

УДК 004.942

А.А.Ящук

Луцький національний технічний університет

ВІЗУАЛІЗАЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ ПРОЦЕСУ СУШІННЯ МАТЕРІАЛУ ЗА ДОПОМОГОЮ БІБЛІОТЕКИ OPENTK ДЛЯ C# В СЕРЕДОВИЩІ VISUAL STUDIO

Ящук А.А. Візуалізація математичної моделі процесу сушіння матеріалу за допомогою бібліотеки OpenTK для C# в середовищі VisualStudio. У статті представлено дослідження можливостей бібліотеки OpenTK для трьохвимірної візуалізації математичної моделі процесу сушіння матеріалу.

Ключові слова: візуалізація, модель, VisualStudio, OpenTK.

Ящук А.А. Визуализация математической модели процесса сушки материала с помощью библиотеки OpenTK для C# в среде VisualStudio. В статье представлено исследование возможностей библиотеки OpenTK для трехмерной визуализации математической модели процесса сушки материала.

Ключевые слова: визуализация, модель, VisualStudio, OpenTK.

Yaschuk A.A. Visualization of the mathematical model of the process of drying of material using OpenTK library for C# in the VisualStudio environment. The research of capabilities of OpenTK library for 3D visualization of the mathematical model of the process of drying of material is submitted in the article.

Keywords: visualization, model, VisualStudio, OpenTK.

Постановка проблеми. Процес сушіння описується складними законами, що характеризують зміну параметрів шару матеріалу в просторі і часі. Основними параметрами, що характеризують матеріал в процесі сушіння є його вологість u , температура θ , а параметрами сушильного агента – вологовміст d і температура t . На процес сушіння впливають властивості матеріалу, зокрема питома теплоємність c_m , пористість ξ та ін. [1].

Одним з підходів до вирішення задач моделювання процесу сушіння товстого шару матеріалу є ступінчастий метод розрахунку [2]. В основі цього методу лежить припущення, що товстий шар матеріалу являє собою сукупність елементарних тонких шарів матеріалу, параметри яких у визначений момент часу є постійними за усім об'ємом кожного з цих тонких шарів.

Даний метод дає можливість отримати залежності для визначення тривалості сушіння, енерговитрат, оцінити вплив параметрів на процес сушіння та встановити деякі інші теплоенергетичні показники.

Реалізація такої моделі вимагає численних розрахунків, які доцільно виконувати з застосуванням комп'ютерної техніки. Результатами розрахунку є набір чисел в табличній формі, що характеризують параметри матеріалу і сушильного агента на різній глибині шару матеріалу в різний час від початку сушіння. Актуальною є проблема візуалізації цих даних з врахуванням особливостей цього процесу. Використання стандартних програмних засобів і компонентів для побудови графіків не дозволяє в повній мірі розкрити суть описаного процесу.

Аналіз останніх досліджень і публікацій. В результаті теоретичних і експериментальних досліджень було отримано математичні залежності, які описують процес сушіння шару насіння льону олійного [1]:

$$u_{i,j} = a \cdot \exp(-k \cdot \tau_j) \cdot (u_0 - u_p) + u_p, \quad (1)$$

$$a = 1,347708 - 0,168 \cdot v - 0,006387 \cdot t_{i,j} + 0,061 \cdot v^2, \quad (2)$$

$$k = 0,179083 - 0,10775 \cdot v - 0,004767 \cdot t_{i,j} + 0,0435 \cdot v^2 + 0,000063 \cdot t_{i,j}^2. \quad (3)$$

$$u_p = \left(-\frac{\ln(1-\varphi)}{1,76 \cdot 10^{-4} \cdot (\theta_{i,j-1} + 56,228)} \right)^{0,525}, \quad (4)$$

$$\varphi = \frac{745 \cdot d_{i,j}}{(622 + d_{i,j}) \cdot 10^{0,622 + 7,5 \cdot t_{i,j} / (238 + t_{i,j})}}, \quad (5)$$

