

УДК 621.37

В.О. Іванов, д-р техн. наук
О.В. Вишнівський**МЕТОДИКА ПОБУДОВИ СІТКОВИХ МОДЕЛЕЙ
ТРИВИМІРНИХ ЕЛЕКТРОДИНАМІЧНИХ ОБ'ЄКТІВ**

Інститут електроніки та систем управління НАУ, e-mail: alexandervishnevsky@rambler.ru

*Розглянуто особливості застосування програмного середовища 3ds Max для побудови сіткових моделей тривимірних електродинамічних об'єктів геометрично складної форми.***Вступ**

Насиченість сучасних літаків, штучних супутників Землі, спеціальних автомобілів тощо радіоелектронними засобами (РЕЗ) різного призначення загострює проблему забезпечення можливості їх одночасного функціонування без взаємних радіозавад.

Однією з причин виникнення електромагнітної несумісності бортових РЕЗ є неминуче виникнення непередбачених електромагнітних зв'язків між окремими антенами, що розміщені понад провідним корпусом рухомого об'єкта того, або іншого типу. При цьому сукупність різноманітних антен незалежних різночастотних РЕЗ і провідний корпус складної конфігурації створюють єдиний електродинамічний об'єкт, здатний збуджувати, відбивати та поглинати енергію електромагнітного поля. Корпус електродинамічного об'єкта доцільно розглядати як своєрідний вібратор складної антени. Розміри цього вібратора неоднаково співвідносяться з довжинами хвиль λ_i , на яких функціонують штатні антени бортових РЕЗ.

Проблеми забезпечення взаємної електромагнітної сумісності (ЕМС) бортових РЕЗ у процесі їх експлуатації і вибору такого оптимального розташування антен понад корпусом електродинамічного об'єкта доцільно розв'язувати ще на етапі проектування об'єкта. Але коректні методики прогнозування і кількісної оцінки можливої якості електромагнітної обстановки (ЕМО) у точці розташування окремої антени обраної конструкції дотепер не створені з причини труднощів урахування конструктивних особливостей електродинамічних об'єктів складної конфігурації.

Точне оцінювання якості ЕМО проводиться експериментально на натурних об'єктах або їх фрагментах. Такий метод є консервативним і дорогим. Тому розробка коректної методики оперативного прогнозування якості ЕМО програмними методами є актуальною і перспективною задачею. Цю задачу можна розв'язати, якщо електродинамічний об'єкт подати у вигляді провідної сіткової моделі [1].

Довжина елементів загальної сіткової моделі не повинна перевищувати значень $0,1\lambda_{\min}$, де λ_{\min} – мінімальна довжина хвилі, на якій досліджуються електромагнітні зв'язки між антенами окремих РЕЗ. Струми в окремих елементах сіткової моделі, що є елементарними збуджувачами поля, пов'язані з напругами на входах окремих антен електродинамічного об'єкта розмірними коефіцієнтами, які мають зміст власних або взаємних опорів.

Ці зв'язки зображують у вигляді математичної моделі електродинамічного об'єкта і утворюють систему лінійних алгебраїчних рівнянь, порядок яких може сягати десятків тисяч. Однак сучасна обчислювальна техніка дозволяє розв'язати і визначити взаємні опори для будь-якої пари штатних антен з урахуванням шунтуючого впливу на кожну з сукупності усіх інших елементів електродинамічного об'єкта, тобто внесених опорів. Зміщуючи одну або декілька антен в інші точки поверхні, можна забезпечити припустиме значення взаємного або внесеного опору, при якому ЕМО в точці розташування антен буде сприятливою для конкретного РЕЗ.

Реалізація задачі щодо конкретної методики визначення якості ЕМО розпочинається зі створення об'ємної сіткової моделі з урахуванням технічної документації на конструктивні особливості електродинамічного об'єкта з урахуванням типів штатних антен.

