GPU QMC OPTIMIZATION

Ming Wong
Tyler McDaniel

Quantum Monte Carlo Simulation

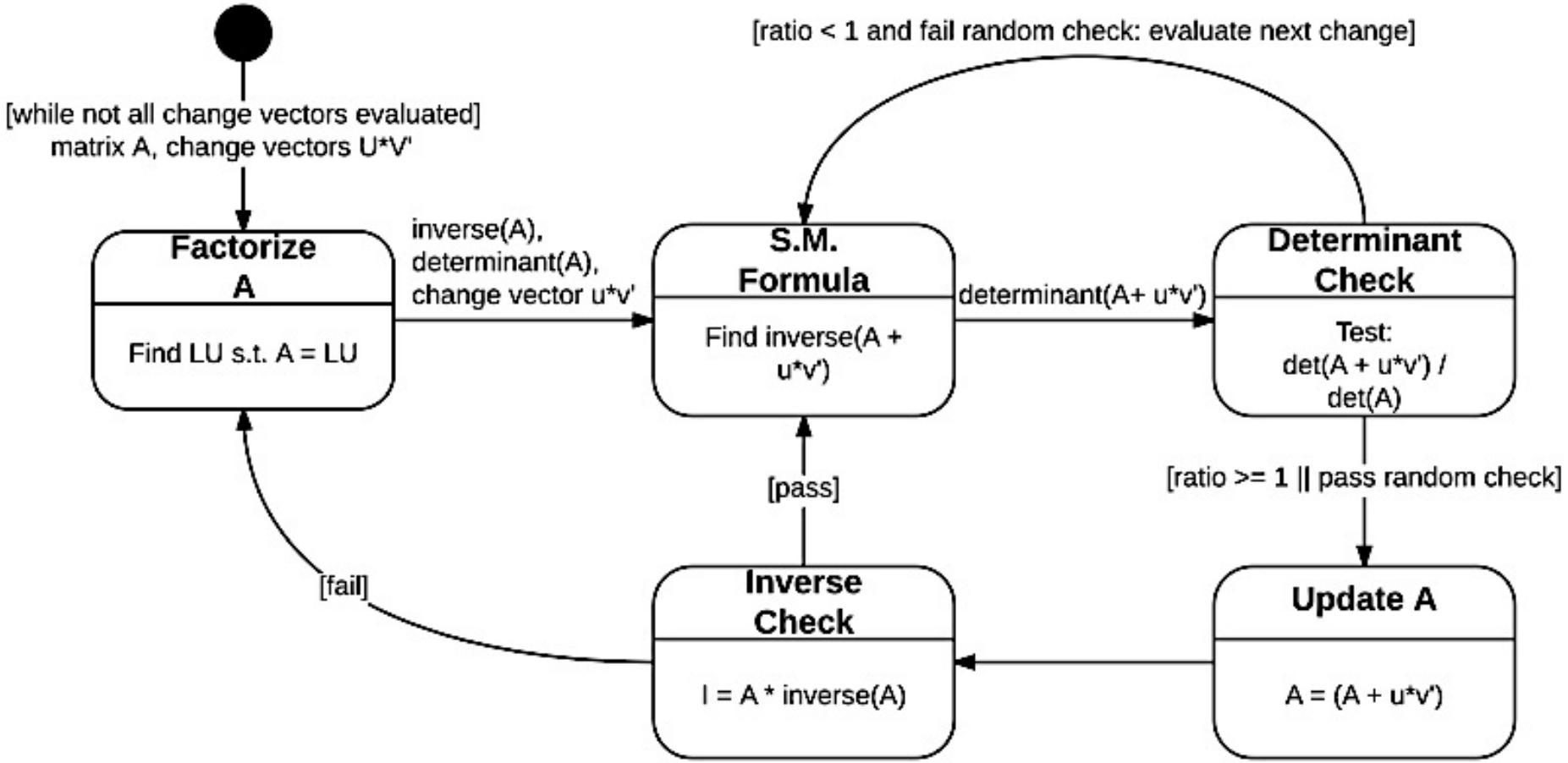
Slater Determinant for N-electrons system

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(\mathbf{x}_1) & \chi_2(\mathbf{x}_1) & \cdots & \chi_N(\mathbf{x}_1) \\ \chi_1(\mathbf{x}_2) & \chi_2(\mathbf{x}_2) & \cdots & \chi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(\mathbf{x}_N) & \chi_2(\mathbf{x}_N) & \cdots & \chi_N(\mathbf{x}_N) \end{vmatrix}$$