$$d_{i,j} = d_{i-1,j} + 0,001 \cdot \frac{(u_{i,j-1} - u_{i,j}) \cdot \rho_{\text{мат.}}}{(100 + u_{i,j-1}) \cdot \rho_{\text{с.н.}} \cdot \xi}, \quad (6)$$

$$t_{i,j} = t_{i-1,j} - \frac{c_{mat} \cdot (\theta_{i-1,j} - \theta_{i,j}) + (u_{i,j-1} - u_{i,j}) \cdot (H_{ni,j} - H_{pi,j}) \cdot (d_{i,j} - d_{i,j-1})}{c_{нов} \cdot (u_{i,j-1} - u_{i,j}) \cdot (1 + d_{i,j})} \quad (7)$$

де $i, i-1$ – індекс, що відповідно позначає порядковий номер поточного і попереднього тонкого елементарного шару, який є складовою товстого шару матеріалу, що піддається сушінню, $j, j-1$ – індекс, що відповідно позначає поточний і попередній короткий проміжок часу, який є складовою загального періоду сушіння,

$u_{i,j}$ – вологість матеріалу в кожному i -му тонкому шарі в кінці короткого проміжку часу $\Delta\tau_j$, %, $\theta_{i,j}$ – температура матеріалу в кожному i -му тонкому шарі в кінці короткого проміжку часу $\Delta\tau_j$, °C, $d_{i,j}$ – вологовміст сушильного агента що проходить через i -й тонкий шар в короткий проміжок часу $\Delta\tau_j$, г/кг с.п.,

$t_{i,j}$ – температура сушильного агента що проходить через i -й тонкий шар в короткий проміжок часу $\Delta\tau_j$, °C,

a і k – розрахункові коефіцієнти сушіння,

u_p – рівноважна вологість, %, φ – відносна вологість,

c_{mat} – питома теплоємність матеріалу, Дж/(кг·K), $c_{нов}$ – питома теплоємність повітря, Дж/(кг·K),

ξ – пористість матеріалу,

$\rho_{mat.}$ – питома маса матеріалу, кг/м³,

$\rho_{с.п.}$ – густина сухого повітря, кг/м³,

$H_{ni,j}$ і $H_{pi,j}$ – відповідно ентальпія пари і рідини, кДж/кг.

Таким чином розрахунок процесу сушіння шару насіння льону олійного зводиться до послідовного розрахунку параметрів u, θ, d і t для кожного тонкого шару матеріалу вкінці кожного короткого проміжку часу, тривалістю $\Delta\tau$, з врахуванням рівноважної вологості насіння льону олійного u_p , швидкості сушильного агента $v_{с.а.}$, а також фізико-механічних властивостей матеріалу.

Вихідними даними для розрахунку в кожному i -му тонкому шарі в проміжок часу $\Delta\tau_j$ є вологість $u_{i,j-1}$ і температура $\theta_{i,j-1}$ матеріалу в кінці попереднього проміжку часу $\Delta\tau_{j-1}$, а також вологовміст $d_{i-1,j}$ і температура $t_{i-1,j}$ сушильного агента на виході з попереднього шару.

Згідно з представленою вище математичною моделлю була розроблена комп'ютерна програма [1], що автоматизує процес розрахунку (рис. 1).

Вихідні дані для розрахунку:

Початкова вологість матеріалу W_0 : 17
 Кінцева вологість матеріалу, %: 7
 Початкова температура матеріалу: 20
 Температура сушильного агента t_a : 45
 Вологовміст сушильного агента d : 6
 Висота шару матеріалу H , м: 0,03
 Швидкість сушильного агента V , м/с: 1,5
 Розрахункові короткі проміжки часу, хв: 1
 Також розраховувати охолодження після сушіння
 Температура повітря для охолодження: 20

Властивості матеріалу і сушильного агенту:

Теплоємність матеріалу C_p : 1,8 за формулою
 Теплоємність повітря C_p : 1,004
 Об'ємна маса матеріалу кг/м³: 640
 Питома вага повітря, кг/м³: 0,98
 Відобразити додаткові параметри

Кількість тонких шарів, для яких проводиться розрахунок i : 10
 За товщину тонкого шару прийняти: 0,003 м
 Прихована теплота пароутворення води g : 2330

