

NGHIÊN CỨU ỨNG DỤNG PHÉP TÍNH VI PHÂN VÀ LẬP TRÌNH PYTHON ĐỂ MÔ PHỎNG CHUYỂN ĐỘNG KHÔNG ĐỀU

Trần Xuân Hối

Trường Đại học Phú Yên

Email: tranxuanhoi@pyu.edu.vn

(Ngày nhận bài: 4/3/2025, ngày nhận bài chỉnh sửa: 24/4/2025, ngày duyệt đăng: 22/5/2025)

TÓM TẮT

Nghiên cứu này trình bày phương pháp ứng dụng phép tính vi phân kết hợp với ngôn ngữ lập trình Python để tạo mô phỏng vật chuyển động không đều. Sau đó tác giả đã áp dụng phương pháp cho ba ví dụ minh họa, gồm vật rơi tự do rồi nảy lên từ mặt đất; dao động của con lắc đơn; phép cộng vận tốc thông qua mô phỏng bài toán chó đuổi thỏ. Kết quả nghiên cứu cho thấy việc sử dụng phương pháp này có nhiều tiện lợi như có thể thực hiện trên cả máy tính và các thiết bị di động, các thông số của hình động có thể điều chỉnh được theo mong muốn. Hơn nữa, phương pháp đã cho thấy ý nghĩa và ứng dụng của phép tính vi phân trong mô phỏng cũng như tích hợp một số môn học khác nhau, và có thể là công cụ để hỗ trợ trong dạy học và nghiên cứu lĩnh vực khoa học tự nhiên và công nghệ hiện nay.

Từ khóa: Vi phân, mô phỏng, Vpython

1. Mở đầu

Vi phân là một khái niệm cơ bản trong toán học giải tích. Vi phân có nhiều ứng dụng trong các lĩnh vực khác nhau. Nó giúp tìm hiểu và giải quyết các bài toán liên quan đến tốc độ thay đổi, bài toán cực trị, cũng như đường tiếp tuyến của các hàm số trong toán học. Vi phân giúp mô tả các hiện tượng vật lý như vận tốc và gia tốc của các vật chuyển động, dòng chảy và dòng điện, tổng quát hơn là khảo sát sự thay đổi của các đại lượng vật lý theo thời gian [1], [2]. Đặc biệt, vi phân còn có thể ứng dụng để tính toán trong mô phỏng các đối tượng trong nghiên cứu và kỹ thuật. Các chương trình máy tính phát triển cao đang cho phép những người làm mô phỏng sử dụng phép tính để tương tác với các đối tượng và tốc độ khác nhau.

Khái niệm gia tốc cũng như các chuyển động biến đổi nói chung là một

trong những nội dung quan trọng trong Vật lý nhưng cũng là nội dung gây khó hiểu đối với học sinh, sinh viên. Sẽ không ngạc nhiên khi sinh viên cảm thấy khó hiểu với nguyên lý cơ bản của vi phân: vận tốc là giới hạn của tỉ số giữa quãng đường đi được và khoảng thời gian ngắn dùng để đi hết quãng đường đó [3]. Có một số nghiên cứu đã ứng dụng vi phân và Mathematica để khảo sát chuyển động biến đổi của vật rơi tự do và dao động của con lắc đơn [2] cũng như dùng Excel để mô phỏng chuyển động của con lắc đơn [3].

Nghiên cứu này sẽ tiếp cận và trình bày cách ứng dụng vi phân và Vpython để tạo một số chuyển động biến đổi, sau đó đưa ra một vài ví dụ áp dụng như mô phỏng vật rơi tự do, mô phỏng sự dao động của một con lắc đơn và minh họa bài toán chó đuổi thỏ. Ưu điểm của phương pháp này là tạo ra các ảnh động

một cách nhanh chóng, hoạt động giống như bản chất của vật cũng như chủ động điều chỉnh được các thông số của chuyển động. Hơn nữa, phương pháp còn có thể thực hiện trực tiếp trên trình duyệt của máy tính hoặc điện thoại mà không cần phải cài đặt.

