

Автоматизация процесса проектирования антенн и устройств СВЧ в современных программных комплексах электродинамического моделирования

Часть 5. Моделирование линзы Люнеберга в САПР Agilent EMPro

Андрей Пластиков (Москва)

На примере многослойной линзовой антенны центральной симметрии – линзы Люнеберга – рассмотрен процесс создания сценария автоматизированного построения модели антенны на языке Python в программной среде Agilent EMPro. В качестве примера смоделирована семислойная линзовая антенная система с облучателем в виде полуволнового вибратора. Для расчётов использован метод конечных разностей во временной области с конформной сеткой разбиения.

ВВЕДЕНИЕ

Важным направлением антенной техники сегодня является разработка многолучевых и сканирующих антенн с относительно большими значениями коэффициента усиления. Построить такую антенную систему можно на основе антенной решётки или антенны оптического типа – линзовой или зеркальной. Среди диэлектрических линзовых антенн широкое распространение получили многослойные антенны центральной симметрии на основе так называемой линзы Люнеберга (ЛЛ). Для синтеза структуры таких линз часто используют приближённые методы расчёта, в то время как для осуществления точного моделирования желательно использовать точные численные методы.

В статье дано краткое описание ЛЛ и её модификаций. Основное внимание

уделено описанию процесса создания программы-сценария автоматизированного построения модели ЛЛ, облучателя и установки параметров разбиения на языке Python в САПР Agilent EMPro. В качестве примера рассмотрена семислойная аппроксимация тела линзы, взятая из литературы. Расчёт проведён методом конечных разностей во временной области с использованием конформной сетки разбиения.

СФЕРИЧЕСКАЯ ЛИНЗА ЛЮНЕБЕРГА

Линза Люнеберга и её модификации [1, 2] принципиально позволяют осуществлять сканирование в полном телесном угле 4π стерадиан. Такая линза представляет собой диэлектрическую сферу с переменным коэффициентом преломления, зависящим только от текущего локального радиуса:

$$n(r) = \sqrt{2 - (r/a)^2}, \quad (1)$$

где a – радиус сферы, r – расстояние от центра линзы.

Линза Люнеберга фокусирует падающий на неё параллельный пучок лучей (плоский фронт волны) в точку на поверхности. Несмотря на то что впервые ЛЛ была предложена около 70 лет назад [3], практический интерес к ней проявился только в последние два десятилетия, что связано, главным обра-

зом, с появлением новых материалов и развитием технологии производства линзовых антенн. При этом для упрощения технологии изготовления ЛЛ переходят от линзы с непрерывным изменением показателя преломления к ступенчатому. Таким образом, сферическое тело линзы формируется набором диэлектрических слоёв с различными значениями относительной диэлектрической проницаемости ϵ (см. рис. 1).

Повышенный интерес к ЛЛ среди разработчиков сканирующих и многолучевых антенных систем (АС) оптического типа привёл к развитию специализированных методов расчёта таких многослойных линзовых антенн центральной симметрии, в том числе и в отечественной литературе (см., например, [4–6]). Для осуществления окончательных расчётов структуры ЛЛ с учётом модели конкретных облучателей и конструкций крепления целесообразно использовать строгие численные методы вычислительной электродинамики.

САПР AGILENT EMPro: ВСТРОЕННЫЙ ЯЗЫК НАПИСАНИЯ СЦЕНАРИЕВ

Программная среда трёхмерного электромагнитного моделирования EMPro [7, 8] расширяет возможности известной платформы разработки электронных устройств и блоков Agilent Advanced Design System (ADS), однако может рассматриваться и как самостоятельный программный продукт для моделирования антенн и устройств СВЧ. Среди особенностей САПР EMPro следует отметить объединение двух различных расчётных модулей – на основе методов конечных элементов в частотной области и конечных разностей во временной

