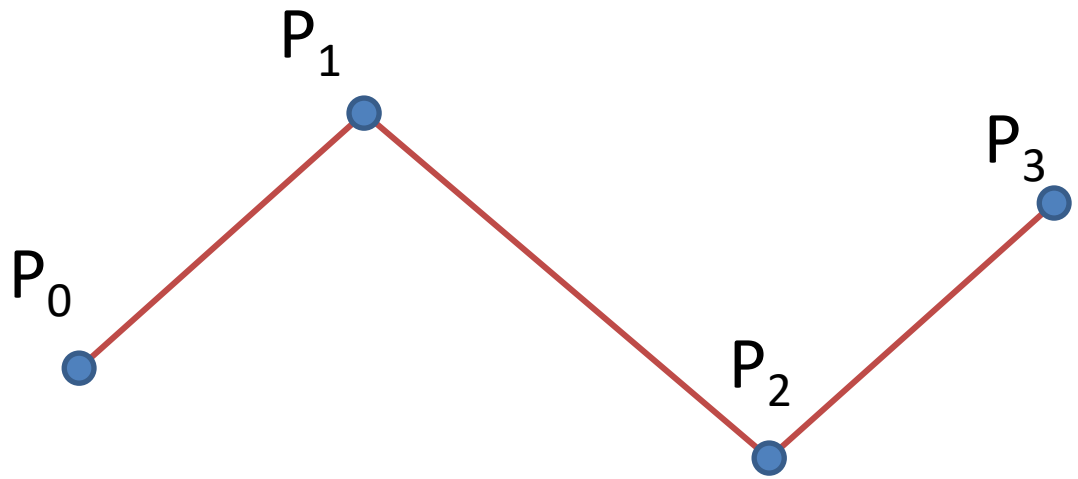
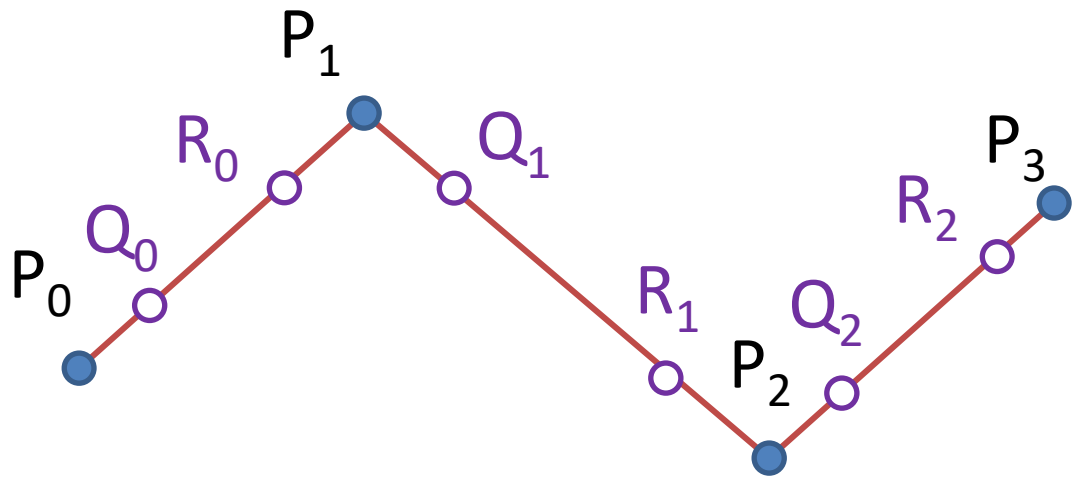
- Chaikin's scheme

(step 0)



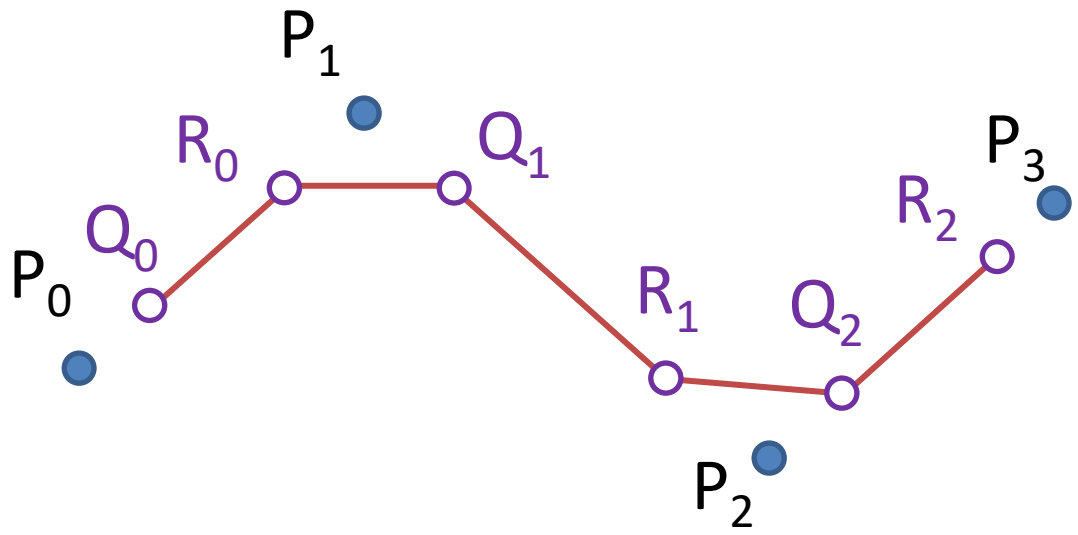
- Chaikin's scheme

Refine 1 Step 1



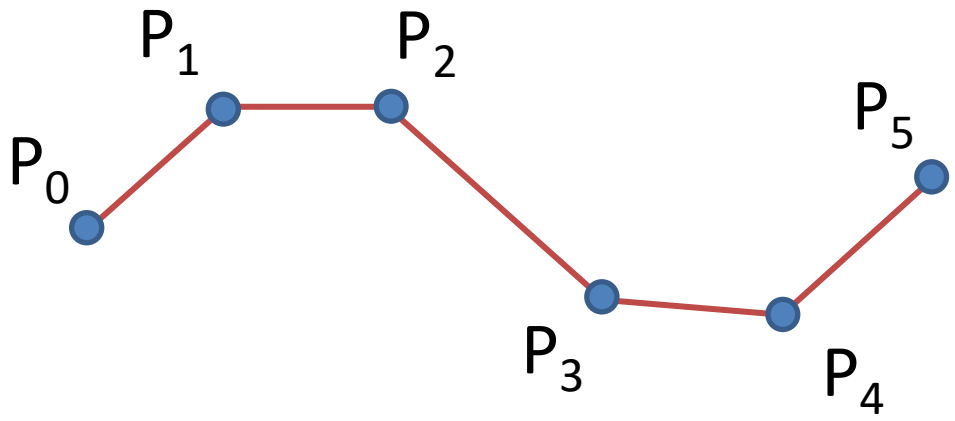
- Chaikin's scheme

Refine 1 Step 2



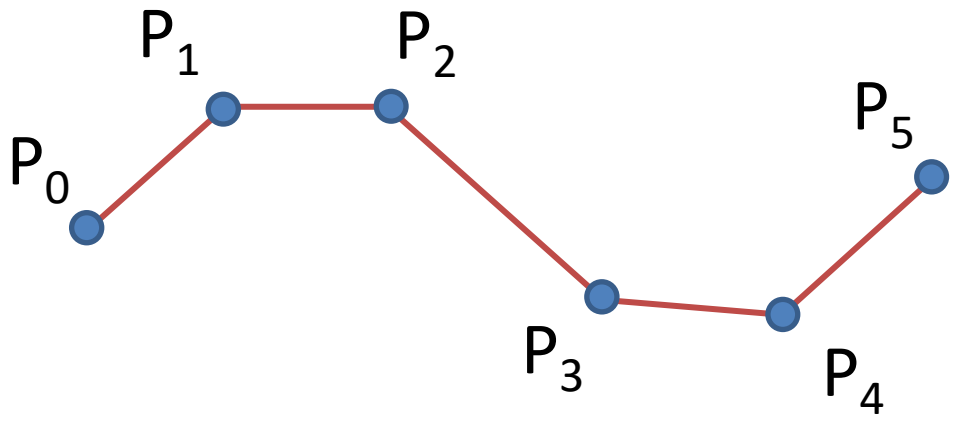
- Chaikin's scheme

Refine 1 Step 3



... restart at step 1 to increase refinement

- Chakin's scheme

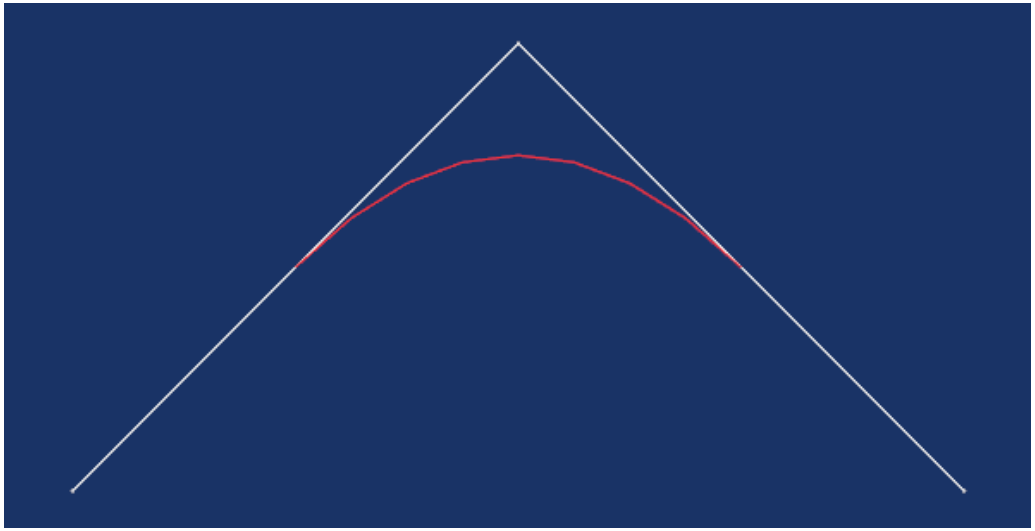


4 CPs → 6 CPs

General case: (n) CPs → (2n-2) CPs

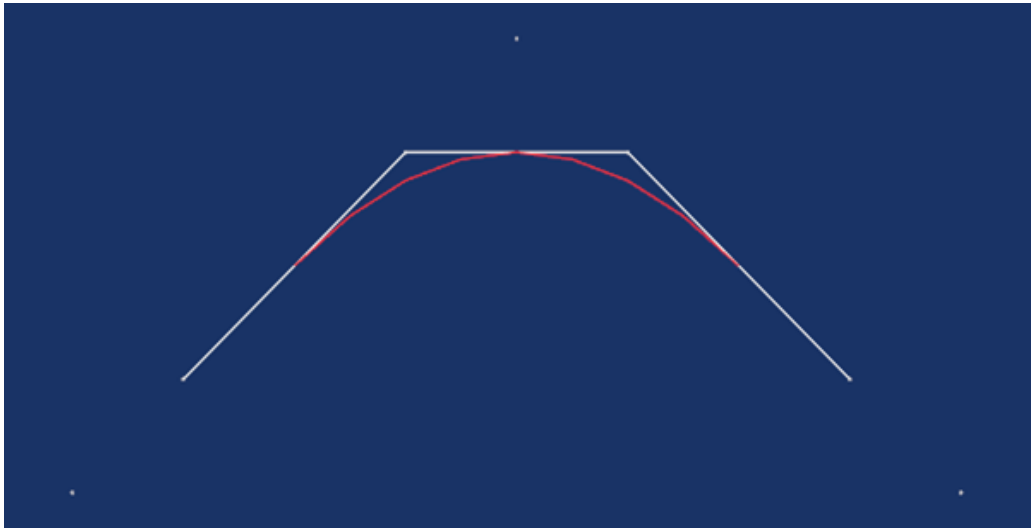
- Chakin’s scheme converges towards a quadratic B-spline with uniform knots.

Step 0



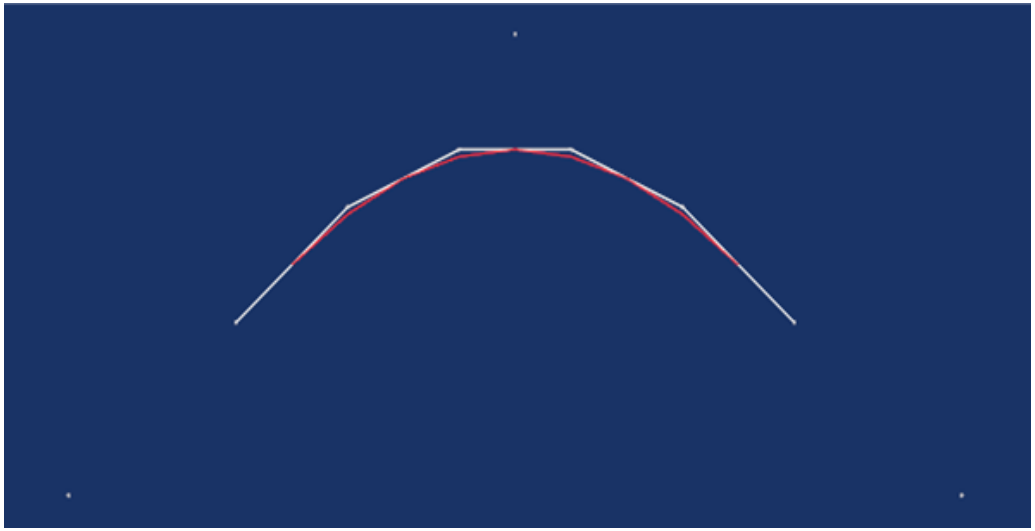
- Chakin’s scheme converges towards a quadratic B-spline with uniform knots.

