Lecture 11

Note Title 2/9/2012

→ Prog part of assign 1 : due Feb 20/21

Input : accept vertices via a file

Output: display polyn on screen

demo the prog ~~with~~ at pre-arranged time

C++ code with opengl

MATLAB

2) Cell decomposition Approach

a) Partition C_{free} into a union of $\boxed{\text{non-overlapping}}$ regions called $\boxed{\text{cells}}$, whose union is C_{free}. &

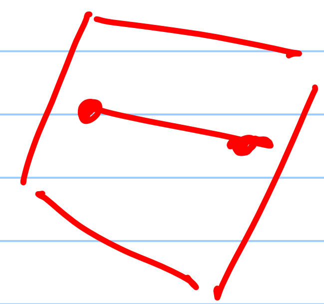
b) Build a "adjacency relationship" between cells – called an "adjacency graph" (connectivity ~~of~~ graph): its nodes are cells, and edges between nodes represent if the corresponding cells are adjacent. A sequence of adjacent cells connecting q_i and q_f will be called a "channel". A path can be extracted from the channel.

overall approach

What constitutes a cell?

1) should be a "simple" shape:

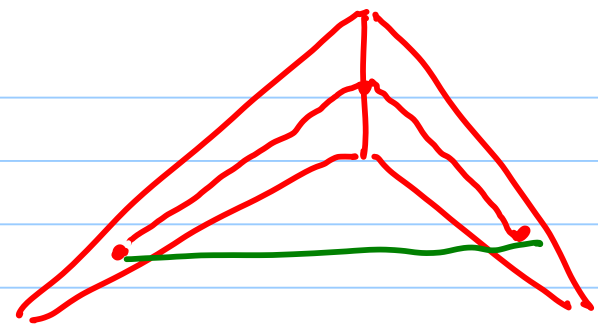
connecting two pts within the cell should be "easy".



"Convex shape"

2) adjacency relationships should be relatively easy to test.

3) constructing a path that crosses two cells should be easy.



Two types of cell decomp.:

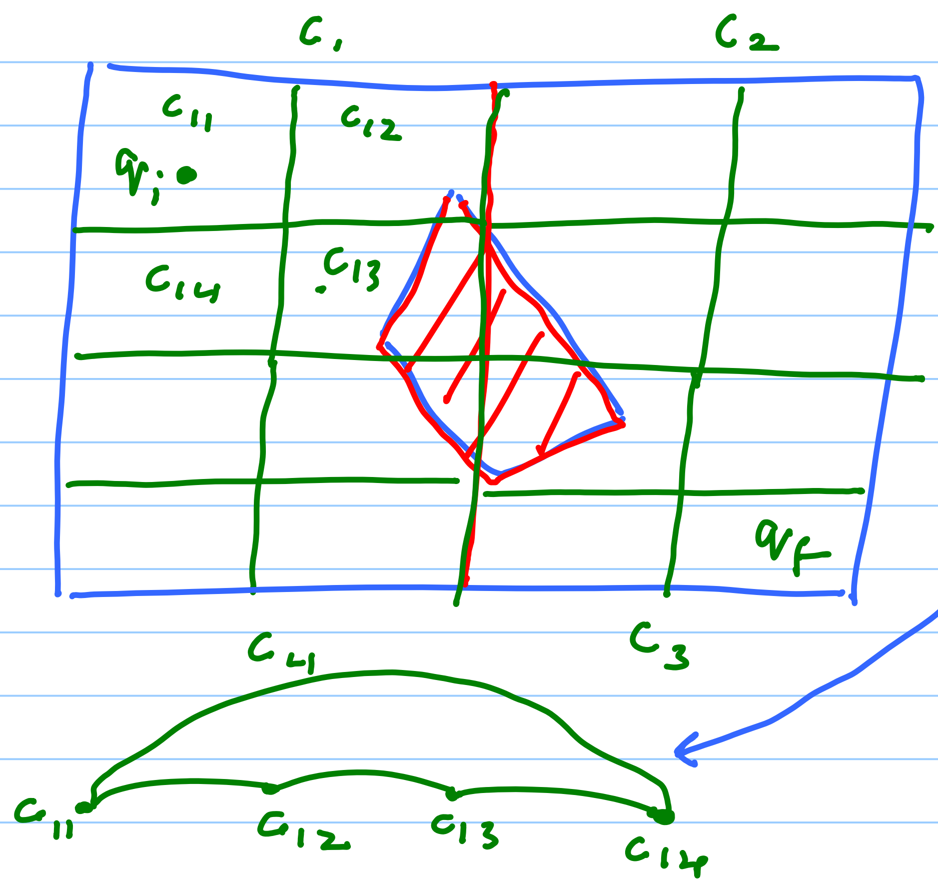
1) Approximate : Cell shapes are predetermined

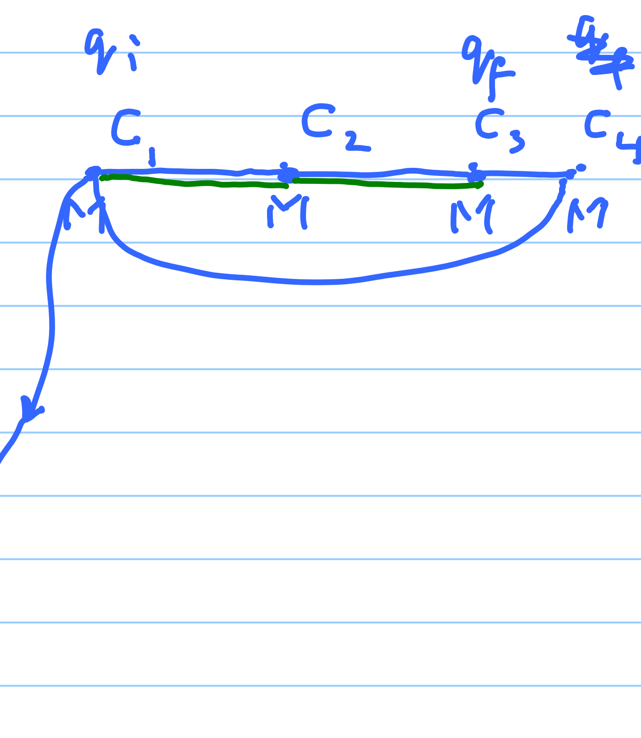
2) Exact : Cell shape is dependent on

"Certain Criticalities"