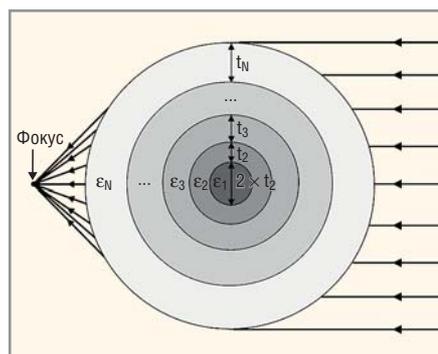


Рис. 1. Многослойная линза Люнеберга

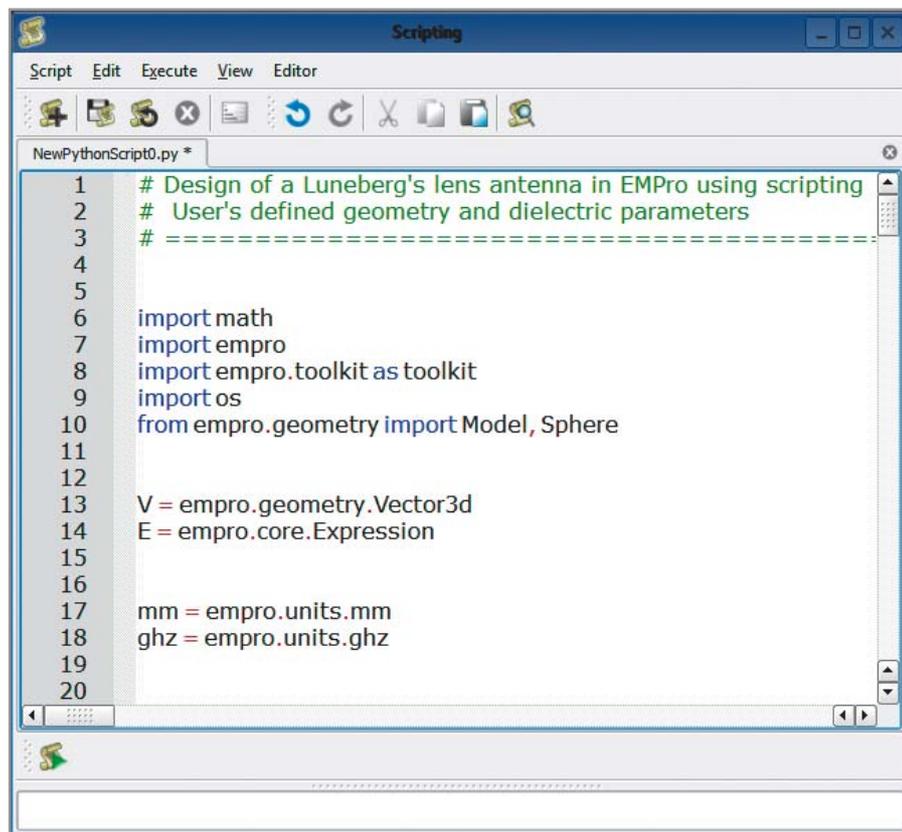


Рис. 2. Окно редактора сценариев

области – в одной программной оболочке. При проведении расчётов методом конечных разностей во временной области (Finite Difference Time Domain, FDTD) имеется возможность использовать конформную сетку разбиения.

Программа EMPro имеет встроенный редактор сценариев на языке Python. Имеющиеся средства программирования предоставляют доступ к набору команд, выполняемых в процессе обычного моделирования в CAD-интерфейсе: построению геометрических примитивов, заданию свойств материалов, установкам разбиения на ячейки, обработке результатов расчётов и др.

Как отмечается [9], «Python (<http://python.org>) является простым и в то же время мощным интерпретируемым объектно-ориентированным языком программирования. Он предоставляет структуры данных высокого уровня, имеет изящный синтаксис...», что делает его идеальным языком для быстрого написания различных приложений,

работающих на большинстве распространённых платформ». Язык Python имеет свободно распространяемый интерпретатор, что, несомненно, повлияло на его достаточно широкое распространение в веб-программировании. Последнее время язык Python начинает активно использоваться и в средах написания макросов/сценариев в программных продуктах различного назначения, в том числе и пакетах электродинамического моделирования.

НАПИСАНИЕ ПРОГРАММЫ-СЦЕНАРИЯ ДЛЯ ПОСТРОЕНИЯ МОДЕЛИ МНОГОСЛОЙНОЙ ЛИНЗЫ ЛЮНЕБЕРГА

Опишем процесс создания сценария на языке Python в программе EMPro, который осуществляет автоматизированное построение модели ЛЛ с заданным числом и параметрами слоёв и облучателя, разбиение модели линзы на ячейки и расчёт радиотехнических характеристик полученной модели АС.

