

УДК 378.147:004.25(07)

П.І.Чернега, І.А.Сліпухіна, Є.С.Поляев
Національний авіаційний університет

СТВОРЕННЯ АНІМОВАНИХ МОДЕЛЕЙ З КУРСУ ФІЗИКИ ЯК ЕЛЕМЕНТ ОСОБИСНІСНО- ОРІЄНТОВАНОГО НАВЧАННЯ (НА ПРИКЛАДІ FLASH PROFESSIONAL CS 5)

Розроблено і створено анімовані моделі з оптики, що є одним із способів реалізації особистісно-орієнтованого навчання (PBL) з використанням міжпредметних зв'язків. Проаналізовано деякі деталі процесу алгоритмізації та візуалізації фізичних задач в оболонці Flash Professional CS 5. Доведено актуальність і перспективність використання даного способу організації навчального процесу як елемента новітнього освітнього середовища

Разработаны и созданы анимированные модели по оптике, что является одним из способов реализации личностно-ориентированного обучения (PBL) с использованием межпредметных связей. Проанализированы некоторые детали процесса алгоритмизации и визуализации физических задач в оболочке Flash Professional CS 5. Доказано актуальность и перспективность использования данного способа организации учебного процесса как элемента новейшего образовательной среды

Designed and developed an animated model of optics, which is one of the ways of realization of personality-based learning (PBL) with the use of interdisciplinary connections. Analyzed some of the details of the algorithms and visualization of physical problems in the shell of Flash Professional CS 5. The relevance and prospects of using this method, the educational process as an element of modern educational environment

Ключові слова: Flash Professional CS 5, ActionScript 3.0, технології моделювання, вивчення фізики, фізичні моделі, особистісно-орієнтоване навчання.

Гуманізація освіти як частина сучасної освітньої парадигми займає важливе місце в багатогранному процесі сучасного суспільства [1, с.156-158].

Гуманістично-орієнтована дидактична система включає: нові цілі освіти, спрямовані на самореалізацію учня і вчителя через засвоєння системи загальнолюдських цінностей; новий зміст освіти, співвіднесений з особистими потребами суб'єктів навчального процесу; систему особистісно-орієнтованих методів і форм навчання, їх вибір учнями та вчителями (або викладачами і студентами); демократичний стиль педагогічного та професійного спілкування; включення діяльності вчення в структуру процесів саморозвитку та життєвого самовизначення особистості учнів. Особистісно-орієнтоване навчання – один з компонентів освітнього середовища, зорієнтованого на врахування актуальних і перспективних потреб навчання і розвитку пізнавальних можливостей студентів [2, с.15].

Проблема міждисциплінарних зв'язків в процесі підготовки фахівців наразі є особливо актуальною, бо вона є безпосереднім віддзеркаленням передових технологічних інновацій, наукове підґрунтя яких знаходиться на межі кількох фундаментальних дисциплін (математики, фізики, хімії, біології, філософії) та похідних від них (фізики тонких плівок, наноструктур, хімії високомолекулярних сполук тощо).

Перенесення мотиву з кінцевого результату навчання на саму навчальну діяльність є виключно важливим фактором, багато в чому визначає ефективність процесу навчання. Найбільший ефективною у навчанні є безпосередня предметна мотивація, але й опосередкована може відігравати істотну роль. Один з таких опосередкованих мотивів, здатних перевернути інтерес учнів на вивчення фізики, пов'язаний з великою популярністю комп'ютерів та інформаційних технологій у сучасних студентів, багато з яких вже на молодших курсах

технічних вузів на певному рівні володіють мовами програмування, включаючи іноді навіть мови високого рівня, наприклад, С або С++.

Теза про підвищення мотивації в навчанні фізики через інтеграцію з фахово-орієнтованими дисциплінами була взята за основу в процесі моделювання деяких фундаментальних законів і явищ з оптики [3, с.125].