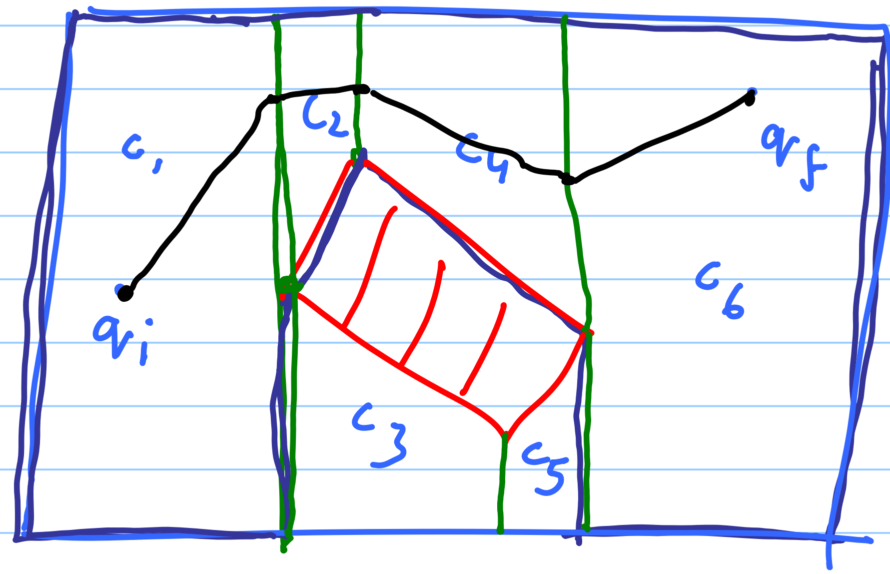
Approximate: in C_{free}.

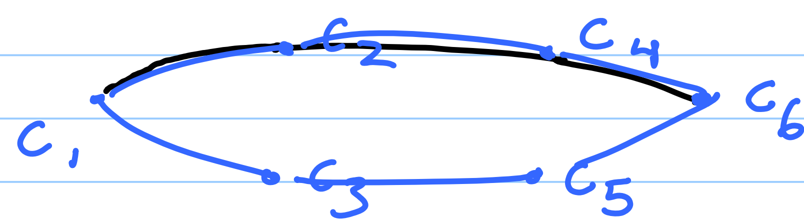
hierarchical "resolution Completeness"





Exact decomp. Trapezoidal decomposition → Complete algorithm.





__General Case:__ "boundaries of robot & obstacles" are rep. as semi alg. sets.

Recall that these are described by union/intersection of polynomials (with rational coeffs) in-equalities

Exact
"Cell decomposition" method for

L decomposing C_{free} into "cells":

Collin's Decomposition

Complexity : 2^{2^d} $\quad$ d : dim of C_{free}

"Schwartz + Sharir"

<u>Approximate Cell Decomposition</u>

1) Represent C_{free} as a union of cells of pre-defined shape,

often "rectangloid"

2) easier to implement

3) provides little insight into structure of C_{free}

4) time + space complexity is exp. in d, the dim of c-space, useful for low-dim c-space ≤ 4.

Free
Obstack
Mixed

ted, the associated connectivity graph, denoted by G_i, is searched for hannel connecting $\mathbf{q}_{init}$ to $\mathbf{q}_{goal}$.

simple first-cut planning algorithm is the following:

1. Compute a rectangloid decomposition $\mathcal{P}_1$ of Ω. Set i to 1.

2. Search the connectivity graph G_i associated with the decomposition $\mathcal{P}_i$ for a channel connecting the initial cell containing $\mathbf{q}_{init}$ to the goal cell containing $\mathbf{q}_{goal}$. If the outcome of the search is an E-channel, return success. If it is an M-channel, proceed to the next step. Otherwise, return failure.

3. Let Π_i be the M-channel generated at Step 2. Set $\mathcal{P}_{i+1}$ to $\mathcal{P}_i$. For every MIXED cell κ in Π_i, compute a rectangloid decomposition $\mathcal{P}^\kappa$ of κ and set $\mathcal{P}_{i+1}$ to $[\mathcal{P}_{i+1} \setminus \{\kappa\}] \cup \mathcal{P}^\kappa$. Set i to $i+1$. Go to Step 2.

