Anisotropically Scaled Grid Cells in SIMION

Overview

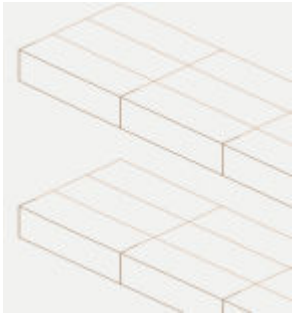

Figure: PA with non-square grid cells.

As of SIMION 8.1, potentials array grid cells can be *anisotropically scaled* (rather than *isotropically scaled*), which means that the grid density in the X, Y, and Z directions can be scaled by different amounts. This allows grid cells to be narrow rectangles (2D) or rectangular boxes (3D), rather than just squares and cubes as permitted in previous versions of SIMION.

The anisotropic scaling support utilizes a new feature in SIMION 8.1.0, also described here, where you can store the X, Y, Z cell lengths in mm inside the PA file (rather than or in addition to using the PA instance scale factor in the workbench IOB). This feature can be useful even for isotropically scaled PAs. It is required, for example, in the *Poisson Solver in SIMION*, and it can provide convenience in Laplace conditions as well.

Purpose of Anisotropic Scaling

The anisotropic scaling feature allows certain geometries, particularly some geometries that are long in one direction, to be more efficiently and/or accurately modeled. As an example, you may have long multipole rods or a narrow lens system where the grid density in the radial direction is much more critical that the grid density in the axial direction, particularly if there are surface errors due to curved electrodes that don't align to grid lines. With anisotropic scaling, you can set the axial grid density to be finer than the radial grid density. (2D planar arrays, such as used in the "quad" example, may be effective too though.) Anisotropic scaling may also in some cases permit better alignment of surfaces to grid units for higher accuracy.

Purpose of Storing Cell Lengths in PA Files

Prior to SIMION 8.1.0, potential array files didn't contain an explicit physical size. Their sizes were measured instead in unitless/dimensionless "grid units". It wasn't until you projected an "instance" of a potential array on a workbench (IOB) and assigned a mm/gu scaling factor on that instance that the physical size became known.

This changed some in SIMION 8.1.0, where the X, Y, Z cell lengths in mm can optionally be stored in the PA file itself. This feature is somewhat necessary for the *Poisson Solver in SIMION*, where the absolute physical size of the array (and its space-charge) is not arbitrary as is true in the Laplace solver. This feature is also helpful when grid cells are anisotropically scaled (i.e. X, Y, Z cell lengths are different), so the Laplace solver needs to know these cell lengths (actually only their relatively lengths, although having absolute lengths doesn't hurt). This feature is also convenient in visualizing PAs not yet added to IOB's–e.g. seeing real mm units in the Modify 3D preview screen.

SIMION 8.1.0 still allows you to interpret these PA cell lengths as relative values, if you wish, and you are still allowed to rescale instances of the arrays on the workbench using a scaling factor. In particular, SIMION 8.1.0 always assumes SIMION 8.0 formatted PA files have cell lengths of 1 mm/gu (i.e. mm = gu), so any PA instance scaling factor that multiplies these lengths can still be understood as a mm/gu scaling factor if you wish, so full backward compatiblity with SIMION 8.0 is maintained.

PA Changes

Three additional values are stored in the potential array: dx_mm, dy_mm, and dz_mm. These are the mm/gu scaling factors in the X, Y, and Z directions respectively. These values default to 1 mm/gu. They are normally limited to the range 1.0e-6 to 900 mm/gu, though this is an abitrary limit. For 2D cylindrical arrays, dz_mm is always equal to dy_mm. For 2D planar arrays, dz_mm indicates the length of a grid unit in the Z direction when an instance of the array is placed into the workbench. These values are defined on the "New PA" screen and can also be changed from the *Set* button on the Modify screen. They can also be defined in the GEM file (see below) or STL import screen (see below).

Please note that storing grid cell sizes other than 1 mm in the PA file will affect the interpretation of various parameters such as PA instance scale factor, PA instance working origin, magnetic PA scaling factor (ng), and GEM file units. These points are discussed more below.

Effect on Refine

The *relative* magnitudes of the three mm/gu values are used by the Refine function to properly calculate the field under the Laplace equation. Strictly speaking, only the *relative* ratios dy_mm/dx_mm and dz_mm/dx_mm matter. The *absolute* values of dx_mm,

dy_mm, and dz_mm are important, however, if you use Refine to solve a system with space-charge under the Poisson equation (*Poisson Solver in SIMION*).