Параметры слоёв линзы Люнеберга

Параметры	Номер слоя						
	1	2	3	4	5	6	7
Толщина слоя t , мм	75	26	63	30	33	33	65
Диэлектрическая проницаемость слоя ϵ	1,76	1,70	1,66	1,60	1,53	1,46	1,39

В качестве метода расчёта выберем FDTD [10].

Рассмотрим вариант построения ЛЛ, представленный в статье [11], в качестве примера. Соответствующая линзовая антенна имеет внешний диаметр 650 мм и состоит из $N = 7$ диэлектрических слоёв. Параметры слоёв в соответствии с обозначениями, введёнными на рисунке 1, занесены в таблицу. Расстояние f от края линзы до точки фокуса составляет 118 мм. Зададимся тангенсом угла диэлектрических потерь $\tan\delta$ всех слоёв, равным 0,001. Для возбуждения ЛЛ используем простейший облучатель в виде полуволнового диполя. Расчёт проведём на частоте 4 ГГц. Поскольку программа EMPro имеет встроенный редактор сценариев, команды вводятся в соответствующем окне (см. рис. 2) и не требуют для выполнения наличия внешнего интерпретатора.

Комментарии в языке Python начинаются с символа # и продолжаются до конца строки. Комментарий может находиться в начале строки или после кода, но не внутри строкового выражения. Заметим, что автору не удалось реализовать возможность написания русскоязычных комментариев (использовалась ОС Windows 7). Однако в строках кода, рассматриваемых далее, мы для удобства будем использовать комментарии, написанные кириллицей.

Написание программы-сценария начнём с объявления импорта модулей, необходимых для правильного функционирования команд. В нашем случае соответствующие строки кода будут выглядеть следующим образом:

```
import math
import empro
import empro.toolkit as toolkit
import os
from empro.geometry import Model, Sphere
```

Модуль math отвечает за использование дополнительных математических операций и функций. Например, для расчёта e^2 следует выполнить команду `math.exp(2)`. В последней строке приведённого выше кода производится импорт классов Model и Sphere из модуля геометрии `empro.geometry`.

Далее введём обозначения, упрощающие используемые в дальнейшем операции объявления векторов в трёх-

мерном пространстве и задание размерностей вводимых параметров:

```
V = empro.geometry.Vector3d
E = empro.core.Expression
mm = empro.units.mm
ghz = empro.units.ghz
```

В результате мы можем определить, например, некую величину x , равную 1 мм, одним из следующих двух способов: $x1=E("1 \text{ mm}")$ или $x2=1*\text{mm}$. При этом мы получим, что данным величинам присвоены значения $x1=1\text{mm}$ и $x2=0.001$ соответственно. В этом можно убедиться, выполнив команду `print x`, осуществляющую вывод значения переменной в окно отладочной информации, находящееся внизу окна редактора сценариев (см. рис. 2). В коде сценария допустимо использовать оба варианта объявления размерности. Объявим переменные сценария, задающие параметры ЛЛ:

```
N=7 # Число слоев линзы
# Толщины слоев, мм
t=[75,26,63,30,33,33,65]
# Диэлектрическая проницаемость слоев
eps=[1.76,1.70,1.66,1.60,1.53,1.46,1.39]
tand=0.001 # tand слоев
# Расстояние от края линзы до фокуса, мм
f=118
```

Размерность величинам f и элементам массива t присвоим позднее при их непосредственном использовании. Заметим, что в языке Python нумерация элементов массивов (списков (lists) в общем случае) начинается с нуля.

Рассчитаем массив R внешних радиусов каждого слоя. Для этого сначала объявим R просто как переменную, равную радиусу первого слоя, выполнив команду $R=[t[0]]$. Затем добавим в R оставшиеся $(N-1)$ элементов следующим образом:

```
for i in range(1,N):
    R.append(R[i-1]+t[i])
```

Заголовок строки объявления тела цикла `for...` завершается двоеточием. Все инструкции, образующие тело цикла, записываются с отступом, начиная со следующей после объявления функции строки. Для такого смещения

можно использовать символ табуляции (клавиша Tab). Эти две особенности синтаксиса языка Python характерны для других видов циклов и при объявлении функций.