Với xu thế dạy học tích hợp và tăng cường ứng dụng công nghệ như hiện nay, nghiên cứu này hi vọng đóng góp phần nào đó trong công cuộc đổi mới phương pháp dạy học, phát triển năng lực người học cũng như ứng dụng công nghệ trong nghiên cứu và kỹ thuật.

2. Phương pháp

2.1. Phép lấy vi phân

Phép lấy vi phân cho ta biết cách một hàm số thay đổi khi biến số của nó thay đổi. Nhờ phép lấy vi phân mà ta có thể xác định được giá trị gần đúng của một hàm số hoặc một đại lượng cần khảo sát theo sự thay đổi của biến số [4]. Trong trường hợp hàm một biến $y = f(x)$ thì khi cho biến x thay đổi một số gia Δx , bằng phép lấy vi phân ta tìm được độ biến thiên của y . Tùy thuộc vào yêu cầu về sai số cho phép của bài toán mà ta có thể điều chỉnh giá trị của số gia cho phù hợp. Khi xác định được độ biến thiên tức thời (độ dốc) của đại lượng cần khảo sát, ta tiếp tục cho biến thay đổi một số gia nữa và lại xác định độ biến thiên tại thời điểm cách thời điểm ban đầu 2 số gia. Cứ như vậy ta có thể tạo vòng lặp tính toán và xác định được tập xác định và tập giá trị của hàm số trong khoảng cần khảo sát. Từ hai tập số này ta có thể vẽ được đồ thị, biết được quỹ đạo chuyển động, hoặc

thậm chí mô phỏng được hoạt động của đối tượng.

Theo định nghĩa đạo hàm, ta có [4]:

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} \quad (1)$$

Nếu Δx đủ nhỏ theo yêu cầu của bài toán thì:

$$f'(x_0) \approx \frac{\Delta y}{\Delta x}$$

Ta thu được độ biến thiên của y là:

$$\Delta y \approx f'(x_0) \cdot \Delta x$$

Hay:

$$f(x_0 + \Delta x) - f(x_0) \approx f'(x_0) \cdot \Delta x$$

Viết lại:

$$f(x_0 + \Delta x) \approx f(x_0) + f'(x_0) \cdot \Delta x \quad (2)$$

Công thức (2) cho phép ta xác định chuỗi các cặp giá trị liên tiếp nhau khi thực hiện vòng lặp tính toán ứng với số gia cho trước. Để minh họa cho công thức (2), sau đây ta áp dụng cho một số ví dụ cụ thể.

2.2. Áp dụng cho vật rơi tự do

Vị trí của vật rơi tự do từ trạng thái đứng yên được xác định theo:

$$h = \frac{gt^2}{2} \quad (3)$$

Áp dụng công thức (2), ta có:

$$h(t + \Delta t) \approx h(t) + v \cdot \Delta t \quad (4)$$

2.3. Áp dụng cho con lắc đơn

Gia tốc góc của con lắc đơn được xác định theo công thức (5), với θ là góc lệch của con lắc:

$$\gamma = \frac{g}{l} \sin \theta \quad (5)$$

Tốc độ góc:

$$\omega = \gamma t \quad (6)$$

Góc lệch:

$$\theta = \frac{\gamma t^2}{2} \quad (7)$$

Áp dụng công thức (2) cho (6) và (7) ta được:

$$(t + \Delta t) \approx \omega(t) + \gamma \cdot \Delta t \quad (8)$$

$$(t + \Delta t) \approx \theta(t) + \omega_t \cdot \Delta t \quad (9)$$

2.4. Áp dụng cho bài toán chó đuổi thỏ

Bài toán: Một con thỏ (T) chạy dọc theo trục x với tốc độ không đổi v thì bị một con chó (C) xuất phát từ một điểm trên trục y đuổi theo với tốc độ không đổi V , biết rằng chó chạy nhanh hơn thỏ và luôn hướng về phía thỏ. Hãy mô phỏng hiện tượng.