Аналіз досліджень

У відомих науково-технічних виданнях відсутні вказівки або рекомендації щодо сіткових моделей з урахуванням програмних технологій.

Постановка завдання

Основною метою статті є розробка методики отримання сіткових моделей об'ємних тіл складної конфігурації на основі їх конструкторської документації з застосуванням програмного середовища *MaxScript*, яке первісно створювалось як інструмент об'єктно-орієнтованого програмування для задач комп'ютерної анімації (*3dsMax*).

Методика побудови сіткових моделей

Спочатку об'єкт, що моделюється, необхідно скомпонувати з набору так званих "стандартних та розширених примітивів", таких як куля, куб, конус і так далі, відповідно до розмірів фрагментів об'єкта, описаного у конструкторській документації. Необхідні зміни та уточнення вносяться за допомогою набору модифікаторів у меню "Modifiers". Шви поміж адаптованими примітивами усуваються за допомогою техніки "weld". Далі об'єкт можна перетворити на один з чотирьох типів: Mesh, Poly, Patch або NURBS. На рис. 1 подано сферу з радіусом 10 м, отриману у формі Mesh.

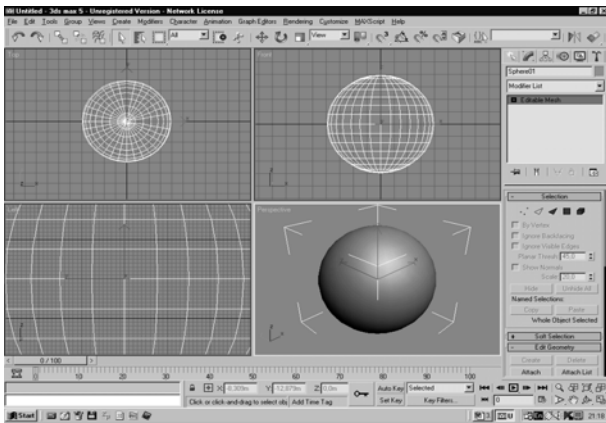


Рис. 1. Інтерфейс 3ds Max та Mesh-сфера

Можна отримати зображення вершин-координат 32-х сегментів цієї сфери і отримати доступ до цих координат у діалоговому вікні Max.

На рис. 2 показано вигляд Mesh-сфери.

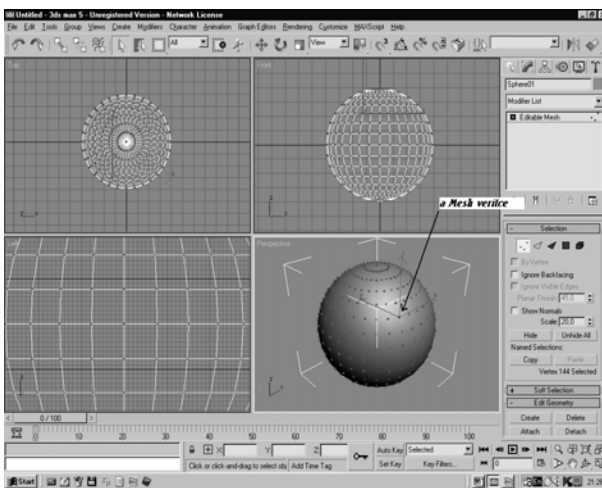


Рис. 2. Mesh-сфера з можливістю прямого доступу до координат

Існує також можливість отримати доступ до координат програмним шляхом. Для цього перетворюємо сферу на Patch-об'єкт та використовуємо вмонтовану в Max мову програмування MaxScript. MaxScript – це об'єктно-орієнтована мова програмування, яку можна порівняти, наприклад, з мовою C++.

Patch-сфера матиме дещо інший вигляд, – з'являться елементарні "трикутники" замість елементарних "квадратів" Mesh-моделі, які потім усуваються програмним шляхом за методикою, розробленою у праці [1]. Сітку можна покращити двома способами: прибрати діагональний елемент (чотирикутна чарунка); додати ще один діагональний елемент (трикутна чарунка). На рис. 3 показано оновлений варіант сіткової моделі сфери.