Теперь зададим переменные, имеющие размерность частоты:

```
# Центральная частота
freq0= 4*ghz
# Максимальная частота диапазона
freq_max=4*ghz
# Задание расчетного диапазона частот проекта
empro.activeProject.parameters().setFormula("minFreq", "4 GHz")
empro.activeProject.parameters().setFormula("maxFreq", "4 GHz")
```

Две последние команды вводят в проект переменные, которые задают диапазон частот расчёта и находятся в окне Parameters. Рассчитаем соответствующие длины волн и зададим переменные, которые будут использоваться далее для установки параметров разбиения на ячейки:

```
# Скорость света
c0=3*10**8
# Расчет длин волн
lambda0= c0/freq0
lambda_min = c0/freq_max
# Параметры разбиения на ячейки
max_cell_size = lambda_min/12
min_cell_size=0.7
```

Далее объявим основную процедуру `main` проекта:

```
def main(autoQueue = True):
    # «Очистка» текущего проекта
    empro.activeProject.clear()
    # Активация оболочки FDTD
    empro.skinKeeper().skin = 'SkinFDTD'
    # Вызов пользовательских процедур
    setupMaterials()
    setupLensGeometry()
    setupDipoleGeometry()
    setupGrid()
    setupSources()
    setupSensors()
    setupSimulation()
```

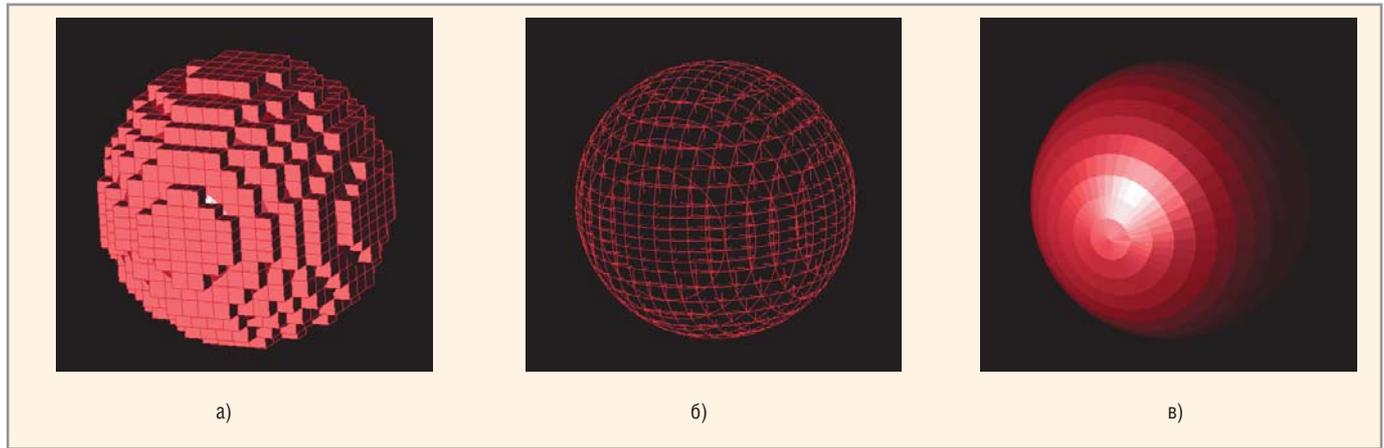


Рис. 3. Пример разбиения модели сферы на ячейки при использовании стандартной ортогональной сетки FDTD (а) и при введении конформной сетки; сетка разбиения (б) и модель сферы (в)

```
# Сохранение проекта
empro.activeProject.save-
ActiveProjectAs()
# Расчет проекта
return empro.activePro-
ject.addSimulationDataToProject(a
utoQueue)
```

Первая строка определения функции, которая называется обычно заголовком функции, обозначается в языке Python ключевым словом def (от англ. define, определять). После def должно следовать название функции

со списком формальных параметров в скобках. Заметим, что в языке Python, как и в языке Си, процедура – это функция, которая не возвращает никакого значения, т.е. не имеет списка выходных параметров. Ниже мы кратко рассмотрим содержание семи пользовательских процедур, вызываемых в теле main.