Giải bài toán ta thu được các kết quả sau [5]:

Vị trí thỏ:

$$x_T = vt \quad (10)$$

ta được:

$$x_T(t + \Delta t) \approx x_T(t) + v \cdot \Delta t \quad (11)$$

Vận tốc của chó tại thời điểm bất kì được xác định theo (12), trong đó R là bán kính vector, $\vec{n}$ là vector đơn vị:

$$\vec{V} = V \frac{\vec{R}_t - \vec{R}_c}{\Delta R} = V \vec{n} \quad (12)$$

Áp dụng công thức (2) cho quãng đường của chó đi được:

$$\vec{s}_C(t + \Delta t) \approx \vec{s}_C(t) + \vec{V} \cdot \Delta t \quad (13)$$

2.5. Tạo mô phỏng cho chuyển động

Theo nguyên tắc tạo hoạt hình, một đoạn hoạt hình gồm nhiều ảnh tĩnh, mỗi ảnh có những thay đổi nhỏ về hình dạng hoặc vị trí, chúng xuất hiện và biến mất một cách liên tiếp trong một khoảng thời gian ngắn tạo ra cảm giác cử động hoặc chuyển động khi ta quan sát. Đối với các đại lượng biến đổi không đều, chẳng hạn như các dao động điều hòa, vật rơi tự do hoặc chuyển động có gia tốc nói chung thì việc tạo hoạt hình giống như thật sẽ phức tạp hơn so với các chuyển động đều bởi quy luật xuất hiện của các ảnh tĩnh là không đều. Các vị trí hoặc hình dạng của vật biến đổi theo một quy luật hoặc hàm số nào đó.

Có nhiều cách để vẽ các ảnh tĩnh như vẽ thủ công, vẽ bằng các phần mềm đồ họa trên máy tính rồi “ghép” chúng lại thành hình động. Đối với các hình động đơn giản và mang tính học thuật thì Web Vpython là một công cụ đáng xem xét. Web Vpython là một phần của thư viện Python, kết hợp giữa ngôn ngữ lập trình Python với các module đồ họa 3D được định nghĩa sẵn bằng các hàm [6]. Ứng dụng thư viện này cho phép người dùng tạo các đối tượng 3D, và điều khiển các đối tượng này chuyển động theo một quy luật hoặc một hàm nào đó. Do vậy, Web Vpython là một công cụ thuận tiện để minh họa và mô phỏng toán học, cũng như các hiện tượng tự nhiên. Đặc biệt Web VPython là một trong những công cụ tốt để hỗ trợ trong dạy học do tính đơn giản và cấp phép nguồn mở của nó. Hơn nữa, nó cho phép người dùng viết và chạy các chương trình trực tiếp trên trình duyệt web máy tính hoặc các thiết bị di động như máy tính bảng và điện thoại di động tại trang <https://glowScript.org>.

Trong nghiên cứu này, các công thức vi phân (4), (8), (9), (11) và (13) được viết theo ngôn ngữ Python và chạy trên trang Web Vpython và được thể hiện trên phụ lục 1, phụ lục 2 và phụ lục 3. Các khai báo về điều kiện ban đầu của hệ cũng như đồ họa của các đối tượng tất nhiên được thực hiện trước khi dùng các hàm để điều khiển đối tượng chuyển động.

3. Kết quả nghiên cứu và thảo luận

3.1. Kết quả nghiên cứu

Các phương trình vi phân (4), (9), (11) và (13) được thể hiện trên Vpython lần lượt là phương trình (4'), (9'), (11')

và (13'), trong đó các đại lượng tương ứng được liệt kê ở bảng 1. Ở đây, vì bài báo tập trung khai thác ứng dụng của vi

phân nên việc tập trung vào bàn luận về các đoạn mã Python liên quan sẽ trình bày ở một nghiên cứu khác.