Effect on the Workbench and View Screen

The PA instance "scale" factor (on the View screen, PAs tab, Positioning panel) will multiply the grid unit sizes (dx_mm, dy_mm, dz_mm) defined in the PA file. In SIMION 8.1, the PA instance scale factor should normally be set to 1, and in some cases–mainly arrays solved with the *Poisson Solver in SIMION*– it can be required to be set to 1. SIMION 8.0 PA files, always have cell sizes of $(1\ mm)^3$, and in that case the PA instance "scale" factor can be interpreted as being a mm/gu unit conversion factor.

It is normally valid to scale arrays sizes on the View screen by identical scaling factors in the X, Y, and Z directions (i.e. scale factor not equal to 1) without re-refining the array. The exception is arrays solved by the Poisson equation since the absolute array size affects the magnitude of the space-charge density, which in turn affects the potentials inside the array (for details, see *Effect of Scaling After Refining*).

For 2D planar arrays, the cell length in the Z direction (dz_mm) affects the length of the corresponding array instance projected onto the workbench. The "nz use" field ("View screen, PAs tab, Positioning panel, nz use field) is in units of grid units in the Z direction, so dz_mm determines how long each of those grid units is.

The PA Instance "PA working origin" (X/Y/ZWo, also on the Positioning panel) is now in units of mm as exists prior to the application of any scaling factor. In SIMION 8.1, the PA instance scale factor should normally be set to 1, so there is no difference between physical mm units (i.e. after application of the scaling factor). For SIMION 8.0 PA files, which always have cell sizes of $(1\ mm)^3$, the PA working origin units can be interpreted as grid units (gu).

If you change the grid unit sizes in the PA after saving an IOB file containing the PA, you will on reloading the IOB get a warning about dimensions being changed, and SIMION might be able to guess what needs to be changed in the corresponding PA instance and offer to make some adjustments. It can be common, for example, to change the density of grid cells in a PA without changing its physical size on the workbench in order to trade-off calculation efficiency for accuracy. If in a PA file you increase the number of grid cells in each dimension by some factor and reduce the grid unit size by the same factor (e.g. as the Double and Halve functions on the Modify screen do), then the overall size in mm of the PA will remain unchanged, and the array will also remain in the same position when the workbench (IOB) is reloaded. In particular, the scale and PA working origin parameters will not need to be changed under the current definitions of these parameters.

Effect on Magnetic Arrays (ng Scaling Constant)

Magnetic potential arrays have a “ng” scaling factor that scales the potential gradients so that they are in units of Gauss. The ng scaling factor is now specified in mm (rather than grid units). If, for example, you have a magnetic poles that are separated by 10 grid units in the Y direction and the Y grid unit size dy_mm = 6, then ng should be set to 60 mm. If in doubt, check by measuring the field with your mouse in the View screen.

If the cell sizes in your PA file are $(1\ mm)^3$, as is always true in SIMION 8.0 PA, the ng may also be interpreted as grid units (gu), so compatibility is maintained with SIMION 8.0.

Effect on the Fly’m

The trajectory quality factor (TQual) defines the time-step size as a function of the minimum grid unit size for the active electric and/or magnetic arrays the particle is currently located inside. For example, TQual = 0 adjusts the time-step size so that the particle travels approximately one grid unit per time-step. With anisotropically scaled arrays, the minimum dimension of the grid cell is used for this purpose.

PA File Format

The older (“version 1”) potential array format used by SIMION 7.0 and 8.0 does not support the dx_mm, dy_mm, and dz_mm scaling values. Therefore, potential arrays where at least one of the dx_mm, dy_mm, or dz_mm values is not 1 are instead saved in “version 2” of the potential array format. Version 2 potential arrays are not readable in SIMION versions prior to SIMION 8.1. You will get an error message “Failed loading potential array file (x.pa). Not a recognized PA file type” if you attempt to load such an array in an earlier version of SIMION.

Advanced note: Version 2 potential array files have their mode field set to -2 (i.e. version 2), and the following values have been appended to the file header:

```
double dx_mm;
double dy_mm;
double dz_mm;
```

IOB Format Changes

IOB files containing PAs stored with grid cell sizes other than $(1\ mm)^3$ are written in IOB format version 2 (rather than 1). IOB format version 2 files loaded into earlier versions of

SIMION will give the (somewhat nonsensical) error “Aborted: Too Many Instances in IOB file”.

Advanced note: IOB format version 2 files contain 52 additional bytes inserted at the beginning of the file. Bytes 0-7 are `"SIMIOB\xFF\x7F"`. Bytes 8-47 are currently ignored. Bytes 48-51 are the integer 0x7fffffff. PA header records in IOB format version 2 files are always in PA format version 2, even if the PAs themselves are formatted in an earlier version.

