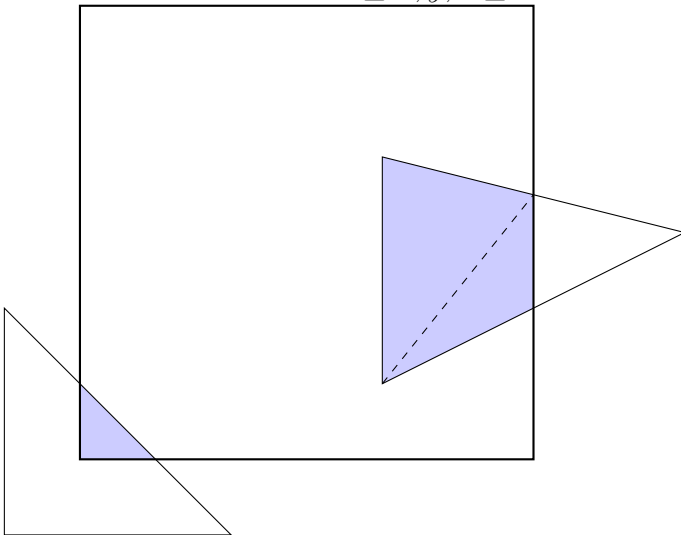


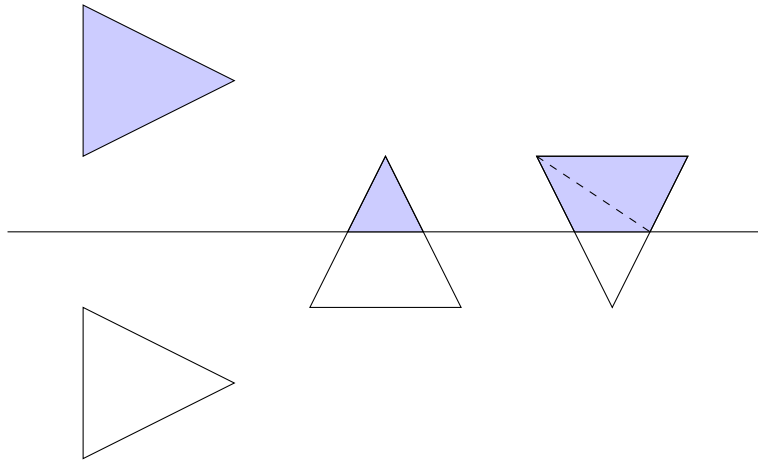
# Clipping Algorithms and Pipeline Integration

## 1 Clipping problem

Given our viewing volume  $[-1, 1]^3$ , we wish to retain only (parts of) triangles inside the viewing area. This volume is a cube bounded by the planes  $x = \pm 1$ ,  $y = \pm 1$ , and  $z = \pm 1$ . The feasible area satisfies  $-1 \leq x, y, z \leq 1$ .



## 2 Clipping cases



There are a relatively small number of possible ways a triangle can be cut with a plane; these can be enumerated. This leaves the problem of cutting clipping triangle edges with a cutting plane to create the new triangle vertices.

To clip against our viewing volume, a cube, we simply clip against each plane one by one. The triangles that survive one plane are passed to the next.

## 3 Segment-Plane Intersection

Cutting edges can be done using the ray-plane intersection logic, but it can also be done much cheaper than that. Consider for example the plane  $x = -1$ . We wish to find a point  $\vec{P} = \vec{A} + \lambda(\vec{B} - \vec{A})$ . If this point lies on the plane, then  $-1 = P_x = A_x + \lambda(B_x - A_x)$ , which is trivially solved for  $\lambda$ . The intersection location is obtained by plugging into  $\vec{P}$ , and vertex data may be interpolated to  $\vec{P}$  using  $\lambda$ .

## 4 Clipping in Homogeneous Coordinates

In homogeneous coordinates, the clipping volume is defined by  $-w \leq x, y, z \leq w$ . Note that if  $x = -w$ , then after the perspective divide,  $\frac{x}{w} = -1$ , which is the intended clipping plane in 3D. In this case,  $\bar{P} = \bar{A} + \gamma(\bar{B} - \bar{A})$ , where  $\bar{P} = \langle P_x, P_y, P_z, P_w \rangle$ . If  $\bar{P}$  lies on the plane  $x = -w$ , then  $A_x + \gamma(B_x - A_x) = -(A_w + \gamma(B_w - A_w))$ . This can simply be solved for  $\gamma$ , from which  $\bar{P}$  can be computed.