he search of G_i at Step 2 can be guided by various heuristics. In

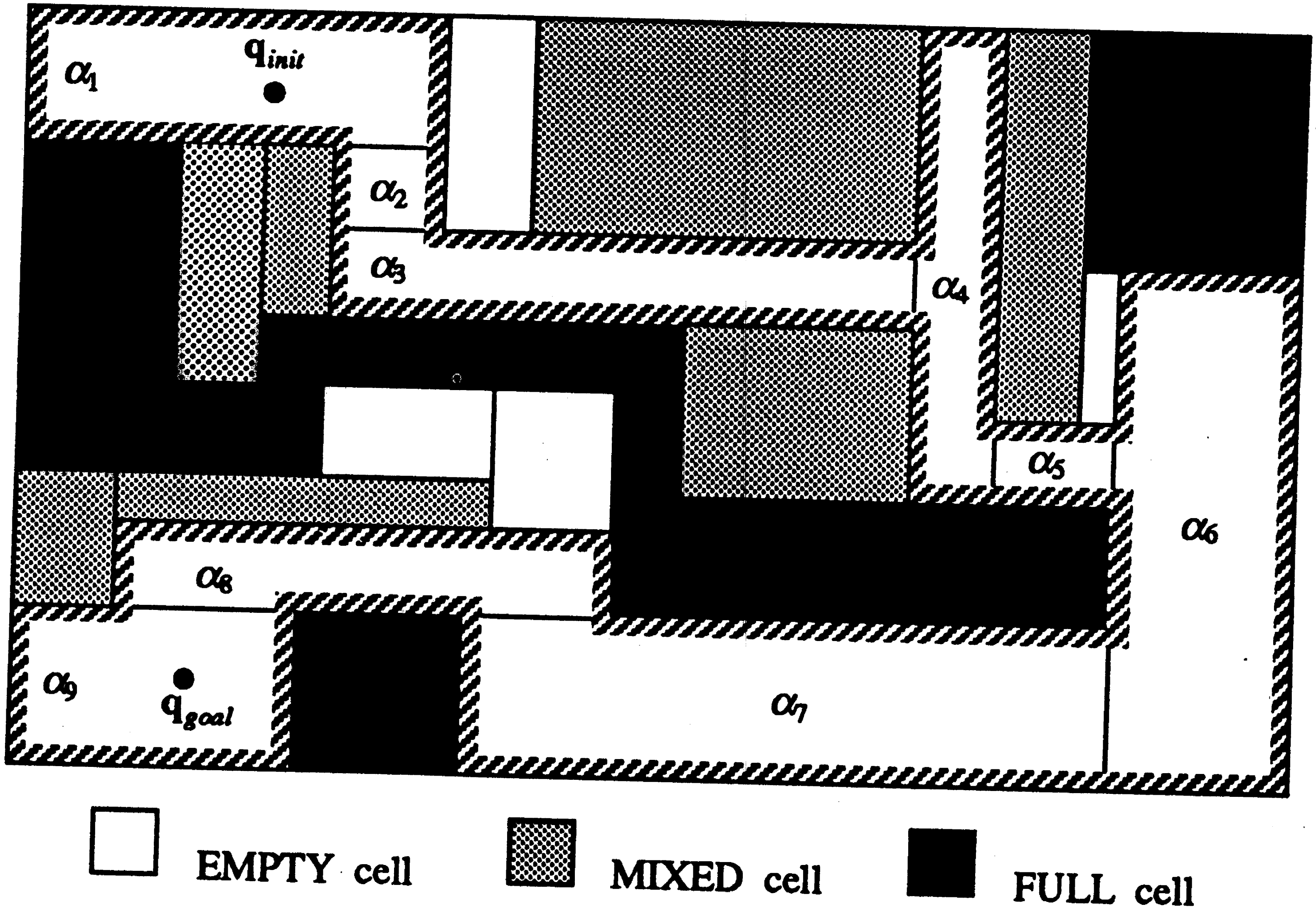
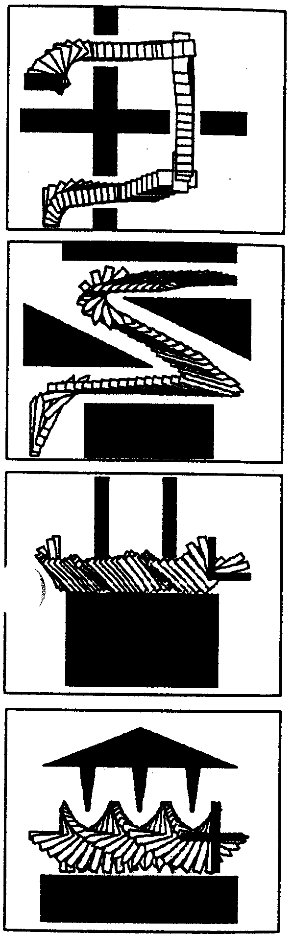


Figure 1. A channel is a sequence of adjacent cells which are either EMPTY or MIXED. If all the cells are EMPTY the channel is said to be an E-channel, otherwise it is said to be an M-channel. This figure shows an E-channel (striped contour) in a two-dimensional space. It connects the two cells that contain the

How to divide a mixed cell
further and label the
subdivided cells?

"Divide & Label"



ɔy a planner based on the
ıs input problems [Zhu and
polygon that can translate
ıtacles.