Зупиняти розрахунок при кількості інтервалів j : 100

Результати розрахунку:

Тривалість сушіння матеріалу для досягнення необхідної вологості: 44,7 хв

Вологість, W_{ij}	Температура агента, t_{ij}	Вологовміст, d_{ij}	Температура матеріалу, T_{ij}					
	n=1	n=2	n=3	n=4	n=5	n=6	n=7	
l=1	15,8403432	16,3104953	16,5601771	16,7075565	16,8001873	16,8609386	16,901977	Wcp38= 7,86293955769439; Wcp39= 7,72109597861512;
l=2	15,4716622	15,6436309	15,9294258	16,1716923	16,3556577	16,4928063	16,595680	Wcp40= 7,58277807665182;
l=3	15,1133019	15,2938999	15,4591879	15,6639342	15,8623879	16,0326219	16,171467	Wcp41= 7,44792260627108;
l=4	14,7649734	14,9537597	15,1253229	15,2812706	15,4446622	15,6042904	15,747793	Wcp42= 7,31646551624019;
l=5	14,4263960	14,6221815	14,7996212	14,9599579	15,1044780	15,2426624	15,372924	Wcp43= 7,18833494609068;
								Wcp44= 7,06345697323086;
								Wcp45= 6,9417643253454;

Рис.1. Реалізація математичної моделі процесу сушіння шару матеріалу

Мета дослідження. Дослідити можливості бібліотеки OpenGL для 3D візуалізації математичних моделей процесу сушіння щільного шару матеріалу на платформі .Net Framework

Виклад основного матеріалу. OpenGL [3] — специфікація, що визначає незалежний від мови програмування крос-платформовий програмний інтерфейс для написання функцій, що використовують 2D та 3D комп'ютерну графіку. Цей інтерфейс містить набір функцій, які можуть використовуватися для малювання складних тривимірних сцен з простих примітивів. Широко застосовується індустрією комп'ютерних ігор і віртуальної реальності, у графічних інтерфейсах (Comviz, Clutter), при візуалізації наукових даних, в системах автоматизованого проектування тощо.

Платформа Microsoft.NET Framework не має вбудованих засобів підтримки бібліотеки OpenGL. Для застосування бібліотеки OpenGL на платформі Microsoft.NET Framework можна використовувати виклик функцій безпосередньо з динамічної бібліотеки Opengl.dll, яка є реалізацією бібліотеки OpenGL в операційній системі Windows. На даний момент існує декілька бібліотек для платформи Microsoft.NET Framework, в яких виконана необхідна реалізація викликів функцій з динамічної бібліотеки Opengl.dll, і які пропонують деякі додаткові можливості, що дозволяють спростити використання бібліотеки OpenGL на платформі Microsoft.NET Framework.

На даний момент найбільш широко використовуються бібліотеки Tao Framework [4] і Open Toolkit Library [5].

Бібліотека Open Toolkit Library (OpenTK) підтримує останні версії бібліотеки OpenGL і має більш зручний інтерфейс виклику функцій бібліотеки OpenGL, який дозволяє уникнути цілого ряду помилок при їх використанні. Бібліотека OpenTK підтримує не тільки Microsoft.NET Framework, а й Mono Framework, який являє собою проект з відкритим вихідним кодом міжплатформенної реалізації Microsoft.NET Framework.

Для використання бібліотеки OpenTK після її встановлення на локальному комп'ютері, у проект C# Windows Forms, створений у VisualStudio, було додано посилання на OpenTK.dll, а також OpenTK.GLControl.dll, що дозволило використовувати компонент GLControl для відображення 3D моделі безпосередньо на формі Form (рис.2).