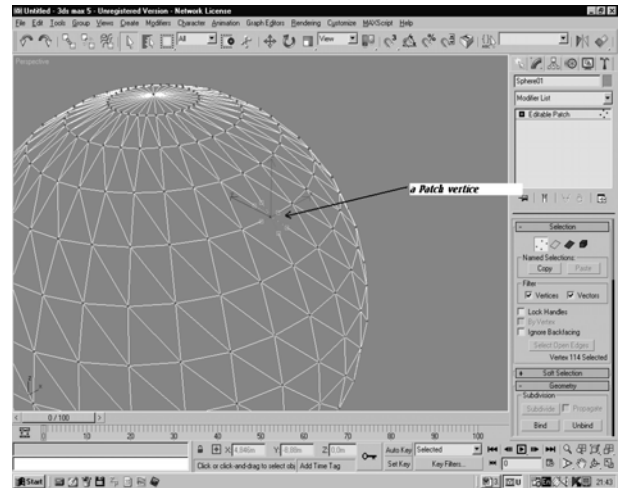


Рис. 3. Patch-версія

Подамо текст оригінальної програми, яка автоматично знаходить координати будь-якого вузла ґратки Patch-тіла:

--Alexander Vishnevsky 2003

--The Patchfire

-- this small program calculates vertices' coordinates

-- of any patch object-----

fin=getNumEdges \$a

for i=1 to fin do

(v1=getEdgeVert1 \$a i+1

v2=getEdgeVert2 \$a i+1

print "edge#"; print i

print "v1: "; print v1

print "v2: "; print v2

print "v1.x: "; _v1=getVert \$a v1;

print _v1.x

print "v1.y: "; -----

print _v1.y

print "v1.z: "; -----

print _v1.z -----

print "v2.x: "; _v2=getVert \$a v1;

print _v2.x ; -----

print "v2.y: "; -----

print _v2.y ; -----

print "v2.z: "; -----

print _v2.x)

Потрібно зазначити, що теоретично для такого тіла, наприклад, фюзеляж літака, краще підходить Mesh-методика, оскільки вона оперує сітковою моделлю, побудованою на “прямокутниках”, у той час як при Patch-методі з’являється ще один діагональний елемент, який є надлишковим. До того ж він геометрично довший за ребро Mesh-моделі, що небажано для вирішення наших задач.

Але специфіка MaxScript спонукає до застосування саме Patch-моделей, оскільки тут є унікальний набір функцій, що дозволяє отримати координати сіткової моделі з прив’язкою до відповідних ребер (диполів у електродинамічному тлумаченні), чого важко або неможливо досягти, працюючи з Mesh-моделями.

При математичному моделюванні антени зручно використати вже Mesh-технологію, оскільки є необхідність в отриманні інформації про просторове положення диполів.

Наведемо ще дві програми. Перша з них створює txt – інтерфейсний файл для подальшої роботи програми Antfree, яка будує результуючі діаграми спрямованості (ДС). Цей txt-файл складається з семи основних елементів: його початок містить число, яке відповідає кількості елементарних “диполів” сіткової моделі; перші три колонки цифр (зліва направо) показують x-y, та z-координати однієї вершини елемента сіткової моделі, – а ті, що залишились – координати іншої вершини (рис. 4):

```
--Alexander Vishnevsky 2003
--The Patchfire for Antfree
-- this small program calculates vertices` coordinates
-- of any object
for i=1 to N $a do
(-----
```