Refine 1



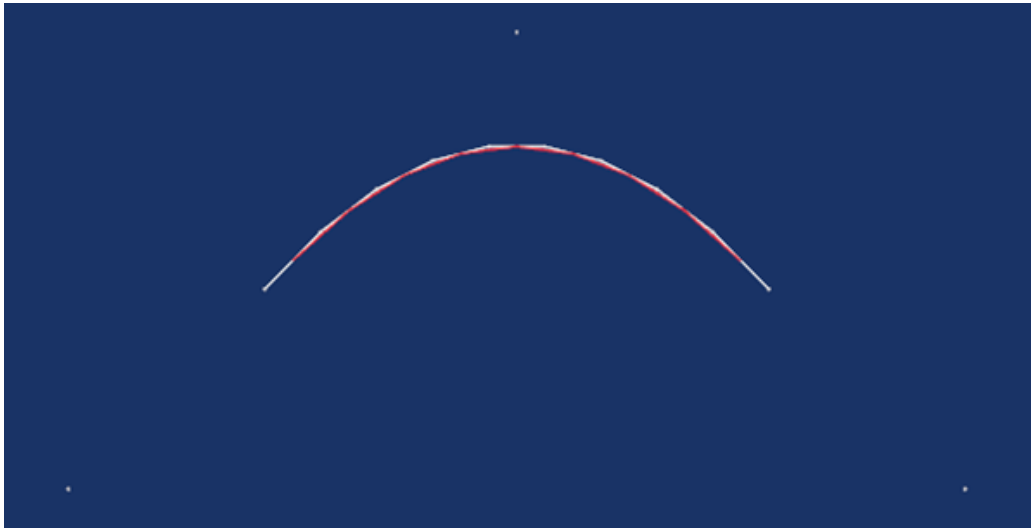
- Chakin’s scheme converges towards a quadratic B-spline with uniform knots.

Refine 2



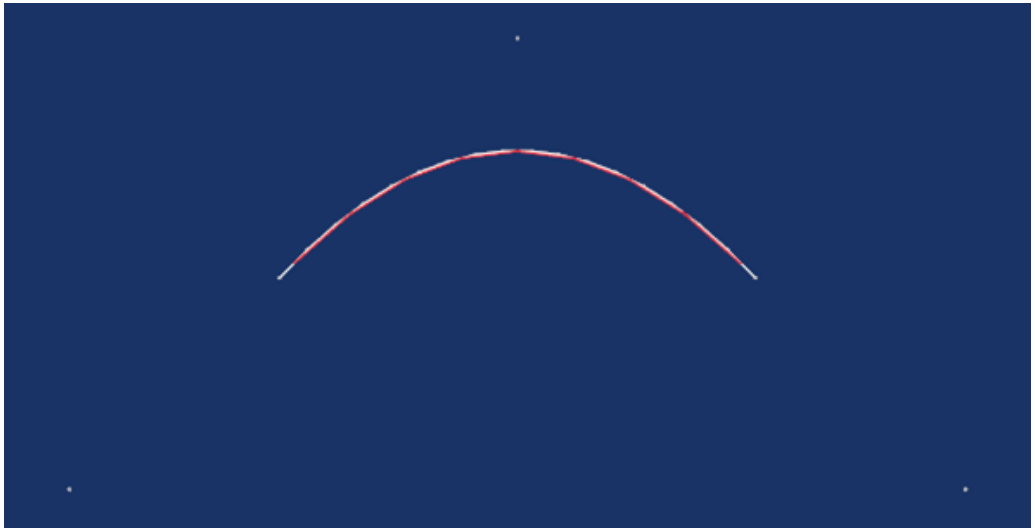
- Chakin’s scheme converges towards a quadratic B-spline with uniform knots.

Refine 3



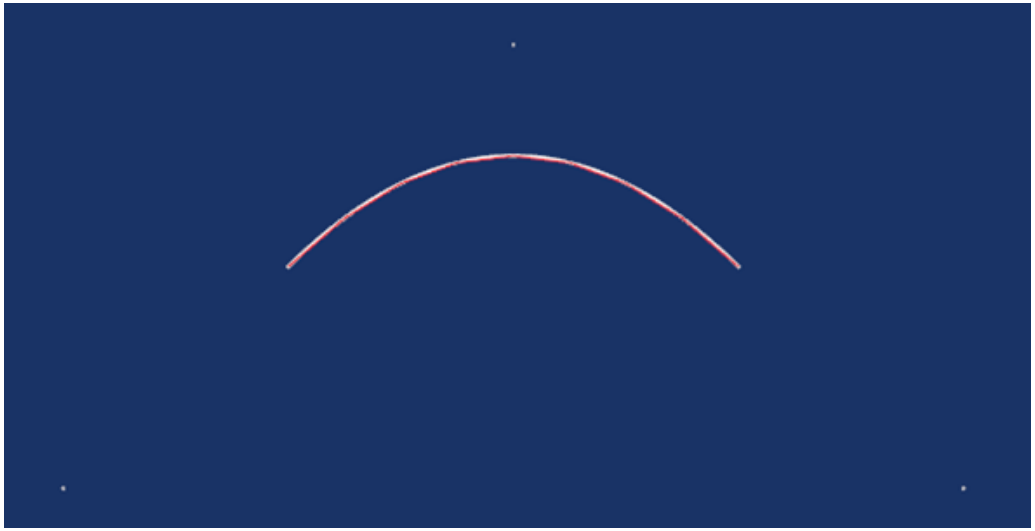
- Chakin’s scheme converges towards a quadratic B-spline with uniform knots.

Refine 4



- Chakin’s scheme converges towards a quadratic B-spline with uniform knots.

Refine 10

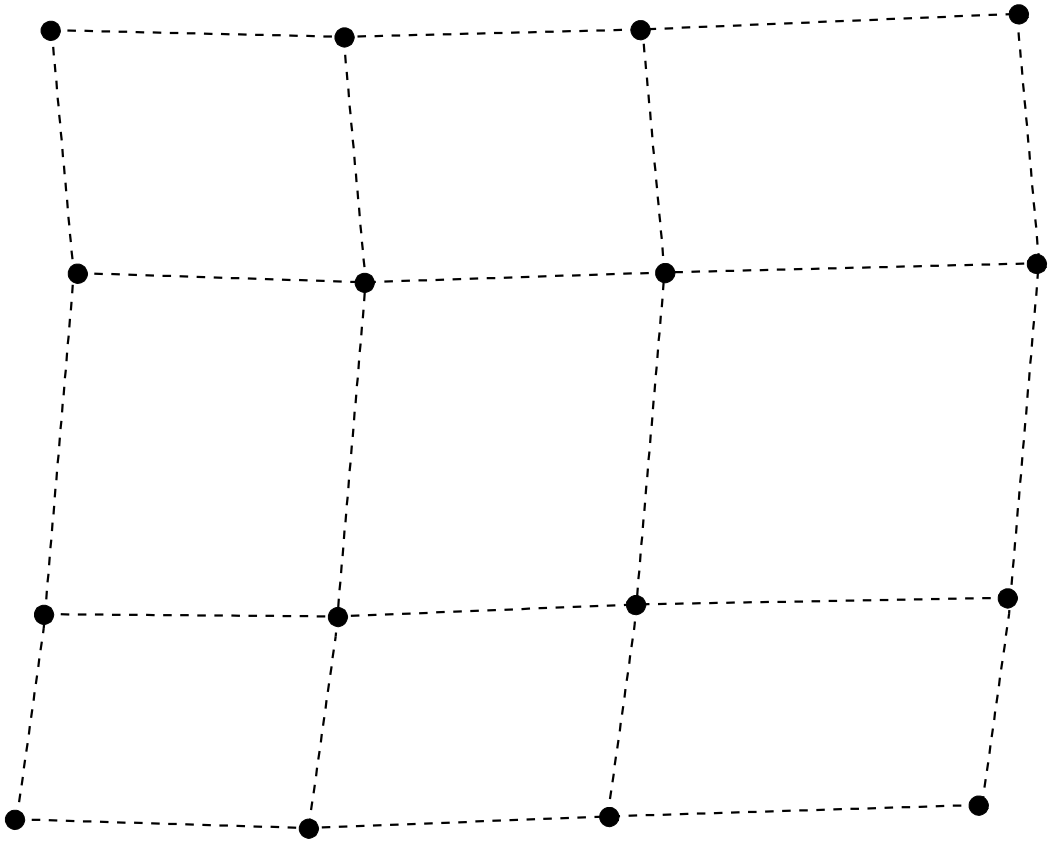


Doo-Sabin's scheme

- Surface subdivision scheme
- Generalize Chaïkin's scheme to meshes with quadrangles.
- (Can be further generalized to any mesh.)