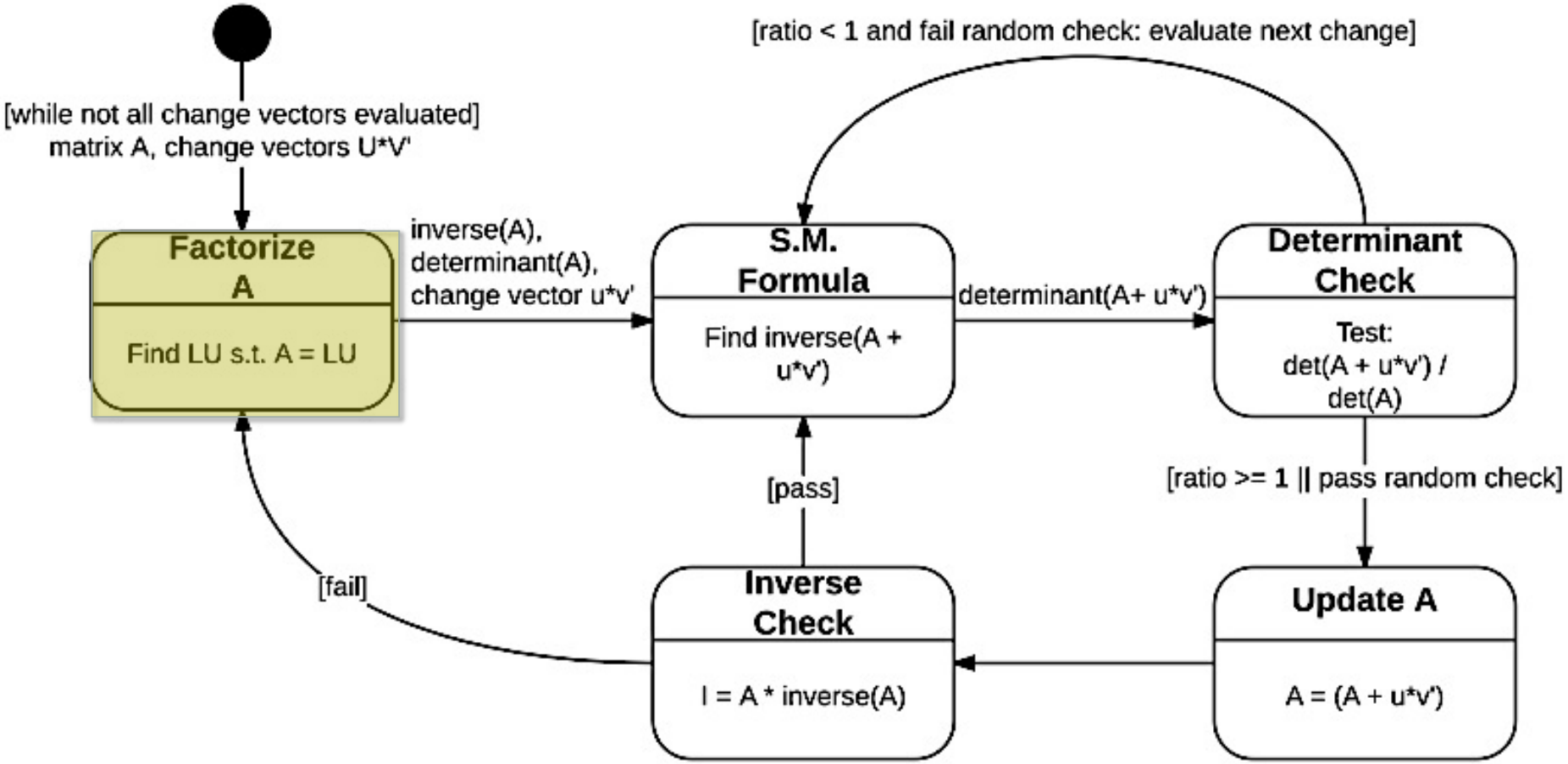
What is QMCPACK?

- Software framework for quantum Monte Carlo simulation
- Written in C++ w/Cuda kernels
- Utilizes Cuda (acceleration) and openMP (parallelization)