Процедура setupMaterials вводит в проект используемые материалы – идеальный проводник PEC для построения диполя и N диэлектрических материалов с названиями diel_layer_i, где

$i = 1, 2, \dots, N$, соответствующими значениями ϵ и $\tan\delta$ и желаемым цветом отображения в окне геометрии проекта:

```
def setupMaterials():
    # PEC
    pec = toolkit.defaultMaterial("PEC")
    empro.activeProject.materials().append(pec)
    # Диэлектрики
    for i in range(1,N+1):
        dielectric =
empro.material.Material()
        dielectric.name =
"diel_layer_%d" % i
        ...
    empro.activePro-
ject.materials().append(dielec-
tric)
```

Процедура setupLensGeometry осуществляет построение геометрии многослойной ЛЛ. Центр линзы совместим с началом системы координат. Для получения элемента i -го слоя для $i > 1$ из построенной сферы соответствующим радиусом R_i вычитаются элементы геометрии предыдущих $(i - 1)$ слоёв, которые предварительно заносятся в список models_for_subtract. Каждому слою ставится в соответствие свой диэлектрический материал.

Процедура setupDipoleGeometry вводит в проект модель полуволнового вибратора в виде бесконечно тонкого идеального провода длиной $\lambda/2$, центр которого совместим с точкой фокуса, расположенной на оси OZ в области $z > 0$:

```
def setupDipoleGeometry():
    dipole_length=lambda/2
    pec = empro.activePro-
ject.materials()["PEC"]
```

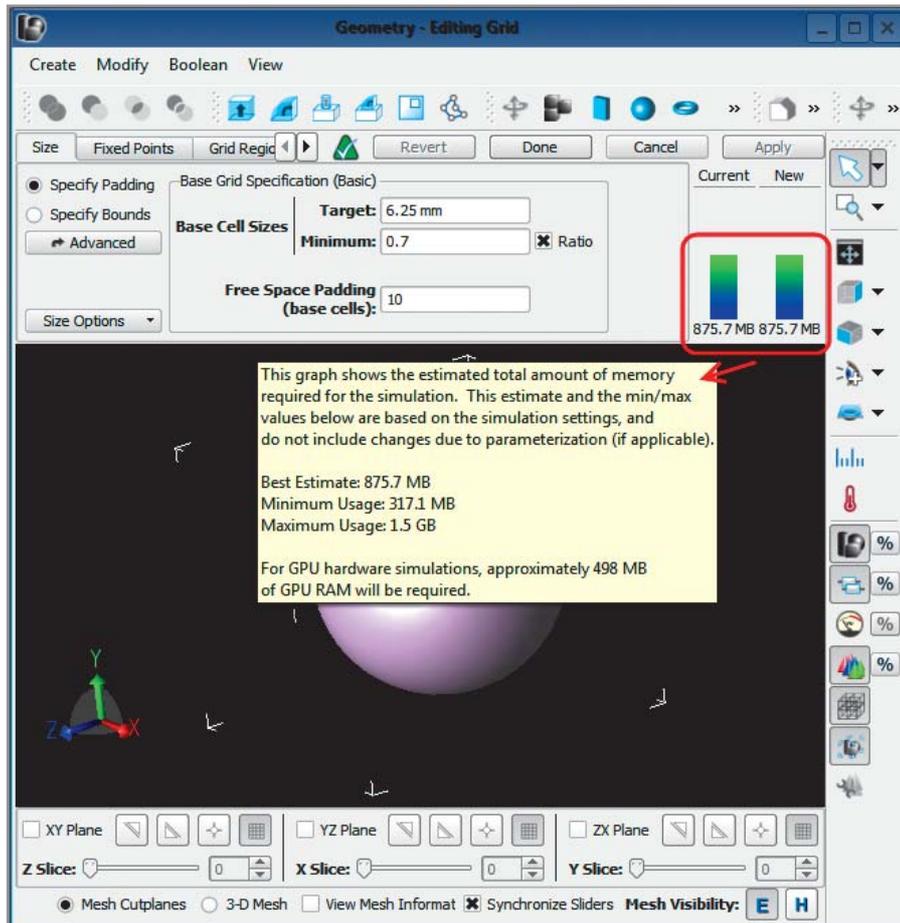


Рис. 4. Информация об ожидаемом объёме необходимой для расчёта памяти

```

tail = V(0, -
dipole_length/2, (f+R[N-1])*mm)
head = V(0,
dipole_length/2, (f+R[N-1])*mm)
wire =
toolkit.geometry.Wire(tail, head)
wire.name = "Dipole"

toolkit.applyMaterial(wire, pec)
empro.activeProject.geometry().append(wire)

```

Процедура `setupDipoleGeometry` устанавливает дискретный порт с импедансом 50 Ом в центре диполя:

```

def setupSources():
    port_length=lambda0/20
    component = empro.components.CircuitComponent()
    ...
    component.tail = V(0, -
port_length/2, (f+R[N-1])*mm)
    component.head = V(0,
port_length/2, (f+R[N-1])*mm)
    ...