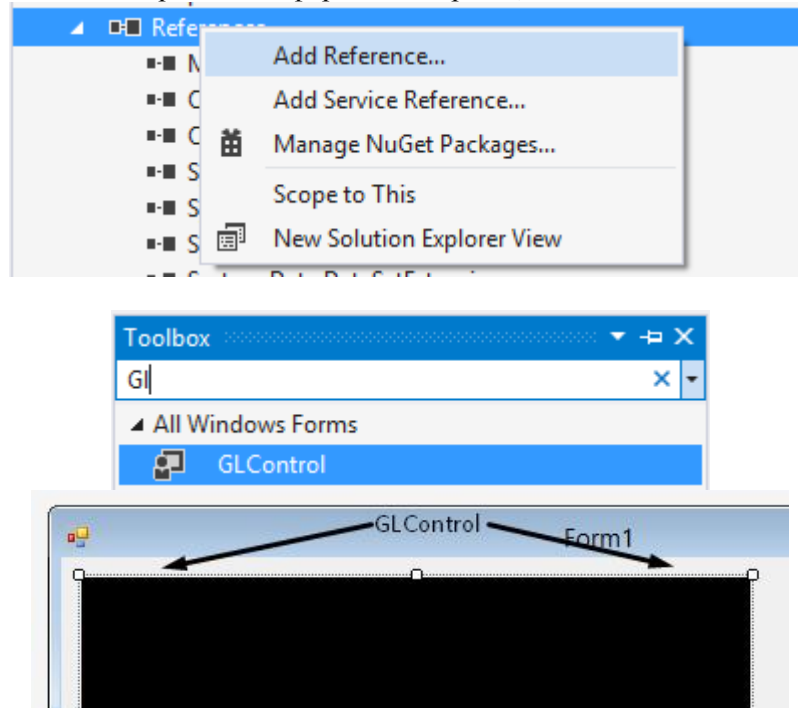


Рис.2. Додавання OpenTK.dll та OpenTK.GLControl.dll до проекту C# Windows Forms

Також потрібно включити в проект наступні простори імен: **using OpenTK;** і **using OpenTK.Graphics.OpenGL;** (рис. 3).

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
using OpenTK;  
using OpenTK.Graphics.OpenGL;
```

Рис.3. Простори імен OpenTK і OpenTK.Graphics.OpenGL

Відображення моделі пов'язане з двома методами. **glControl1_Load** - метод, який обробляє подію Load (завантаження) і **glControl1_Paint** - метод, який обробляє подію Paint, спрацьовує, наприклад, коли ми приховуємо, а потім знову відкриваємо вікно або, наприклад, змінюємо розміри вікна (рис.4).

```
1 reference  
private void glControl1_Load(object sender, EventArgs e)  
{  
    //...  
}  
  
1 reference  
private void glControl1_Paint(object sender, PaintEventArgs e)  
{  
    //...  
}
```

Рис.4. Методи, пов'язані з подіями Load і Paint компонента glControl

Далі наводиться код для події Load.

```
private void glControl1_Load(object sender, EventArgs e)  
{  
    Loaded = true; // Control1 завантажений  
    // колір заднього фону  
    GL.ClearColor(Color.Gray);  
    // об'єкти, що лежать ближче, перекривають ті, які знаходяться далі  
    GL.Enable(EnableCap.DepthTest);  
  
    /* матриця Frustum - визначає область видимості моделі, кут огляду - 75,  
    * відношення довжини до висоти -1,  
    * відстань до першої грані - 20, відстань до дальньої грані - 500 */  
    Matrix4 M4 = Matrix4.CreatePerspectiveFieldOfView((float)(75 * Math.PI /  
    180), 1, 20, 500);  
    GL.MatrixMode(MatrixMode.Projection);  
    GL.LoadMatrix(ref M4);  
  
    // положення камери  
    Matrix4 modelView = Matrix4.LookAt(75, 75, 75, 0, 0, 0, 0, 1, 0);  
    GL.MatrixMode(MatrixMode.ModelView);  
    GL.LoadMatrix(ref modelView);  
}
```

Відображення моделі реалізовано при натисканні кнопки Button. Оскільки процес динамічний, то для відображення зміни параметрів в часі застосовано компонент **Timer**. При натисканні кнопки викликається метод `InitializeTimer()`, в якому ініціалізуються початкові параметри таймера, після чого він активується.