Current QMC implementation



Current QMPC implementation

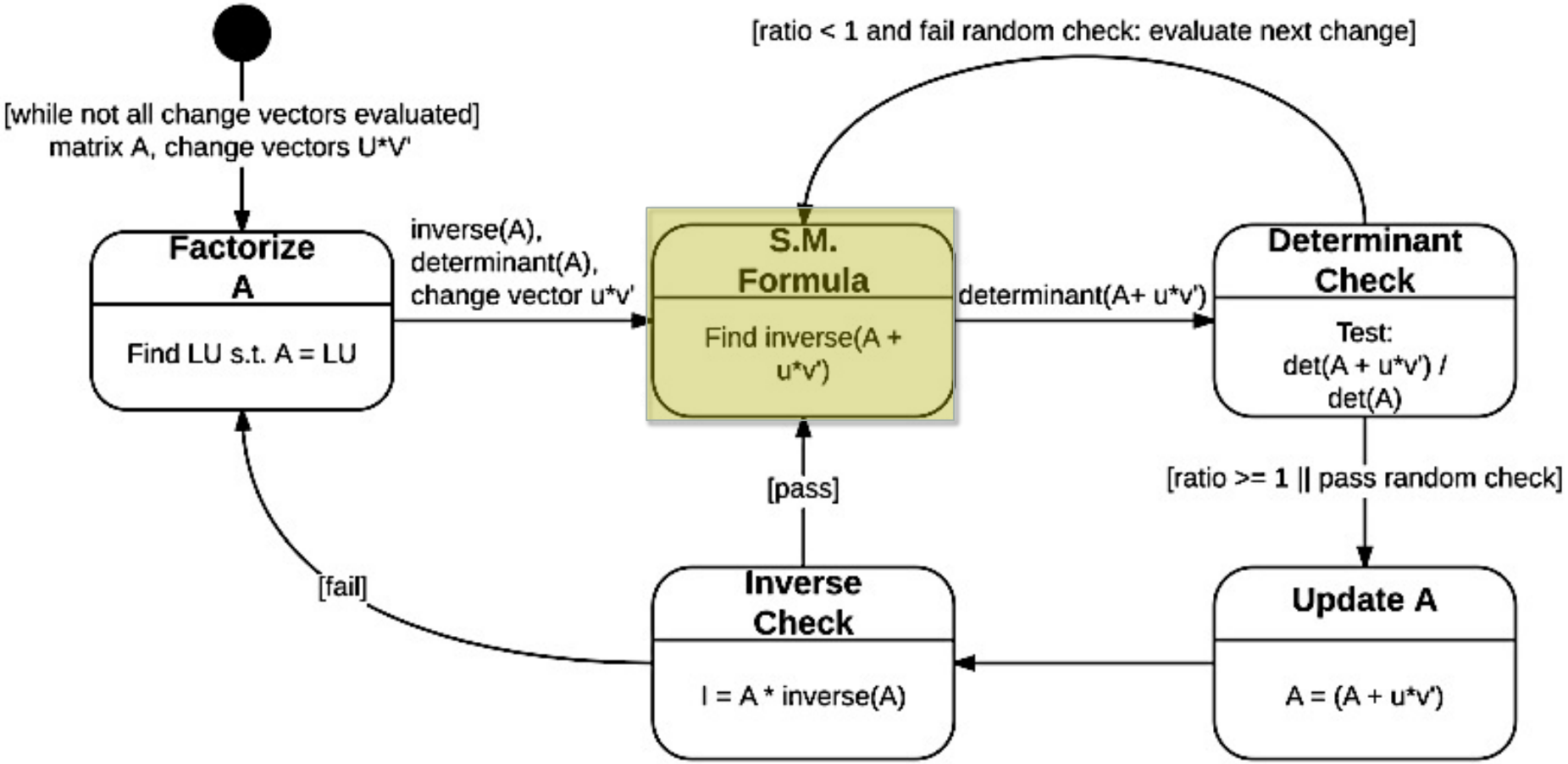


LU Decomposition

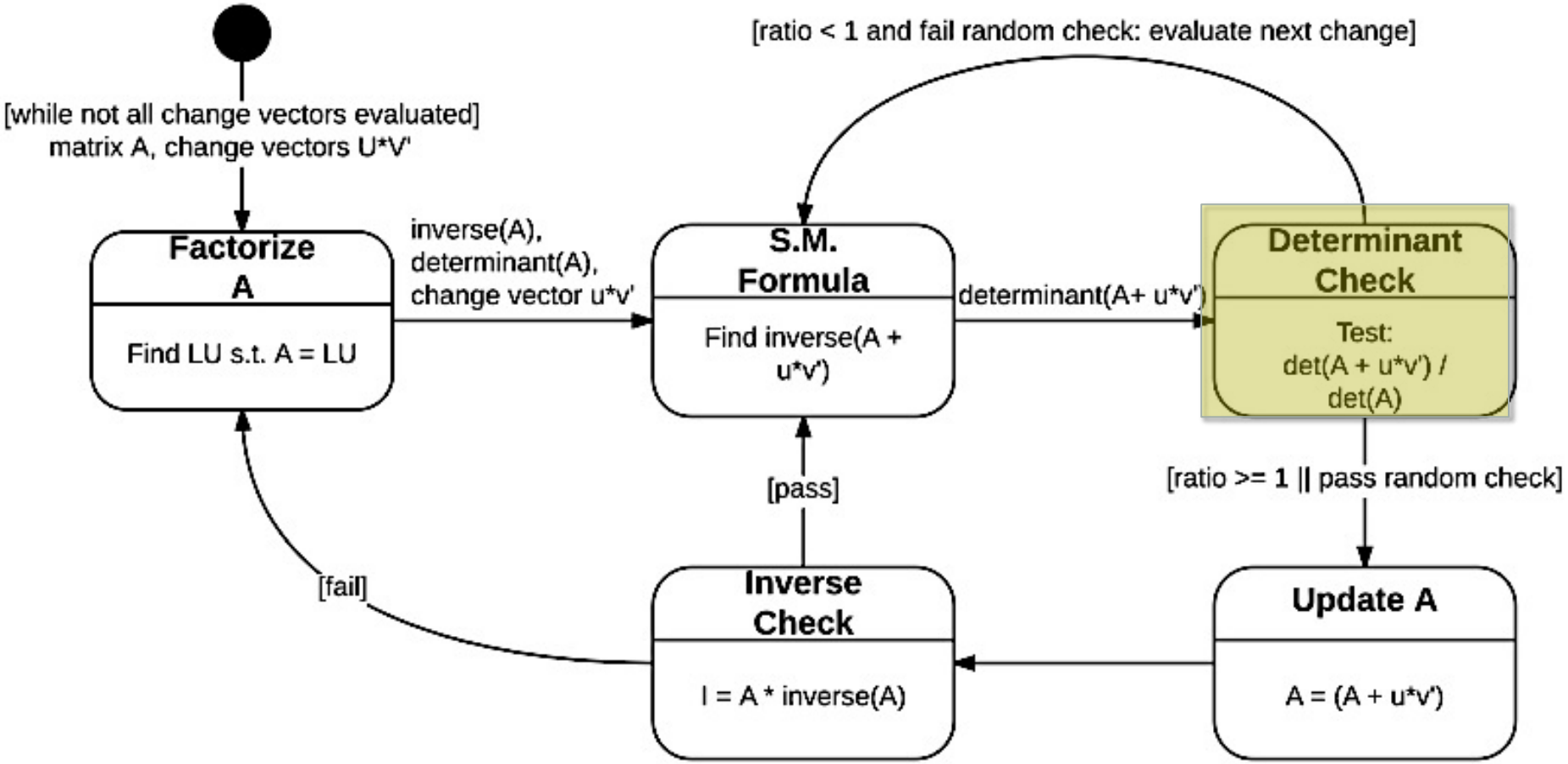
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

