



版本更新: Sim4Life V9.6 桌面版



https://zmt.swiss/assets/downloads/Sim4Life/linked_download/9-6/releasenotes.txt



Sim4Life V9.6 桌面版更新亮點

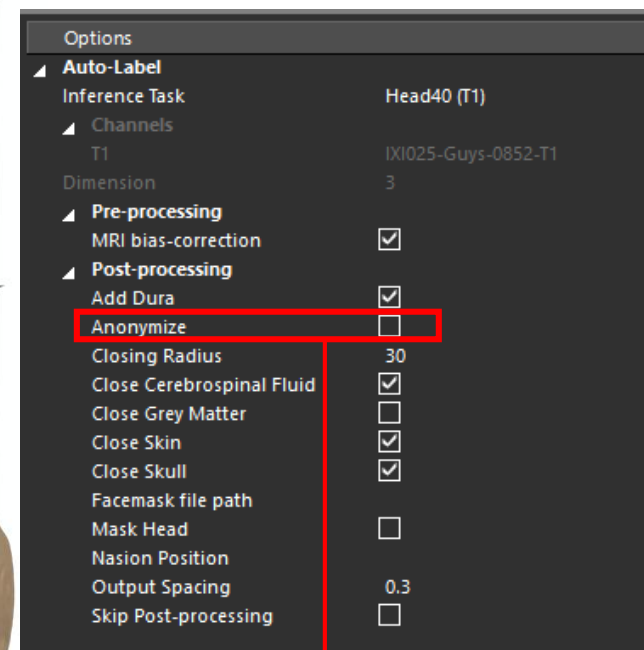
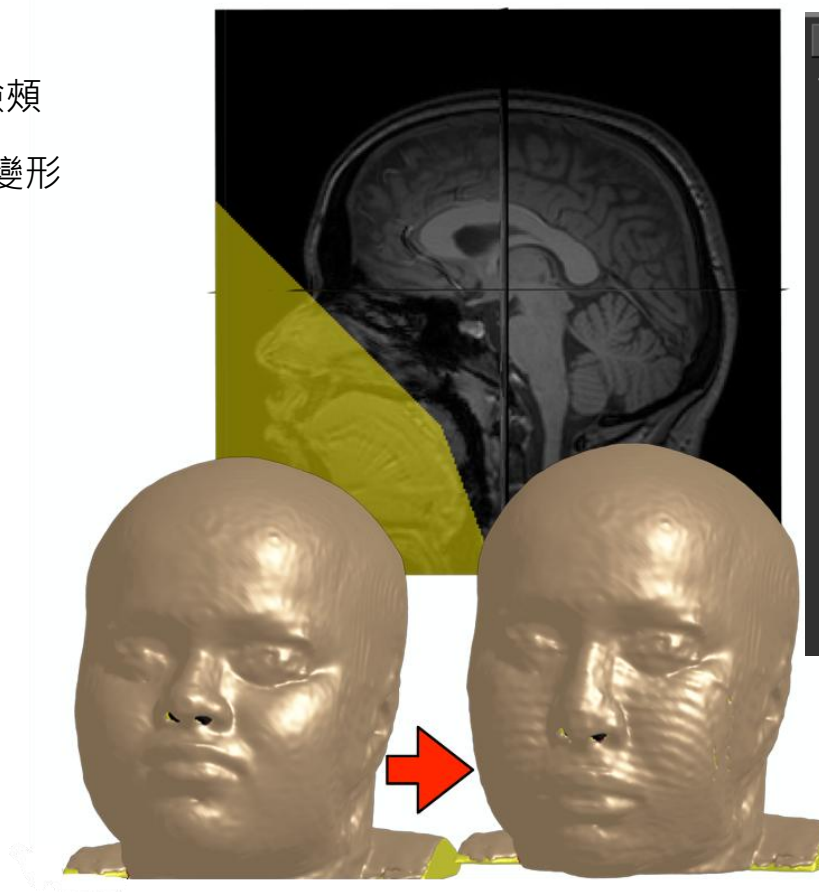
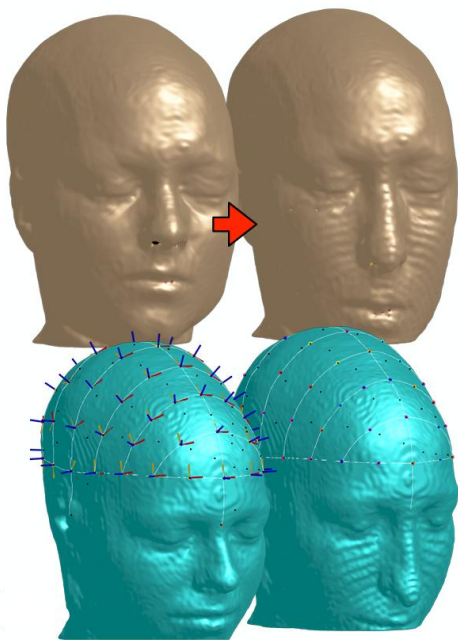
- **更快的神經反應預測:** 廣義活化功能(GAF)現已支援臨床真實的 MRG 雙線軸突模型，能在降低神經元準確度的同時實現計算時間從小時/天到秒/分鐘不等
- **自動化招募曲線分析:** 新的整合工具能在不同神經族群中產生並比較招募曲線，加速神經刺激裝置的評估與治療規劃
- **提升效能與可擴展性:** 專案啟動速度更快，大型解剖/神經模型處理更佳，並支援 NVIDIA Blackwell 雲端 GPU 以應付高要求的優化工作負載
- **提升使用者體驗與自動化:** 更新 Python 腳本環境，並改善了專案載入與初始化時的回饋