Візуалізація процесів – найбільш оптимальний спосіб розвитку у студентів практики аналізу і синтезу з використанням можливостей алгоритмування та програмування за допомогою сучасних продуктів.

Для реалізації зазначеної мети були використані можливості середовища Flash Professional CS 5, однією з переваг якого є на-

явність векторного морфінгу, або, простіше кажучи, плавного перетікання одного ключового кадру у інший [4, с.25-47]. Цей факт дозволив створити складні мультиплікаційні сцени. Застосування мови програмування ActionScript 3.0, що підтримується Flash CS 5, забезпечило інтерактивну взаємодію користувача і програми через керуючі елементи: кнопки, повзунки, перемикачі тощо.

В основі моделювання явища фотоефекту використано експеримент, який має в більшості наукових джерел назву «дослід Столетова»: освітлення катода вакуумного діода світлом різної частоти, інтенсивності визначає закономірності фотоефекту (рис.1).

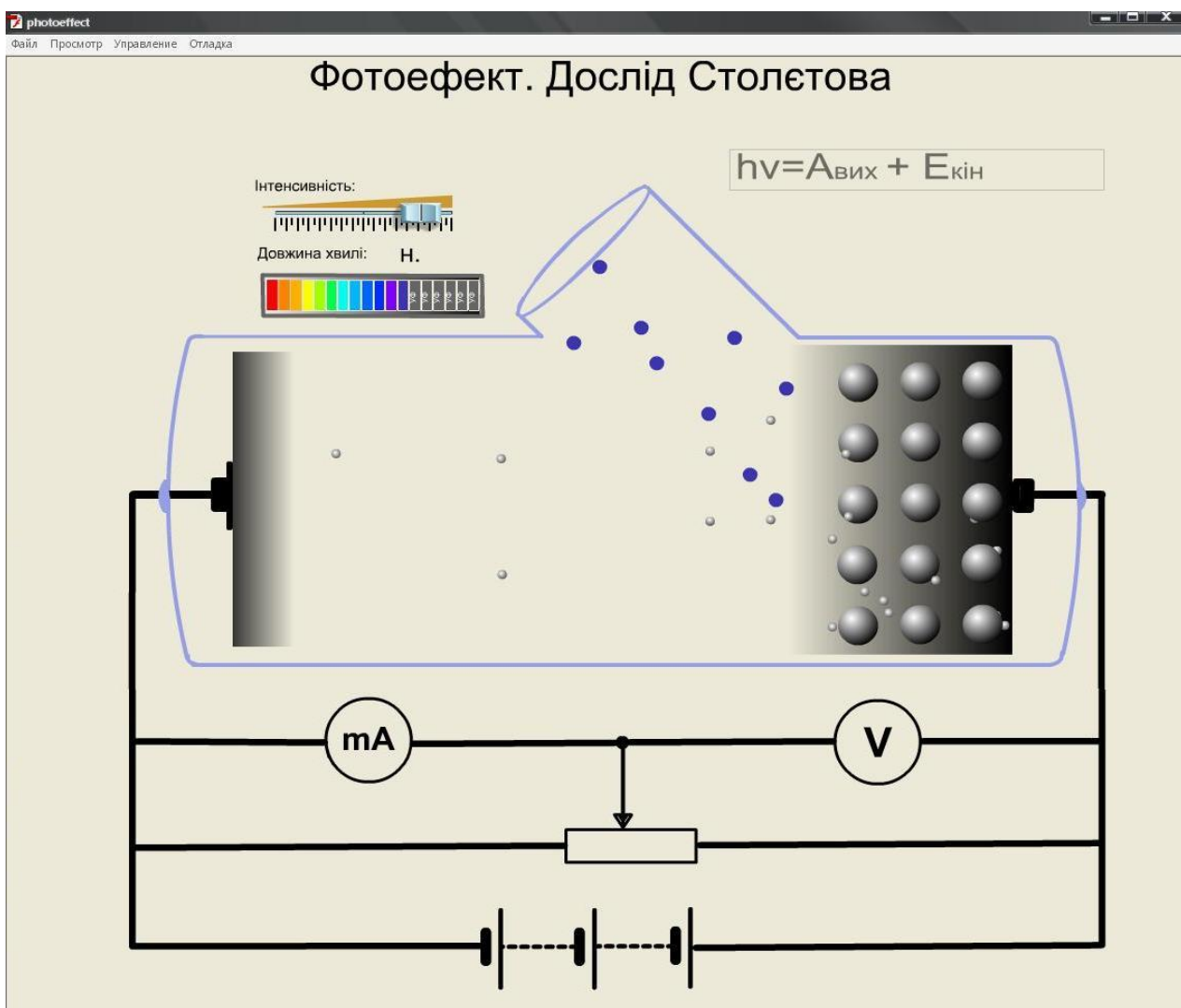