dren of a cell κ have the
: each edge of κ into two

f m = 3, it is called an
t of a two-dimensional

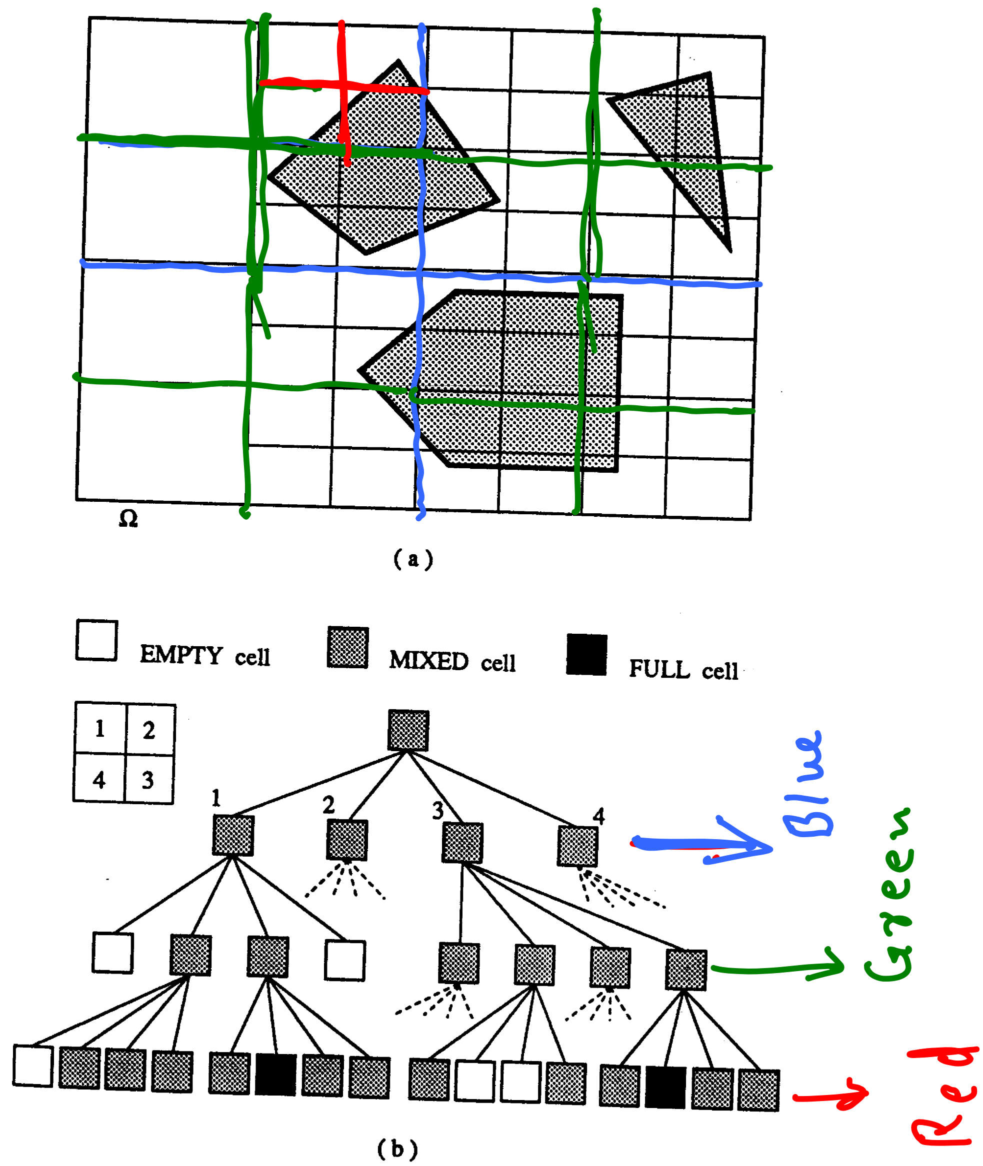


Figure 4. A quadtree decomposition of Ω is obtained by recursively dividing Ω and the generated MIXED cells into smaller cells. The division of a cell creates four new rectangloid cells of equal dimensions. Figure a shows the quadtree decomposition at depth 3 of a simple configuration space. Figure b shows a subset of the corresponding tree.

Note Title 1/28/2010

Cell Labelling :