模型建置(Modeling)

- 支援**模型匿名化**，確保資料處理符合隱私保護需求，且可與 TIP 工具搭配使用
- 只對與身分辨識相關的臉部區域進行匿名化，同時保留腦部、頭骨分割與電極位置，使匿名化前後的腦部電磁場分布盡可能一致
- 主要方法包括：

- (1) 臉部區域遮罩: 眼睛、嘴巴、下巴、臉頰
- (2) 在遮罩區域內進行影像與分割資料的變形



Anonymize

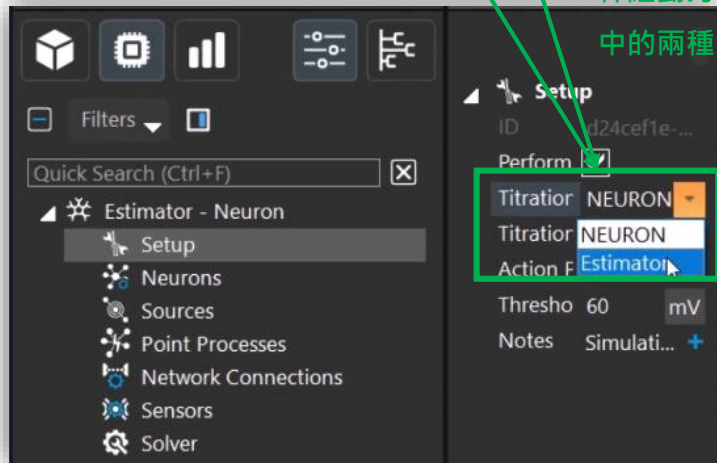
If enabled, the face region is morphed to anonymize the subject.

模擬(Simulation)

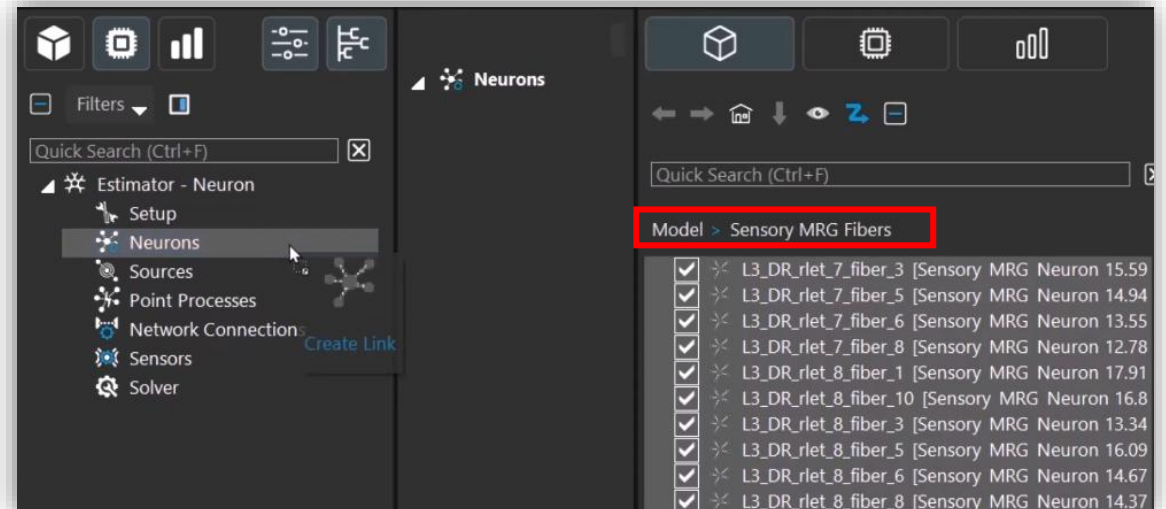
(1) 神經求解器(T-Neuro)