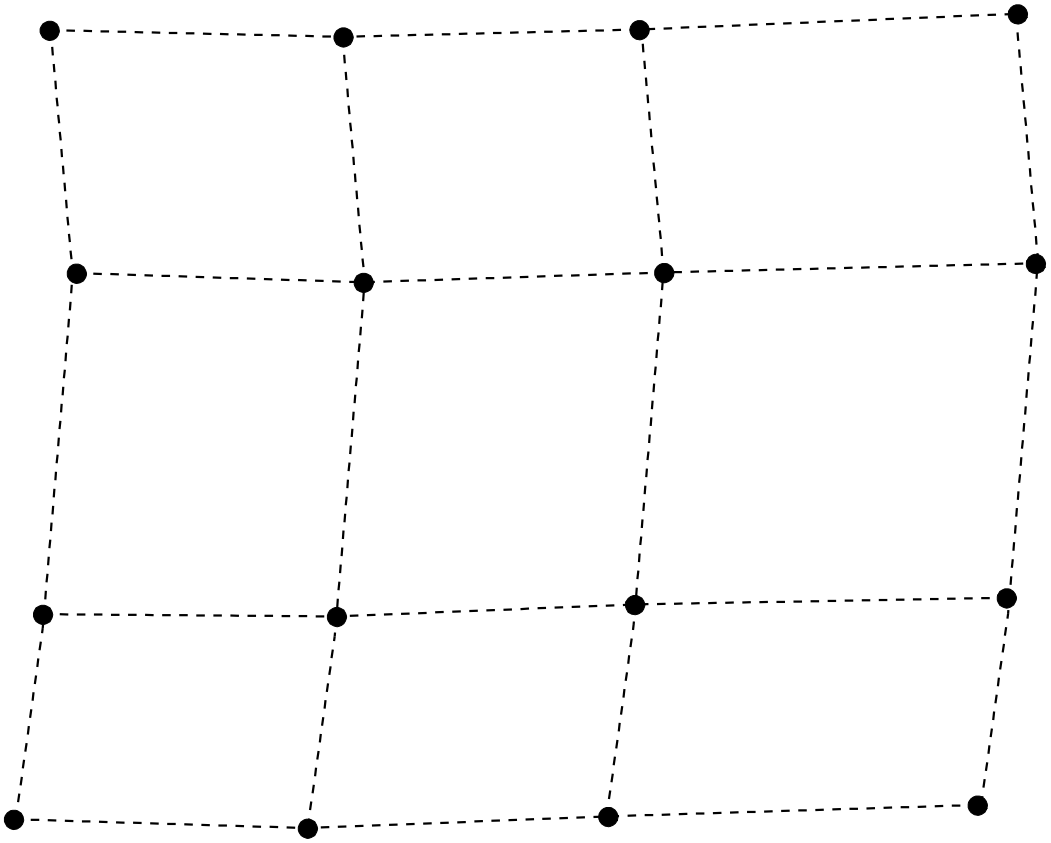
Doo-Sabin's scheme

- General principle



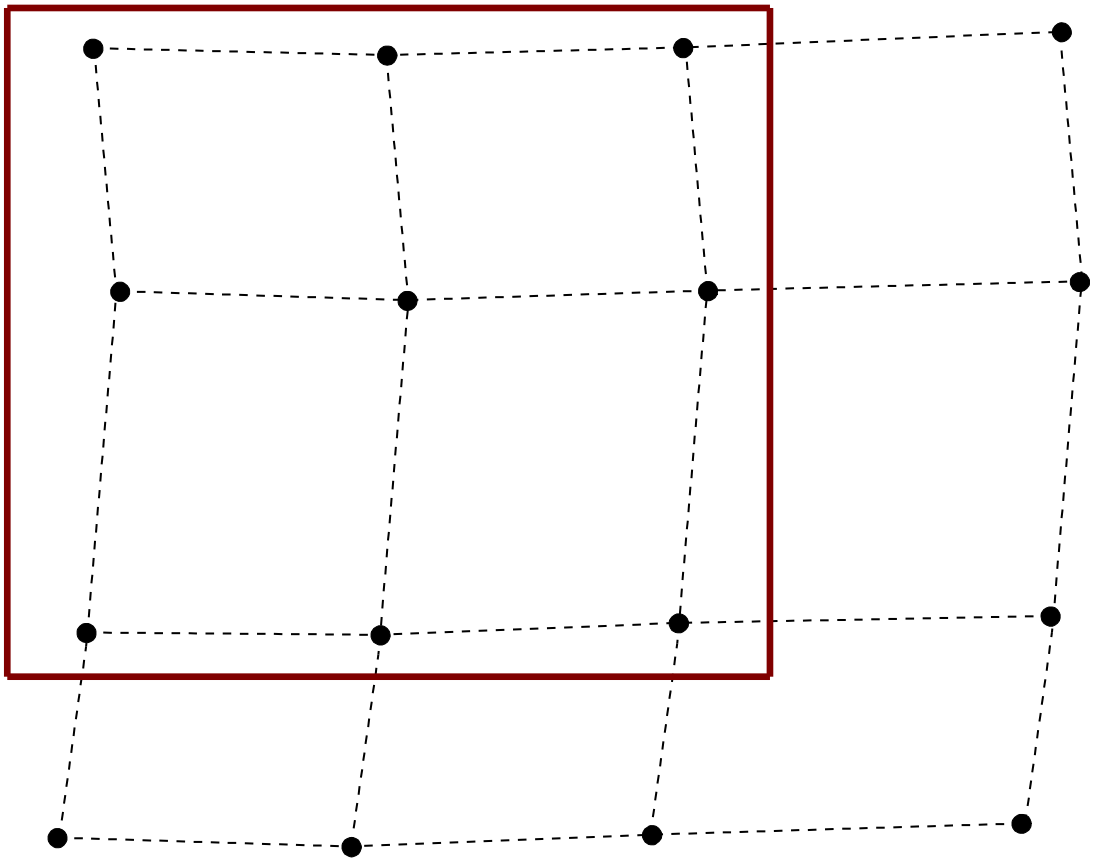
Doo-Sabin's scheme

- Start from a mesh



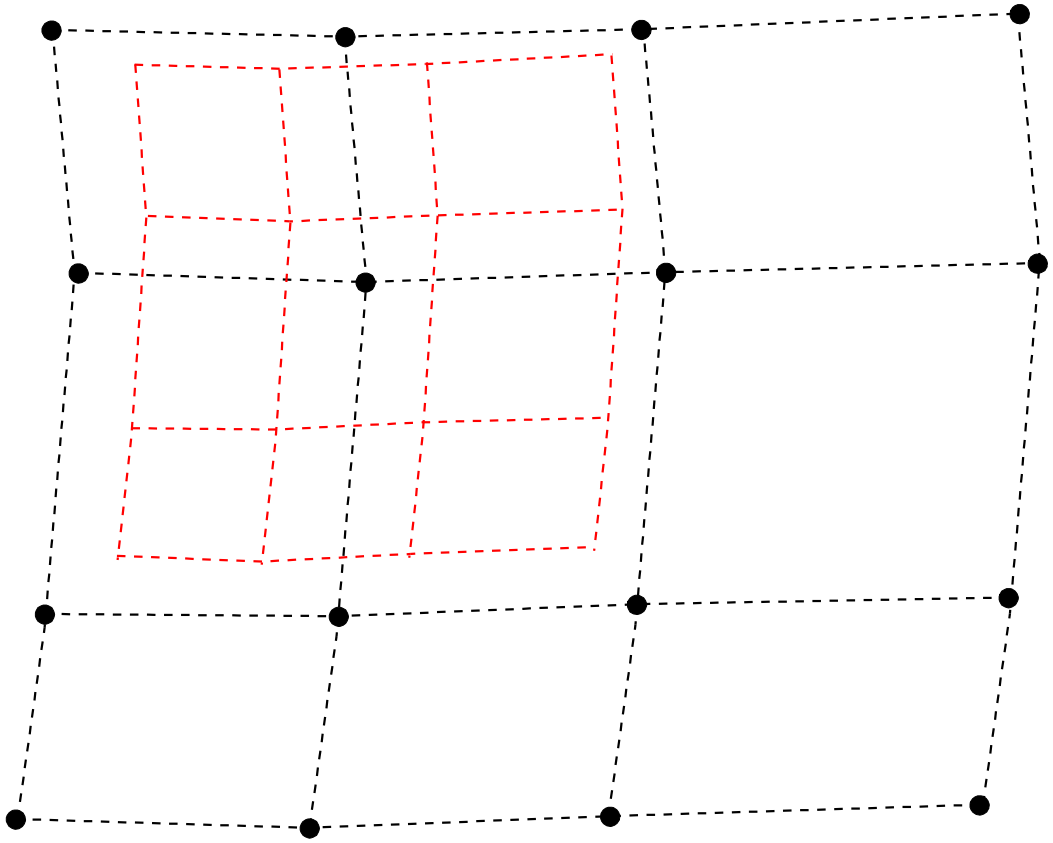
Doo-Sabin's scheme

- On a 3x3 « patche »



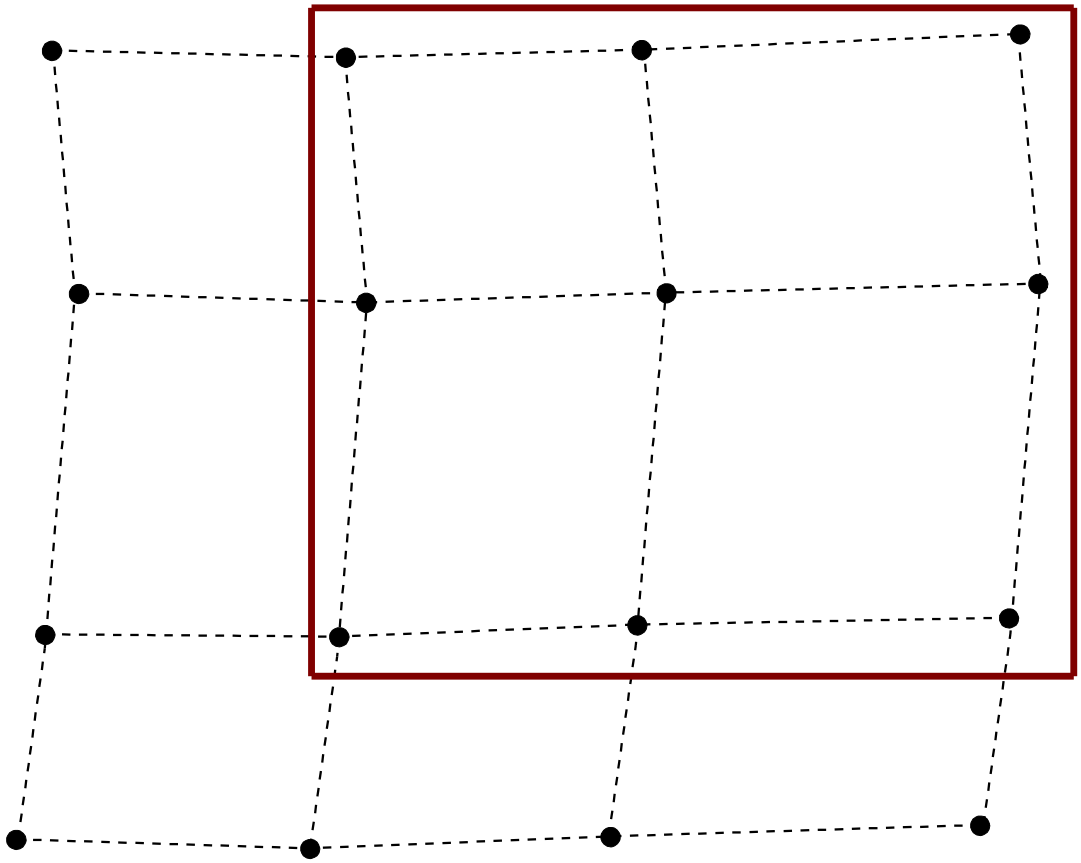
Doo-Sabin's scheme

- Compute a refined 4x4 patch*

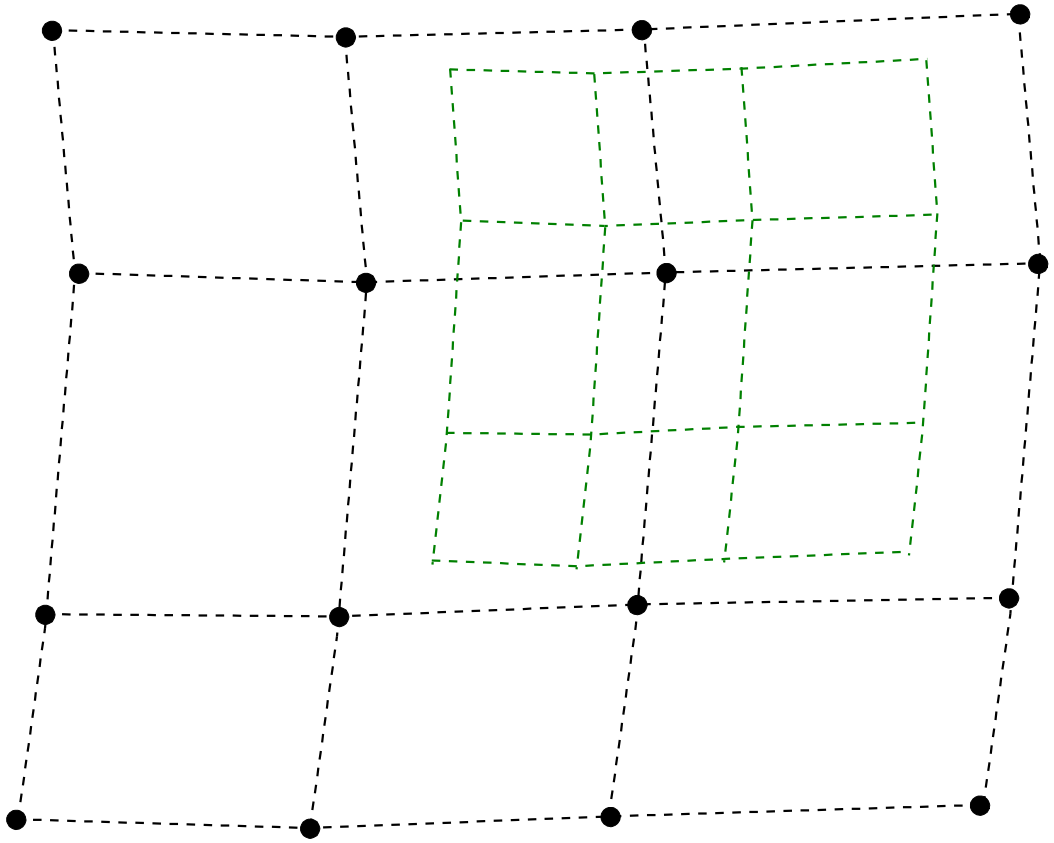


Doo-Sabin's scheme

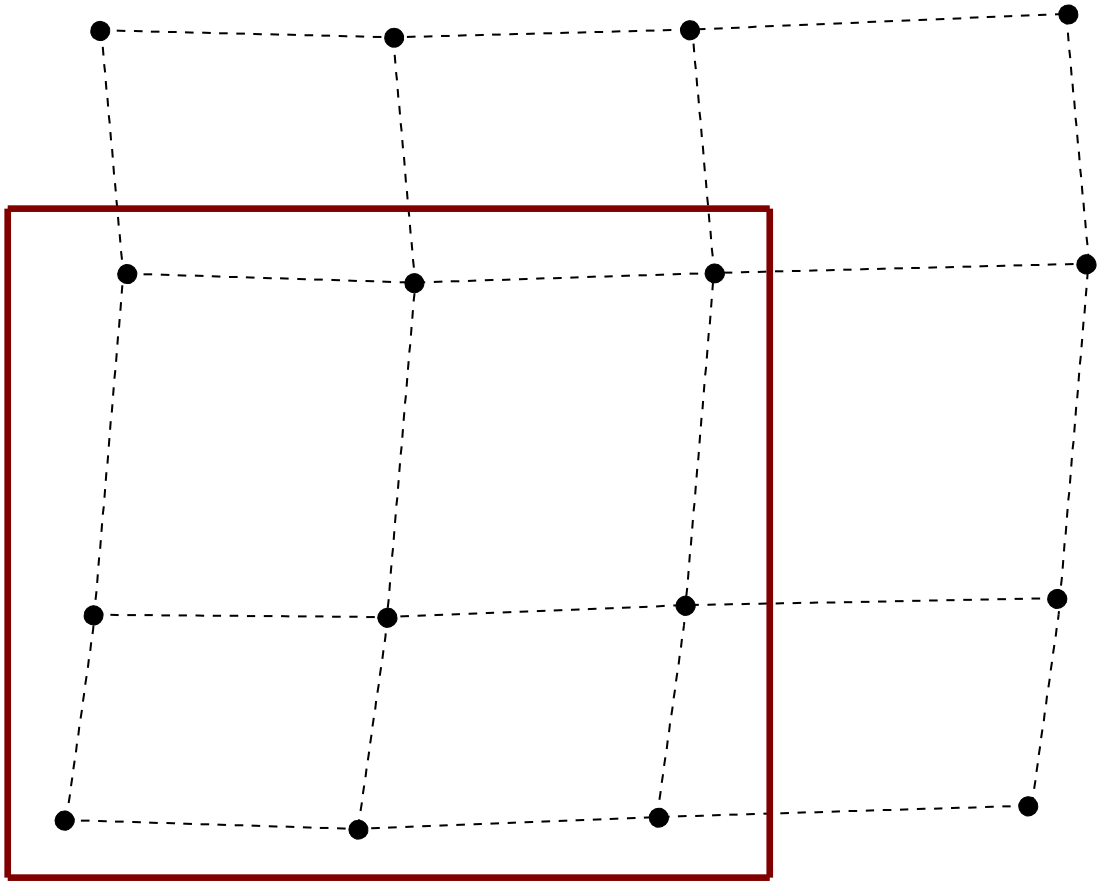
- Loop over all 3x3 « patches »



