Convertir avec Z-set

```
****mesher
 ***mesh toto
  **import gmsh toto.msh
****return
```

Convertir un maillage C2D3 avec un script python

Il peut arriver que la technique ci-dessus ne fonctionne pas ou mette un temps considérable. Auquel cas, autant y aller “à la main” avec un code comme celui ci-dessous:

```
def from_gmsh_to_geof(msh_name, geof_name, dim = 2):
    """Convert from gmesh format (msh) to Zebulon formet (geof).
    Only works for c2d3 elements for now.
    This is messy as hell and have to be rewritten to be able to
    handle physical element (ie, kind of nsets and elsets), count the
correct number of elements (gmsh
    counts also 1D element), etc. Also, fixed count of number of element
tags..

    - msh_name: filename of the msh file, ex: 'maillage.msh'

    returns: write geof file

    """
    msh_f = open(msh_name, 'r')
    msh_c = msh_f.readlines()
    msh_c = [m[:-1] for m in msh_c]

    output = '***geometry\n**node\n'
    nb_nodes = int(msh_c[4])
    output += msh_c[4] + ' {:.0f}\n'.format(dim)

    node_line_start = 5
    node_line_end = node_line_start + nb_nodes
    for i in range(node_line_start, node_line_end):
        #zebulon want 0. and not 0, so convert..
        line = msh_c[i].split()
        output += '{:.0f} {:.16e} {:.16e}\n'.format( float(line[0]),
float(line[1]), float(line[2]) )

    output += '**element\n'
```

```
    nbel = int(msh_c[node_line_end + 2])

    el_line_start = node_line_end + 3
    el_line_end = el_line_start + nbel

    elements = ''
    dummy_1D_counter = 0
    nsets = collections.defaultdict(set)

    for i in range(el_line_start, el_line_end):
        line = msh_c[i].split()
        if int(line[1]) == 1:
            dummy_1D_counter += 1  # ok because we know that 1D and 0D
elements come before
            nsets[line[3]].add(line[5])  # add the tow node of this line
            nsets[line[3]].add(line[6])
        elif int(line[1]) == 2:  # if we have 2D elements (ignore 1D
elements)
            elements += '{:.0f} c2d3 {:.0f} {:.0f} {:.0f}\n'.format(
int(line[0])-dummy_1D_counter, int(line[5]),int(line[6]),int(line[7]) )

    output += '{:.0f}\n'.format(nbel-dummy_1D_counter)
    output += elements
    msh_f.close()

    output += '***group\n'
    for key in nsets:
        output += '**nset {:s} \n'.format(key)
        for val in nsets[key]:
            output += '{:s} '.format(val)
        output += '\n'

    output += '***return'
    with open(geof_name, 'w') as f:
        f.write(output)

    return
```