Input Geometry format

The geometry can be input either using an `hdf5` file format or a MATLAB struct stored as .mat format. The geometry is comprised of two half infinite half spaces, and a bunch of smaller components. The geometry is defined through a collection of vertices, curves and regions. Curves terminate, pass through or end at vertices, and regions are defined as the signed union of curves.

0.1 Types of curves

- `curvetype`=1: the curve is defined as the union of line segments
- `curvetype`=2: a sector of a circle passing through (a, y_0) and (b, y_0) with opening angle θ
- `curvetype`= 3: a sine curve with n half-wiggles passing through $(a, 0)$ and $(b, 0)$ and prescribed amplitude

$$(x(t), y(t)) = \left(a + (b-a) \cdot \frac{t}{n\pi}, Asin(t)\right) \quad t \in (0, n\pi)\,. \tag{0.1}$$

Note that for the last two curve types, the y components of the vertices must be identical. Support for rotated versions will be made available soon. See figure section 0.4 for some illustrative curves.

There will also be two sem-infinite lines $(-\infty, 0) \to (a, 0)$ and $(b, 0) \to (\infty, 0)$ where $(a, 0)$ and $(b, 0)$ are specified by the user. These should not be input as curves, and will be automatically generated later, only the left and right vertices need to be specified.

0.2 hdf5

The hdf5 file is structured as a collection of attributes for the geometry, a group called `curves` where each curve is a subgroup of the that group, and a group called `regions` where each region is a subgroup of that group. For an example, see `test1.h5`.

0.2.1 Attributes

Attributes unless stated otherwise are mandatory.

- `ndomain`: Number of domains
- `ncurve`: Number of curves
- `verts`: (2,nverts), (x, y) coordinates of the vertices. The dataset must contain at least one vertex.
- `rnr`: (ndomain,1), real part of the refractive indices of each domain (optional, default value will be set to 1)
- `rni`: (ndomain,1), imaginary part of the refractive indices of each domain (optional, default value will be set to 0)

- `mode`: 'te' or 'tm', if unspecified or doesn't match either of the two options, the default mode will be 'te' (optional).
- `lambda`: wavelength: Wavelength of the incident light. If unspecified, default value is 0.38.
- `lvert`: verts(:,lvert) is the left vertex defining the negative half infinite line, i.e. the half line will extend from $(-\infty, 0) \to (\text{verts}(1, \text{lvert}), 0)$
- `rvert`: verts(:,rvert) is the right vertex defining the positive half infinite line, i.e. the half line will extend from $(\text{verts}(1, \text{rvert}), 0) \to (\infty, 0)$
- `xylim`: (4,1) `xmin`, `xmax`, `ymin`, `ymax` for plotting the fields, optional, if unspecified, let d denote the size of the smallest square containing all vertices and let c denote the center of that square, then `xylim` will be set to a square centered at c of size $2d$.
- `ngr`: number of targets in each dimension. If unspecified, `ngr` = 300.

0.2.2 Curves group

The `curves` group, has `ncurve` subgroups labelled $1, 2, \ldots$ `ncurve`. Each subgroup has the following attributes

Curve attributes

- `curvetype`: Current supported curvetypes are 1, 2, or 3, see above.
- `curve_id`: curve identifier, must be between $1, 2, \ldots$ `ncurve`.
- `vert_list`: For curve type 1, must be the list of vertices defining the union of line segments. For example, if `vert_list` = [1, 4, 3] for curve type 1, then the curve would be defined as the line segments $\text{verts}(:, 1) \to \text{verts}(:, 4) \to \text{verts}(:, 3)$. If `curvetype` 2 or 3, vert_list must contain two vertices, the points (a, y_0) and (b, y_0) are given by $(a, y_0) = \text{verts}(:, \text{vert\_list}(1))$ and $(b, y_0) = \text{verts}(:, \text{vert\_list}(2))$.
- `theta`: If `curvetype`=2, θ defines the opening angle (optional, if unspecified, it is set to pi/2).
- `ifconvex`: If `curvetype`=2, `ifconvex` = 0, implies the center of the circle is below the y coordinate of the prescribed points, and `ifconvex` = 1, implies the center of the circle is above (optional, if unspecified `ifconvex` =0). Note: if `ifconvex` = 0, then the curve is traversed from $(b, y_0) \to (a, y_0)$. Be mucho careful in this setting.
- `nwiggles`: If `curvetype`=3, `nwiggles` = n above (optional, if unspecified, `nwiggles` = 1).
- `amplitude`: If `curvetype`=3, `amplitude` = A above (optional, if unspecified, `amplitude` = 1).

0.2.3 Regions group

The `regions` group, has `ndomain` subgroups labelled $1, 2, \ldots$ `ndomain`. Each subgroup has the following attributes

Region attributes

- `region_id`: region identifier, must be between $1, 2, \ldots$ `ndomain`.
- `icurve_list`: list of curves which define the region. The sign of the curve denotes the orientation in which it is traversed. For example `icurve_list` $= [1, -4, 3]$ implies that the region is defined by traversing curve 1 forward, traversing curve 4 backwards followed by traversing curve 3 forwards.