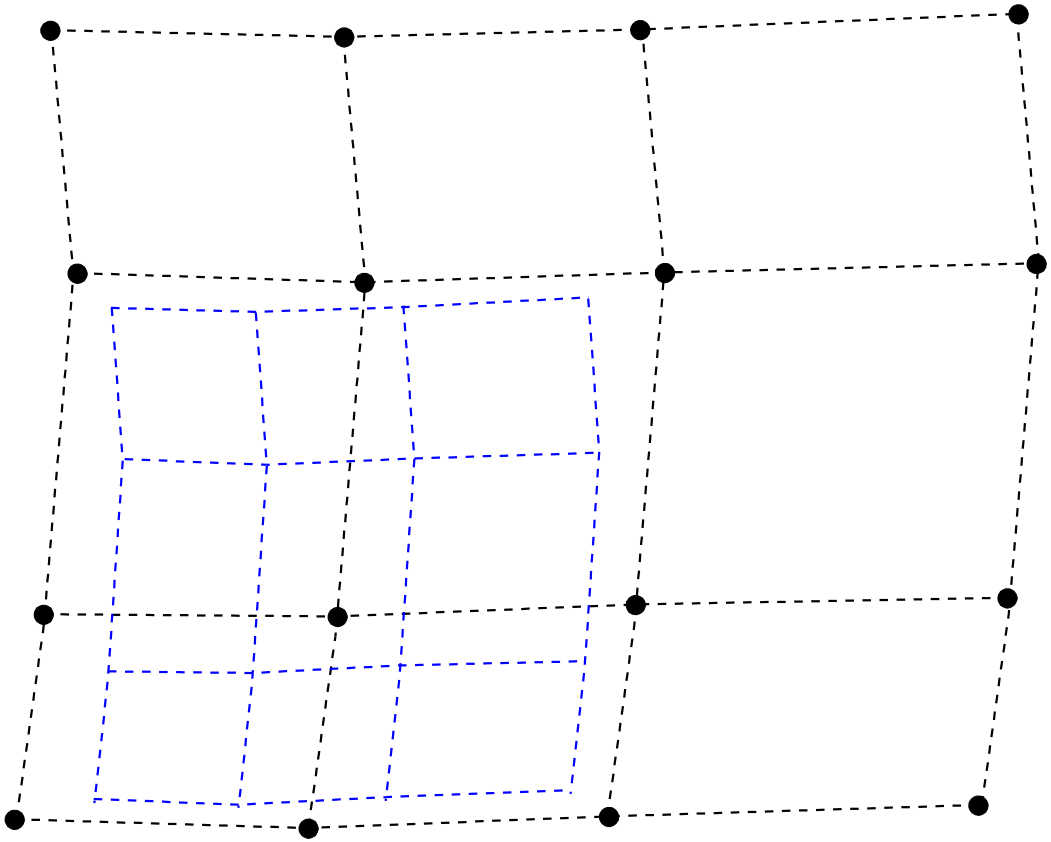
Doo-Sabin’s scheme



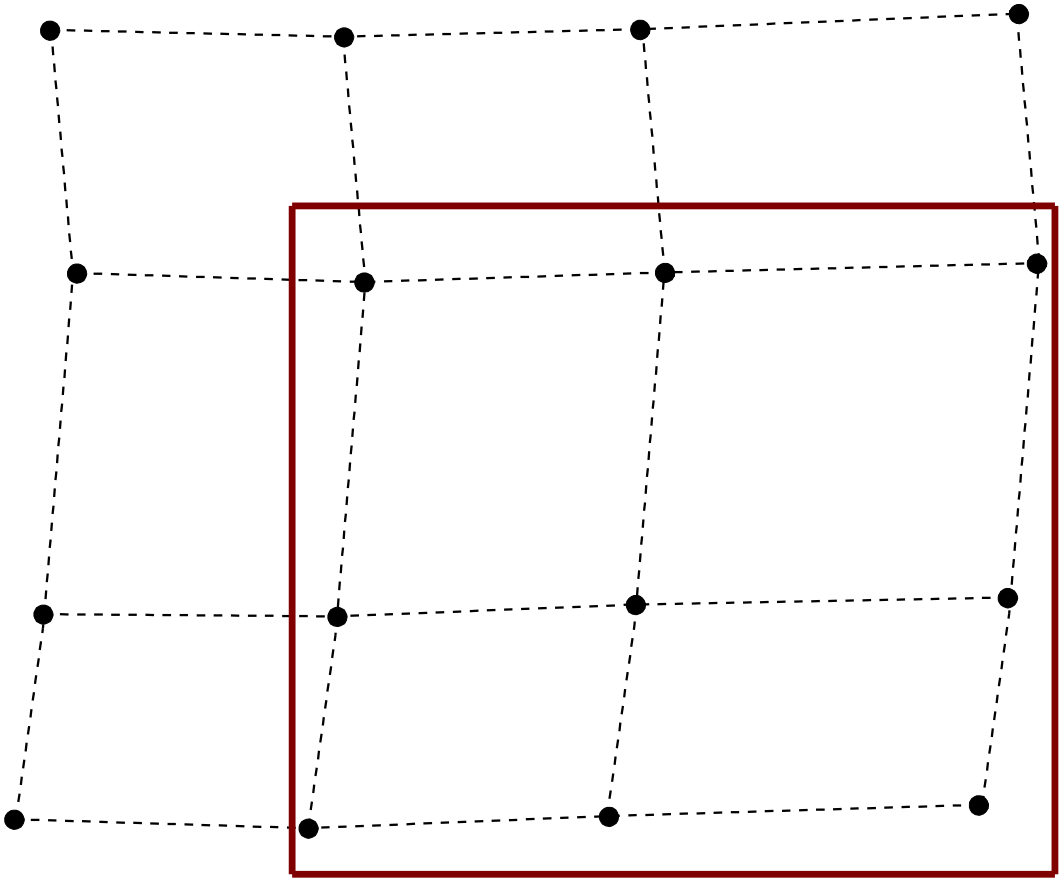
Doo-Sabin's scheme



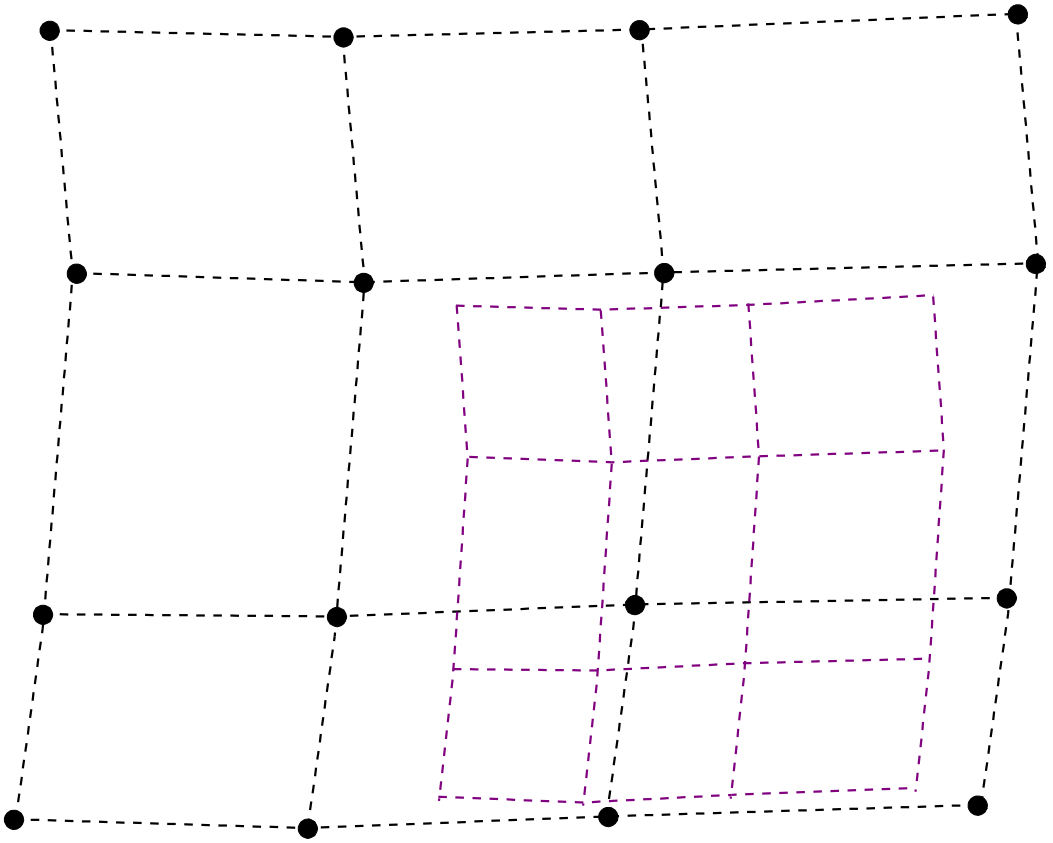
Doo-Sabin’s scheme



Doo-Sabin's scheme

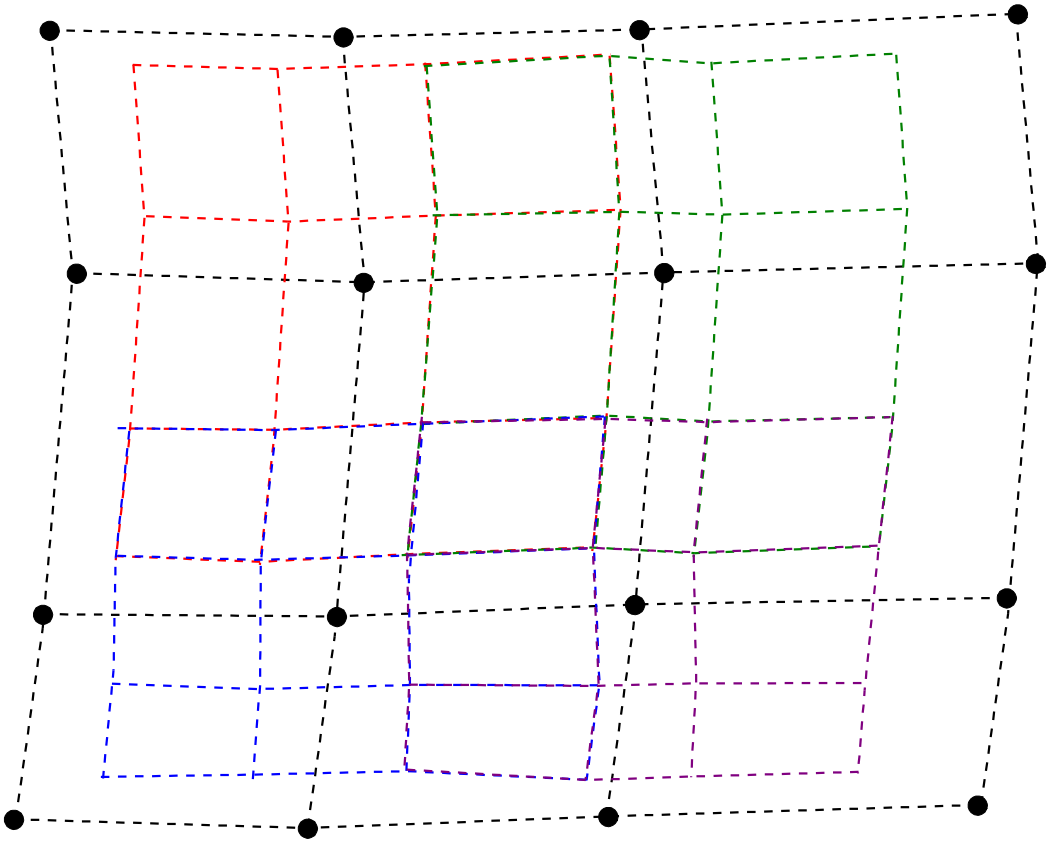


Doo-Sabin’s scheme



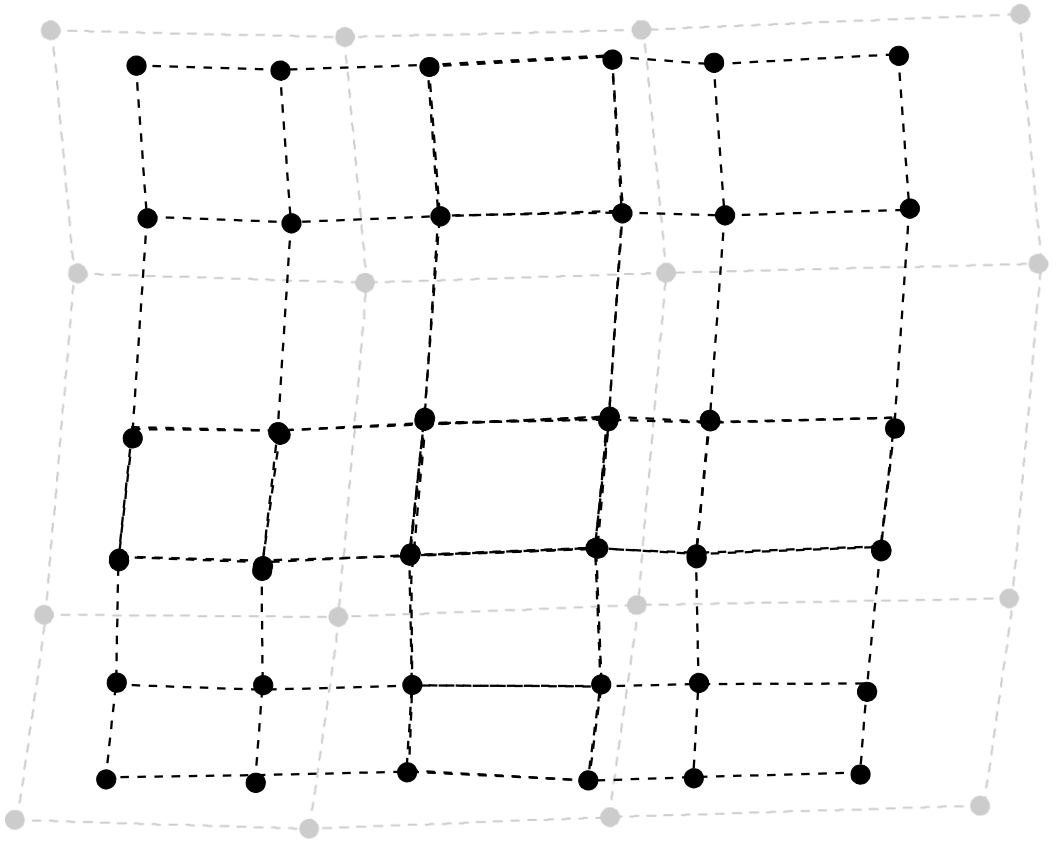
Doo-Sabin's scheme

- Assemble the refined 4x4 patches together

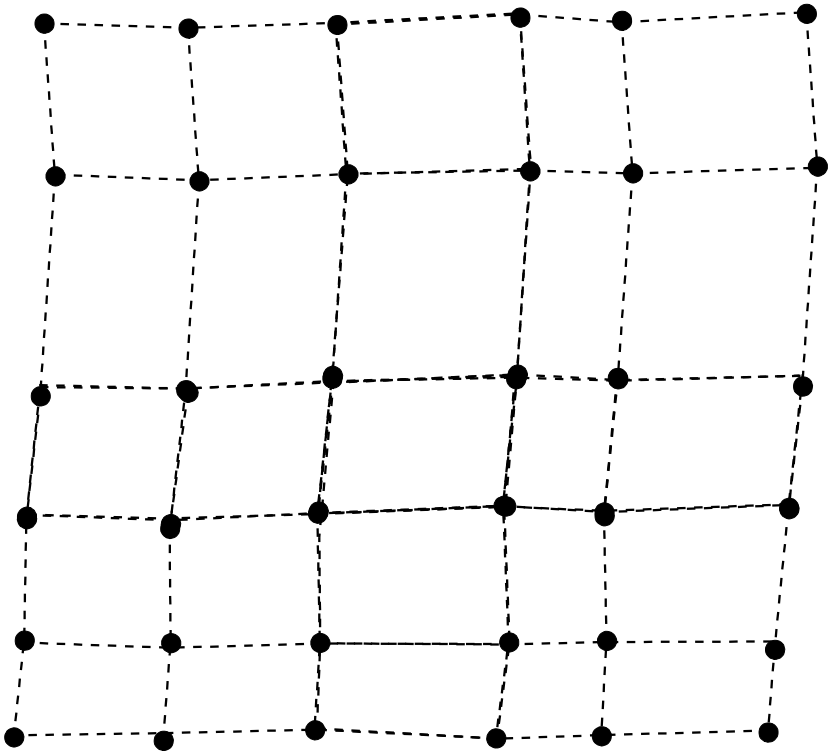


Doo-Sabin's scheme