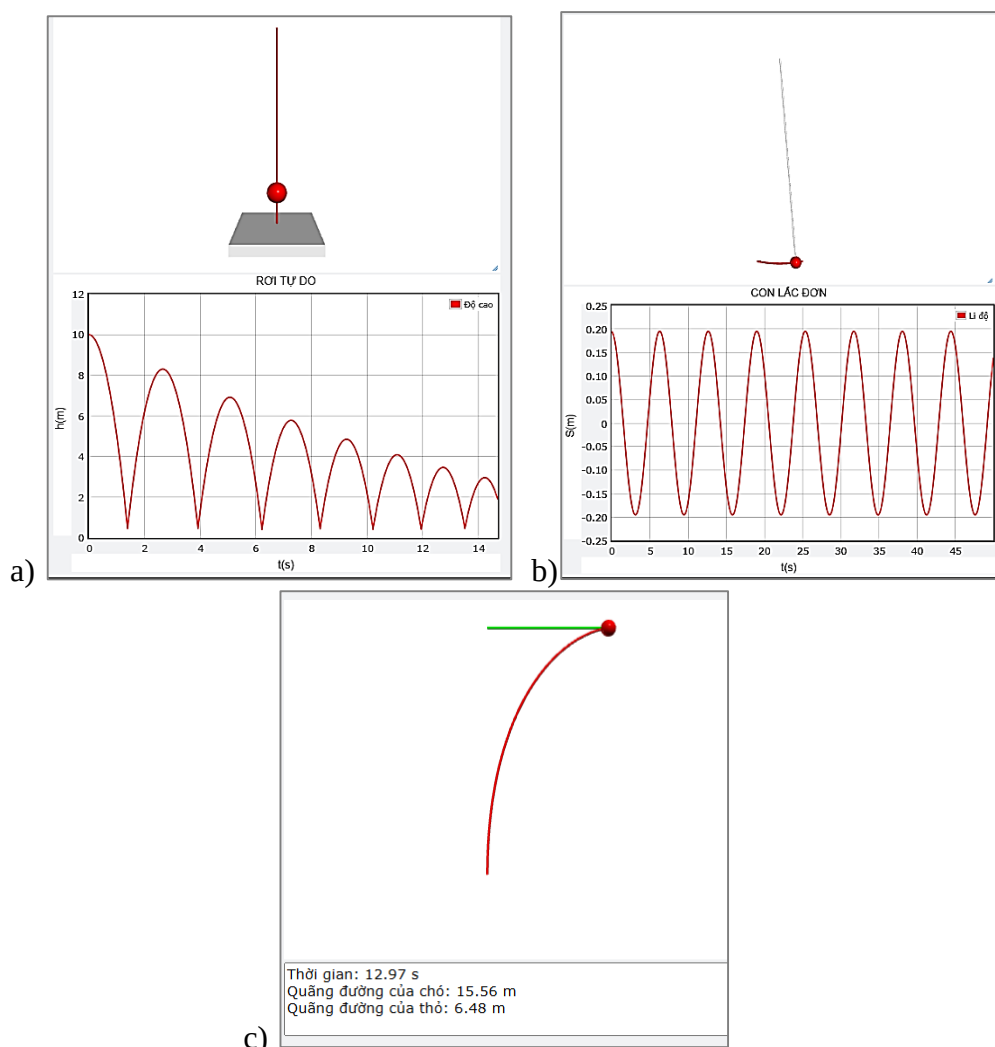
Bảng 1: Các biến và phương trình thể hiện trong Vpython