Рис. 1

Інтерактивний вибір користувачем за допомогою повзунка світла оптичного діапазону, а також регулювання потенціометром різниці потенціалів між електродами дають можливість якісної та кількісної оцінки вольт-амперної

характеристики $I(U)$ (ВАХ) фотоелемента, що включає в себе визначення величини струму насичення, запірної напруги, роботи виходу електронів. Також дослід та як і програма яскраво демонструє явище червоної межі (рис.2).

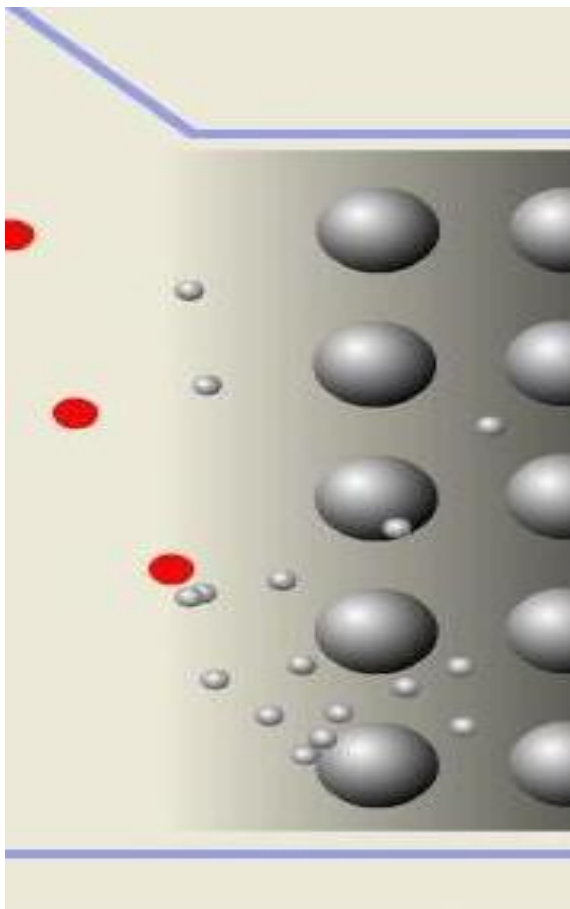


Рис.2

Для реалізації поставленої мети в головному програмному коді Flash CS 5 відкривається цикл, в якому відносно заданих початкових параметрів уже працює анімація. В тілі цього циклу працює команда «слухач подій», котра реагує на натискання чи переміщення кнопок на екрані користувача.

При натисканні клавіші змінюється значення попереднього параметра i , коли програмний код у тілі циклу дійде до рядка звертання до цього параметра, анімація змінюється у відповідності до нього.