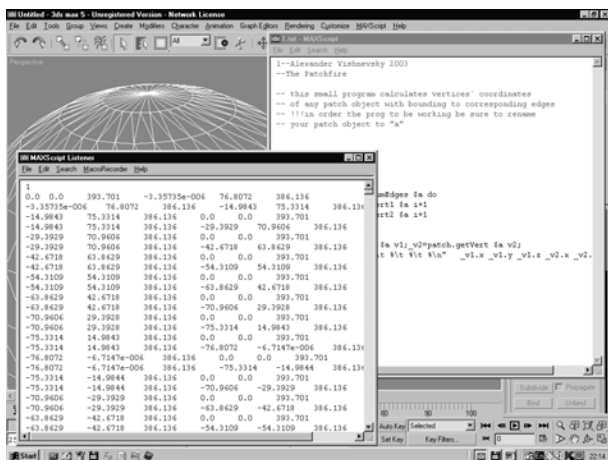


Рис. 4. Приклад роботи програми “Patchfire for Antfree”

```
--for start
v1=N1 $a i+1
v2=N2 $a i+1
_v1=getVert $a v1;_v2=getVert $a v2;
format "%\t %\t %\t %\t %\t %\tn"
_v1.x _v1.y _v1.z _v2.x _v2.y _v2.z
--for end
)-----
```

Друга програма створює такий самий txt – інтерфейс для Antfree, але працює вже з антеною:

```
coords=#(); -----
centercoords=#(); -----
lengths=#(); -----
_alpha=#()
betha=#()
for i=1 N1 $a
do( ----- do
start
coords[i]=getVert $a I -----
)--end of 1 for -----
for i=2 to getNumVerts $a
do(
bar=(coords[i-1].x+coords[i].x)/2
foo=(coords[i-1].y+coords[i].y)/2
barfoo=(coords[i-1].z+coords[i].z)/2
lengths[i-1]=sqrt(((coords[i].x-coords[i-1].x)^2+((coords[i].y-coords[i-1].y))^2+((coords[i].z-coords[i-1].z))^2)
alpha[i-1]=acos(sqrt(((coords[i].x-coords[i-1].x)^2+((coords[i].y-coords[i-1].y))^2)/(lengths[i-1]^2))
betha[i-1]=acos(sqrt(((coords[i].x-coords[i-1].x)^2+((coords[i].z-coords[i-1].z))^2)/(lengths[i-1]^2))
-- centercoords -----
centercoords[i-1]=[bar,foo,barfoo]
)--end of 2 for
for i=1 to getNumVerts $a-1 do(
format "%\t %\t %\t %\t %\t %\tn" centercoords[i].x centercoords[i].y centercoords[i].z
lengths[i] _alpha[i] betha[i]
)--endof 3 for
--//
--prog end -----
--pg filler
--:)
```

Проілюструємо також і її роботу на основі рис. 5, де маємо ті самі сім структурних елементів: число диполів, три просторові координати центрів кожного з них, їхню довжину, кути повороту відносно площин *XOY* та *XOZ*.

Потрібно відзначити, що розроблена оригінальна методика отримання сіткових моделей, не описана в літературі.

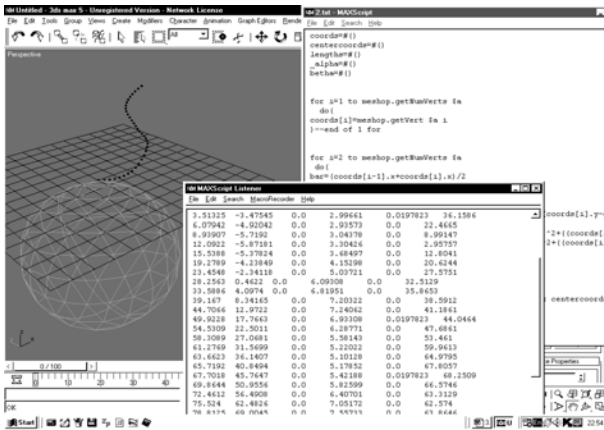


Рис. 5. Приклад роботи програми “Meshfire for Antfree”

Її сутність полягає в тому, що тіло геометрично складної форми покривається мережею променів, що й утворюють спрощену сітку.