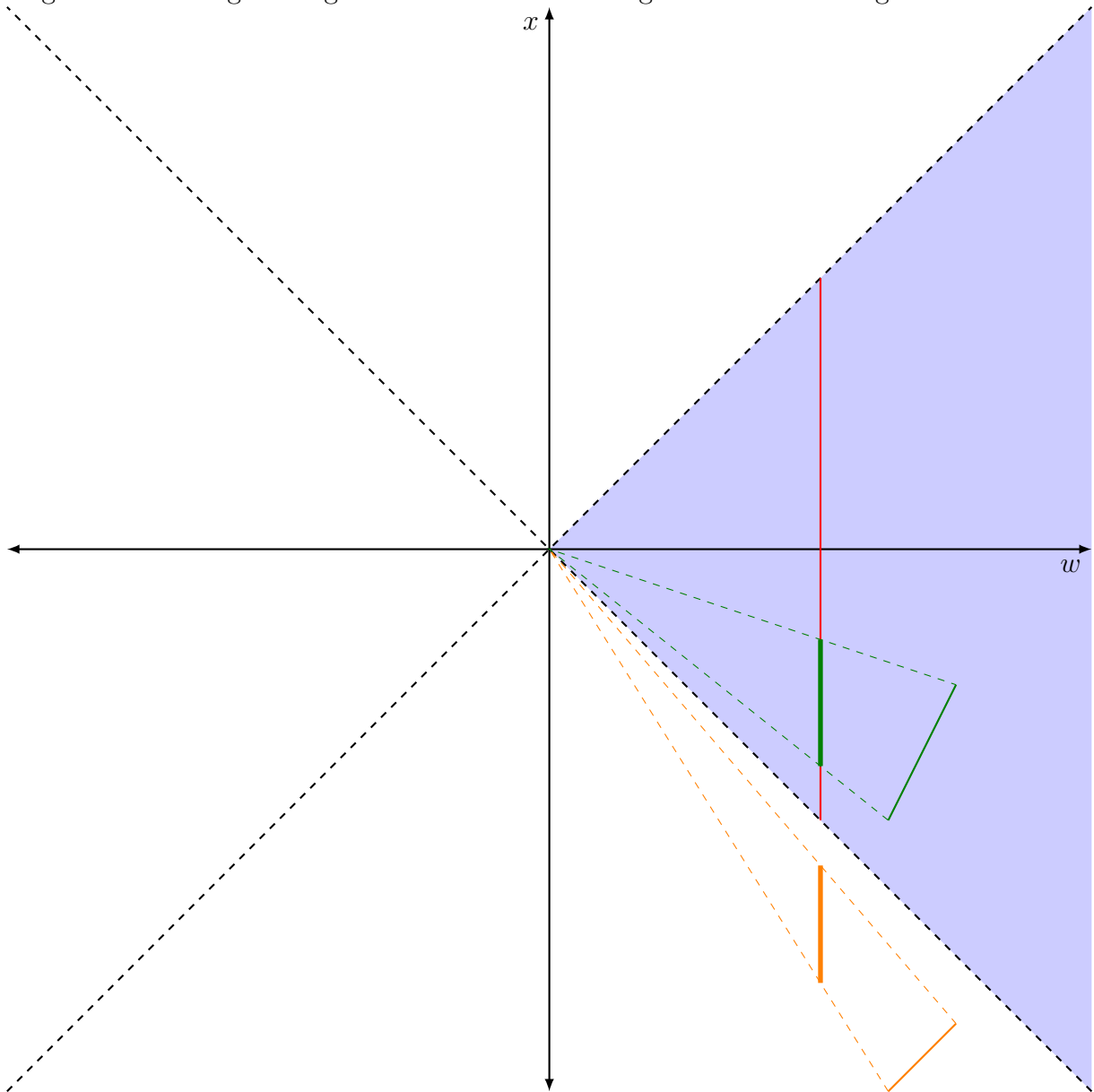
Note that in general  $\lambda \neq \gamma$ . The weight  $\lambda$  is the image space barycentric coordinate; it should be used for interpolating things like depth. It corresponds to OpenGL **no-perspective** interpolation.  $\gamma$  is the world space barycentric coordinate, which should be used for colors and uv values. It corresponds to OpenGL **smooth** interpolation.