В ході створення програми виникали деякі проблеми, наприклад, із написанням функцій

універсальних функцій. Для кожної дії були поставлені у відповідність свої функції, які спрощували програмний код. Це дозволило використовувати, наприклад, функцію «Vilet()», яка вміщує у собі різні масиви та майже ніяких констант (усі константи, які видно нижче, потрібні для порівняння або ж для плавного руху програми), багато разів, лише звертаючись до неї. Наприклад, для кожного значення інтенсивності та довжини хвилі, які вибере користувач, потрібно було б писати код, який наведено нижче:

```
function Vilet()
{
    acsel = _root.polus * pot * 95000;
    vmgn[i] = vmgn[i] + acsel * 1 / 800;
    xmgn[i] = xmgn[i] + vmgn[i] * 1 / 800;
    ymov = yel[i] + (t - tmov[i]) * speedy[i];
    setProperty("el" + i, _x, xmgn[i]);
    setProperty("el" + i, _y, ymov);
    if (xmgn[i] < 200 | ymov > 600 | ymov < 290 | xmgn[i] >
    730)
```

```
{
    removeMovieClip ("el" + i);
    enel[i] = 0;
} // end if
} // End of the function
```

Але за допомогою масивів та ряду універсальних команд, використовуючи лише ключову назву функції Vilet() та деякі посилання, можна вирішити цю проблему.

В процесі написання програма кілька раз модифікувалася. В початковій версії програми ряд функцій, доступних наразі, були лише у

розробці. У ході покращення програми виступали різні нові підходи, ідеї, нові функції та інші різні частини програмного коду. Ось, наприклад, для створення динамічності та реалістичності було використано оператор Math.random(), що у програмному коді має вигляд:

```
for (i = 1; i < 6; i++)
{
    for (j = 1; j < 4; j++)
    {
        xe = 810 + 85 * Math.cos(10 * Math.random());
        ye = 450 + 130 * Math.sin(10 *
Math.random());
        setProperty("ion" + i + j, _x, x);
        setProperty("ion" + i + j, _y, y);
        setProperty("fel" + i + j, _x, xe);
        setProperty("fel" + i + j, _y, ye);
    } // end of for
} // end of for
```

Він відповідає за генерацію псевдовипадкових чисел, що дозволяє використати у програмі ефект хаотичності, як це видно в різно-

манітності траєкторій руху фотоелектронів та квантів світла.

Для демонстрації теплового руху іонів було використано команди :

```
setProperty ("ion"+i+j, _x,x); setProperty ("ion"+i+j, _y,y);
Для розрахунку траєкторії електронів було використано такі рядки :
```

```
x = 750 + Math.random() / 2 + (j - 1) * 60 + Math.cos(t * i)
* Math.cos(t * (50 + i * 2) + i * 3);
y = 330 + Math.random() / 2 + (i - 1) * 60 + Math.sin(t * j)
* Math.sin(t * (50 + j * 2) + j * 3);
```

Цікаві деталі виявилися в процесі анімації досліду, який доводить поперечну властивість світлових (електромагнітних) хвиль. Природний світловий промінь по чергово пропускають через два поляроїди (поляризатор й аналізатор), вимірюючи при цьому інтенсивність світла, яке пройшло через кожний з них. Внаслідок відповідних вимірювань переконаються в справедливості закону Малюса: $I = \frac{1}{2} I_0 \cos^2 \varphi$, де I , I_0 – інтенсивності світла, яке пройшло поляризатор і аналізатор та інтенсивність природного світла; φ – кут між поляризатором і аналізатором. Візуально спостерігач має зареєструвати

сталу інтенсивність світла за поляризатором, рівну $\frac{1}{2} I_0$, а також зростання і спадання інтенсивності від максимального значення до нуля за аналізатором між положеннями «паралельно» та «схрещено». Інтерфейс програми наведено на рис.3.

За допомогою простого алгоритму та створення відповідних функцій, з постійною перевіркою отриманого результату було створено програму візуалізації закону Малюса.