- 為治療規劃流程中使用的高精度生物物理模型提供快速神經反應預測功能
- 加速版 GAF 預測器(Estimator)現已支援 MRG 類型雙電纜(Double-Cable)軸突模型，可用於模擬感覺神經纖維與運動神經纖維
- 與傳統神經元模擬(NEURON)相比，神經活化閾值的預測速度最高可提升 1000 倍以上

神經動力學求解器
中的兩種滴定策略



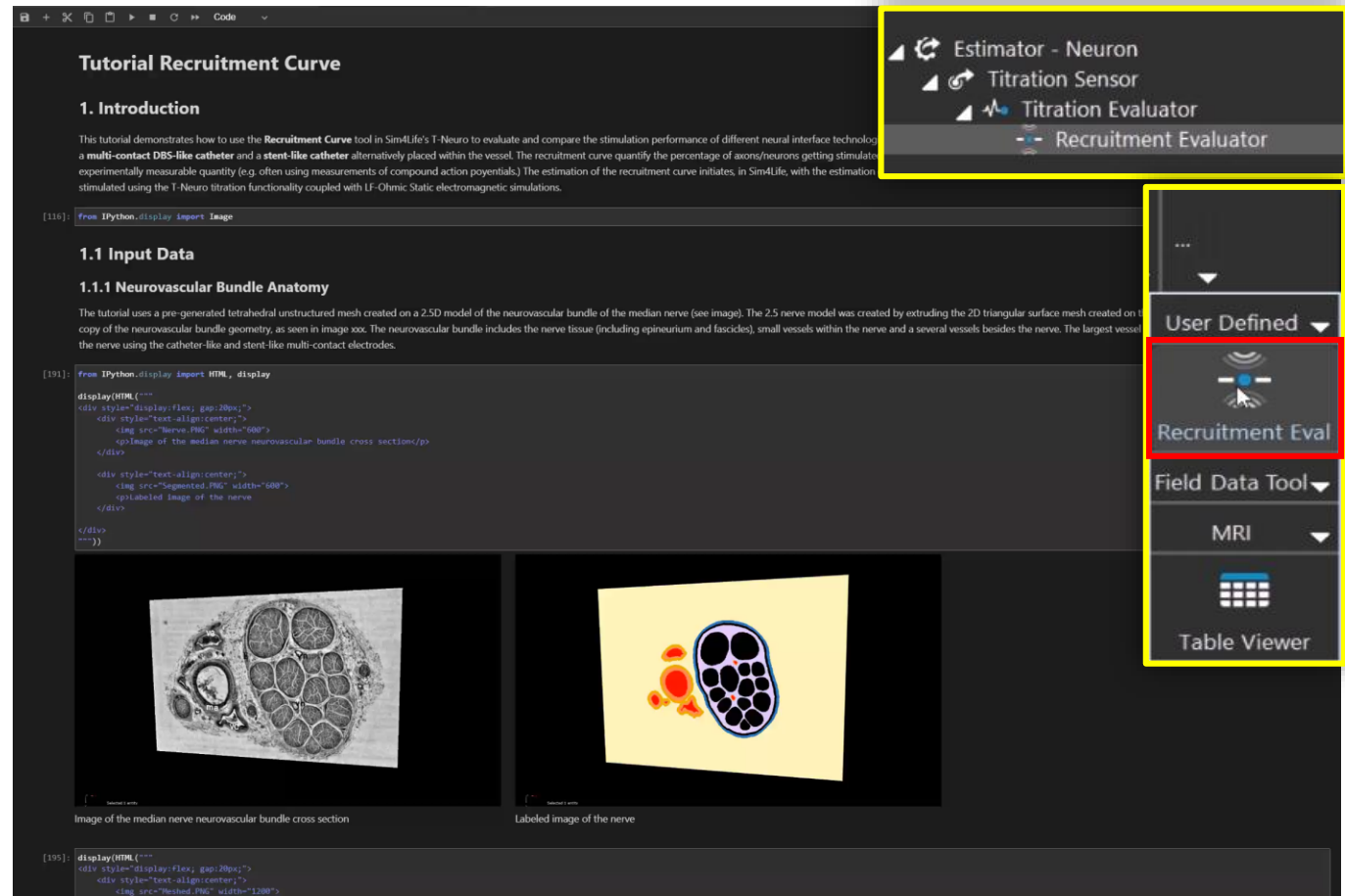
驗證實驗重現已發表的臨床脊髓刺激(SCS)規劃流程，在不到 10 分鐘內可完成完整的 16 電極募集圖，而使用傳統 NEURON 方法則需要超過 24 小時