- Assemble the refined 4x4 patches together

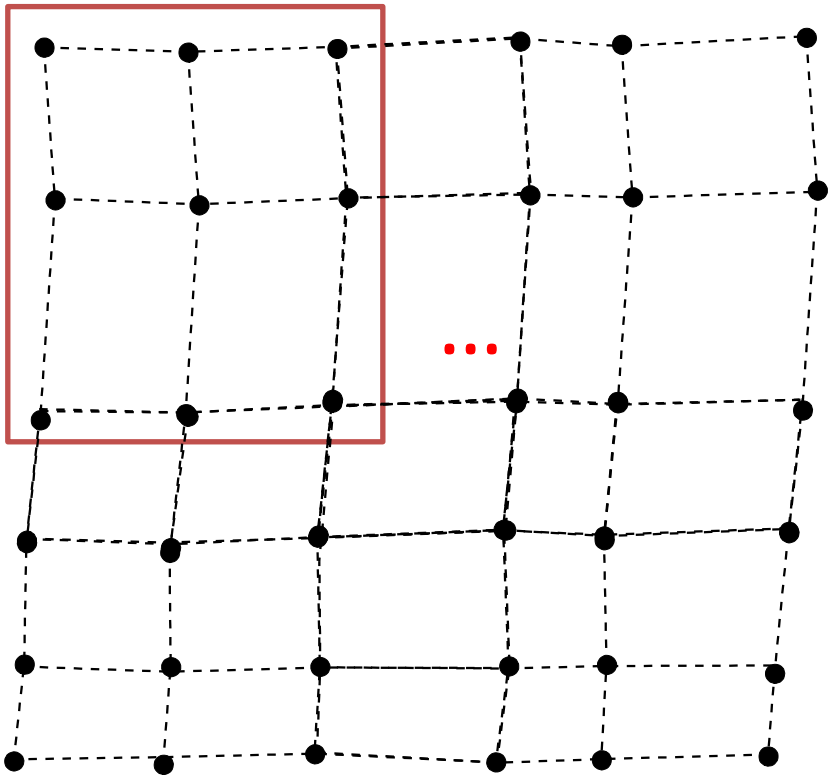


Doo-Sabin’s scheme



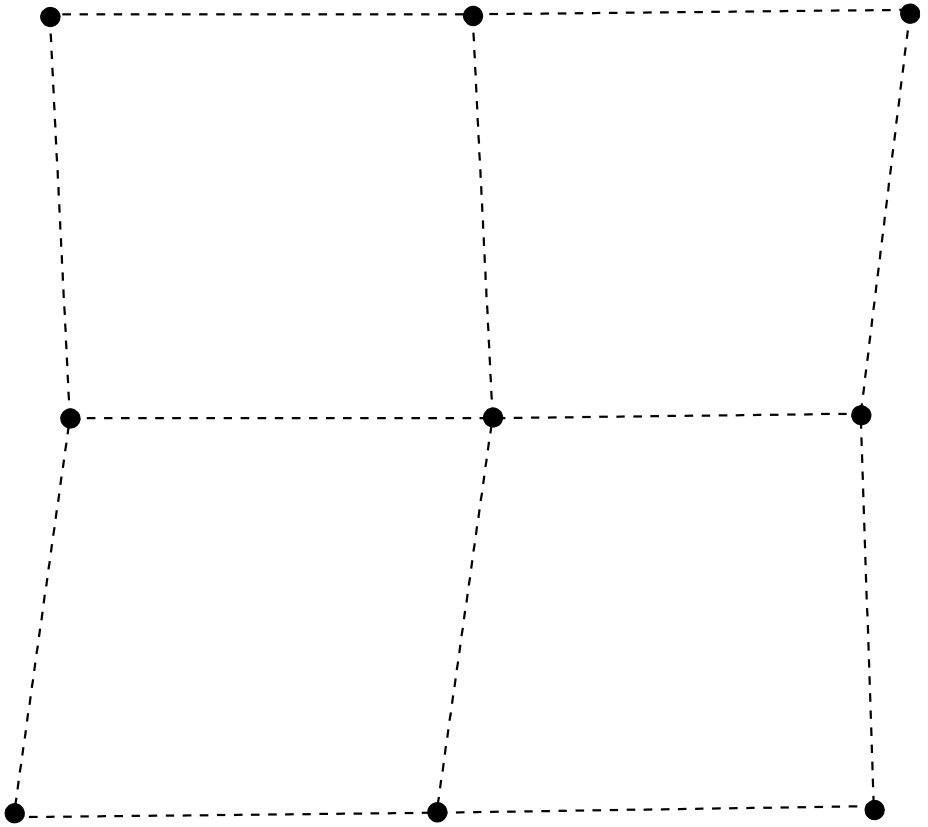
Doo-Sabin's scheme

- Repeat refinement : loop over the 3x3 patches and redo the preceeding operations…



Doo-Sabin's scheme

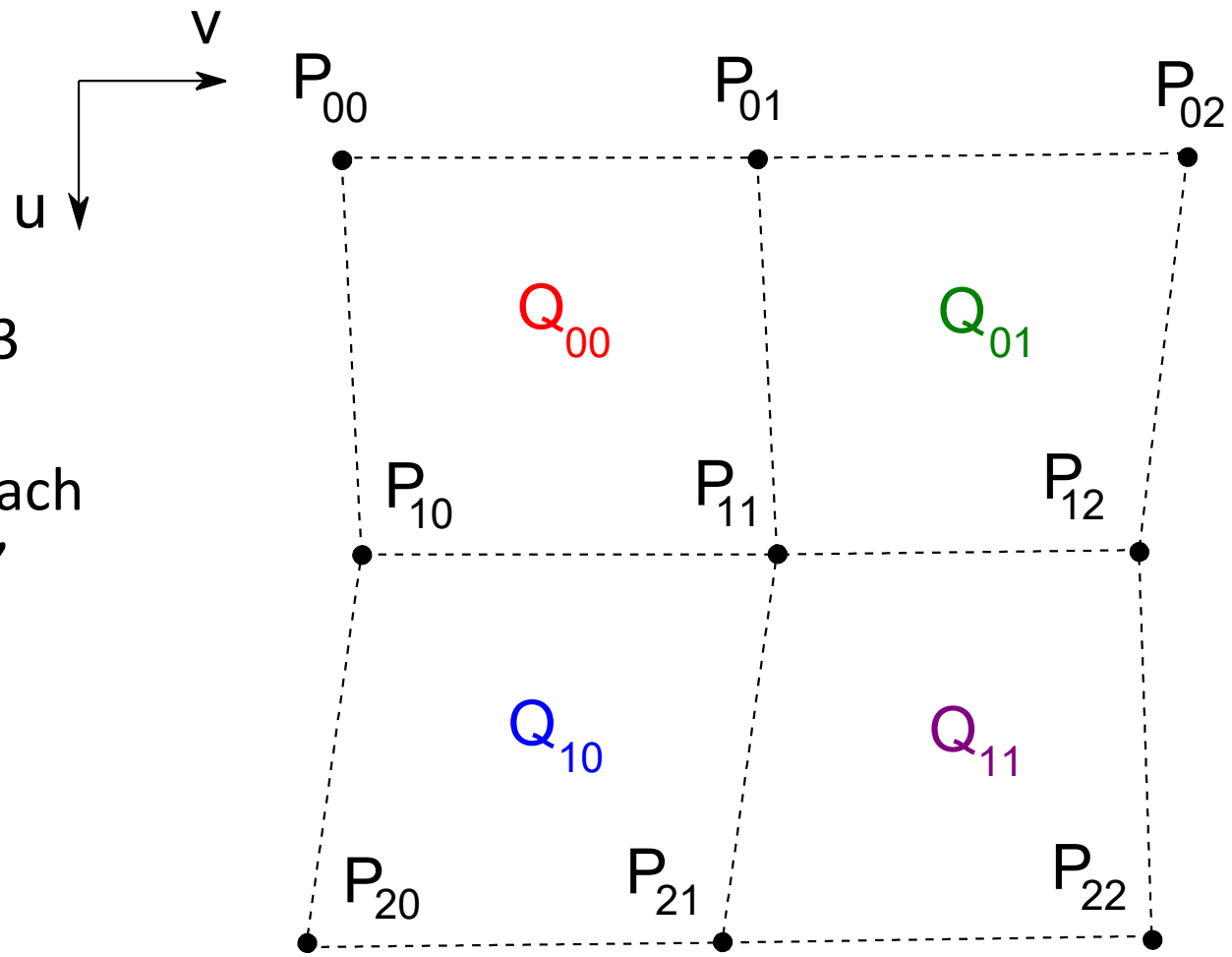
- Computing a refined 4x4 patch from a 3x3 patch:
  - Extract the wanted 3x3 patch*



*This step is already implemented in the provided code.