Refine

The Refine function has been extended to support solving the Laplace equation on anisotropically scaled PA files.

Geometry Files

Four new optional parameters were added to the pa_define command in GEM files:

```
pa_define(nx,ny,nz,symmetry,mirroring,type,ng,dx_mm,dy_mm,dz_mm,gu
```

`dx_mm`, `dy_mm`, `dz_mm` - mm/gu scaling factors in x, y, and z directions. If `dx_mm` is omitted, it defaults to 1. If `dy_mm` is omitted, it defaults to `dx_mm`. If `dz_mm` is omitted, it defaults to `dy_mm`. For 2D cylindrical arrays, `dz_mm` must be equal to `dy_mm`. For 2D planar arrays, `dz_mm` should still be specified since this affects the interpretation of the Z cell size (`nz use` parameter) when the array is placed in the workbench. Using any scaling value than other than 1 will make the PA be saved in PA format version 2, which is not compatible with older SIMION versions.

If the literal string `gu` argument is provided, the default unit scale for objects in the GEM file will be in grid units; otherwise it will be mm.

Three new optional parameters were added to the locate command in GEM files:

```
locate(x,y,z,scale, az,el,rt, scalex,scaley,scalez)
```

`scalex`, `scaley`, `scalez` - X, Y and Z scaling factors. Any of these omitted default to 1 (no scaling). If any of these are specifed, then scale must be omitted: `locate(x,y,z,, az,el,rt, scalex,scaley,scalez)`.

Note that all scaling factors are applied after any rotations (as described in the SIMION

manual in the GEM chapter in the section on `locate`).

The `within`/`notin` commands (unlike `within_inside*`/`notin_inside*`) shift the surfaces by 0.5 gu in the direction of the surface normal, as described in the manual section "I.2.1. Classes of Instructions: Within class". With anistropically scaled grid cells, the size of 0.5 gu will depend on the direction of each surface normal. (This is properly handled in versions >= 8.1.0.22.)

A simple example is located at `examples\geometry\sphere_aniso.gem`:

```
; sphere_aniso.gem - This is a simple demonstration of creating an
; anisotropically scaled PA with a GEM file.
; This example requires SIMION >= 8.1.0.

# local R_mm  = 10    -- sphere radius, mm
# local dx_mm = 0.5   -- x cell size, mm
# local dy_mm = 0.1   -- y cell size, mm
# local dz_mm = 2     -- z cell size, mm

pa_define($(R_mm/dx_mm+1),$(R_mm/dy_mm+1),$(R_mm/dz_mm+1),
          planar, xyz, electrostatic,, $(dx_mm),$(dy_mm),$(dz_mm))

locate(0,0,0,, 0,45,0) {
  electrode(1) { fill { within { sphere(0,0,0, $(R_mm)) } } }
}
    ; note: locate performs rotions, then scalings, then translati
```

The following is equivalent if the "gu" option is used:

```
# local R_mm  = 10    -- sphere radius, mm
# local dx_mm = 0.5   -- x cell size, mm
# local dy_mm = 0.1   -- y cell size, mm
# local dz_mm = 2     -- z cell size, mm

pa_define($(R_mm/dx_mm+1),$(R_mm/dy_mm+1),$(R_mm/dz_mm+1),
          planar, xyz, electrostatic,, $(dx_mm),$(dy_mm),$(dz_mm),

locate(0,0,0,, 0,45,0, $(1/dx_mm),$(1/dy_mm),$(1/dz_mm)) {
  electrode(1) { fill { within { sphere(0,0,0, $(R_mm)) } } }
}
```

If you do this often, you may want to define some macros that eliminate the repetition:

```
# -- First define some macros to simplify definitions....
# function simple_array(t)
#   _put(('pa_define(%d,%d,%d,planar,xyz,electrostatic,,%g,%g,%g,g
#         'locate(,,,,,,, %g,%g,%g) {\n'):format(
#           t.x_mm/t.dx_mm+1,t.y_mm/t.dy_mm+1,t.z_mm/t.dz_mm+1,
#           t.dx_mm,t.dy_mm,t.dz_mm, 1/t.dx_mm,1/t.dy_mm,1/t.dz_m
# end
# function end_simple_array() _put "}\n" end

# -- Now use the macros...
# local R_mm  = 10    -- sphere radius, mm
# simple_array{x_mm=R_mm,y_mm=R_mm,z_mm=R_mm, dx_mm=0.5,dy_mm=0.1,
electrode(1) { fill { within { sphere(0,0,0, $(R_mm)) } } }
# end_simple_array()
```

New options were added to the `gem2pa` batch mode command:

```
simion.command("gem2pa --dx_mm 0.5 --dy_mm 2.0 --dz_mm " ..
               "0.3333333333333333 --scalex 2.0 --scaley 0.5 --sc
```

`dx_mm`, `dy_mm`, `dz_mm` - mm/gu scaling factors in x,y,z directions. If any of these are omitted, they default to the values defined or assumed in the GEM file (pa_define). The GEM file defaults to 1 mm/gu. Using any value other than 1 will make the PA be saved in PA format version 2, which is not compatible with older SIMION versions.