Note that the inequalities above imply  $w \geq 0$ ; points with  $w < 0$  are outside the viewing volume. In this way, scaling homogeneous points by a *positive* number does not change the

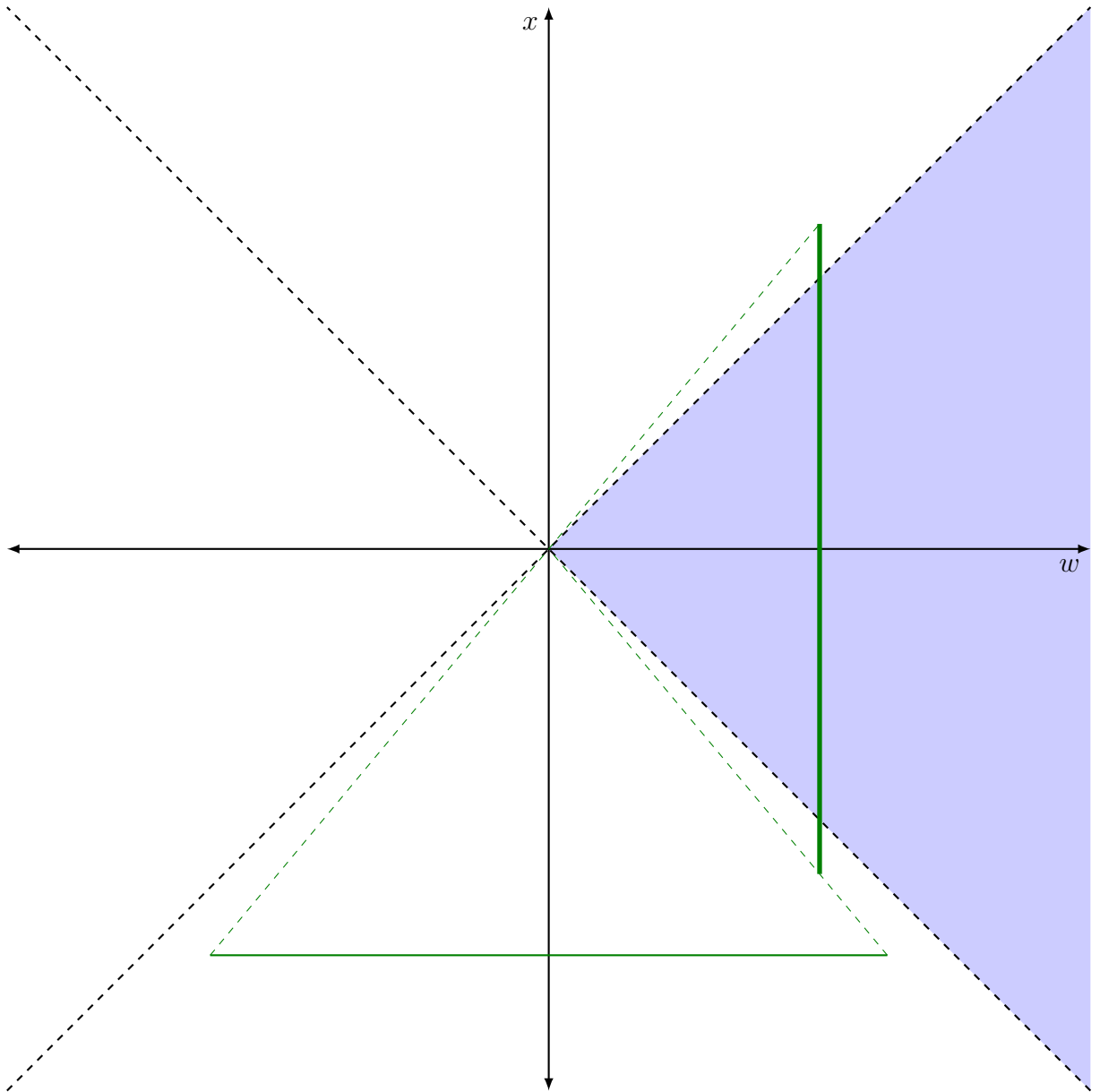
corresponding 3D point, but scaling by a *negative* number will make a point invisible.

## 5 Homogeneous vs 3D clipping

Consider clipping in homogeneous coordinates in 1D. Here, the blue region is the viewing volume. After the perspective divide, points are mapped to the red segment, which is our 1D viewing volume. The green segment is inside the viewing area and the orange one is outside.



However, strange things can happen if  $w < 0$ .



In this case a segment that should be completely out of view, after the perspective divide, covers the whole screen. In this case, clipping after the perspective divide fails. We must thus clip in homogeneous coordinates. Clipping in 3D coordinates does not work.