模擬(Simulation)

(1) 神經求解器(T-Neuro)

- 新增**自動化募集曲線(Recruitment Curve)**分析功能，可根據刺激振幅評估神經纖維的活化情況，並支援不同刺激策略之間的直接比較，簡化神經刺激研究流程
- 最新教學範例透過比較三種**正中神經(Median Nerve)**刺激神經介面技術，展示 Sim4Life 如何進行**自動化進行募集曲線(Recruitment Curve)**分析
 - 袖套式電極(Cuff Electrode)
 - 支架型血管內電極
(Stent-like Endovascular Electrode)
 - 導管型血管內電極
(Catheter-like Endovascular Electrode)

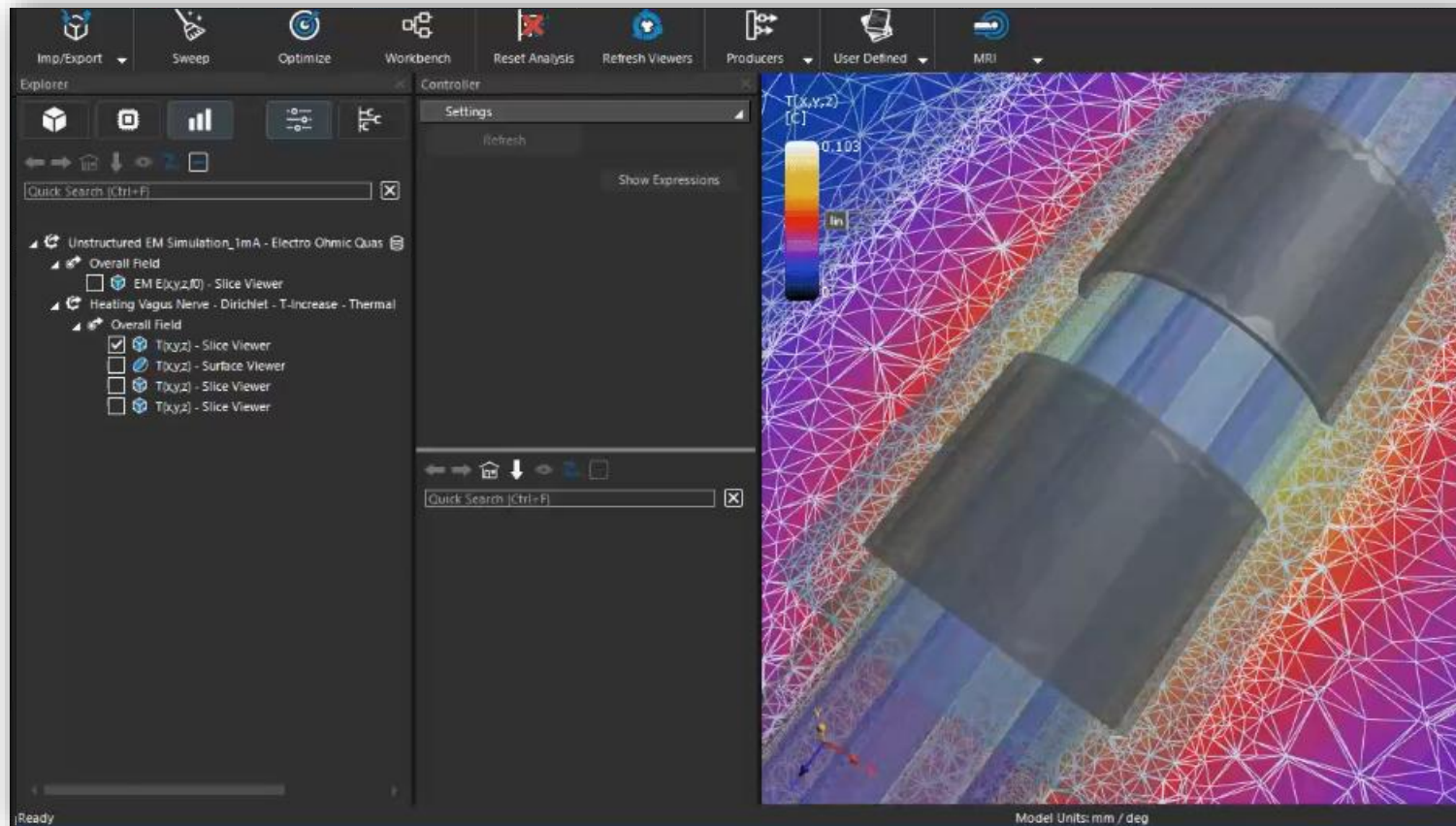


- ✓ 全新的 **Recruitment Curve Evaluator (募集曲線評估器)** 已整合至 T-Neuro 模組，並結合電磁與電生理耦合模擬技術，能夠更完整地反映神經刺激的實際效果
- ✓ 使用者可自動產生**神經募集曲線(Recruitment Curves)**、**量化評估刺激效能與神經募集特性**、使用系統提供的**預設評估指標**或建立符合特定研究或應用需求的**自訂評估指標**

模擬(Simulation)

(2) 熱求解器

- 強化內插(Interpolation)處理機制



Sim4Life V9.6 中的非結構化熱模擬網格處理

- ✓ 改良非結構化網格生成技術，以及優化直角座標網格(Rectilinear Grid)與非結構化網格(Unstructured Grid)之間的內插運算，有效提升模擬結果的數值精度
- ✓ 新增多項合理性驗證機制，進一步提高模擬結果的可靠性與可信度



模擬(Simulation)

(2) 熱求解器

- 新增**合理性檢查**功能，用於驗證作為熱模擬來源的電磁模擬區域，是否與熱模擬區域存在重疊，以確保模擬設定的有效性

The screenshot displays the Auden software interface. On the left, a tree view shows the simulation setup: Heating - Thermal Transient, Setup, Materials (1), Initial Conditions (1), Sources (1), Boundary Conditions (1), Sensors (2), Grid 78 x 78 x 78 = 474.552 kCells (1), Voxels (1), and Solver. The 'Sources (1)' folder is expanded, and 'Transient Heat Source (1)' is selected. The 