  A note on specifying the infinite regions. Recall that the user doesn't need to define the left and right semi-infinite lines defining the half planes. For the upper half plane, the user should define the traversal of the rest of the curves such that when the left line segment is appended to the start of the `curve_list` and the right line segment is appended to the end of it, then that union would define the positively oriented boundary of the upper half plane. For the lower half plane, the right line segment traversed backward will be appended to the start of the list and the left line segment traversed backward will be appended to the end of the list.
- `is_inf`: Whether the region is infinite or not. For bounded regions, `is_inf` =0, for the region containing $y \to +\infty$, `is_inf` = 1, and for the region containing $y \to -\infty$, `is_inf` = -1. Note that there should be exactly one region with `is_inf` = 1, and one region with `is_inf` = -1.

0.3 MATLAB struct

The same data can be provided as a MATLAB struct as well. The `.mat` file should contain a single struct called `geom_class`. The groups `curves, regions` are cell arrays, and the attributes are variables of the struct or the cell array.

For an example, see `test1.mat`.

0.4 Example

Consider the geometry in `test1.h5` or `test1.mat` shown in section 0.4. We have 6 vertices as labelled in the figure, 8 `curves` and 5 `regions`. The `attributes` or `variables` are defined to be as follows

- `ndomain` = 5
- `ncurve` = 8
- `lvert` = 1
- `rvert` = 2

For the group or cell `curves`, the attributes or variables are given by:

- Group /`curves`/1 or `curves`{1}

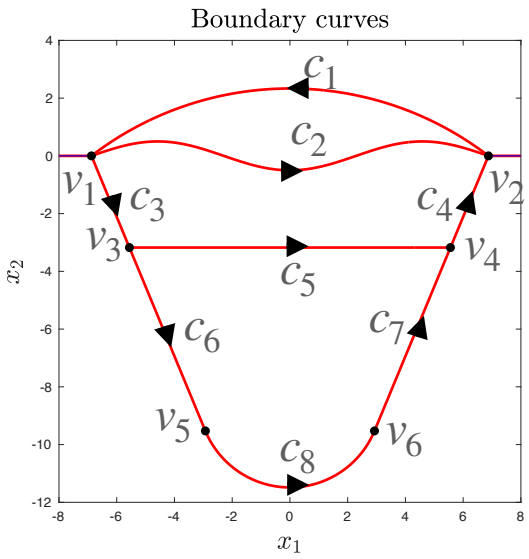

Figure 1: Example geometry in test1.h5 and test1.mat

  - `curvetype` = 2
  - `curve_id` = 1
  - `vert_list` = $[1, 2]$
  - `ifconvex` = 0
  - `theta` = $\pi/2.4$

- Group `/curves/2` or `curves{2}`
  - `curvetype` = 3
  - `curve_id` = 2
  - `vert_list` = $[1, 2]$
  - `nwiggles` = 3
  - `amplitude` = 0.5

- Group `/curves/3` or `curves{3}`
  - `curvetype` = 1
  - `curve_id` = 3
  - `vert_list` = $[1, 3]$

- Group `/curves/4` or `curves{4}`
  - `curvetype` = 1
  - `curve_id` = 4
  - `vert_list` = $[4, 2]$

- Group `/curves/5` or `curves{5}`

  - curvetype $= 1$
  - curve_id $= 5$
  - vert_list $= [3, 4]$

- Group /curves/6 or curves{6}
  - curvetype $= 1$
  - curve_id $= 6$
  - vert_list $= [3, 5]$

- Group /curves/7 or curves{7}
  - curvetype $= 1$
  - curve_id $= 7$
  - vert_list $= [6, 4]$

- Group /curves/8 or curves{8}
  - curvetype $= 2$
  - curve_id $= 8$
  - vert_list $= [5, 6]$
  - ifconvex $= 1$
  - theta $= 3\pi/4$

For the group or cell regions, the attributes or variables are given by:

- Group /regions/1 or regions{1}
  - region_id $= 1$
  - icurve_list $= [-1]$
  - is_inf $= 1$

- Group /regions/2 or regions{2}
  - region_id $= 2$
  - icurve_list $= [-4, -7, -8, -6, -3]$
  - is_inf $=$ -1

- Group /regions/3 or regions{3}
  - region_id $= 3$
  - icurve_list $= [1, 2]$
  - is_inf $= 0$

- Group /regions/4 or regions{4}
  - region_id $= 4$

  - `icurve_list` $= [-2, 3, 5, 4]$
  - `is_inf` $= 0$

- Group /`regions`/5 or `regions`{5}
  - `region_id` $= 5$
  - `icurve_list` $= [-5, 6, 8, 7]$
  - `is_inf` $= 0$