Щоб користувач взаємодіяв із програмою та обертав поляроїди довільним чином було створено керуючі кнопки. Нижче наведено про-

грамний код кнопки, яка додає 10° до кута по-

вороту поляризатора:

```
on (press)
{
  PP.text=int(PP.text)+10
  if (int(PP.text)==180 ) {PP.text=0}
}
```

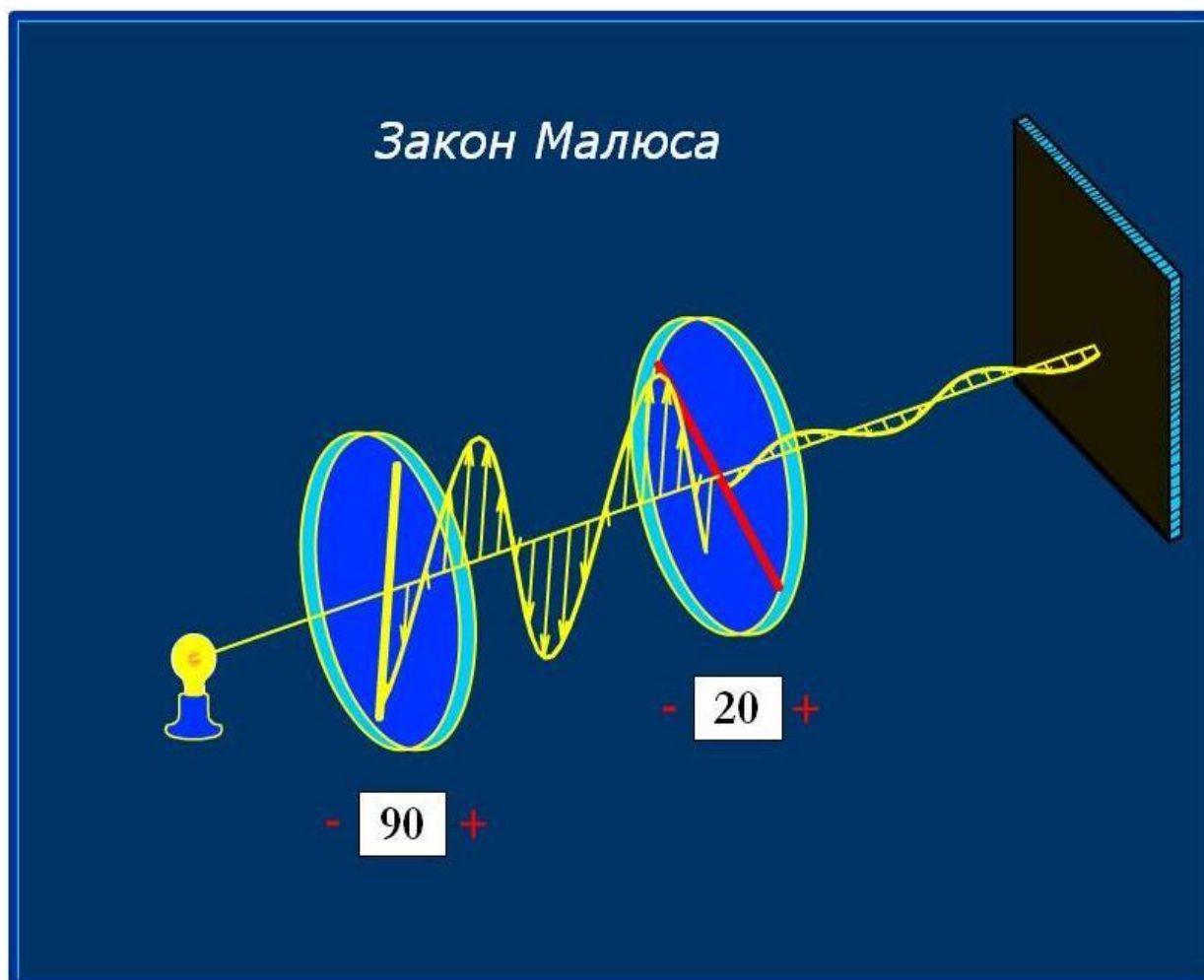


Рис.3

Тобто, при натисканні кнопки «+» константи PP присвоюється значення на 10° більше від попереднього і задається циклічність при повороті на кут, більший за 180° .