'Controller' window for this source is open, showing various settings. The 'Allow Partial Domain Overlap' checkbox is checked and highlighted with a red box. A red arrow points from this checkbox to a text box on the right titled 'Allow Partial Domain Overlap' which explains the function.

Controller

Transient Heat Source

Source Type: Simulation Link

Treat as Port: ☐

Allow Partial Domain Overlap: ☒

Normalize: ☐

Coherent Fields: ☐

Frequency: 8.35e+08 Hz

Interval Mode: User Defined Interval

Start Time: 1800 s

Stop Time: 3600 s

Dipole / Field Sensor Settings (Overall Field)

Amplitude Factor: 1

Allow Partial Domain Overlap

Controls how the interpolation of heat sources handles domain boundaries. When unchecked, the thermal grid must be fully enclosed by the heat source grid (EM simulation, cache file, or analysis pipeline). When checked, partial overlap is allowed, and regions outside the heat source grid will use a default heat-source contribution of 0.

模擬(Simulation)

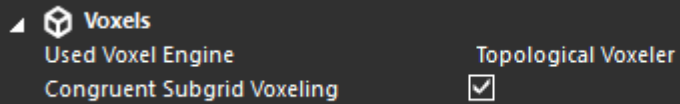
(3) FDTD 電磁求解器

- 新增檢查機制，以確認遠場感測器(Far-Field Sensor)周圍是否保留足夠的緩衝區(Padding)

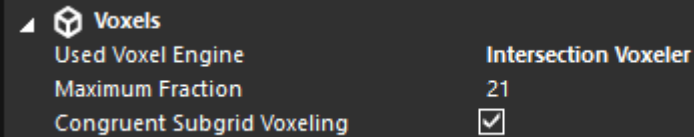
A dark grey rectangular button with a circular arrow icon on the left and the text "New Far-Field Sensor" in blue.