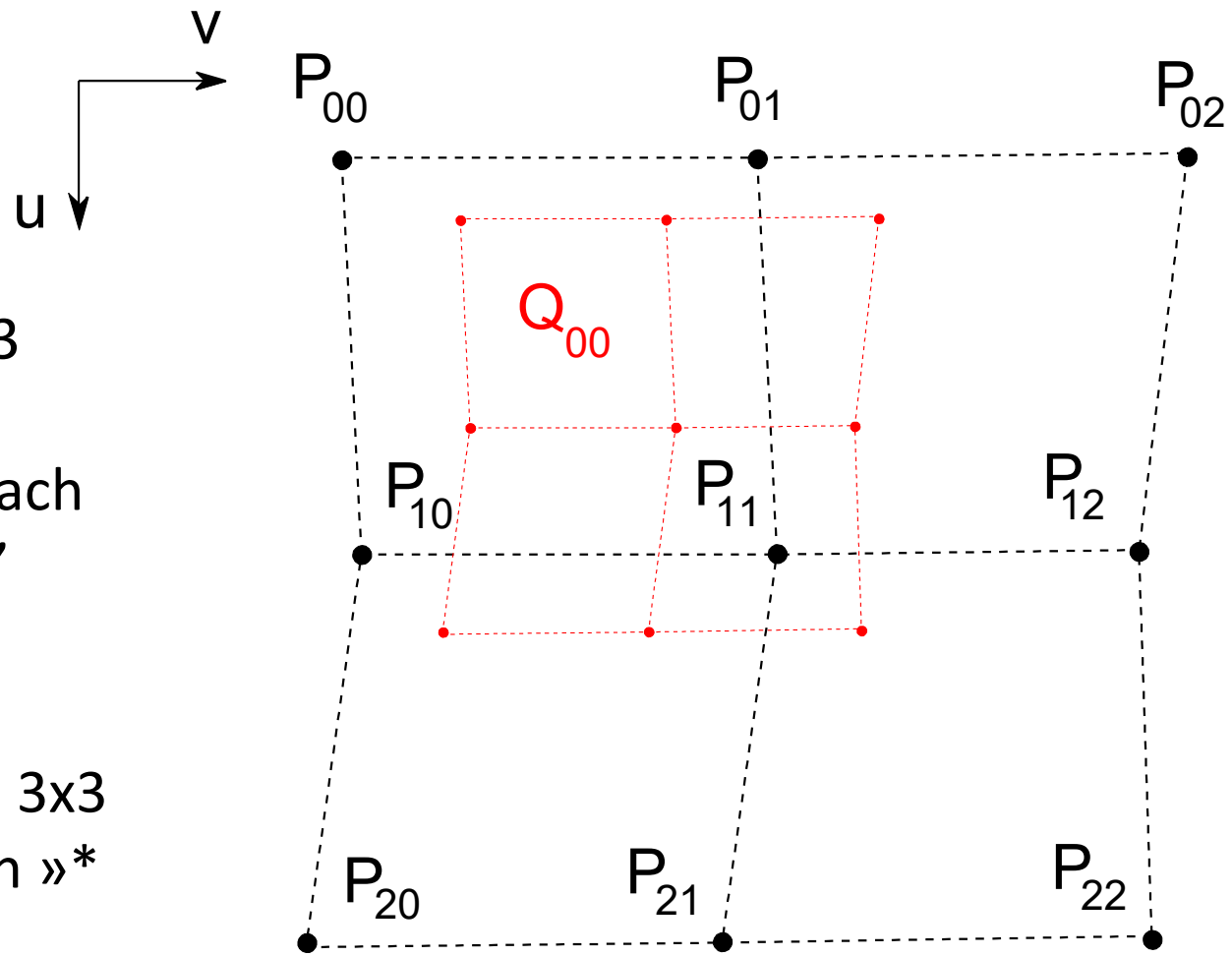
Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:
  - Extract the wanted 3x3 patch
  - Consider each "quadrant"



Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:

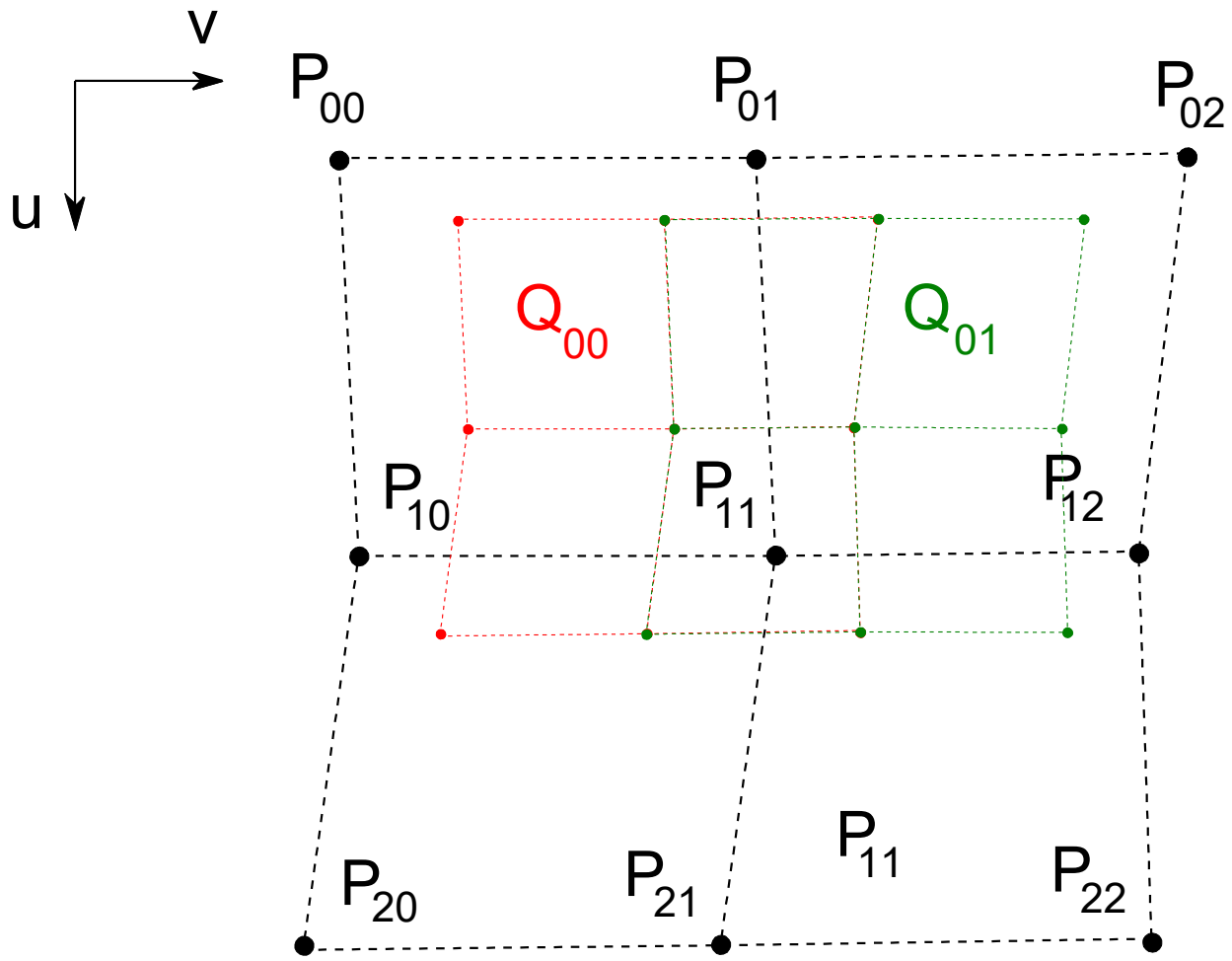


  - Extract the wanted 3x3 patch
  - Consider each "quadrant"
  - For each quadrant, Compute a 3x3 « sub-patch »*

*You must implement this step.

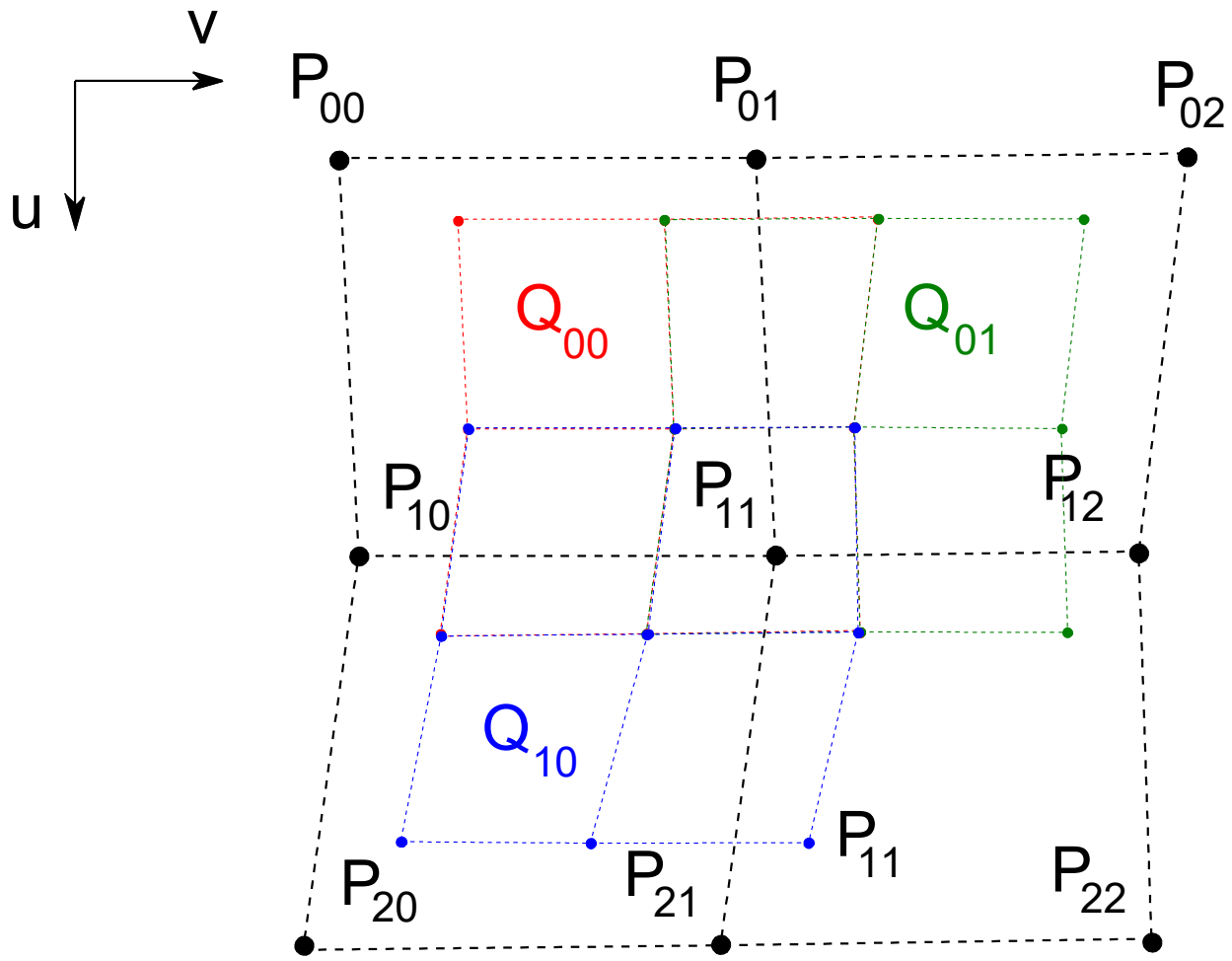
Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:



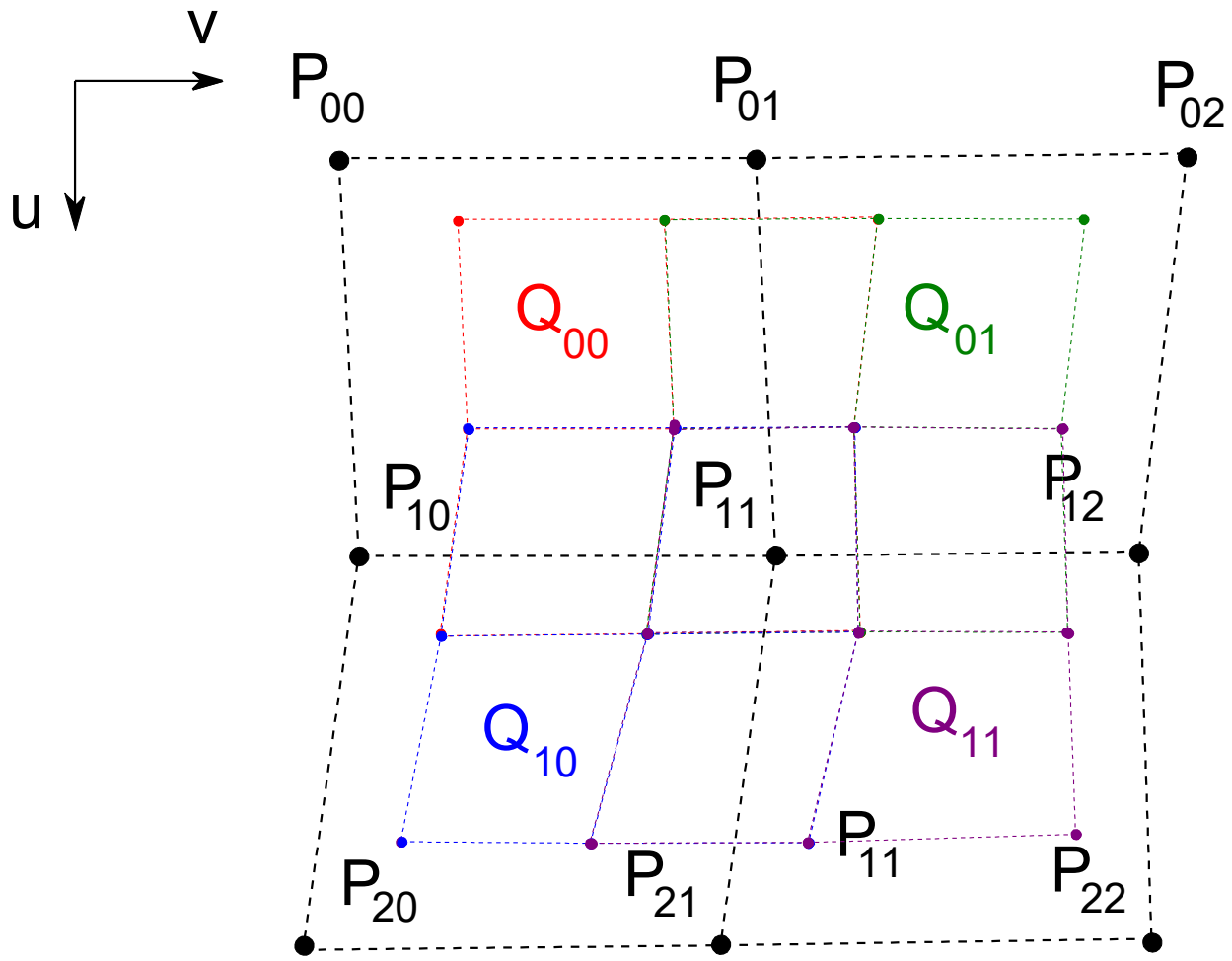
Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:



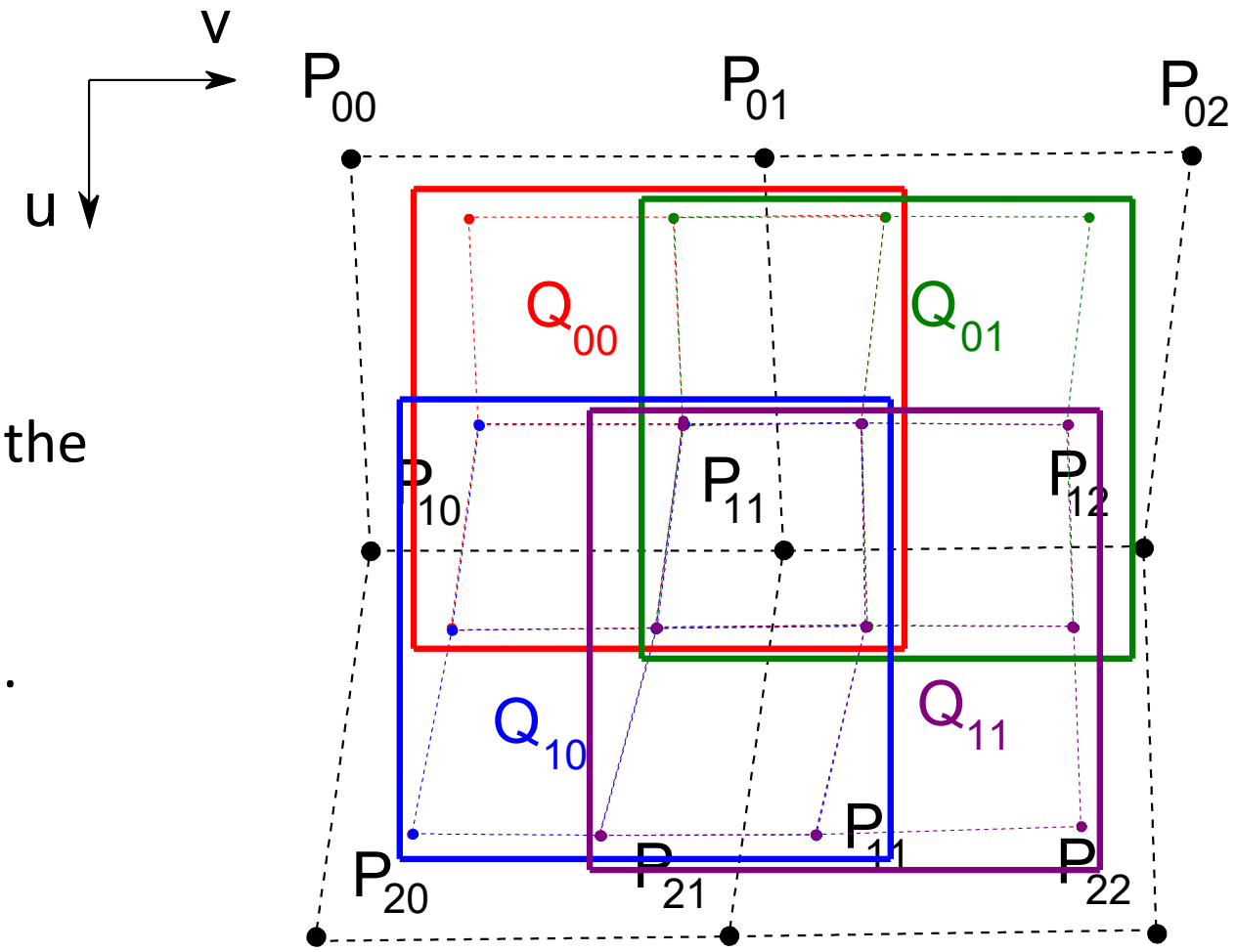
Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:



Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:



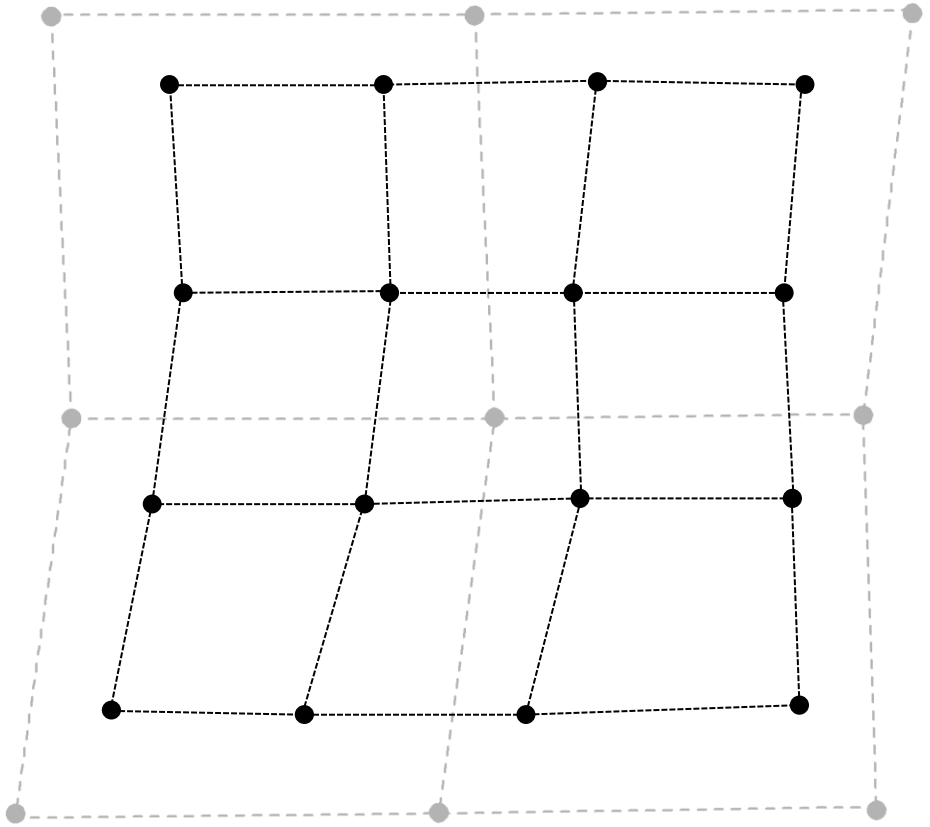
- Finally, assemble the 3x3 sub-patches together*.

*You must implement this step.

Doo-Sabin's scheme

- Computing a refined 4x4 patch from a 3x3 patch:

- Finally, assemble the 3x3 sub-patches together*.



*You must implement this step.

Doo-Sabin's scheme

- Computing a 3x3 sub-patch:
  - Start from the points of the initial 3x3 patch.
  - Compute the four 3x3 matrices S_u, S_v, S_u^T and S_v^T

  (see lecture 5, pp. 61-62.)
  - Apply these matrices to the 3x3 matrix **P** of points of the initial 3x3 patch.

  Reminder: The choice of the applied matrices on **P** will determine on which quadrant your are computing a sub-patch.

$$Q_{00} = S_u \, P \, S_u^T$$
$$Q_{01} = S_u \, P \, S_v^T$$
$$\cdots$$

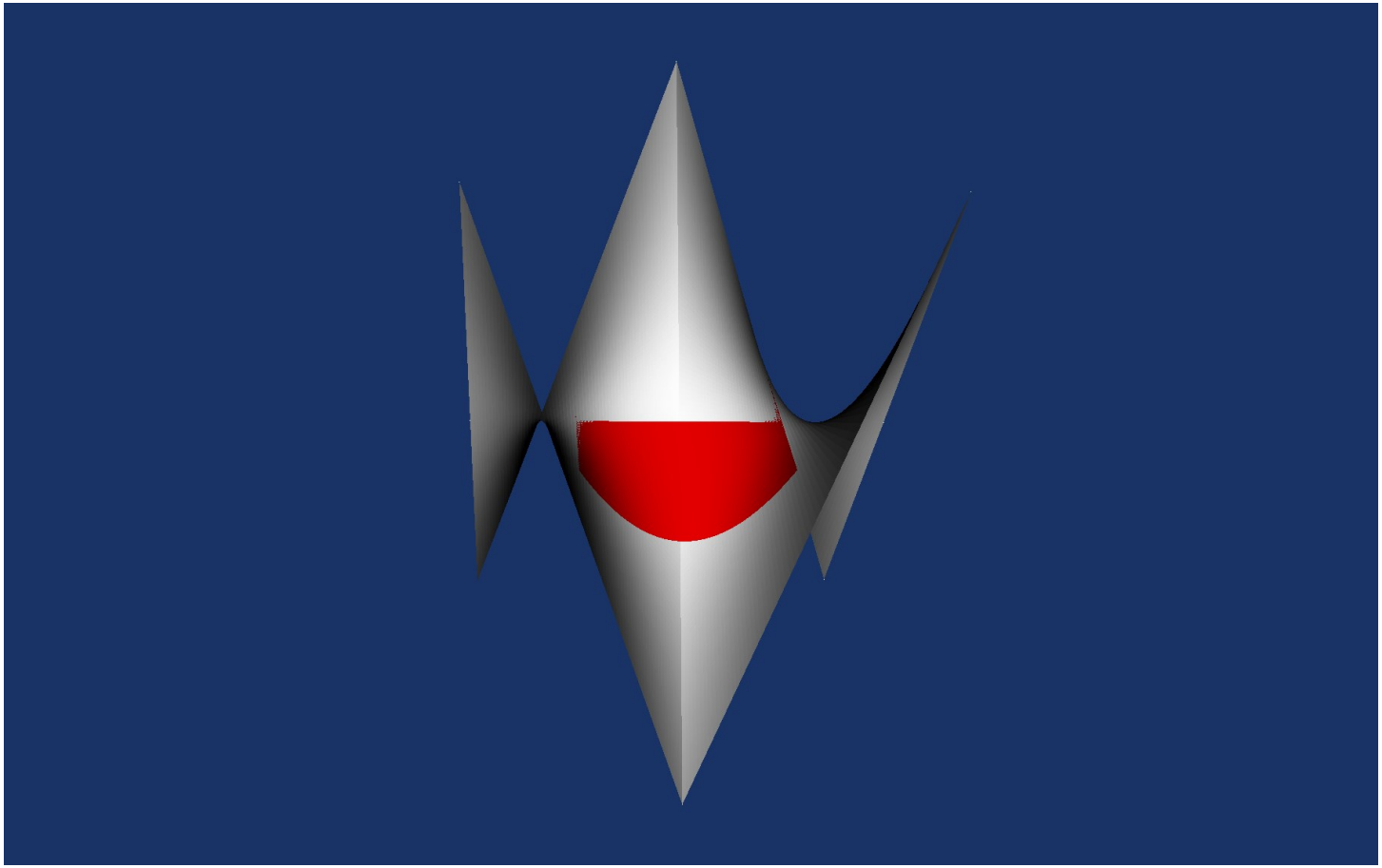
Tip: use the class `Square_Matrix` (in linear_algebra.h) for performing matrix-matrix multiplications.