Аналогічні дії виконано для кнопки «-». Також тут можна побачити «огорожу», тобто

лінію, за котру користувач не має права виходити. У нашому разі межею є 180° .

Форму синусоїди світла в залежності від кута між поляризатором і аналізатором задаємо функцією Wave11():

```
function Wave11()
{
  ly1 = lx1 * Math.tan(alpha);
  fi = pi * phi / 180 + pi / 6;
  sina = Math.sin(fi);
  cosa = Math.cos(fi - pi / 8);
  sinx = Math.tan(alpha);
}
```

```

yo = Ax * Math.sin(omegax * pii - kcx * t);
yo = Ax * Math.sin(-kcx * t);
for (i = 1; i <= Xendl; i = i + kx)
{
    sin = Ax * Math.sin(omegax * i * pii - kcx * t);
    xi = 5.000000E-001 * sin * cosa;
    yi = sin * sina;
    _root.createEmptyMovieClip("line" + i, 3250 + i * 10);
    eval("line" + i).lineStyle(2, 16776960, 100);
    eval("line" + i).moveTo(C + lx1 + i, D - ly1 - i *
sinx);
    eval("line" + i).lineTo(C + i + 5.000000E-001 * sin *
cosa + lx1, sin * sina + D - i * sinx - ly1);
    j = i + kx;
    arrows.duplicateMovieClip("arrows1" + j, j + 4150);
    eval("arrows1" + j)._x = C + i + 5.000000E-001 * sin *
cosa + lx1;
    eval("arrows1" + j)._y = sin * sina + D - i * sinx -
ly1;
    a = 180 * Math.atan(yi / xi) / pi;
    if (xi < 0)
    {
        a = a + 180;
    } // end if
    eval("arrows1" + j)._rotation = a;
} // end of for
i = Xendl;
sin = Ax * Math.sin(omegax * i * pii - kcx * t);
_root.createEmptyMovieClip("line" + i, 2150 + i * 10);
eval("line" + i).lineStyle(2, 16776960, 100);
eval("line" + i).moveTo(C + lx1 + i, D - i * sinx - ly1);
eval("line" + i).lineTo(C + i + 5.000000E-001 * sin * cosa
+ lx1, sin * sina + D - i * sinx - ly1);
} // End of the function

```

Яскравість поля екрану задавалась за допомогою створення міні анімації light та накладання на поле екрану. В залежності від констант, що надаються програмі, шляхом зміни кута нахилу поляроїда, колір, що накладається на екран змінюється.

В моделюванні досліду Юнга (інтерференція від двох щілин) були використані описані вище прийоми і можливості Flash (рис.4).

Організація та процес створення анімованих моделей з фізики продемонстрували високу ефективність такого методичного прийому в

сучасному освітньому середовищі. Надалі планується не тільки розробка інших фізичних моделей, а й об'єднання їх у повноправний інтерактивний ресурс з метою використання фізичних моделей для поглиблення розуміння студентами суті фундаментальних законів, явищ, процесів, оптимізації лабораторних робіт для студентів заочних та дистанційних форм навчання, більш яскравого сприйняття останніми філософії науки в цілому.