(4) Voxeling (體素化)

- 統一 Topological Voxeler 與 Intersection Voxeler 的邊界法向量(Edge Normal)計算方式，並加入一致性驗證測試(Parity Validation Tests)
- 提升 Topological 與 Intersection 體素化方法在熱模擬工作流程中的相容性與結果一致性

A dark grey settings panel for "Voxels". It has a small triangle icon on the left. The text "Used Voxel Engine" and "Congruent Subgrid Voxeling" are on the left, and "Topological Voxeler" with a checked checkbox is on the right.

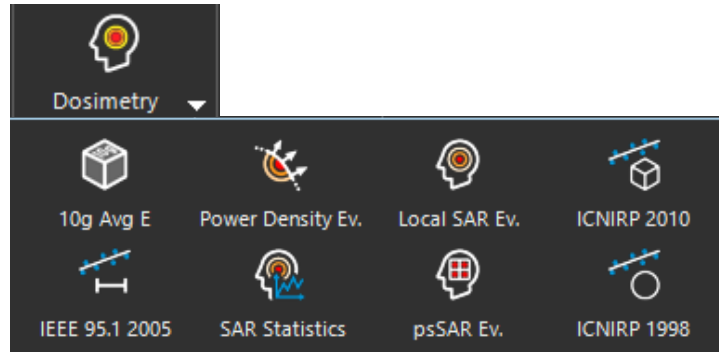
▲ Voxels
Used Voxel Engine
Congruent Subgrid Voxeling Topological Voxeler ☒

A dark grey settings panel for "Voxels". It has a small triangle icon on the left. The text "Used Voxel Engine", "Maximum Fraction", and "Congruent Subgrid Voxeling" are on the left. "Intersection Voxeler" is on the right, with "21" below it and a checked checkbox to its right.

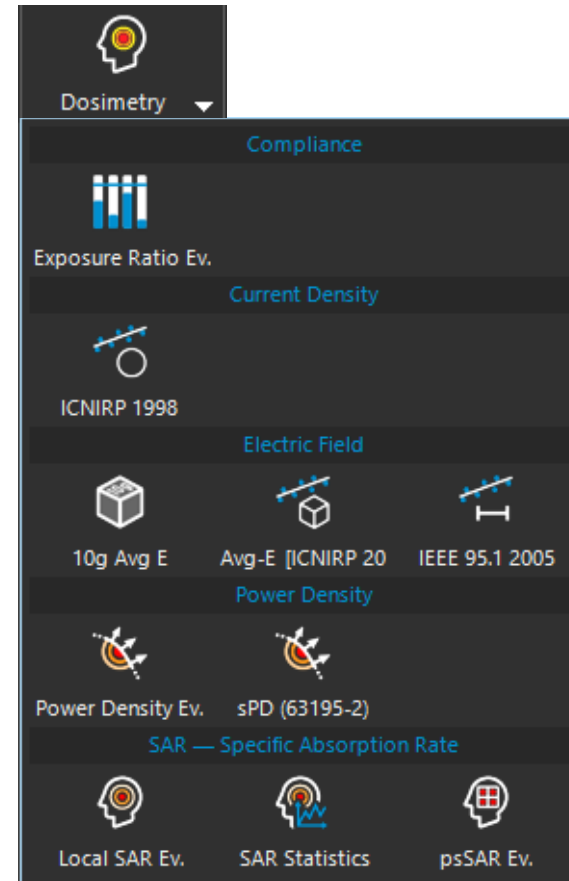
▲ Voxels
Used Voxel Engine
Maximum Fraction
Congruent Subgrid Voxeling Intersection Voxeler
21 ☒

分析(Analysis)

