MỘT SỐ GIẢI PHÁP PHÁT TRIỂN TƯ DUY MÁY TÍNH CHO HỌC SINH THÔNG QUA DẠY HỌC STEM TÍCH HỢP TIN HỌC - VẬT LÍ

Nguyễn Thị Anh Thi, Nguyễn Thị Ý Dung
Khoa Sư phạm, Trường Đại học Quy Nhơn

Tóm tắt: *Chương trình Giáo dục phổ thông 2018 đặt trọng tâm vào việc định hình và phát triển năng lực tư duy máy tính cho học sinh thông qua giáo dục STEM. Tuy nhiên, thực tiễn dạy học lập trình tích hợp khoa học tự nhiên đang đối mặt với rào cản lớn do sự chênh lệch về trình độ công nghệ của học sinh và tính trừu tượng của kiến thức. Nghiên cứu này đề xuất một số giải pháp phát triển tư duy máy tính cho học sinh thông qua dạy học STEM tích hợp Tin học - Vật lí, cụ thể thông qua chủ đề mô phỏng "Chuyển động thẳng biến đổi đều". Thay vì áp dụng lối gõ mã lệnh truyền thống, nghiên cứu sử dụng ngôn ngữ Python kết hợp chiến thuật "mã nguồn có lỗi" (Buggy Code) để rèn luyện kỹ năng kiểm thử và gỡ lỗi thuật toán, và sử dụng thư viện Matplotlib của Python để trực quan hóa dữ liệu. Kết quả thực nghiệm tại Trường THPT Quốc Học Quy Nhơn minh chứng hệ thống các giải pháp này đã giải quyết bài toán phân hóa năng lực, giúp học sinh thấu hiểu kiến thức Vật lí và hình thành rõ nét 4 thành tố cốt lõi của tư duy máy tính (phân rã, nhận dạng mẫu, trừu tượng hóa và thiết kế thuật toán).*

Từ khóa: *Tư duy máy tính, giáo dục STEM, Buggy Code, trực quan hóa dữ liệu, Python.*

SOME SOLUTIONS FOR DEVELOPING COMPUTATIONAL THINKING IN STUDENTS THROUGH INTEGRATED STEM INSTRUCTION (INFORMATICS AND PHYSICS)

Abstract: *The 2018 General Education Program places a strong emphasis on shaping and developing students' computational thinking capabilities through STEM education. However, the practical teaching of programming integrated with natural sciences is facing major barriers due to the disparity in students' technological proficiency and the abstract nature of the knowledge. This study proposes several solutions to develop students' computational thinking through STEM education integrating Informatics and Physics, specifically through the simulation topic of "Uniformly accelerated rectilinear motion". Instead of applying the traditional coding approach, the research utilizes the Python language combined with the "Buggy Code" strategy to train algorithm testing and debugging skills, and uses Python's Matplotlib library for data visualization. Experimental results at Quoc Hoc Quy Nhon High School prove that this system of solutions has solved the capability differentiation problem, helping students deeply understand Physics knowledge and clearly form the four core components of computational thinking (decomposition, pattern recognition, abstraction, and algorithm design).*

Keywords: *Computational thinking, STEM education, Buggy Code, data visualization, Python.*

Nhận bài: 07/06/2026 *Phản biện: 07/07/2026* *Duyệt đăng: 10/07/2026*

I. ĐẶT VẤN ĐỀ

Trong bối cảnh kỷ nguyên số, Tư duy máy tính (Computational Thinking - CT) đã vươn tầm khỏi ranh giới của ngành Khoa học máy tính để trở thành một năng lực nhận thức nền tảng của mọi công dân. Nắm bắt xu thế tất yếu này, Chương trình Giáo dục phổ thông (GDPT) 2018 đã xác định việc định hình và phát triển năng lực tư duy máy tính là hạt nhân cốt lõi của môn Tin học [1], đồng thời xem giáo dục STEM là không gian thực tiễn lý tưởng nhất để học sinh (HS) rèn luyện năng lực này thông qua việc giải quyết các bài toán liên môn.

Tuy nhiên, việc đưa lập trình vào các môn khoa học tự nhiên như Vật lí đang vấp phải những rào cản sư phạm không nhỏ. Về khía cạnh chuyên môn, kiến thức thuộc chương "Động học" (Vật lí 10) mang tính trừu tượng cao, khiến HS gặp khó khăn nếu chỉ tiếp cận qua các công thức tính toán tĩnh trên giấy. Về khía cạnh tổ chức dạy học, thực tiễn cho thấy 100% giáo viên thừa nhận sự chênh lệch lớn về trình độ công nghệ và kỹ năng lập trình của HS trong cùng một lớp học là một thách thức khổng lồ. Nếu áp đặt HS phải tự thiết kế một chương trình mã nguồn từ con số không, phương pháp này dễ gây ra sự quá tải và làm mất đi bản chất của giáo dục STEM.

Để giải quyết bài toán trên, nghiên cứu này đề xuất một số giải pháp phát triển tư duy máy tính cho học sinh thông qua dạy học STEM tích hợp Tin học - Vật lí. Cụ thể, bài viết ứng dụng ngôn ngữ Python với chiến thuật "Mã nguồn có lỗi" (Buggy Code) và công cụ trực quan hóa dữ liệu Matplotlib để thiết kế chủ đề STEM. Mục tiêu của bài viết là chứng minh cách thức các giải pháp sư phạm này tạo ra giúp phân hóa năng lực người học, qua đó thúc đẩy mạnh mẽ sự phát triển các thành tố của tư duy máy tính (phân rã, trừu tượng hóa, thuật toán) cho học sinh THPT.