```
private void InitializeTimer()
{
    // інтервал таймера - швидкість анімації
    timer1.Interval = int.Parse(textBox6.Text);
    timer1.Tick += new EventHandler(timer1_Tick);
// ...
    timer1.Enabled = true; // увімкнення таймера
}
```

Такі параметри матеріалу як температура і вологість в заданий момент часу доцільно представити у вигляді того чи іншого кольору (напр. висока температура – червоний, низька – синій). В тілі методу timer1_Tick() реалізовано інтерпретацію числових значень параметрів математичної моделі, отриманих внаслідок розрахунку, у кольори.

```
int RGB_R, RGB_G, RGB_B;
//...
Color clr = Color.FromArgb(RGB_R, RGB_G, RGB_B);
```

Як було сказано вище, математична модель описує процес сушіння товстого шару матеріалу, який умовно розбивається на тонкі шари, в межах яких вологість і температура є постійними в заданий момент часу. Відповідно, для відображення 3D моделі доцільно використати сукупність геометричних фігур певної товщини, розміщених одна над одною. Кожній з цих фігур у заданий момент часу буде присвоюватися колір, що є інтерпретацією числових параметрів фізичних величин того чи іншого шару матеріалу.

Для кожного виклику події **Tick** компонента **Timer** в компоненті **glControl1** повинно відображатися ряд плоских геометричних фігур однакової товщини певного кольору. Кожен наступний виклик події **Tick** через встановлений інтервал timer1.Interval відповідає кінцю кожного наступного короткого проміжку часу, за який параметри цих шарів матеріалу змінилися, відповідно зміниться і графічне представлення цих шарів.

Відображення кожного тонкого шару здійснюється викликом спеціально створеного методу **Refr()**. Методу передається ряд аргументів – геометричні розміри, умовна відстань між тонкими шарами, колір шару, стиль відображення.

```
public class Layer
{
    // метод, що додає шар матеріалу до 3D моделі
    public void Refr(int width, int length, int height, int level, int distance,
Color clr, bool contour)
    {
        /* width, length, height - геометричні розміри,
        * level - рівень (висота від 0 координати і
        * умовна відстань між тонкими шарами) */

        #region генерація шару
        // Малюємо паралелепіпед. Задаємо 6 граней у просторі
        /*нижня*/
        GL.Color3(clr); // колір
        GL.Begin(BeginMode.Polygon); // тип об'єкта - полігон
        GL.Vertex3(-width, level, length);
        GL.Vertex3(width, level, length);
        GL.Vertex3(width, level, -length);
        GL.Vertex3(-width, level, -length);
        GL.End();
        // аналогічно 5 інших граней з іншими координатами
        // ...
        if (contour) {
            // якщо аргумент contour – true, то вимальовувати ребра
            GL.Color3(Color.Black);
            GL.Begin(BeginMode.LineLoop); // тип об'єкта - замкнена лінія
            GL.Vertex3(-width, level, length);
```