- 依輸出的場量類型重新組織劑量評估(Dosimetry)演算法(例如: 比吸收率 SAR 與電場相關演算法)，使分類更加清晰且易於使用

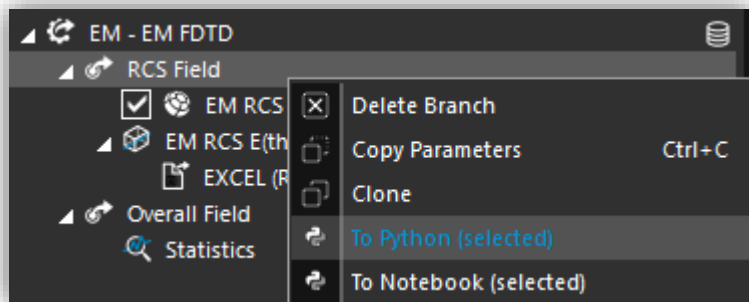


新增分類



Python API

- 擴充「To Python」匯出功能，可支援更複雜且符合實際應用情境的分析流程，包括與 RCS (雷達截面積) 相關的分析路徑及統計評估器(Statistics Evaluator)輸出



```
# Create SarStatisticsEvaluator
inputs = [em_sensor_extractor.Outputs["EM E(x,y,z,f0)"]]
sar_statistics_evaluator = analysis.em_evaluators.SarStatisticsEvaluator(inputs=inputs)
sar_statistics_evaluator.Snapshot = u"1e+10"
sar_statistics_evaluator.UpdateAttributes()
document.AllAlgorithms.Add(sar_statistics_evaluator)

# Access table output as a dict:
# table = sar_statistics_evaluator.Outputs[0].Data.ToDict()
```

```
1 # This script was auto-generated by Sim4Life version 9.6.0.22570
2
3 import numpy as np
4 import s4l_v1.analysis as analysis
5 import s4l_v1.document as document
6 import s4l_v1.model as model
7 import s4l_v1.units as units
8 from s4l_v1 import ReleaseVersion
9 from s4l_v1 import Unit
10
11 try:
12     # Define the version to use for default values
13     ReleaseVersion.set_active(ReleaseVersion.version9_6)
14
15     # Creating the analysis pipeline
16     # Create SimulationExtractor
17     simulation = document.AllSimulations["EM"]
18     simulation_extractor = simulation.Results()
19
20     # Create EmSensorExtractor
21     em_sensor_extractor = simulation_extractor["RCS Field"]
22     document.AllAlgorithms.Add(em_sensor_extractor)
23
24 except Exception:
25     ReleaseVersion.reset()
26     raise
27
```

Python API

- 擴充波導(Waveguide)相關 API，提升先前較難以透過腳本實現的模式選擇(Mode Selection)功能支援

class s4l_v1.simulation.emfdtd.WaveguideSourceSettings 

WaveguideSourceSettings(raw_obj=None, version=None)

Bases: ApiSettings

Creates settings for a waveguide source in an EM-FDTD simulation

Methods:

- Clone** – Create shallow clone of this setting. Does not copy any links.

Attributes:

- Name** (str) – Get/set name of object
- raw** (ConceptSettings) – raw property
- Bandwidth** (float) – Value of: Bandwidth
- BandwidthProp** (ApiProperty) – Property: Bandwidth
- CenterFrequency** (float) – Value of: Center Frequency
- CenterFrequencyProp** (ApiProperty) – Property: Center Frequency
- SelectedModes** (Iterable[int]) – Value of: Selected Modes
- SelectedModesProp** (ApiProperty) – Property: Selected Modes

Examples:

```
import s4l_v1.model as model
import s4l_v1.simulation.emfdtd as emfdtd
import s4l_v1.units as units

# SOURCE
source_settings = emfdtd.WaveguideSourceSettings()
source_settings.CenterFrequency = 12.5, units.GHz
source_settings.Bandwidth = 2000.0, units.MHz
# indices to modes sorted by cut-off frequency, i.e. the fundamental mode is index 1
source_settings.SelectedModes = [1] # 1-based indices

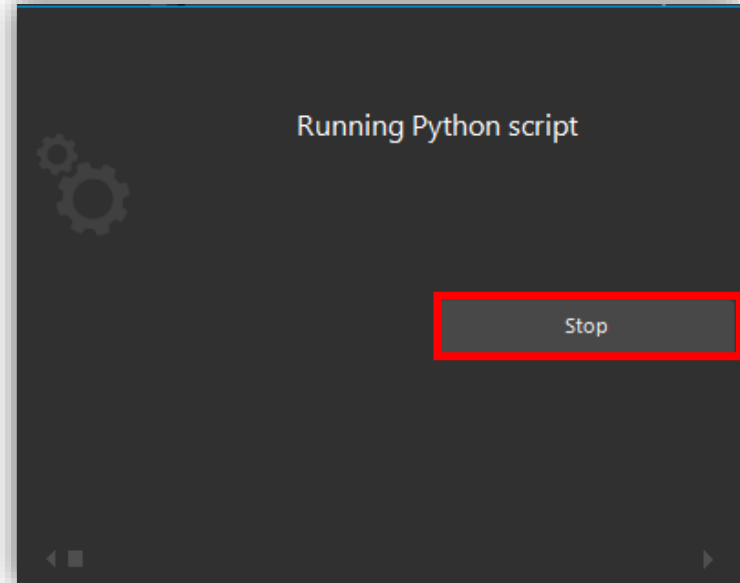
# SENSOR
sensor_settings = emfdtd.WaveguideSensorSettings()
# measure the same modes you excite
sensor_settings.SelectedModes = source_settings.SelectedModes
# inverts the default direction of measurements
sensor_settings.InvertMeasurementDirection = False
```