$a_{ij} x + b_{ij} y + c \le 0$

Recall CB : $V_i \cancel{V_j} \, e_{ij}$

$i \in$ # of obs

$\underbrace{}_{\text{one convex obs.}}$

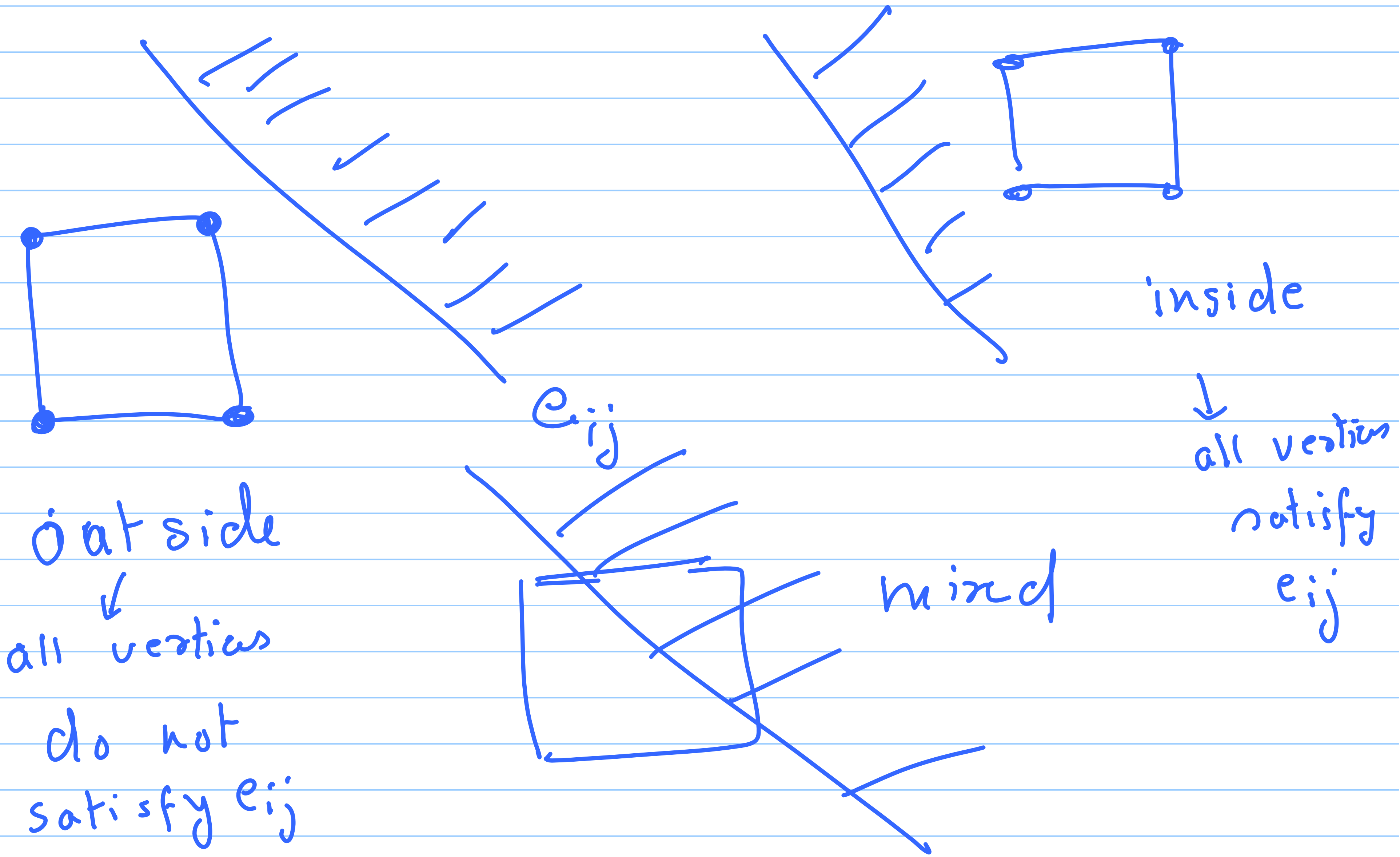
$j \in$ # of vertices

first in R^N : polyg. case

no rot.

"Cells are convex"

So need to test only vertices



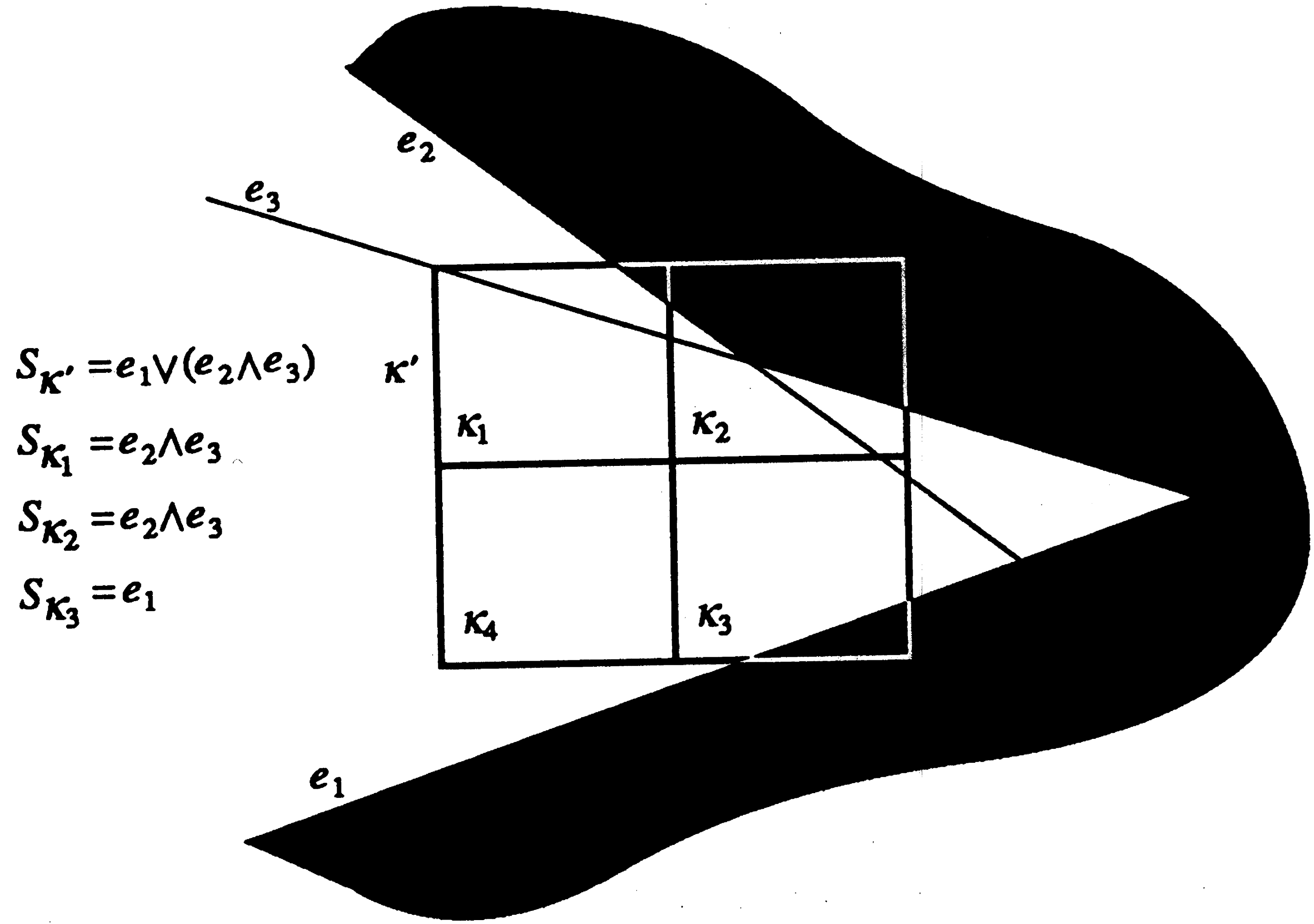
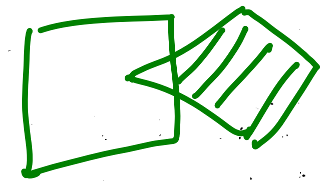