$$A = L * U$$

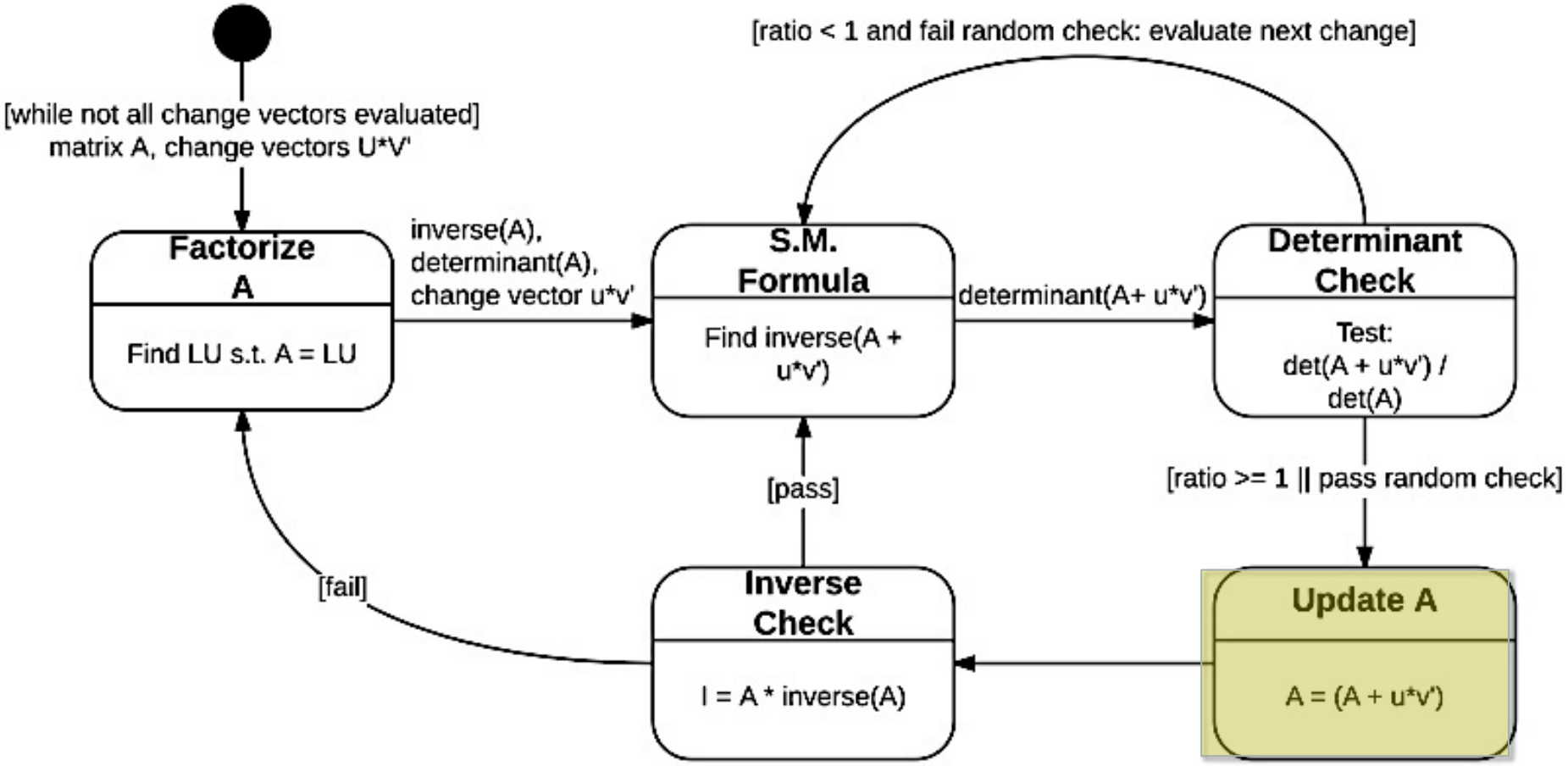
Current QMPC implementation



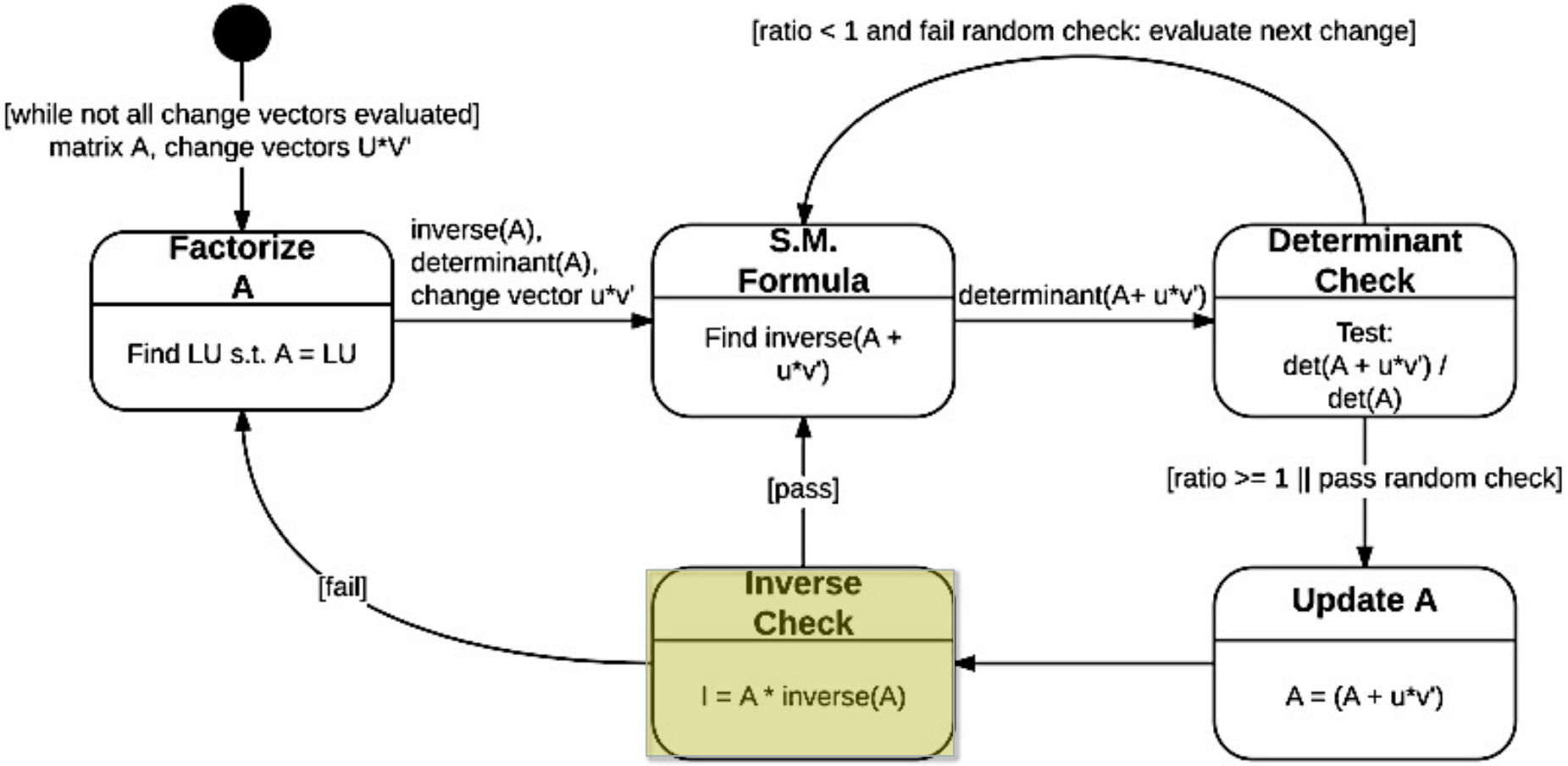
Current QMPC implementation



Current QMPC implementation



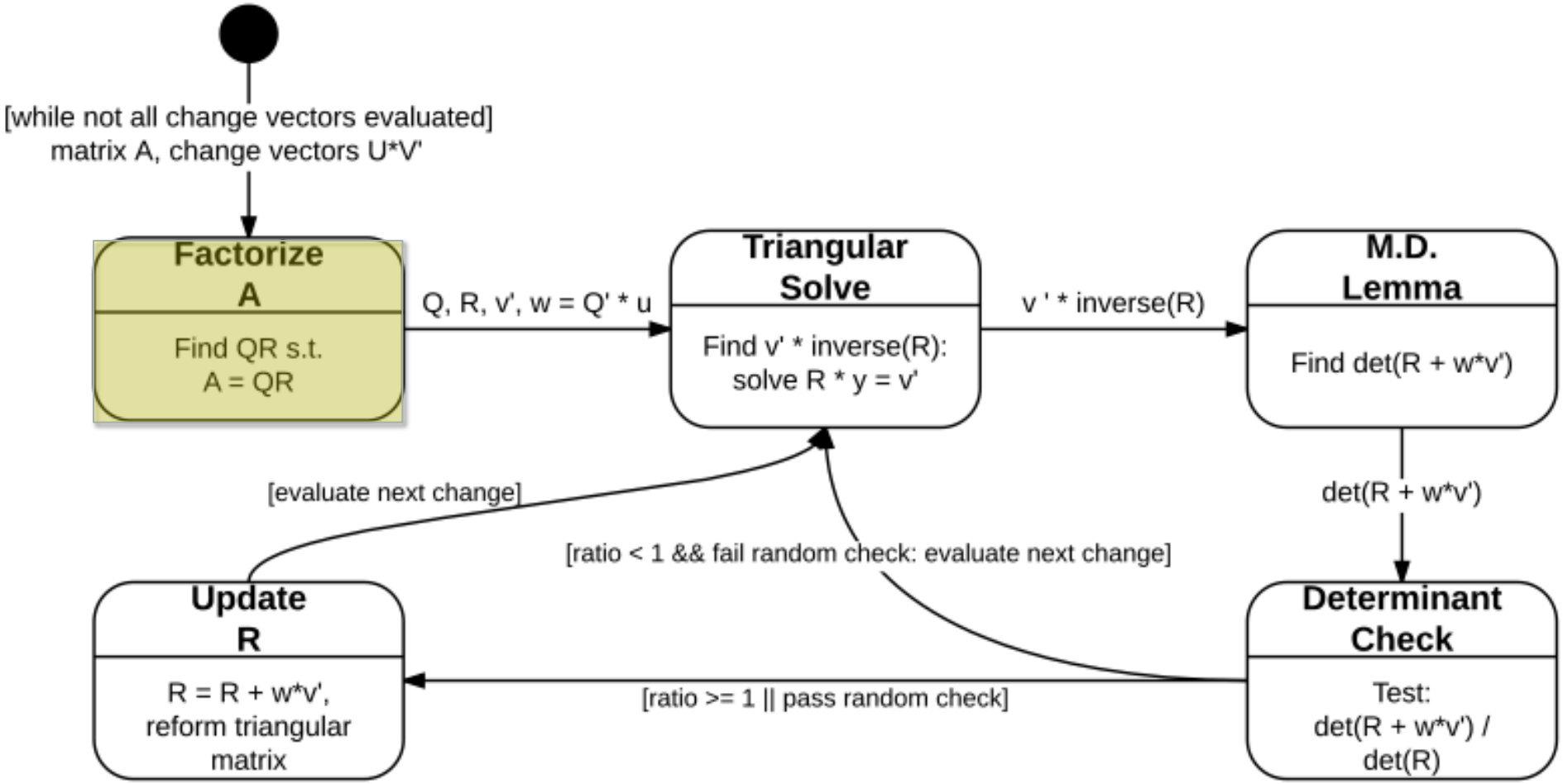