```

В вызываемой далее процедуре `setupGrid` мы задаём параметры разби-

ения на ячейки. Базовые параметры разбиения устанавливаются следующими командами:

```

grid = empro.activeProject.grid-
Generator()
grid.cellSizes.target =
V(max_cell_size,max_cell_size,max
_cell_size)
grid.cellSizes.minimum =
V(min_cell_size,min_cell_size,min
_cell_size)
grid.cellSizes.minimumType =
("RatioType", "RatioType", "Ratio-
Type")
grid.padding.lower = V(10,10,10)
grid.padding.upper = V(10,10,10)

```

Затем для каждого из диэлектрических слоёв зададим свои параметры разбиения, «улучшенные» в $\sqrt{\epsilon}$ раз, и введём конформную (неортогональную) сетку командой вида `part.meshParameters.conformal=True`, где `part` – элемент текущего слоя. Использование конформной сетки разбиения позволит более точно учесть геометрические особенности сферической поверхности слоёв по сравнению с класси-

ческой ортогональной сеткой при аналогичном числе ячеек разбиения (см. рис. 3). В теле этой же процедуры зададим границы окружающего пространства как PML (Perfectly Matched Layer) с помощью команды `empro.toolkit.defineAllBoundariesAsPMLs(5)`.

В процедуре `setupSensors` зададим требуемые мониторы ближнего и дальнего полей. Например, задать расчёт трёхмерной диаграммы направленности (ДН) можно командами:

```

sensors =
empro.activeProject.farZoneSen-
sors()
sensors.append(toolkit.sensor.mak-
eFarZoneThetaPhi())

```

Процедура `setupSimulation` устанавливает параметры решающего устройства. В простейшем случае она может состоять из одной команды вида `toolkit.simulation.setupSteadyStateSimulation([freq0], maxTime = "10000*timestep")`, в которой задаётся максимальное число временных шагов при расчёте переходного процесса во временной области. Здесь `timestep` явля-