Đại lượng trong phương trình vi phân	Đại lượng tương ứng trong Vpython	Ghi chú
h	ball.pos.y	
v	ball.velocity.y	
Δt	dt	
θ	bob.angle	
ω	bob.omega	
x_T	rabbit.pos	
v	rabbit_velocity	
s_C	distance_travelled_by_dog	
V	mag(dog_velocity)	
Phương trình (4)	ball.pos.y += ball.velocity.y * dt	(4')
Phương trình (9)	bob.angle += bob.omega * dt	(9')
Phương trình (11)	rabbit.pos += rabbit_velocity * dt	(11')
Phương trình (13)	distance_travelled_by_dog mag(dog_velocity) * dt	+= (13')

Phụ lục 1 là đoạn mã Vpython mô phỏng hoạt hình của vật rơi tự do phụ lục 2 là của dao động con lắc đơn, và phụ lục 3 là mô phỏng bài toán chó đuổi thỏ. Trong các đoạn mã bao gồm các thành phần như khai báo thông số về điều kiện ban đầu của hệ, dựng các khối hình học, định nghĩa các đại lượng có sử dụng trong các phương trình vi phân và các thành phần điều khiển.

Các đoạn hoạt hình sẽ được phát ra ngay khi chạy trực tiếp chương trình trên trình duyệt web. Trên hình 1 là ảnh chụp màn hình của các đoạn hoạt hình

vật rơi tự do (hình 1a), ảnh chụp dao động của con lắc đơn (hình 1b) và bài toán chó đuổi thỏ (hình 1c). Các đồ thị biểu diễn vị trí cũng được hiển thị tương ứng với trạng của vật. Đối với vật rơi tự do, để mô tả cả quá trình vật đi xuống và đi lên, trong ví dụ này vật được thả không vận tốc đầu và nảy lên khỏi mặt đất. Hệ số va chạm – là tỉ lệ giữa tốc độ lúc sau và tốc độ lúc trước khi va chạm với mặt đất – được lấy là 0,9, tức va chạm này là không đàn hồi. Đối với con lắc đơn, biên độ dao động được lấy là $\pi/16$ (khoảng 11°).



Hình 1: Ảnh chụp từ hình động (a) là vật rơi tự do, (b) là con lắc đơn và (c) là bài toán chó đuổi thỏ cùng với các đồ thị hoặc quỹ đạo vị trí chạy tương ứng với trạng thái của vật trên hình động*

*Hình a), vật rơi từ độ cao 10m với gia tốc rơi tự do $g = 9,81\text{m/s}^2$ và hệ số va chạm $e = 0,9$. Hình b), con lắc đơn có chiều dài 10m, dao động với biên độ góc $\pi/16$.

3.2. Thảo luận

Kết quả của hai đoạn hoạt hình về vật rơi tự do và dao động của con lắc đơn được thể hiện khi chạy chương trình. Vật rơi tự do với vận tốc nhanh dần đều khi đi xuống và chậm dần đều khi đi lên. Quỹ đạo vật rơi được qua thời gian trên hình động cũng được vẽ đồ thị tương ứng với trạng thái chuyển động của vật. Đồ thị này có dạng là các đường parabol

như thể hiện trên hình 1a. Các số liệu tọa độ trên đồ thị này được lấy từ phép tính vi phân để vẽ ra, thể hiện trên dòng lệnh 26 của phụ lục 1. Điều này chứng tỏ hình dạng của đồ thị này phù hợp với công thức lý thuyết của quỹ đạo vật rơi tự do, có dạng $h = gt^2/2$. Ví dụ, theo lý thuyết thì thời gian rơi tự do từ độ cao 10m xuống đất tại nơi có gia tốc trọng

trường $g = 9,81 \text{ m/s}^2$ là $t \approx 1,43\text{s}$, đối chiếu trên đồ thị hình 1a ta thấy phù hợp.