Current QMPC implementation



Proposed Implementation

- Using QR factorization
- Rank-k update
- Triangular solve

Proposed Implementation



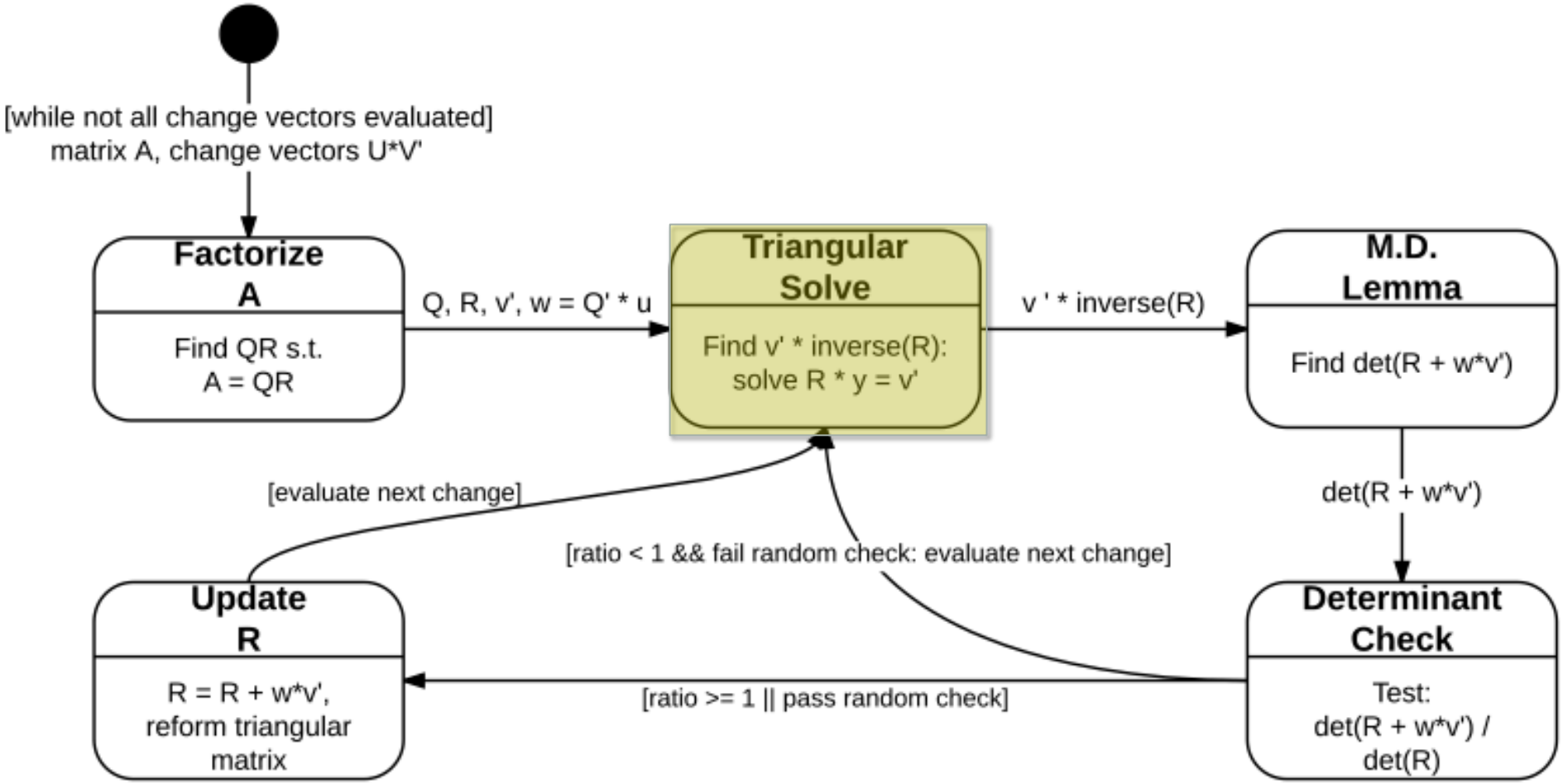
QR Decomposition

$$A = QR,$$