`scalex`, `scaley`, `scalez` - scaling in X, Y, and Z directions. Any of these omitted default to 1 (no scaling). If any of these are specified, then scale cannot be specified.

Note: `x`, `y`, and `z` shifts are in grid units.

STL Import

The SL Tools (sltools.exe) STL <-> PA functions have been extended to support mm/gu scale factors in the X, Y, and Z direction. In STL -> PA import, uncheck the “square cells (x=y=z)” option. The PA will be saved in SIMION 8.1.0 PA version format if necessary.

User Programs

The `ion_mm_per_grid_unit` reserved variable reflects the minimum grid dimension of the electric and magnetic potentials arrays that the particle is currently active in. This is the value that the trajectory quality is calibrated against. If you need the mm/gu for a particular dimension of a particular array, instead use the Lua potential array interface

(below).

The `ion_bfield(xyz)_(gu|mm)` reserved variables correctly handle the anisotropic scaling. These are always in units of Gauss.

Lua Potential Array Interface

The Lua potential array interface has been extended to support anisotropically scaled arrays:

```
local pa = simion.pas[1]

-- mm/gu scaling factors can be get or set
print(pa.dx_mm)
print(pa.dy_mm)
print(pa.dz_mm)
pa.dx_mm = 10
pa.dy_mm = 20
pa.dz_mm = 50
```

SL Libraries

The SL Libraries have been updated to support anisotropically scaled arrays. The newer Lua interface also supports anisotropically scaled arrays (see above).

Double and Halve Functions

The Double and Halve functions have been generalized to support multiplying or dividing the X, Y, and Z directions independently by positive integers.

3D Visualization

The SIMION Modify “3D” (and SL Tools View) and View screen display arrays with proper anisotropic scaling. The Modify XYZ screen displays without anisotropic scaling because this is faster. The Fast Adjust screens also display without anisotropic scaling.

In the View screen 3D Iso view, the “Constr(ai)n” optimization (checkbox on the left of the screen) is not applied for anitropically scaled PAs. Therefore, rendering the 3D Iso view is slower and the display quality not as high as when isotropically scaled PAs are displayed. The assumptions made by this optimization (i.e. lines through far corners of grid cell cubes are perpendicular to the screen) are not really compatible in general with

anisotropically scaled arrays.

Gradient Contour Plots

The potential contours, gradient contours, and PE views should work with anisotropically scaled arrays.

Other Assorted Changes

- The ”?” button on the View screen, PAs panel displays effective mm/gu values in each dimension.
- GEM import screen x/y/z scaling factors were added.
- In the View screen 3D Iso view, arrays with different scales in the X, Y, and Z directions are always drawn as if the “Constrn” checkbox is unchecked.
- *simionx.FieldArray - 3D vector field* (SIMION Example: field_array) was extended for anisotropic grid units.
- In *FLY2 files*, if “coordinates relative to” is a PA instance, then particle positions are taken in x, y, and z grid units. However, any direction vectors take relative mm/gu into account. For example, if a PA is 10 mm/gu in X and 1 mm/gu in Y, and a FLY2 uses PA coordinates with a direction vector(1,1,0), then this is still a 45 degree angle on the workbench. This behavior might be subject to change in the future.

Other [FIX-TODO] Items

- Changes to array scaling via *simion.pas* inside the View screen may not properly update workbench and graphics. This is not specific to anisotropically scaled PAs (e.g. other symmetry/size changes also have this issue).
- In PA API, possibly add pa.format_version (1 or 2) variable.

See Also

See *Electrode Surface Enhancement / Fractional Grid Units* for more accurately handling surfaces that do not align to PA grid units (e.g. curved surfaces).

Changes

- 8.1.1.1 - SLTools(STL->PA): Fixed non-“square cells” with “region rotate” defined. mm/gu sizes were being applied to wrong axes, causing obvious distortions to geometric aspect ratio.
- 8.1.0.22 - Fixed GEM file handling of 0.5 grid unit (gu) surface adjustment in

within/notin commands when using anisotropically scaled grid cells. See above and *SIMION Software Change Log* for more details.