1 Introduction

This will provide an introduction to the tissue model generation function, skin_create. It is important to note that the function’s name is a little misleading because this function is flexible enough that it can be utilized to create many types of biological tissue. It is important to note that this manual assumes a basic knowledge of MATLAB programming.

2 Getting Started

The code skin_create creates and discritizes a tissue model with set optical properties. The spatial location and number of cellular components are randomized each time.

3 Calling the program

Call the FDTD program as follows:
[media] = skin_create(media,Ewave,options);

The inputs media, and options are structures that are described in Sections 3.1, and 3.2, respectively. A brief description of each input is provided in Table 1. The structure Ewave only requires a single field.
Ewave.lambda0 – wavelength of the electromagnetic wave to be simulated.
### 3.1 media

This structure contains all the basic information that describes the model space size

- `.depth` – The depth of the medium
- `.width` – The width of the medium
- `.randstate` (optional) – This provides `skin_create` with a randomstate to begin with. This is provided so a user can regenerate a particular medium.

Note: all these fields are called as follows: 
`media.XXXX`
where `XXXX` is the field described above.

### 3.2 options

This structure is a ‘catch all’ that allows the user maximum versatility with the `skin_create` function without having to edit the actual code. The options structure can change the number of subcellular components, number of layers, starting place for each layer and so on. Because `options` allows the user to completely specify the tissue geometry, it has many substructures that will be explained below.

#### 3.2.1 options.cell

This substructure contains all the information about the cellular components. These are listed in Table 3.2.1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>media</td>
<td>Structure that stores information about medium.</td>
</tr>
<tr>
<td></td>
<td>(See Section 3.1)</td>
</tr>
<tr>
<td>Ewave.lambda0</td>
<td>Wavelength of field that will propagate through medium.</td>
</tr>
<tr>
<td>options</td>
<td>structure specifies options for the user</td>
</tr>
<tr>
<td></td>
<td>(See Section 3.2)</td>
</tr>
</tbody>
</table>

Table 1: The inputs for the function `execute_FDTD`.
### Variable | Purpose | Default
---|---|---
.posd | Initial depth for the cell layers within the tissue. | 0
.drad0 | Maximum radius in the depth dimension. | 5 µm
.xrad0 | Minimum radius in the x-direction | 20 µm
.rad | Final radius of the cell | 10 µm

### 3.2.2 options.indicies

### 3.2.3 options.mito

### 3.2.4 options.mel

### 3.2.5 options.DE

## 4 Outputs

After execution, `execute_FDTD` will automatically store the structures in `media` and `Ewave`.

### 4.1 Ewave

In addition to the input fields, the structure `Ewave` will also contain the following:

- `.Ez_field` – real electric field (z-direction) for the final time-step of the
<table>
<thead>
<tr>
<th>Variable</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>.mlrad</td>
<td></td>
</tr>
<tr>
<td>.nummelanin</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>.depth</td>
<td></td>
</tr>
<tr>
<td>.period</td>
<td></td>
</tr>
<tr>
<td>.DEamp</td>
<td></td>
</tr>
</tbody>
</table>

• .Hx_field – real magnetic field (x–direction) for the final time-step of the simulation

• .Hy_field – real magnetic field (y–direction) for the final time-step of the simulation

If you also specified the necessary information for a detector, you will also find the signal at the detector–depth versus time in Ewave.detector.timesig