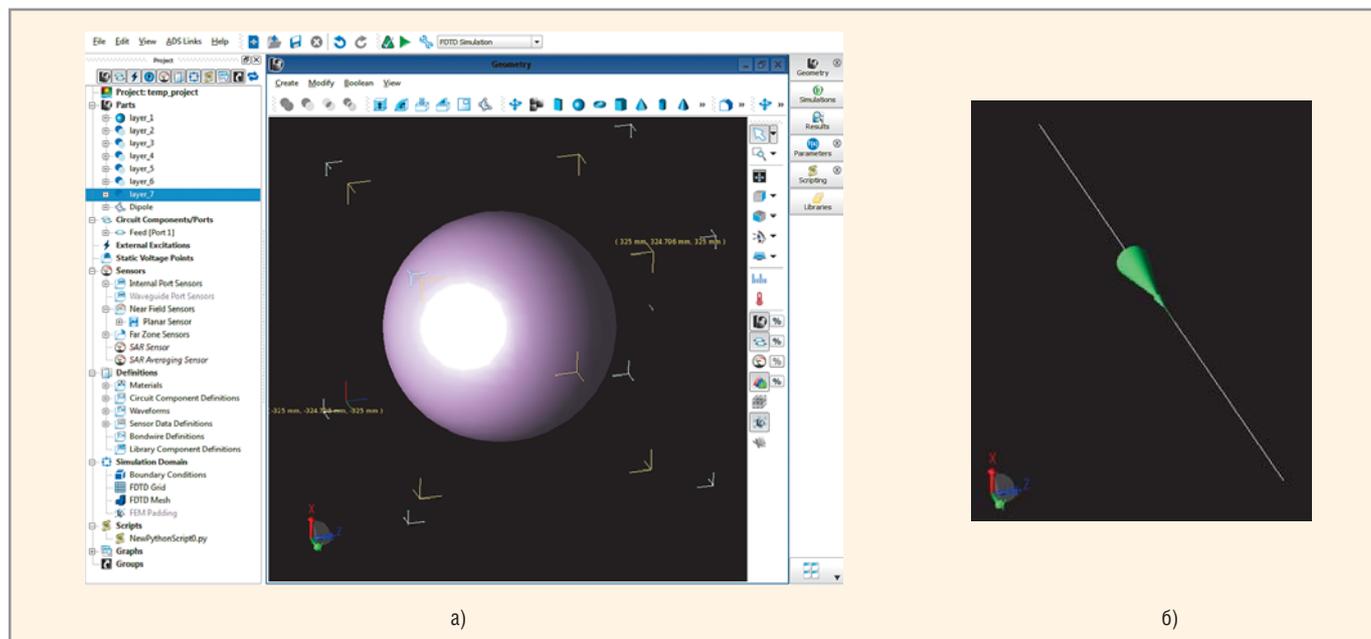


Рис. 5. Построенная модель антенны в интерфейсе программы EMPro (а) и модель вибратора (б)

ется переменной проекта, которая рассчитывается автоматически, исходя из используемой сетки разбиения и частотного диапазона расчёта с целью получения устойчивых вычислений. Расчёт переходного процесса заканчивается при достижении заданного уровня отражённого сигнала (по умолчанию –30 дБ) или через установлен-

ное максимальное число временных шагов.

РЕЗУЛЬТАТЫ РАСЧЁТА

Для запуска написанного сценария необходимо выбрать в меню редактора пункт *Execute* → *Execute Script* или нажать на кнопку . После выполнения действий, заданных в семи вве-

дённных пользовательских процедурах, появится диалоговое окно сохранения проекта в файл. После этого сценарий запустит решающее устройство на расчёт.

Отметим, что созданная сценарием модель содержит около 4,4 млн. ячеек. Программа EMPro выводит информацию об ожидаемом объёме необходимой для расчёта оперативной или видеопамяти ещё до начала расчёта. Соответствующие данные находятся в окне *Geometry – Editing Grid* (см. рис. 4). В нашем случае для расчёта потребуется около 876 Мб (максимум 1,5 Гб) оперативной памяти или около 317 Мб видеопамяти при использовании видеокарты, совместимой с инструкциями CUDA. Для смягчения требований к аппаратным ресурсам и ускорения расчётов в рассматриваемой задаче можно ввести две плоскости симметрии – электрическую и магнитную.

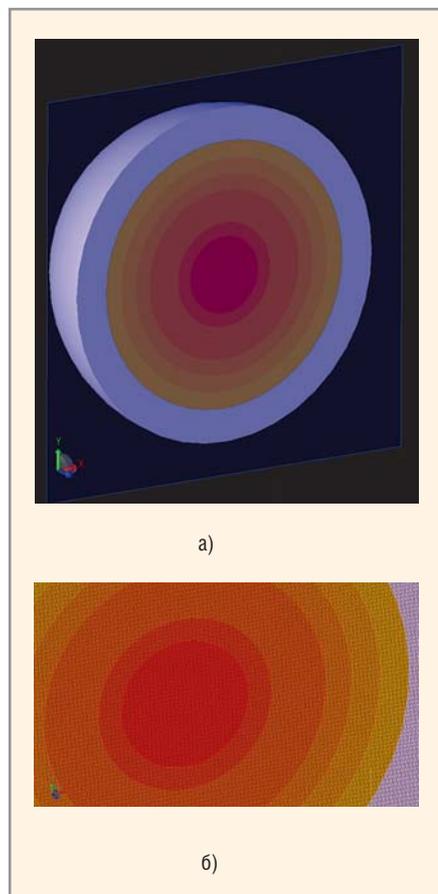


Рис. 6. Модель линзы в разрезе (а) и с отображённой сеткой разбиения (б)

