabilizing” means that the feedback

$$K = (R + B^T P B)^{-1} B^T P A, \quad u = -Kx \quad (3)$$

makes $A - BK$ Schur: all its eigenvalues have modulus less than one.

Set $G = BR^{-1}B^T$. Eliminating the control, or using the matrix inversion lemma, rewrites Equation 2 as

$$P = Q + A^T P (I + GP)^{-1} A. \quad (4)$$

Order matters: these are matrices, not scalars. In particular, $P(I + GP)^{-1} = (I + PG)^{-1}P$.

For a finite horizon of N stages and **zero terminal cost**, let P_N be its value matrix. Then $P_0 = 0$ and

$$P_{N+1} = Q + A^T P_N (I + GP_N)^{-1} A. \quad (5)$$

Thus ordinary iteration computes horizons 1, 2, 3, ... SDA will compute $P_1, P_2, P_4, P_8, \dots$ without visiting every intermediate horizon.

2. The missing object: a block with two boundary variables

The value matrix alone describes the zero-terminal-cost case. To join blocks, we need a summary that also accepts a terminal costate λ_N : the gradient of the downstream value with respect to x_N . An N -step block has relations

$$x_N = E_N x_0 - G_N \lambda_N, \quad \lambda_0 = H_N x_0 + E_N^T \lambda_N. \quad (6)$$

These are **boundary relations**, not a forward state transition: we prescribe x_0 and λ_N , then recover x_N and λ_0 .

- E_N transmits the initial state across the optimized block.
- G_N measures how a terminal costate changes its terminal state.
- H_N is the initial value curvature when $\lambda_N = 0$; hence $H_N = P_N$.

For one step, stationarity and the costate equation are

$$R u_0 + B^T \lambda_1 = 0, \quad \lambda_0 = Q x_0 + A^T \lambda_1. \quad (7)$$

Substitute $u_0 = -R^{-1} B^T \lambda_1$ into the dynamics. Comparing with Equation 6 gives $E_1 = A$, $G_1 = G$, $H_1 = Q$. This needs no inverse of A .

2.1. Join two identical blocks

Temporarily omit the horizon subscript. Two copies of Equation 6 give

$$x_N = E x_0 - G \lambda_N, \quad \lambda_0 = H x_0 + E^T \lambda_N, \quad (8)$$

$$x_{2N} = E x_N - G \lambda_{2N}, \quad \lambda_N = H x_N + E^T \lambda_{2N}. \quad (9)$$

Eliminate the shared costate in the first relation:

$$(I + GH) x_N = E x_0 - G E^T \lambda_{2N}. \quad (10)$$

Define $W = I + GH$. Then substitute $x_N = W^{-1} E x_0 - W^{-1} G E^T \lambda_{2N}$ into the two outer relations. For the initial costate, use $I - H W^{-1} G = (I + HG)^{-1}$; for the terminal state, use $W^{-1} G = G(I + HG)^{-1}$. The result again has form Equation 6, with

$$E_+ = E(I + GH)^{-1} E, \quad (11)$$

$$G_+ = G + E G (I + HG)^{-1} E^T, \quad (12)$$

$$H_+ = H + E^T H (I + GH)^{-1} E. \quad (13)$$

Initialize $(E_0, G_0, H_0) = (A, BR^{-1}B^T, Q)$ and apply these updates simultaneously. The iteration index k now means a horizon of 2^k : $H_k = P_{2^k}$. Subscripts hereafter index **doublings**, not time steps.

3. What structure survives the doubling?

3.1. Symmetry, positivity, and a monotone value

Suppose G, H are symmetric positive semidefinite. Although $I + GH$ need not be symmetric, it is nonsingular: GH has nonnegative real eigenvalues (its nonzero eigenvalues coincide with those of $H^{\frac{1}{2}}GH^{\frac{1}{2}}$). Moreover,

$$H(I + GH)^{-1} = H^{\frac{1}{2}}(I + H^{\frac{1}{2}}GH^{\frac{1}{2}})^{-1}H^{\frac{1}{2}} \succeq 0. \quad (14)$$

This identity remains valid for singular H ; no inverse of H is needed. The analogous identity holds for $G(I + HG)^{-1}$. Therefore Equation 13 preserves symmetry and positivity, and

$$H_+ \succeq H, \quad G_+ \succeq G. \quad (15)$$

The first inequality has an optimal-control interpretation: adding nonnegative stage costs cannot reduce the zero-terminal-cost optimal value. Under our LQR assumptions, H_k increases to the stabilizing P .

3.2. The symplectic connection, without assuming invertible dynamics

Let $J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$, and write the boundary relation as a matrix pencil:

$$M \begin{pmatrix} x_0 \\ \lambda_0 \end{pmatrix} = L \begin{pmatrix} x_N \\ \lambda_N \end{pmatrix}, \quad M = \begin{pmatrix} E & 0 \\ -H & I \end{pmatrix}, \quad L = \begin{pmatrix} I & G \\ 0 & E^T \end{pmatrix}. \quad (16)$$

For symmetric G, H , direct multiplication gives

$$MJM^T = LJL^T = \begin{pmatrix} 0 & E \\ -E^T & 0 \end{pmatrix}. \quad (17)$$

This is the symplectic matrix-pair structure that the doubling retains in exact arithmetic. If E is invertible, $S = L^{-1}M$ is an ordinary symplectic state-costate transition: $S^TJS = J$. The pair formulation and SDA still make sense when E is singular; forming that transition is unnecessary.

