```
def intersectLinePlane(line=None, plane=None):

    # Make all numpy available via shorter 'num' prefix
    import numpy as np
    import math as m

    #INTERSECTLINEPLANE Intersection point between a 3D line and a plane
    #
    #   PT = intersectLinePlane(LINE, PLANE)
    #   Returns the intersection point of the given line and the given plane.
    #   LINE:  [x0 y0 z0 dx dy dz]
    #   PLANE: [x0 y0 z0 dx1 dy1 dz1 dx2 dy2 dz2]
    #   PT:    [xi yi zi]
    #   If LINE and PLANE are parallel, return [NaN NaN NaN].
    #   If LINE (or PLANE) is a matrix with 6 (or 9) columns and N rows, result
    #   is an array of points with N rows and 3 columns.
    #
    #   PT = intersectLinePlane(LINE, PLANE, TOL)
    #   Specifies the tolerance factor to test if a line is parallel to a
    #   plane. Default is 1e-14.
    #
    #   Example
    #     % define horizontal plane through origin
    #     plane = [0 0 0   1 0 0   0 1 0]
    #     % intersection with a vertical line
    #     line = [2 3 4  0 0 1]
    #     intersectLinePlane(line, plane)
    #     ans =
    #        2   3   0
    #     % intersection with a line "parallel" to plane
    #     line = [2 3 4  1 2 0];
    #     intersectLinePlane(line, plane)
    #     ans =
    #       NaN  NaN  NaN
    #
    #   See also:
    #   lines3d, planes3d, points3d, clipLine3d
    #
    #   ---------
    #   author : David Legland
    #   INRA - TPV URPOI - BIA IMASTE
    #   created the 17/02/2005.
    #

    #   HISTORY
    #   24/11/2005 add support for multiple input
    #   23/06/2006 correction from Songbai Ji allowing different number of
    #       lines or plane if other input has one row
    #   14/12/2006 correction for parallel lines and plane normals
    #   05/01/2007 fixup for parallel lines and plane normals
    #   24/04/2007 rename as 'intersectLinePlane'
    #   11/19/2010 Added bsxfun functionality for improved speed (Sven Holcombe)
    #   01/02/2011 code cleanup, add option for tolerance, update doc
    #   07/06/2018 translated to python by Valerie Martin

    #TEST
    #if 1:
    #        plane = np.asarray([0., 0., 0.,  1., 1., 0.,  2., 1., 4.])
    #        line = np.asarray([2., 3., 4., 1., 0., 1.])

    # extract tolerance if needed
    tol = 1e-14

    # unify sizes of data
    nLines = line.shape
    nPlanes = plane.shape

    # N planes and M lines not allowed
    if nLines != nPlanes and min(nLines, nPlanes) > (1,):
        error='MatGeom:geom3d:intersectLinePlane','Input must have same number
of rows, or one must be 1'

        # plane normal
        #n = np.cross(np.subtract(plane[3:6], plane[0:3]),
np.subtract(plane[6:9], plane[3:6]))
        n = np.cross(plane[3:6], plane[6:9])

        # difference between origins of plane and line
        dp = np.subtract(plane[0:3], line[0:3])

        # dot product of line direction with plane normal
        denom = np.dot(line[3:6], n)

        # relative position of intersection point on line (can be inf in case of
a
        # line parallel to the plane)
        t = np.dot(dp, n)/ denom

        # compute coord of intersection point
        point = np.add(t*line[3:6], line[0:3])

        # set indices of line and plane which are parallel to NaN
        if abs(denom) < tol:
            point = (float('nan'), float('nan'), float('nan'))

    return point
```