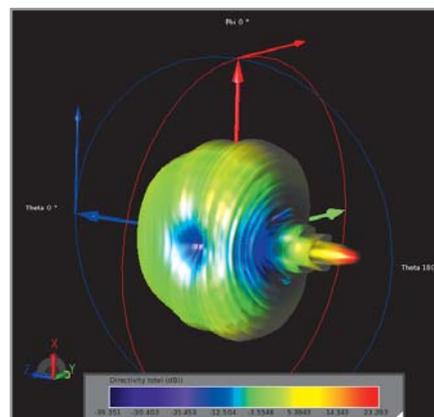


Рис. 7. Трёхмерная ДН АС

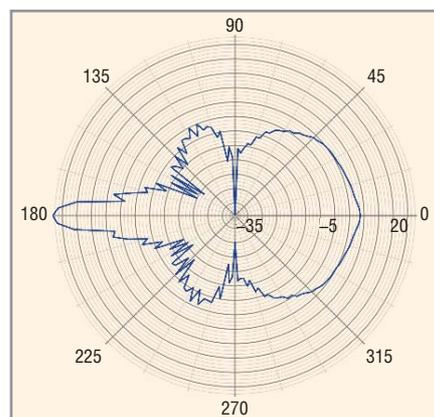


Рис. 8. ДН АС в плоскости YZ в полярной системе координат

На рисунке 5а изображена построенная сценарием модель ЛЛ в интерфейсе программы EMPro. Увеличенная модель вибратора приведена на рисунке 5б. Дискретный порт возбуждения отображён зелёным конусом в центре облучателя. Сечение линзы и сетка разбиения в плоскости XY показаны на рисунке 6. На рисунках 7 и 8 изображены рассчитанные трёхмерная и двухмерная ДН АС. Коэффициент направленного действия (КНД) рассмотренной ЛЛ составляет около 24 дБ. Повысить КНД АС (при неизменных теле линзы и частоте расчёта) и снизить уровень бокового излучения можно при использовании более подходяще-

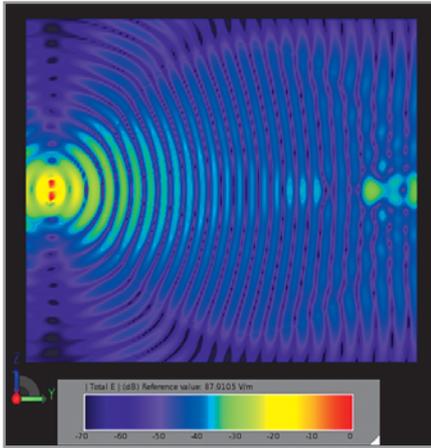


Рис. 9. Распределение ближнего поля в логарифмическом масштабе

го облучателя, например, рупорной антенны или диэлектрической антенны бегущей волны. На рисунке 9 приведено распределение ближнего поля в плоскости YZ.

ЗАКЛЮЧЕНИЕ

Таким образом, на примере моделирования многослойной линзы Лüneберга показана возможность использования сценариев на языке Python в программе EMPro для построения мо-

дели антенны и задания параметров разбиения на ячейки. Расчёт проведён методом конечных разностей во временной области с использованием конформной сетки разбиения. В качестве облучателя использован полуволновый вибратор.

Такой подход к моделированию удобно использовать для проведения многократных исследований с различными вариантами аппроксимации тела линзы. При необходимости написанный сценарий можно дополнить, например, возможностью статистического анализа, показывающего влияние разброса диэлектрических параметров на характеристики антенны.

Автор выражает благодарность компании Agilent за предоставление пробной версии программы EMPro 2011.12.

ЛИТЕРАТУРА

1. Жуж М.С., Молочков Ю.Б. Проектирование линзовых, сканирующих, широкодиапазонных антенн и фидерных устройств. Энергия, 1973.
2. Morgan S.P. General Solution of the Luneburg Lens Problem. J. of Applied Physics. 1958. V. 29. PP. 1358–1368.

3. Luneberg R.K. Mathematical Theory of Optics. Brown Univ., Providence, R.I., 1944.
4. Кузьмин С.В. Математическая модель многослойной линзы Лüneберга. Автореф. канд. дисс., СПб, 2004.
5. Панченко Б.А., Гизатуллин М.Г., Лебедева Е.В. Рассеяние электромагнитных волн многослойной линзой Лüneберга. Антенны. 2009. № 11. С. 30–33.
6. Панченко Б.А., Лебедева Е.В. Антенные характеристики линзы Лüneберга. Антенны. 2010. № 12. С. 5–9.
7. <http://www.home.agilent.com>.
8. Ян Х.-С., Ли Х.-С. Уникальные возможности трёхмерного электромагнитного моделирования в САПР Advanced Design System. Современная электроника. 2011. № 2. С. 66–69.
9. Россум Г., Дрейк Ф.Л.Дж., Откидач Д.С. Язык программирования Python. Под ред. Д.С. Откидач. 2001.
10. Yee K.S. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. on AP. 1996. V. 14. № 4. PP. 302–307.
11. Huang et al. Multiobjective Optimization and Design of a Luneburg Lens Antenna with Multiband Multi-Polarized Feed-System. PIERS. 2012. V. 129. PP. 251–269. ©