3.3. Why convergence accelerates

The discrete Riccati map has derivative at its stabilizing solution

$$D\mathcal{F}_{P[\Delta]} = A_c^T \Delta A_c, \quad A_c = (I + GP)^{-1}A = A - BK. \quad (18)$$

Since A_c is Schur, ordinary iteration converges geometrically asymptotically. For any fixed r with $\rho(A_c) < r < 1$, sufficiently large horizons admit a bound of the form $\|P - P_N\| \leq Cr^{2N}$. Sampling at $N = 2^k$ gives

$$\|P - H_k\| \leq Cr^{2 \cdot 2^k}. \quad (19)$$

This explains the quadratic-type convergence in the doubling index; a bound alone is not an exact error recurrence. The formal SDA convergence analysis is given in Section 2 of [1]. Modes close to the unit circle can delay the fast asymptotic regime. Structure preservation is not a guarantee of perfect conditioning or of exact symmetry in floating-point arithmetic.

4. A scalar example you can check by hand

Take $A = 1.1$, $B = Q = R = 1$. Then $G = 1$ and

$$p = 1 + \frac{1.21p}{1+p}, \quad p^2 - 1.21p - 1 = 0. \quad (20)$$

The positive solution is

$$p_\star = \frac{1.21 + \sqrt{1.21^2 + 4}}{2} = 1.773770721741... \quad (21)$$

Its closed-loop multiplier is $\frac{1.1}{1+p_\star} \approx 0.39657$, inside the unit circle. The negative root is not the stabilizing solution.

Write the scalar SDA variables as (e, g, h) . Each iteration uses

$$d = 1 + gh, \quad e_+ = \frac{e^2}{d}, \quad g_+ = g + e^2 \frac{g}{d}, \quad h_+ = h + e^2 \frac{h}{d}. \quad (22)$$

Starting at $(1.1, 1, 1)$, the first update is $(e_+, g_+, h_+) = (0.605, 1.605, 1.605)$. The next h is $1.605 + 0.605^2 \cdot \frac{1.605}{1+1.605^2} \approx 1.769280206374$.

Iteration	Horizon	H_k	Absolute error
0	1	1.000000000000	7.738×10^{-1}
1	2	1.605000000000	1.688×10^{-1}
2	4	1.769280206374	4.491×10^{-3}
3	8	1.773767969365	2.752×10^{-6}
4	16	1.773770721740	1.030×10^{-12}
5	32	1.773770721741	2.220×10^{-16}

Table 1: Double-precision Octave results. After five doublings the error reaches the floating-point floor; further squaring cannot add reliable digits.

A visual measure: correct decimal digits. The bar length below represents $-\log_{10}(|p_\star - H_k|)$, capped at 16. Each iteration eventually gains roughly twice as many digits, until rounding dominates.

$k = 0$		0.11
$k = 1$		0.77
$k = 2$		2.35
$k = 3$		5.56
$k = 4$		11.99
$k = 5$		15.65

The speedup is in **iteration count**, not in matrix size. In the dense case, each doubling still requires $O(n^3)$ arithmetic and $O(n^2)$ storage.

5. Implementation: solve systems and check the original problem

The following Octave/MATLAB loop implements Equation 13. The transpose solve uses $(I + GH)^T = I + HG$; it does not assume $GH = HG$.

```
G = B * (R \ B'); H = Q; E = A;
G0 = G; n = size(A, 1); converged = false;
tol = 1e-12; maxit = 50;
for k = 1:maxit
    W = eye(n) + G*H;
    U = W \ E;
    V = W' \ E';
    En = E*U;
    Gn = G + E*G*V;
    Hn = H + E'*H*U;
    % All right-hand sides above use OLD E, G, H.
    E = En; G = Gn; H = Hn;
    % Remove only roundoff-level antisymmetric components.
    G = (G+G')/2; H = (H+H')/2;
    T = A'*H*((eye(n)+G0*H) \ A);
    residual = norm(Q + T - H, 'fro');
    scale = max(1, norm(Q, 'fro') + norm(T, 'fro') ...
                + norm(H, 'fro'));
    K = (R+B'*H*B) \ (B'*H*A);
    stable = max(abs(eig(A-B*K))) < 1;
    if residual/scale < tol && stable
        converged = true; break;
    end
end
if ~converged, error('SDA failed validation'); end
P = H;
```

Why these checks? A small increment can mean convergence or stagnation. The residual tests the original DARE, while the stability check distinguishes the stabilizing solution from other algebraic solutions. A small residual is not a forward-error certificate for an ill-conditioned problem; near-unit-circle eigenvalues also need a more careful stability assessment than this teaching loop. Symmetrization removes rounding noise, not an incorrect matrix ordering.

A noncommuting check. With

$$A = \begin{pmatrix} 1.1 & 0.3 \\ 0 & 0.95 \end{pmatrix}, \quad B = \begin{pmatrix} 0.2 \\ 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 2 & 0.4 \\ 0.4 & 1 \end{pmatrix}, \quad R = 0.7, \quad (23)$$

eight raw SDA updates (without symmetrization) gave a Frobenius difference of 2.2×10^{-15} from Octave's dare solution and an absolute compact-form residual of 6.3×10^{-16} . Unlike the scalar example, this checks matrix order.

Scope. These formulas are for the standard positive-semidefinite DARE. Continuous-time, indefinite, descriptor, and large sparse problems need suitable transformations or specialized variants, not blind reuse of this initialization.

Further reading

- [1] E. K.-W. Chu, H.-Y. Fan, W.-W. Lin, and C.-S. Wang, “Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations,” *International Journal of Control*, vol. 77, no. 8, pp. 767–788, 2004, doi: 10.1080/00207170410001714988.