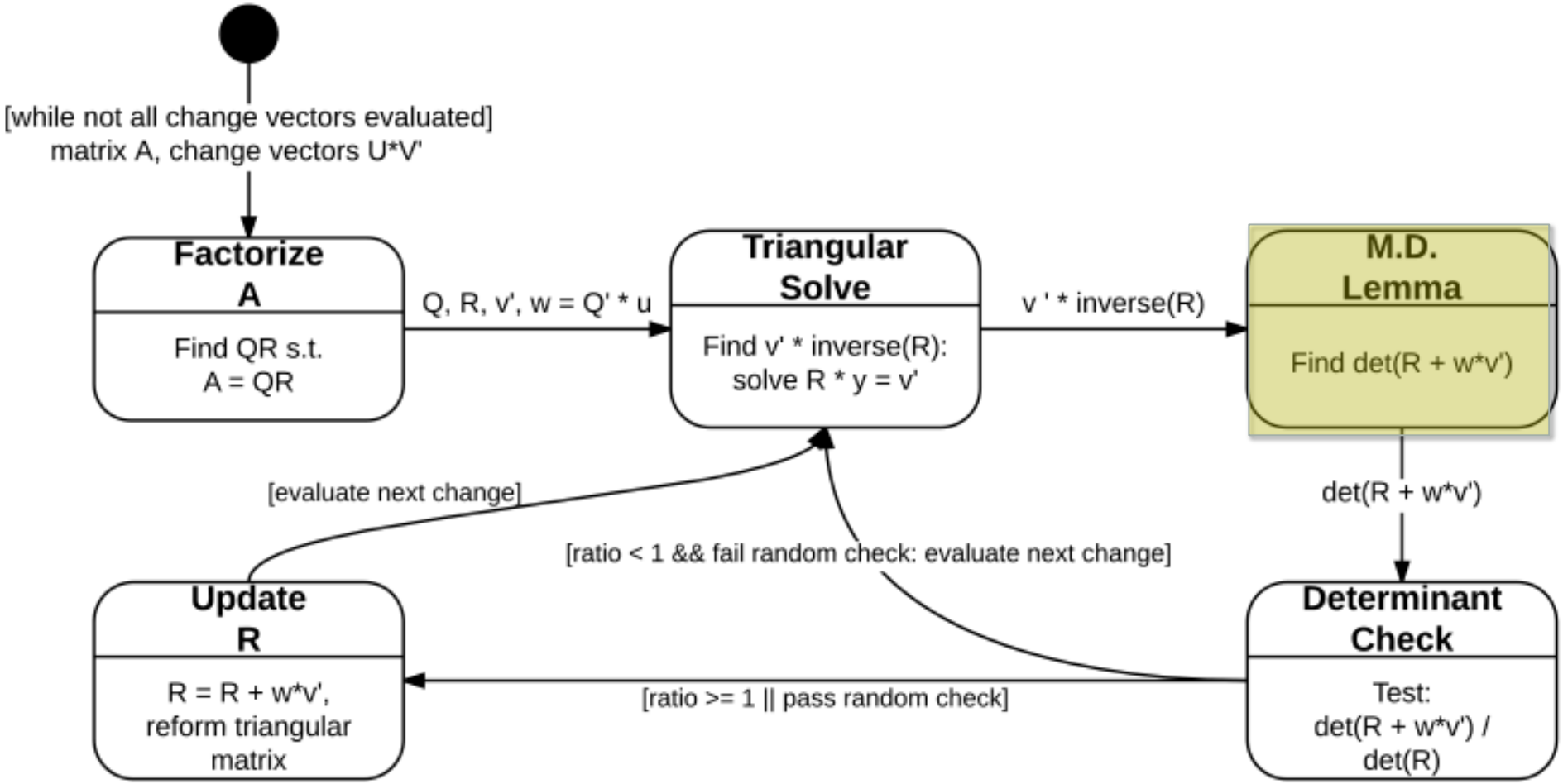
$$\begin{pmatrix} q_1 & \cdots & q_n \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & & 0 & r_{nn} \end{pmatrix}$$

Note that Q is orthonormal and R is upper triangular.