Figure 6. This figure illustrates the simplification of a C-sentence when new cells are created and labeled. The sentence $S_{\kappa'} = e_1 \vee (e_2 \wedge e_3)$ is associated with the MIXED cell κ'. When this cell is decomposed (in a quadtree fashion), four new cells denoted by κ_1 through κ_4 are generated. Both κ_1 and κ_2 are



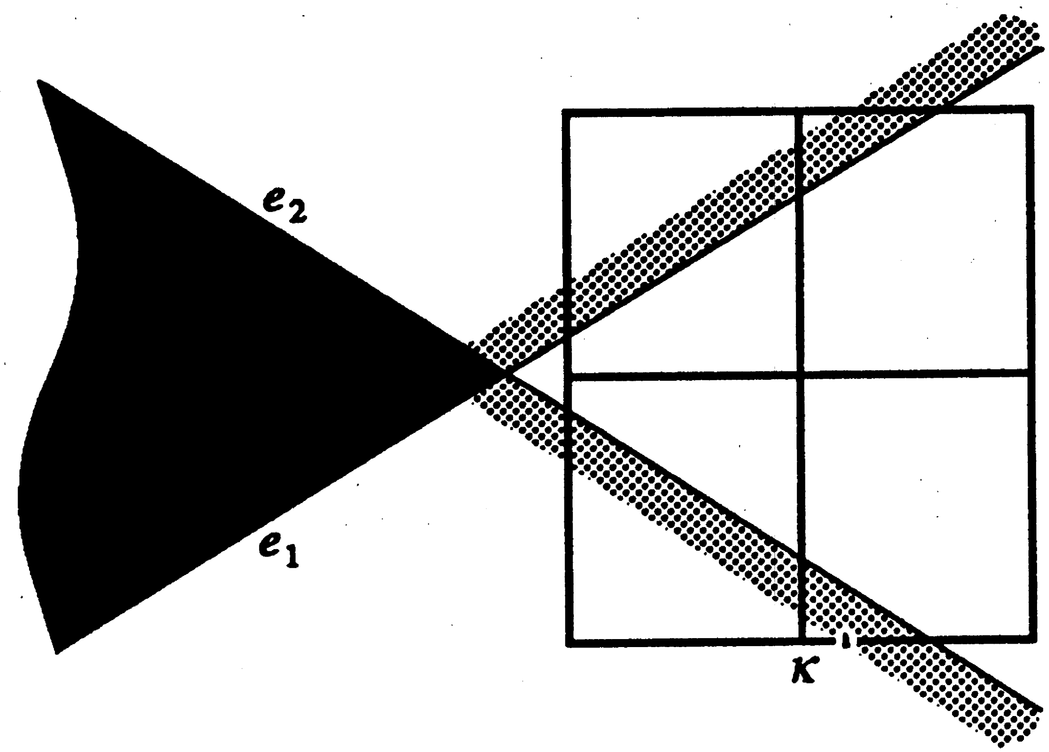
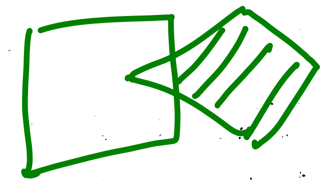


Figure 7. This figure illustrates how a cell κ gets labeled as MIXED though it has no intersection with the C-obstacle region. Assume that the C-sentence associated with the parent cell of κ is $e_1 \wedge e_2$. Since κ is cut by both e_1 and e_2, κ is labeled as MIXED and the C-sentence $e_1 \wedge e_2$ is associated with it. However, no point in κ satisfies e_1 and e_2 *simultaneously*. This incorrect (but conservative) labeling results from the fact that each C-constraint is individually considered as a half-plane. The error is eventually corrected at a deeper level in the quadtree decomposition. (The "inside" side of a C-constraint line is shown