Đối với dao động của con lắc đơn, vật nặng cũng dao động nhanh dần khi về vị trí cân bằng, chậm dần khi về biên. Trên đồ thị cũng chỉ ra tương ứng li độ của vật nặng biến thiên theo quy luật đường sin. Trong ví dụ này, con lắc có độ dài 10m, lấy $g = 9,81\text{m/s}^2$ thì chu kì con lắc theo lí thuyết là $T \approx 6,34\text{s}$. Đối với hiện tượng chó đuổi thỏ thì đoạn hoạt hình cũng cho thấy hướng chạy của chó luôn hướng về vị trí của thỏ, giúp học sinh hiểu được công thức cộng vận tốc trong động học. Kết quả của phương pháp này cho thấy bản chất của hiện tượng vật lí được diễn tả một cách trực quan, hỗ trợ và xoay được ở góc nhìn 3D, do đó có thể dùng để hỗ trợ tư liệu dạy học cũng như trong các ứng dụng có liên quan.

Về đánh giá sai số, như đã đề cập ở mục “Phương pháp”, phép lấy vi phân (không sử dụng tích phân) sẽ cho ra kết quả không chính xác tuyệt đối. Phương pháp này chỉ thuận tiện để tính gần đúng. Sai số của phép tính phụ thuộc vào độ lớn của số gia. Số gia càng nhỏ thì kết quả càng chính xác nhưng phép tính sẽ dài hơn. Thật vậy, đối với trường hợp vật rơi tự do từ trạng thái đứng yên theo công thức lí thuyết thì trong 1 giây đầu vật rơi được 4,905m (lấy $g = 9,81\text{m/s}^2$). Nếu tính bằng phương pháp vi phân theo công thức (4), nếu ta lấy số gia $dt = 0,1\text{s}$ thì quãng đường vật rơi được sau 1s là 5,396m, khi đó sai số là 10% (xem bảng 2); nếu lấy $dt = 0,05\text{s}$ thì quãng đường là 5,15m, sai số giảm còn 5%; $dt = 0,01\text{s}$ thì sai số xấp xỉ 1%. Như vậy, tùy vào yêu cầu của bài toán mà ta có thể chọn số gia phù hợp khi dùng phương pháp vi phân.

Bảng 2: Quãng đường vật rơi được tương ứng số gia thời gian

dt = 0,1		dt = 0,05		dt = 0,01	
t	h	t	h	t	h
0,1	0,098	0,05	0,025	0,01	0,001
0,2	0,294	0,10	0,074	0,02	0,003
0,3	0,589	0,15	0,147	0,03	0,006
0,4	0,981	0,20	0,245	0,04	0,010
0,5	1,472	0,25	0,368	0,05	0,015
0,6	2,060	...	...	...	...
0,7	2,747	0,85	3,752	0,97	4,663
0,8	3,532	0,90	4,194	0,98	4,759
0,9	4,415	0,95	4,660	0,99	4,856
1,0	5,396	1,00	5,150	1,00	4,954

Phương pháp ứng dụng vi phân để dựng hoạt hình có nhiều ưu điểm. Người dùng dễ dàng điều chỉnh các tùy chọn như thay đổi kích thước, hình dạng của

các thành phần hình học; điều chỉnh các thông số vật lí của hệ để điều khiển hoạt hình một cách chủ động theo như ý muốn. Có thể áp dụng phương pháp này

để minh họa nhiều hiện tượng khác như các bài toán va chạm; minh họa định luật bảo toàn động lượng, đặc biệt là trường hợp vật có khối lượng thay đổi liên tục theo thời gian như chuyển động của hỏa tiễn; mô phỏng hoạt hình cho các bài toán cộng vận tốc có hướng luôn thay đổi; chuyển động của piston và trục khuỷu trong động cơ.