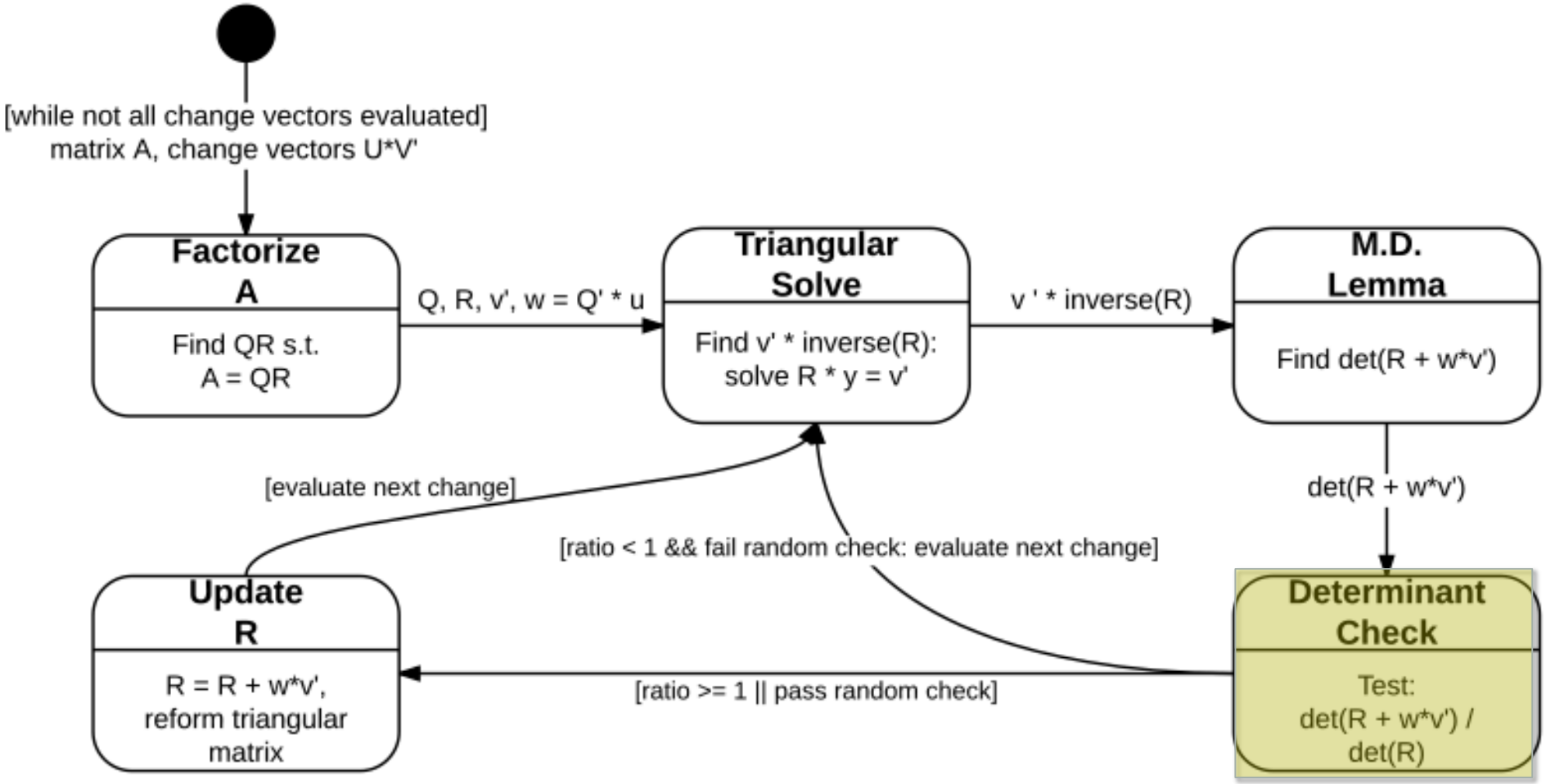
Proposed Implementation



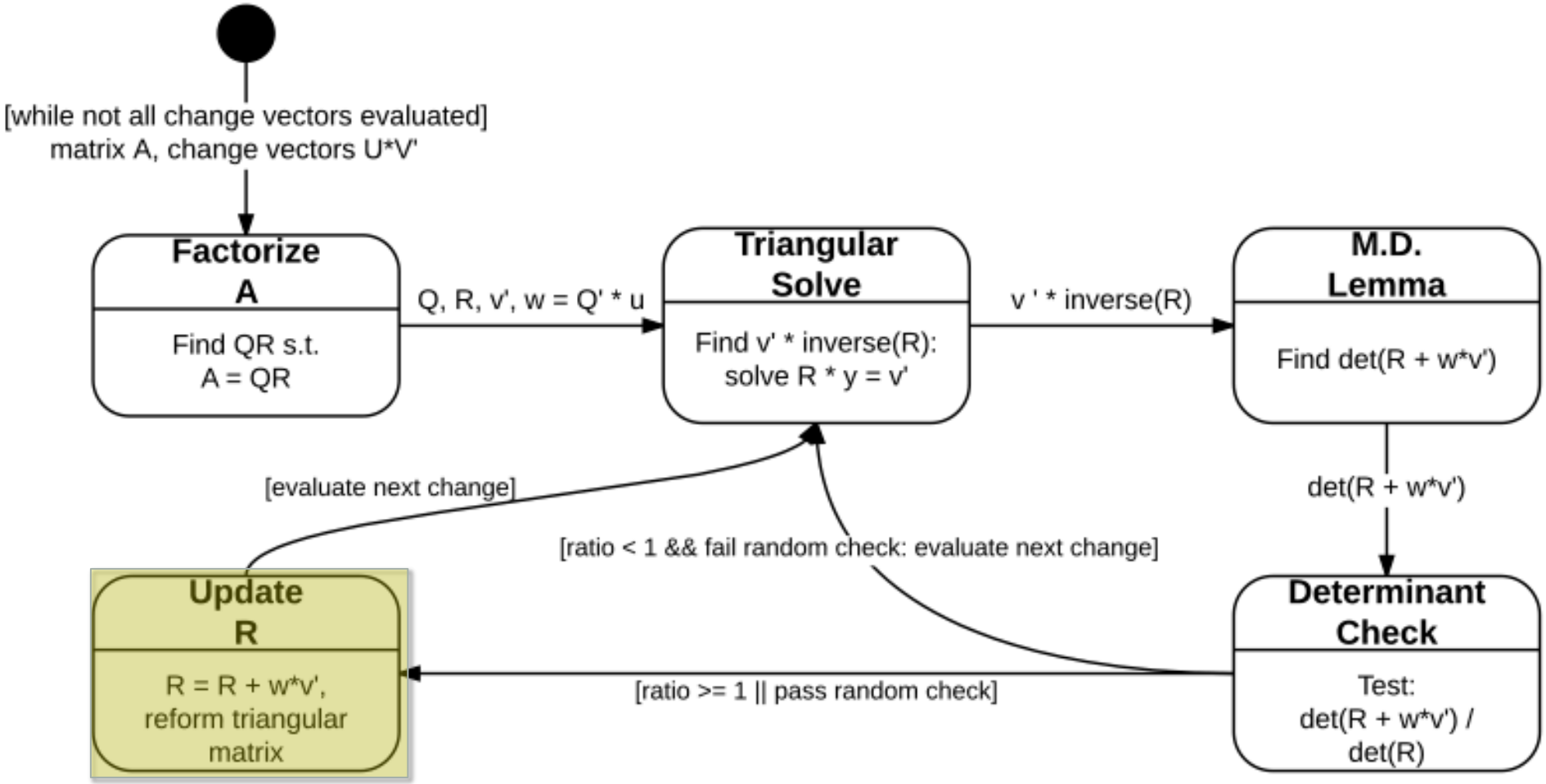
Proposed Implementation



Proposed Implementation



Proposed Implementation



Given's Rotation

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}^T \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}, \quad r = \sqrt{a^2 + b^2}.$$

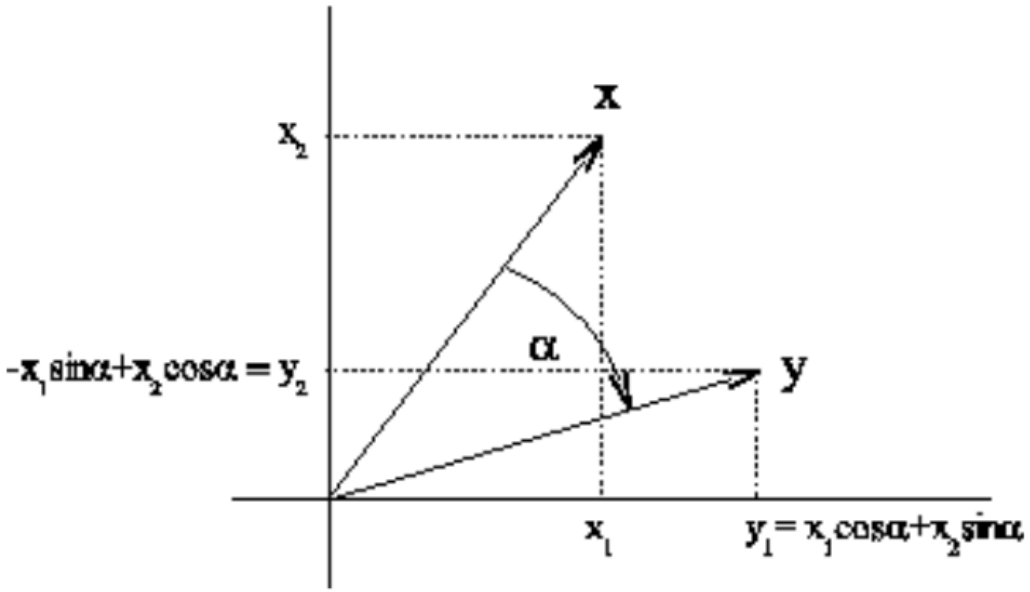
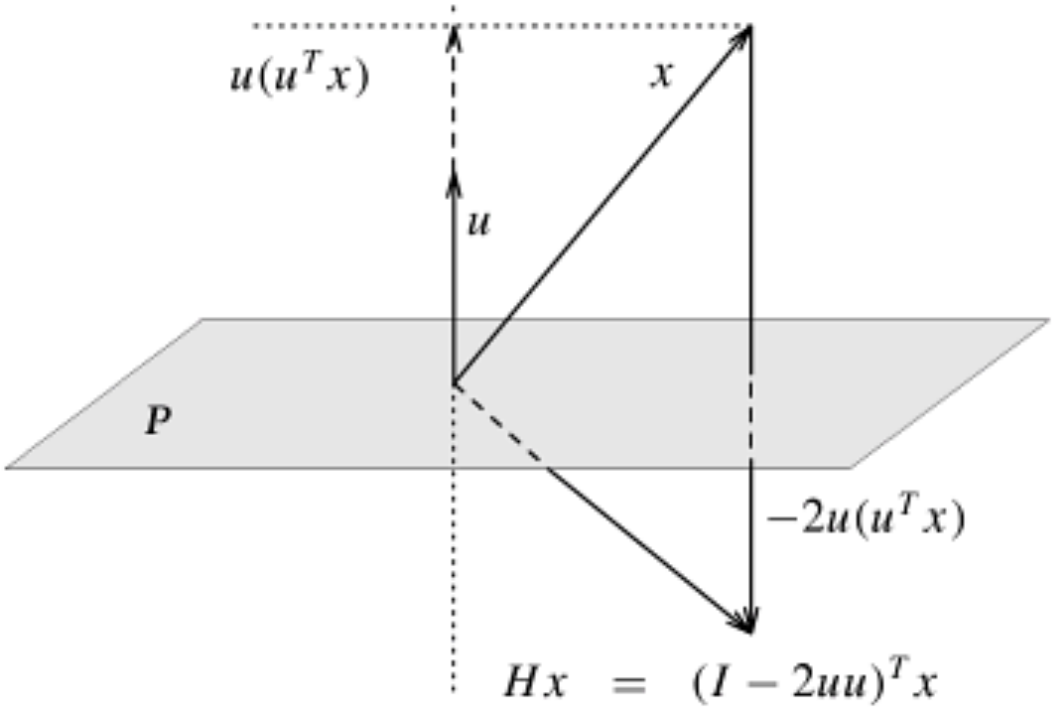


Figure 4.3: Rotation of x in a plane by an angle α

Householder Reflection



$$\begin{aligned} Hx &= (I - 2uu)^T x \\ &= x - 2u(u^T x) \end{aligned}$$

Completed Work

- Design our algorithms, background work, version control repository
- Serial MATLAB implementations for rank-1 and rank-k updates
- Serial C/LAPACK/BLAS for rank-1 update
- cuBLAS for rank-1 update

In Progress

- Serial C/LAPACK/BLAS for rank k
- C++/Cuda kernels for rank-1/rank k update

Acknowledgements

- We greatly appreciate help from our mentors:
  - Dr. Ed D'Azevedo from ORNL
  - Dr. Ying Wai Li from ORNL
  - Dr. Kwai Wong from UTK