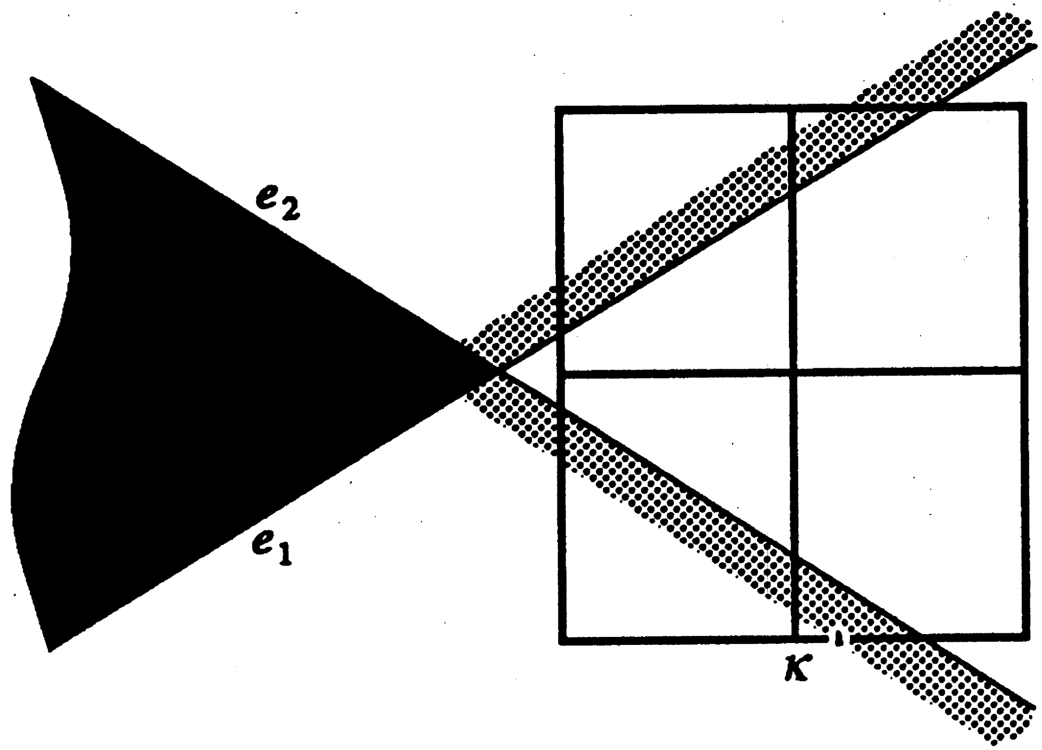


Figure 7. This figure illustrates how a cell κ gets labeled as MIXED though it has no intersection with the C-obstacle region. Assume that the C-sentence associated with the parent cell of κ is $e_1 \wedge e_2$. Since κ is cut by both e_1 and e_2, κ is labeled as MIXED and the C-sentence $e_1 \wedge e_2$ is associated with it. However, no point in κ satisfies e_1 and e_2 *simultaneously*. This incorrect (but conservative) labeling results from the fact that each C-constraint is individually considered as a half-plane. The error is eventually corrected at a deeper level in the quadtree decomposition. (The "inside" side of a C-constraint line is shown

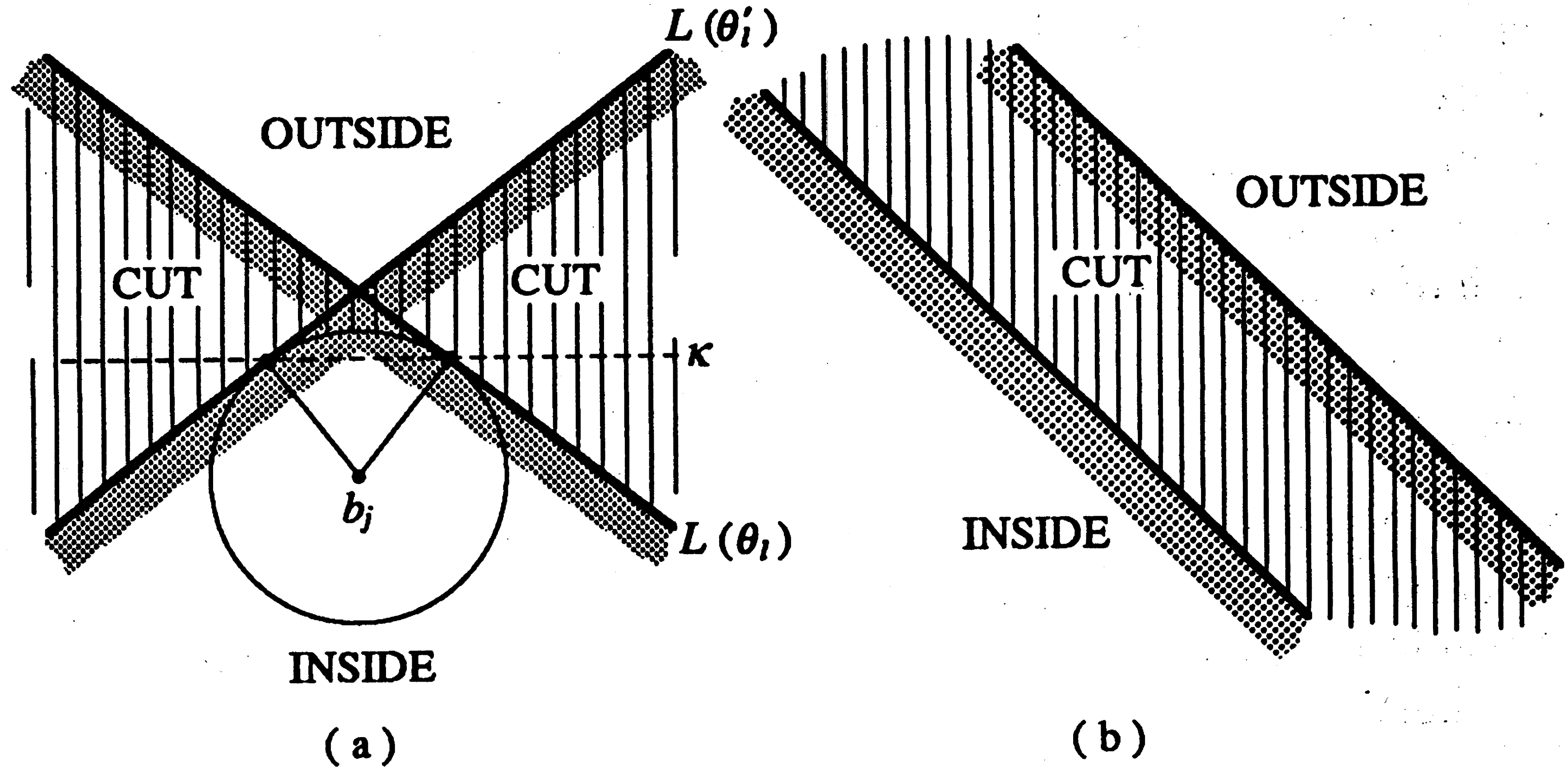
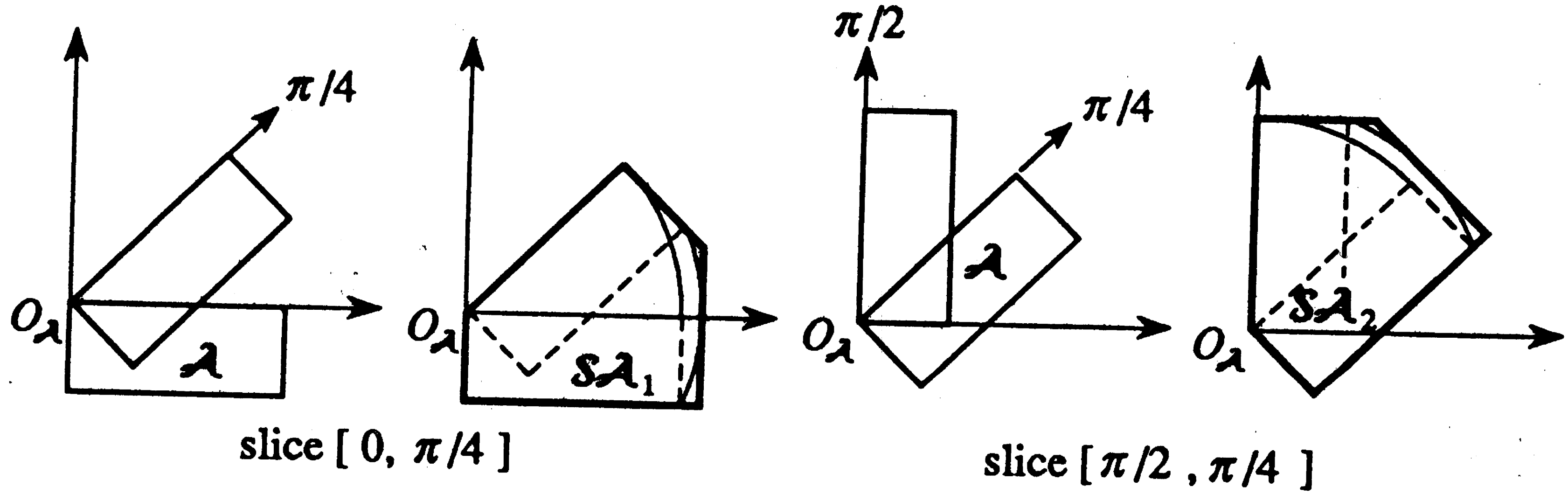