4. Kết luận

Bài báo đã khai thác được ứng dụng của vi phân và VPython để tạo hoạt hình, sau đó áp dụng cho ba ví dụ minh họa. Phương pháp này có ưu điểm là người dùng chủ động điều chỉnh được các thông số của hình động để phù hợp với bản chất của hiện tượng. Các thông số chuyển động cũng có thể được truy suất và hiển thị trên đồ thị một cách đồng bộ với hình động và được chạy trực tiếp trên

trình duyệt web máy tính hoặc thiết bị di động mà không cần phải cài đặt phần mềm. Hơn nữa, phương pháp cũng cho kết quả hoạt hình 3D có thể xoay được, và cũng tích hợp một số môn học khác nhau như để phục vụ cho dạy học, phù hợp với xu thế hiện nay. Tuy nhiên, khi sử dụng phương pháp ta nên chú ý về sai số gây ra bởi bản chất của phép lấy vi phân. Các đại lượng cần khảo sát hoặc đối tượng cần mô phỏng chuyển động phải định lượng được hoặc biểu diễn được bằng hàm số. Phương pháp này phù hợp để mô phỏng các chuyển động hoặc khảo sát các đại lượng biến đổi liên tục phức tạp hơn. Bài báo hy vọng đóng góp phần nào trong công cuộc đổi mới phương pháp và ứng dụng công nghệ trong dạy học cũng như ứng dụng cho các lĩnh vực khác.

TÀI LIỆU THAM KHẢO

- [1] M. Bourne, “Applications of Differentiation,” 27 Feb 2025. [Online]. Available: <https://www.intmath.com/applications-differentiation/applications-of-differentiation-intro.php>.
- [2] S. V. Samsonau, “Computer simulations combined with experiments for a calculus-based physics laboratory course,” *Physics Education*, vol. 5, no. 53, 2018.
- [3] M. Fowler, “Using Excel to Simulate Pendulum Motion and Maybe Understand Calculus a Little Better,” *Sci Educ*, no. 13, p. 791–796, 2004.
- [4] F. O. Ochieng, “Applications of Differential Calculus in real life,” 16 Mar 2023. [Online]. Available: <https://www.slideshare.net/slideshow/applications-of-differential-calculus-in-real-life/256532677>.
- [5] Z. K. Silagadze and O. I. Chaschina, “The dog-and-rabbit chase problem as an exercise in introductory kinematics,” *Latin-American Journal of Physics Education*, vol. 3, no. 3, pp. 111-115, 2009.
- [6] Vpython, “Vpython,” 2025. [Online]. Available: <https://vpython.org/>. [Accessed 27 Feb 2025].

Phụ lục

```

1 Web VPython 3.2
2 #VẬT RƠI TỰ DO VÀ VA CHẠM VỚI MẶT ĐẤT
3 ball_radius = 0.5
4 initial_height = 10
5 g = 9.81
6 e = 0.9
7 t = 0
8 scene = canvas(title='RƠI TỰ DO VÀ VA CHẠM VỚI MẶT ĐẤT',
9               width=600, height=400, center=vector(0, initial_height / 2, 0),
10              background=color.white)
11 floor = box(pos=vector(0, 0, 0), size=vector(4, 0.5, 4),
12            color=color.white)
13 ball = sphere(pos=vector(0, initial_height, 0), radius=ball_radius,
14              color=color.red, make_trail=True)
15
16 ball.velocity = vector(0, 0, 0)
17 dt = 0.01
18 |
19 graph1 = graph(title='RƠI TỰ DO', xtitle='t(s)', ytitle='h(m)')
20 curve = gcurve(graph=graph1, color=color.blue)
21
22 Position_m = gcurve(color=color.red, label="Độ cao")
23
24 while True:
25     rate(50)
26     ball.pos.y += ball.velocity.y * dt
27     ball.velocity.y -= g * dt
28     Position_m.plot(t, ball.pos.y)
29
30     if ball.pos.y < ball_radius and ball.velocity.y < 0:
31         ball.velocity.y = -ball.velocity.y * e
32     if ball.velocity.y < 1e-2 and ball.pos.y < ball_radius:
33         ball.pos.y = initial_height
34     t += dt