```

GL.Vertex3(width, level, length);
GL.Vertex3(width, level, -length);
GL.Vertex3(-width, level, -length);
GL.End();
// ...
#endregion
}}}
```

Виклик методу **Refr()** в тілі методу **timer1_Tick()** поміщений в тіло циклу, число ітерацій якого рівне кількості умовних тонких шарів матеріалу. При кожній наступній ітерації циклу передаються числові параметри кожного наступного тонкого шару, відмінні від параметрів попереднього, відповідно кожен наступний тонкий шар візуально буде відрізнятися від попереднього. Градієнт температур і вологостей буде інтерпретуватися як градієнт кольору в 3D моделі. Число ітерацій методу **timer1_Tick()** буде відповідати числу коротких проміжків часу, на які розділена загальна тривалість сушіння.

Приклад реалізації візуальної моделі на основі математичної показаний на рис. 5.

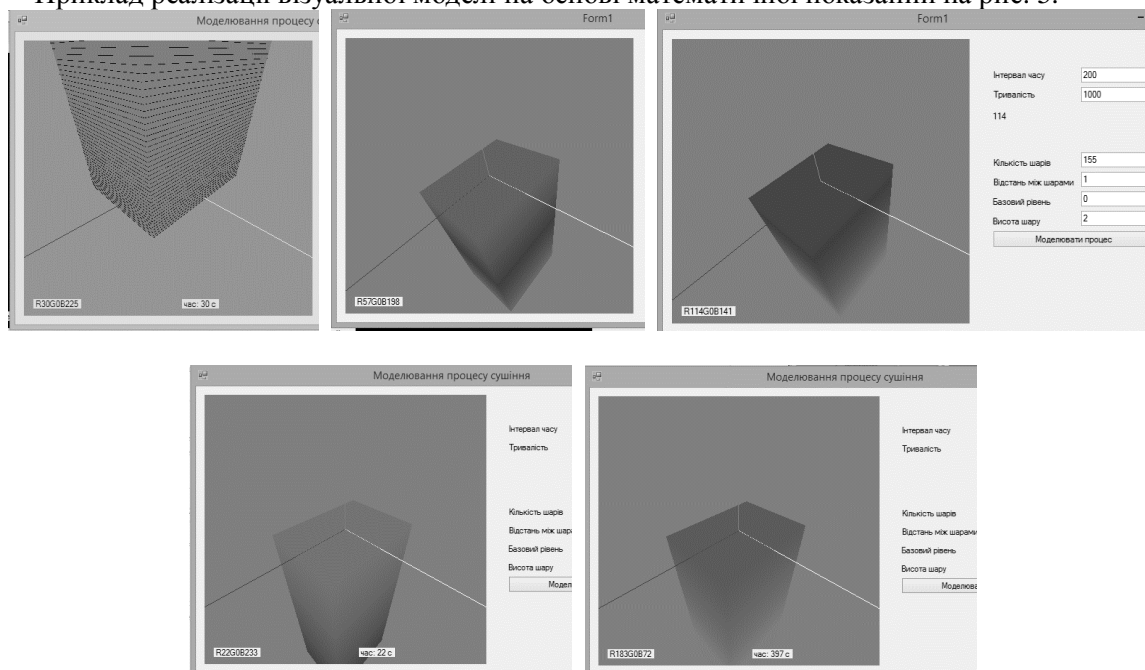


Рис.5. Візуальне моделювання зміни стану шару матеріалу в процесі його сушіння на основі математичної моделі

Велика популярність OpenGL, його широкі можливості, а також ряд переваг, таких як надійність, стабільність, простота використання, переносимість на різні платформи, розширюваність, роблять перспективними подальші дослідження в цій галузі.

Висновки. Досліджено можливості бібліотеки OpenTK для графічного відображення математичної моделі. Створена програма, яка демонструє зміну стану шару матеріалу в процесі його сушіння. Подальші дослідження дозволять більш широко розкрити потенціал OpenGL, для його застосування в даному напрямку. Розробка власних засобів графічної візуалізації доцільна у випадку необхідності представлення специфічних даних, коли існуючі засоби від сторонніх розробників не забезпечують в повній мірі бажаного результату.

Список використаних джерел

1. Ящук А.А. Моделювання процесу сушіння насіння льону олійного / А.А. Ящук, Р.В. Кірчук // Збірник наукових праць Вінницького національного аграрного університету[Текст] : серія: Технічні науки. – Вінниця : Видавничий центр ВНАУ, 2012. – Вип. 11. Т.1. (65). – С. 305 – 309.
2. Окунь Г.С. Расчет продолжительности и энергоёмности процесса сушки зерна в слое с помощью ЭВМ / Г.С.Окунь, И.И.Верцман, Ю.В. Есаков // Труды ВИМ, 1984, т. 100. – С. 73-80.
3. OpenGL. The Industry's Foundation for High Performance Graphics [Electronic Resource]. – Mode of access : URL : <https://www.opengl.org> . – Title from the screen.
4. The Tao Framework [Electronic Resource]. – Mode of access : URL : <http://sourceforge.net/projects/taoframework/> . – Title from the screen.
5. OpenTK [Electronic Resource]. – Mode of access : URL : <http://www.opentk.com/> . – Title from the screen.