Doo-Sabin's scheme

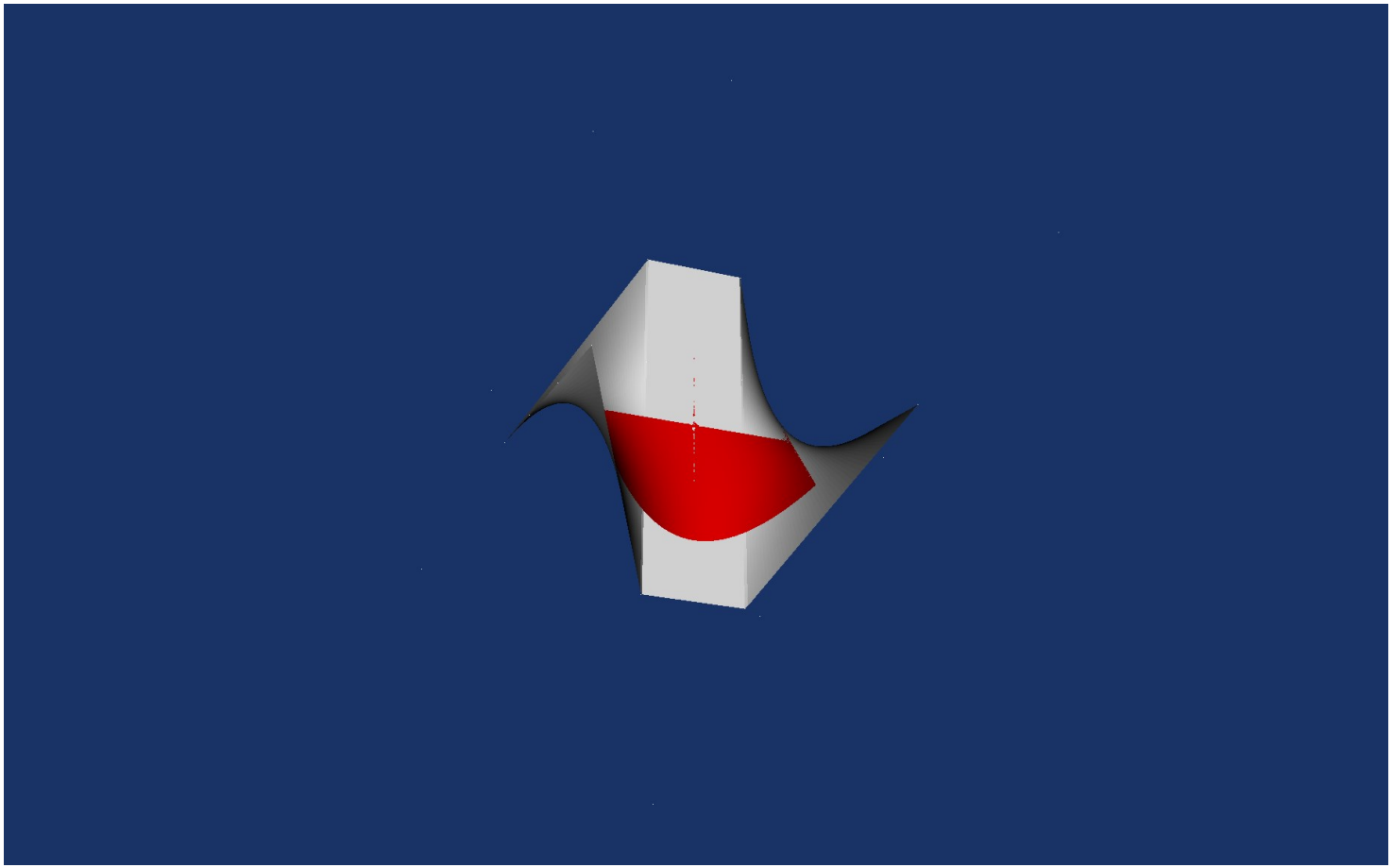
- Result: you should obtain a subdivision surface that tends to a degree-2 B-Spline surface with uniform nodal sequences.

(Given as a red surface in the code.)

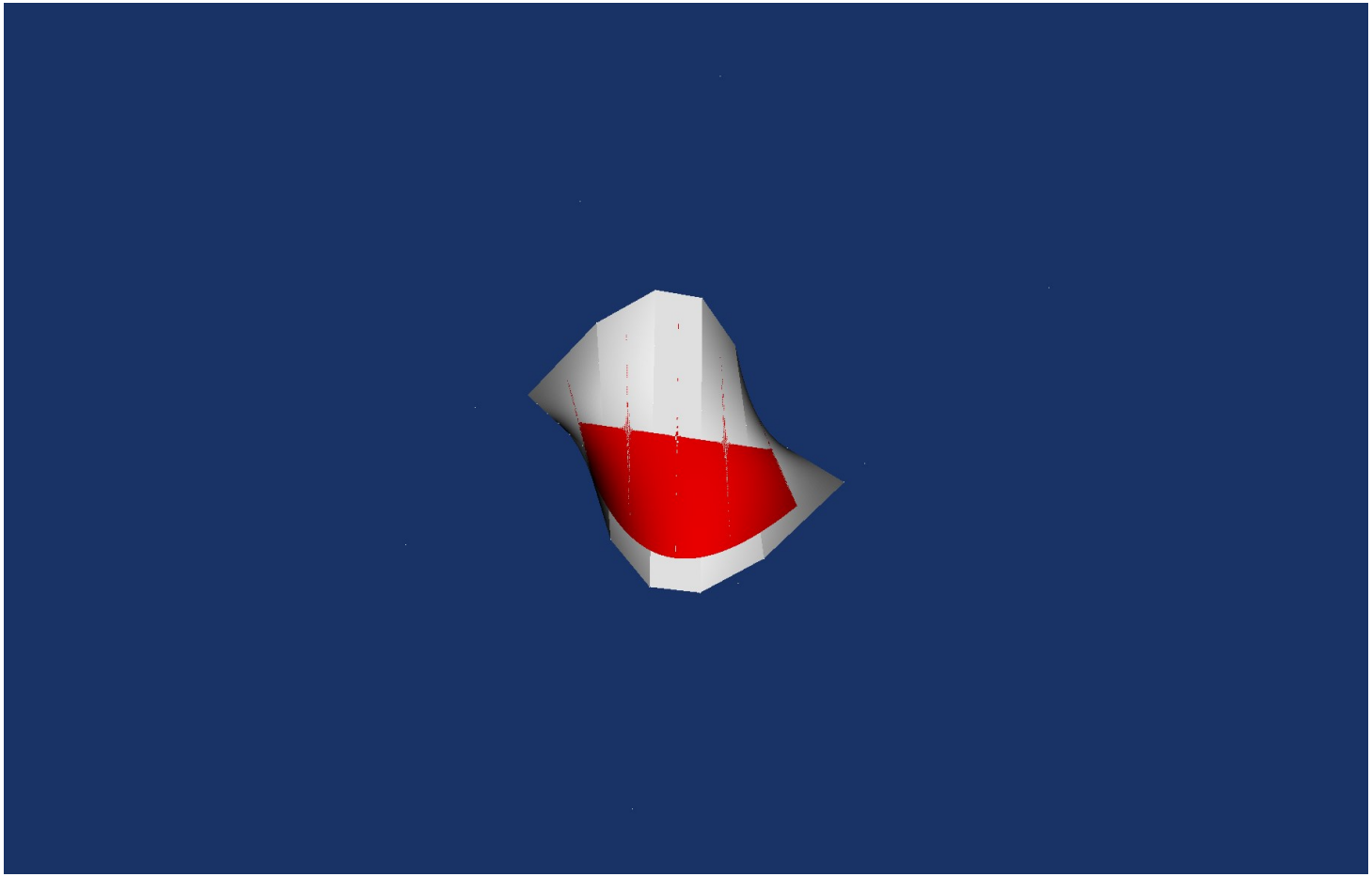
Doo-Sabin’s scheme



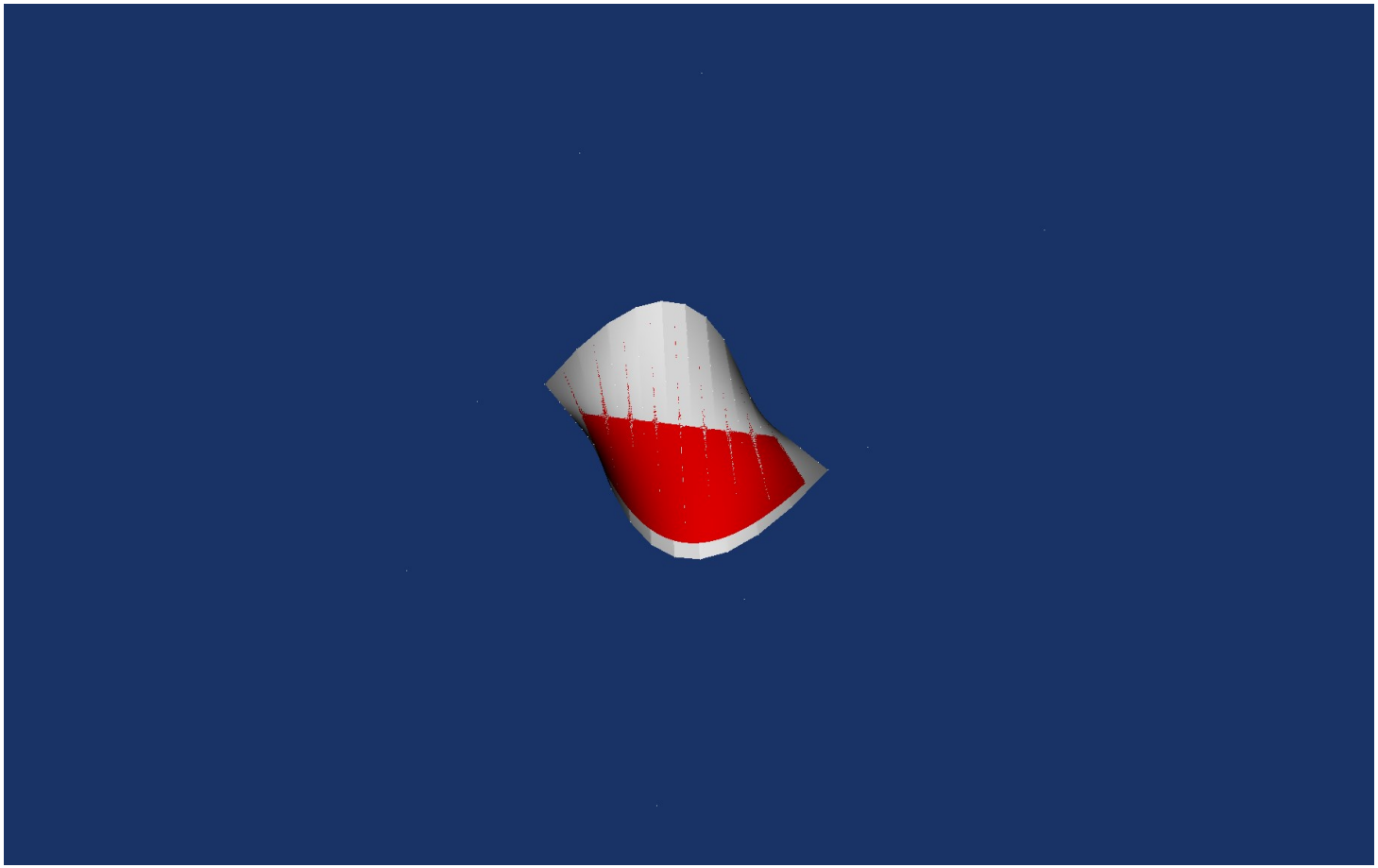
Doo-Sabin's scheme



Doo-Sabin’s scheme



Doo-Sabin’s scheme



Doo-Sabin’s scheme