Оригінальність полягає в тому, що промені „охоплюють” тіло за допомогою внутрішніх сплайн-функцій 3ds max, що дозволяє швидко і зручно побудувати мережу променів навколо тіла геометрично складної форми. Промені виходять з точки установки антени. Таким чином можливо, використовуючи перший закон Кірхгофа та відомий набір електродинамічних формул, знайти шукану ДС товстого вібратора.

Кожний промінь розбивається на деяку кількість сегментів, причому максимальна відстань між двома сусідніми променями не повинна перевищувати $\frac{\lambda}{4}$, де λ – довжина хвилі.

Цей метод доцільно застосовувати для тіл округлої форми, наприклад, для автомобіля або штучного супутника Землі з встановленою на ньому антеною.

Наведені деякі приклади, ілюструють можливості програмного середовища 3ds Max для створення сіткових моделей тіл геометрично складної форми.

Як приклад розглянемо сіткову модель літака типу 7J7, довжина якого значно перевищує довжину хвилі, з Mesh-антеною, встановленою зверху на фюзеляжі (рис. 6). Для побудови цієї моде-

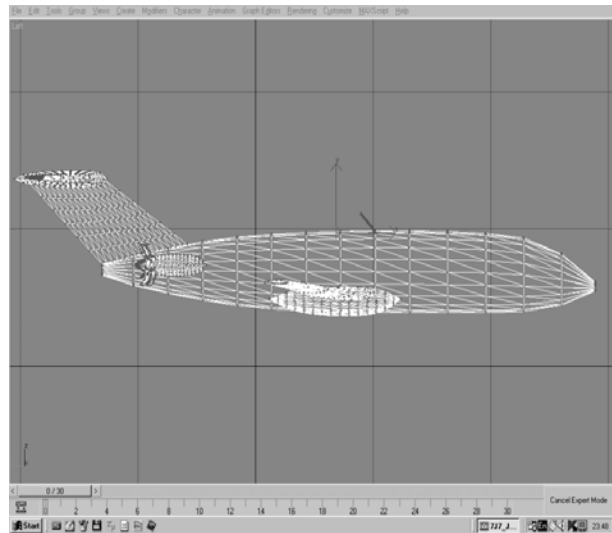


Рис. 6. Приклад сіткової моделі літака типу 7J7

лі було використано майже 10000 елементів, довжина яких не перевищує 0,1 λ .

Наведені сіткові моделі можна застосовувати для розрахунку ДС системи типу „штатна антена – корпус літака” за методом моментів. Подібні моделі можна також застосовувати для прогнозування вирішення проблем електромагнітної сумісності у процесі проектування літаків, космічних об’єктів. Можливе їх використання у метеородіолокації (моделювання хмар), а також у задачах, пов’язаних з математичним моделюванням електромагнітного впливу блискавки на літак, де також потрібно знаходити розподіл струму по тілу геометрично складної конфігурації.

Висновки

Перспективним середовищем для сучасного сіткового моделювання об’єктів є 3ds Max (комп’ютерна анімація). Пасивні товсті вібратори моделюються із застосуванням Patch – технології, а активні вібратори – із застосуванням Mesh-технологій. Координати окремого елемента сіткової моделі зручно знаходити на основі програм, наведених у статті.

Список літератури

1. Мумра Р. Вычислительные методы в электродинамике. – М.: Мир, 1977.– 488 с.

Стаття надійшла до редакції 03.06.04.

В.А. Иванов, А.В. Вишневський

Методика построения сеточных моделей трехмерных электродинамических объектов

Рассмотрены особенности применения программной среды 3ds Max для построения сеточных моделей трехмерных электродинамических объектов геометрически сложной формы.

V.A. Ivanov, A.V. Vishnevsky

The method of constructing 3d wire-grid models of electrodynamic objects

3d Max programming environment peculiarities for constructing 3d wire-grid models of electrodynamic objects have been considered.