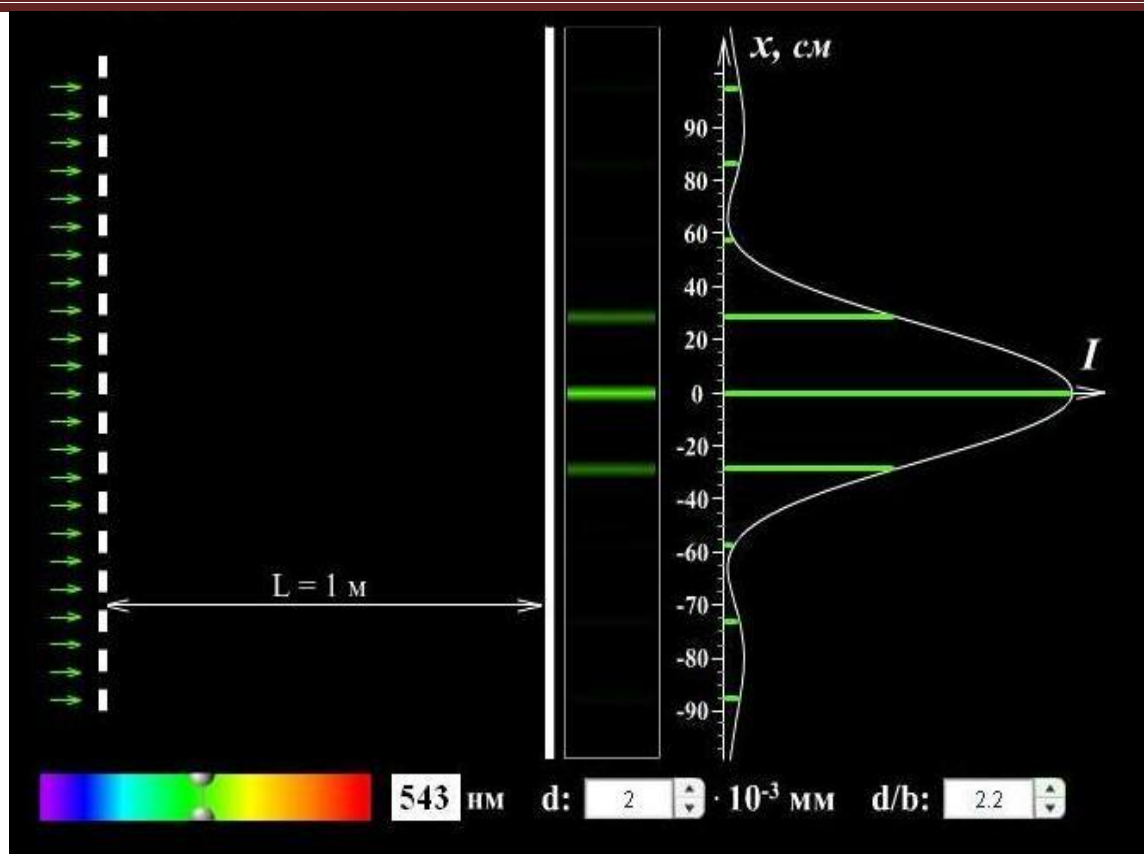


Рис. 4

Список літератури:

1. Енциклопедія освіти/ Акад.пед.наук України; головний редактор В.Г.Кремень.– К.: Юрінком Інтер, 2008.– 1040 с.
2. Якиманская И.С. Разработка технологии личностно-ориентированного обучения //Вопросы психологии, 1995.–№2. – с.13-21.

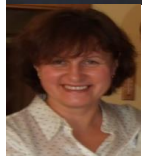
3. Гулд Х., Тобочник Я. Компьютерное моделирование в физике: В 2 т. Пер. с англ. – М: Мир, 1990. – 349 с.

Использование ADOBE® FLASH® PROFESSIONAL CS5 & CS5.5// http://help.adobe.com/ru_RU/flash/cs/using/flash_cs5_help.pdf/ вільний доступ

Відомості про авторів



Чернега Петро Іванович – к.ф.-м.н., доцент, доцент кафедри загальної фізики Національного авіаційного університету. Розробник навчально-методичного комплексу з курсу загальної фізики за ECTS. Автор багатьох наукових та методичних праць. Захоплення: командні види спорту, автомобілізм.



Сліпухіна Ірина Андріївна – к.ф.-м.н., доцент, доцент кафедри загальної фізики Національного авіаційного університету. Коло наукових інтересів: фізика напівпровідників, методика викладання фізики у вищій школі.



Поляєв Євген Сергійович - студент Інституту інформаційно-діагностичних систем Національного авіаційного університету. Коло наукових інтересів включає вивчення особливостей сучасних мов програмування.

Стаття надійшла до редакції 10.04.2011р.