Problem 1: [Tracing rays through uniform spatial subdivisions]

Consider the problem of tracing a ray through a uniform 3D spatial subdivision, and iterating over
the ordered list of intersected cells. Assume that the subdivision is axis aligned, that cells are cubical
of length $h$, and that the array dimensions are $N_x \times N_y \times N_z$. Denote the 3D ray by $r(t) = e + dt$,
where $e = (e_x, e_y, e_z)$ and $d = (d_x, d_y, d_z)$.

One approach for determining the next intersected cell is to evaluate $t$ values for all axis-aligned
slabs of each axis (see figure) and then determining whether the next $t$ value is from an $x$, $y$, or $z$
slab, and then intersecting the corresponding cell.

Instead of computing all values at once, an incremental DDA-type algorithm is preferrable in prac-
tice. In this problem, you will state concisely an efficient incremental algorithm to iterate over
cells pierced by the advancing ray by effectively determining which of the $x$, $y$, or $z$ dimensions
has the smallest $t$ value. Assume that you can output a intersected cells by issuing the command
output(i,j,k).

Similar to efficient DDA rasterizers, your algorithm should only require simple addition and com-
parison operations in the inner loop. State your initial setup operations, assuming that the first
boundary cell the ray hits is $(i_0, j_0, k_0)$—here indices are 0-based and run from $0, \ldots, N - 1$—at
position $e$ and $t = 0$ for simplicity.

Problem 2: [Rasterization]

Rational linear interpolation (RLI) is used to interpolate texture coordinates in screen space without
the artifacts introduced by linear interpolation of $(u, v)$ values in screen space [Heckbert and More-
ton 1991; http://citeseer.ist.psu.edu/36582.html]. Describe the screen space val-
ues obtained if RLI were used to interpolate $(x, y, z)$ values in screen space (Hint: consider the case
where the texture coordinates are spatial coordinates). Clearly state and compare how the RLI $z$
value differs from the screen-space depth value $z'$ used in the Z buffer algorithm. Why not use RLI
$z$ values for the Z buffer algorithm?