```

Phụ lục1: Đoạn mã mô phỏng hoạt hình của vật rơi tự do

```

1 Web VPython 3.2
2 #SỬ DỤNG VI PHÂN ĐỂ DỰNG DAO ĐỘNG CỦA CON LẮC ĐƠN
3 length = 10
4 g = 9.81
5 theta0 = pi/16
6 #e = 0.3
7 t = 0
8 scene = canvas(title='CON LẮC ĐƠN', width=600, height=400,
9               background=color.white)
9 pivot = vector(0, 8, 0)
10 bob = sphere(pos=vector(length * sin(theta0),
11                       -length * cos(theta0), 0), radius=0.5,
12             color=color.red, make_trail=True)
13 rod = cylinder(pos=pivot, axis=bob.pos - pivot, radius=0.05)
14
15 bob.velocity = vector(0, 0, 0)
16 bob.angle = theta0
17 bob.omega = 0
18 dt = 0.05
19 |
20 graph1 = graph(title='CON LẮC ĐƠN', xtitle='t(s)', ytitle='S(m)')
21 displace_m = gcurve(color=color.red, label="Li độ")
22
23 while t<50:
24     rate(50)
25     alpha = -g / length * sin(bob.angle)
26     bob.omega += alpha * dt
27     bob.angle += bob.omega * dt
28     displace_m.plot(t, sin(bob.angle))
29
30     bob.pos = vector(length*sin(bob.angle), -length*cos(bob.angle), 0)
31     rod.axis = bob.pos - pivot
32     t += dt

```

Phụ lục 2: Đoạn mã mô phỏng hoạt hình dao động của con lắc đơn


```

1 Web VPython 3.2
2 from vpython import *
3
4 scene = canvas(width=600, height=400, center=vector(0, 0, 0),
5               background=color.white)
6
7 dog = sphere(pos=vector(-5, -5, 0), radius=0.4,
8             color=color.red, make_trail=True)
9 rabbit = sphere(pos=vector(-5, 8, 0), radius=0.2,
10            color=color.green, make_trail=True)
11 dog_velocity = vector(0, 1.2, 0)
12 rabbit_velocity = vector(0.5, 0, 0)
13
14 dt = 0.01
15 t = 0
16 distance_travelled_by_dog = 0
17 |
18 while mag(rabbit.pos - dog.pos) > 0.1:
19     rate(100)
20
21     distance = mag(rabbit.pos - dog.pos)
22     direction = norm(rabbit.pos - dog.pos)
23
24     dog_velocity = direction * mag(dog_velocity)
25     dog.pos += dog_velocity * dt
26     rabbit.pos += rabbit_velocity * dt
27
28     t += dt
29     distance_travelled_by_dog += mag(dog_velocity) * dt
30
31 # Print the results
32 print(f"Thời gian: {round(t,2)} s")
33 print(f"Quãng đường của chó: {round(distance_travelled_by_dog,2)} m")
34 print(f"Quãng đường của thỏ: {round(rabbit.pos.x+5,2)} m")
35

```

Phụ lục 3: Đoạn mã mô phỏng hoạt hình bài toán chó đuổi thỏ

RESEARCH ON CREATING NON-UNIFORM MOTION ANIMATION USING DIFFERENTIAL CALCULUS AND PYTHON PROGRAMMING

Tran Xuan Hoi

Phu Yen University

Email: tranxuanhoi@pyu.edu.vn

(Received: 4/3/2025, Revised: 24/4/2025, Accepted for publication: 22/5/2025)

ABSTRACT

This study presents a method of applying differential calculus in mathematics combined with the Python programming language to create animations of non-uniform motion. The author then applied the method to three illustrative examples, including a free-falling object bouncing off the ground, the oscillation of a simple pendulum, and Velocity addition simulating dog chasing rabbit problem. The results show that using this method has many advantages, such as being able to perform on both computers, and mobile devices, and the parameters of the animation can be adjusted as desired. Furthermore, the method demonstrates an application of differential calculus in animation construction, as well as integrating a number of different subjects This method can also serve as a useful tool for teaching and research in the fields of natural sciences and technology today.

Keywords: *Differential calculus, simulation, animation, Vpython*