Figure 8. This figure illustrates the projection of the portion of the C-surf $a(\theta)x + b(\theta)y + c(\theta) = 0$ which is comprised in the angular interval $[\theta_l, \theta_l']$ i the xy-plane. When the C-surface is of type A, the projection is the reg swept out by a line rotating around an obstacle vertex b_j (Figure a). When C-surface is of type B, the projection is the region swept out by a translat line parallel to an obstacle edge $E_j^{\mathcal{B}}$ (Figure b). In both cases, the project divides the plane into three regions designated by OUTSIDE, INSIDE, a

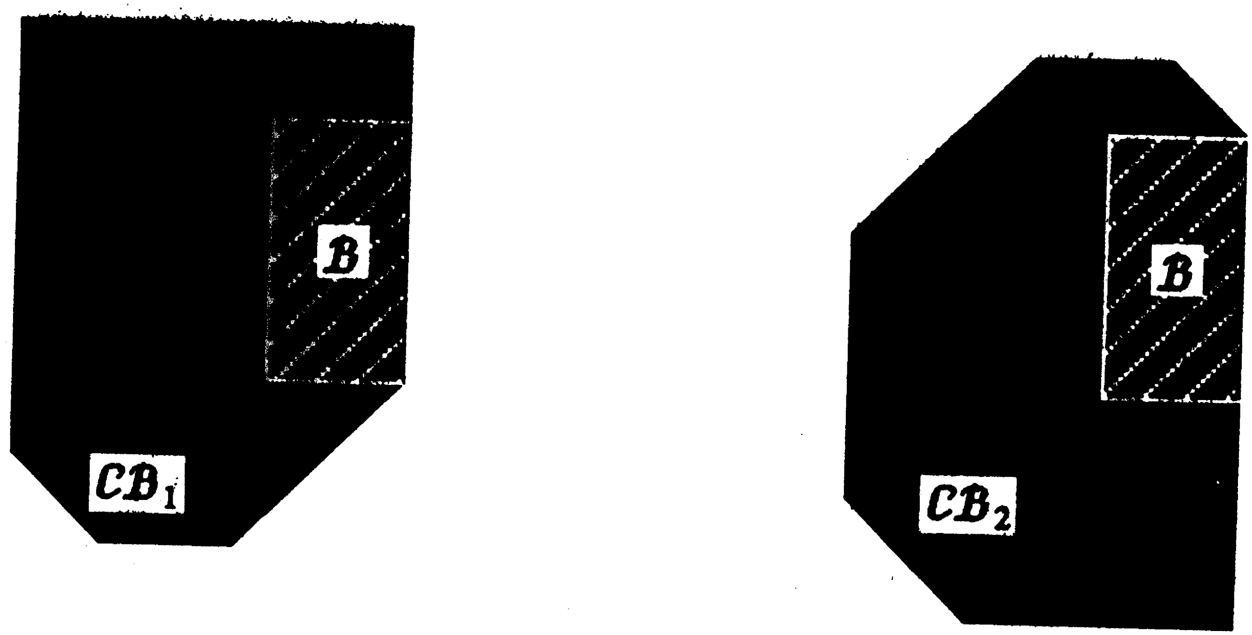
ORIENTATION SLICING:

"Approximate" approach



π/4
O_λ
λ
slice [0, π/4]
O_λ
$S\lambda_1$
π/2
π/4
O_λ
λ
slice [π/2 , π/4]
O_λ
$S\lambda_2$

(a)



B
CB_1
B
CB_2

(b)