Python API

- 擴展聲學(Acoustics) API 的封裝(Wrapper)與繫結(Binding)功能，提升建立模擬設定流程的腳本化能力
- Python 腳本執行期間，現可透過彈出視窗中的 Stop 按鈕中斷執行

```
NAME
  s4l_v1.simulation.acoustic - Acoustic simulator API

CLASSES
  s4l_v1._api.simwrappers.ApiSettings(s4l_v1._api.apiwrappers.ApiObject)
  s4l_v1.simulation._acoustic_impl.BoundarySettings
  s4l_v1.simulation._acoustic_impl.FieldSensorSettings
  s4l_v1.simulation._acoustic_impl.GlobalBoundarySettings
  s4l_v1.simulation._acoustic_impl.MaterialSettings
  s4l_v1.simulation._acoustic_impl.PointSensorSettings
  s4l_v1.simulation._acoustic_impl.SetupSettings
  s4l_v1.simulation._acoustic_impl.SolverSettings
  s4l_v1.simulation._acoustic_impl.SourceSettings
  s4l_v1.simulation._gridder_implAutomaticGridSettings
  s4l_v1.simulation._gridder_impl.GlobalGridSettings
  s4l_v1.simulation._gridder_impl.ManualGridSettings
  s4l_v1.simulation._voxeler_implAutomaticVoxelerSettings
  s4l_v1.simulation._voxeler_impl.GlobalVoxelerSettings
  s4l_v1.simulation._voxeler_impl.ManualVoxelerSettings
  s4l_v1._api.simwrappers.ApiSimulation(s4l_v1._api.apiwrappers.ApiObject)
  s4l_v1.simulation._acoustic_impl.Simulation
```



Python API

- Analysis 匯出的 Python 腳本現已與評估器輸出結果及使用者介面中設定的表格式結果保持一致
- 針對多項 Python API 的易用性問題進行改善，包括支援更嚴格的字型處理，以及新增相容性輔助機制，以降低既有腳本因版本變更而失效的風險
- 強化與材料資料庫(Material Database)相關的 API 路徑，並加入回歸測試覆蓋，提升穩定性與可靠性
- 對非啟用(不可見)設定項目的寫入操作，現可透過明確的策略控制(ignore、warn、raise)，不再以無提示方式忽略或失敗
- 當指派不存在的設定屬性時，系統現在會立即拋出 `AttributeError`，讓 API 或設定結構錯誤更容易被發現與除錯

使用者體驗

- 網格生成(Meshing)與渲染(Rendering)等耗時作業現可採用**非阻塞式執行**，使用者在作業進行期間仍可持續操作系統
- 提升新專案與既有專案的開啟速度，縮短等待時間
- 雲端最佳化工作流程現已支援 NVIDIA Blackwell GPU
- 改善專案載入與初始化過程中的狀態回饋，讓執行進度更加清晰
- 提升 3D 視圖的操作流暢度與回應速度
- 透過雲端遠端執行的模擬結果，現在可更快速地下載至本機

自 V9.4.4 B21915 版本後的主要問題修復

- 修正大型網格(Mesh)儲存為 ACIS 檔案格式(SAB/SAT 及 .smash)時，可能產生檔案損毀的問題
- 修正啟用 Export tissue list 選項匯出分割(Segmentation)資料時，組織清單(Tissue List)未儲存至正確目錄的問題
- 修正分析(Analysis)頁籤中 RCS (雷達截面積) 評估器流程的「To Python」程式碼生成功能
- 改善幾何吸附(Geometry Snapping)與選取(Picking)功能對頂點、邊線及模型實體的辨識可靠性，提升幾何操作的精確度與一致性
- 修正 Topological Voxeler 中的法向量(Normal)生成問題，並修正邊界法向量(Edge Normal)的計算方式
- 修正非結構化網格(Unstructured Mesh)中 Patch 區域選取異常的問題