II. NỘI DUNG NGHIÊN CỨU

2.1. Cơ sở lý luận về năng lực tư duy máy tính và vai trò của các chiến thuật sư phạm công nghệ

Bản chất của Tư duy máy tính (TDMT) trong giáo dục STEM: Việc tư duy máy tính không đơn thuần là kỹ năng gõ mã lệnh (coding), mà là một quá trình nhận thức và lập luận logic nhằm định hình vấn đề sao cho các giải pháp có thể được thực thi tự động bởi máy tính. Cấu trúc của TDMT được hình thành từ 4 thành tố cốt lõi: Phân rã vấn đề (Decomposition), Nhận dạng mẫu (Pattern Recognition), Trừu tượng hóa (Abstraction) và Thiết kế thuật toán (Algorithm Design) [5]. Trong dạy học STEM, TDMT đóng vai trò là "động lực thúc đẩy", cho phép HS mô hình hóa các hiện tượng khoa học phức tạp, chuyển đổi các em từ trạng thái "người tiêu dùng công nghệ" sang "người sáng tạo công nghệ".

Chiến thuật "Mã nguồn có lỗi" (Buggy Code) và tư duy kiểm thử: Trong khoa học máy tính, gỡ lỗi (Debugging) và kiểm thử là những hoạt động bộc lộ rõ nhất tư duy phản biện và khả năng truy vết thuật toán. Đối diện với thực trạng HS yếu dễ bỏ cuộc khi lập trình, chiến thuật Buggy Code ra đời như một giải pháp phân hóa xuất sắc. Thay vì yêu cầu viết code từ đầu, giáo viên cung cấp một đoạn mã Python gần hoàn chỉnh nhưng cố tình chứa các lỗi logic (sai tên biến, sai công thức vật lí, lỗi thụt lề). Hoạt động này buộc HS phải hóa thân thành các "kỹ sư kiểm thử", dùng tư duy máy tính để phân rã từng dòng lệnh, đối chiếu với quy luật vật lí để "bắt bệnh" và sửa lỗi. Khung mã có sẵn giúp HS yếu có điểm tựa, trong khi quá trình gỡ lỗi lại tạo ra thách thức kích thích tư duy bậc cao ở HS khá giỏi.

Trực quan hóa dữ liệu (Data Visualization) như một công cụ nhận dạng mẫu: Sự kết nối giữa Khoa học và Công nghệ chỉ đạt yêu cầu khi những dữ liệu tính toán khô khan được biểu diễn trực quan. Bằng việc tích hợp các thư viện mạnh mẽ như Matplotlib của Python, HS có thể lập trình để chuyển đổi hàng ngàn con số tính toán thành đồ thị. Khả năng phản hồi tức thời của phần mềm khi thay đổi các tham số vật lí giúp HS dễ dàng đối chiếu đồ thị hiển thị với lý thuyết động học, từ đó củng cố khả năng trừu tượng hóa và nhận dạng mẫu quy luật tự nhiên.

2.2. Mô hình hóa các định luật động học bằng Python

Chủ đề "Mô phỏng chuyển động thẳng biến đổi đều" là môi trường lý tưởng để HS luyện tập tư duy máy tính thông qua việc chuyển ngữ từ Khoa học Tự nhiên sang Khoa học Máy tính. Quá trình mô hình hóa này đòi hỏi sự vận dụng toàn diện 4 thành tố của tư duy máy tính:

✓ Trừu tượng hóa (Abstraction) kiến thức Vật lí: Trong Vật lí 10, chuyển động nhanh dần đều được chi phối bởi gia tốc hằng số (a), vận tốc v=v0+at, và độ dịch chuyển d=v0t+(1/2)at2. HS phải trừu tượng hóa các đại lượng này thành các Biến (Variables) trong không gian bộ nhớ của Python, loại bỏ các yếu tố vật lí không cần thiết (như lực cản, khối lượng xe).

✓ Phân rã vấn đề (Decomposition): Thế giới vật lí là một dòng thời gian liên tục. Để máy tính có thể xử lý, HS buộc phải dùng tư duy phân rã, băm nhỏ quá trình chuyển động thành các khoảnh khắc rời rạc (t=0,1,2...). Cấu trúc vòng lặp for...in range() trong Python chính là công cụ hoàn hảo để hiện thực hóa sự phân rã này, buộc máy tính lặp lại phép tính tọa độ tại từng giây.

✓ Thiết kế thuật toán và Nhận dạng mẫu (Algorithm & Pattern): Dữ liệu sinh ra từ vòng lặp phải được thu thập có hệ thống. HS thiết kế thuật toán sử dụng cấu trúc Danh sách (List) kết hợp lệnh .append() để lưu trữ chuỗi dữ liệu tọa độ. Cuối cùng, để nhận dạng quy luật động học, HS gọi các hàm plt.plot() và plt.show() của Matplotlib. Máy tính sẽ nội suy mảng dữ liệu này lên hệ trục tọa độ, vẽ ra một đường cong Parabol hoàn hảo. Việc tự tay viết ra thuật toán biến đổi công thức vô hình thành biểu đồ hữu hình là minh chứng rõ nhất cho sự phát triển về năng lực tư duy máy tính của người học.

2.3. Đề xuất giải pháp sư phạm phát triển tư duy máy tính qua chủ đề mô phỏng

Để phát triển tối đa các thành tố của tư duy máy tính (phân rã, nhận dạng mẫu, trừu tượng hóa và thiết kế thuật toán), đồng thời khắc phục rào cản về sự chênh lệch năng lực công nghệ của học sinh, nghiên cứu đề xuất 3 giải pháp thiết kế sư phạm trọng tâm:

Giải pháp 1: Ứng dụng chiến thuật "Mã nguồn có lỗi" (Buggy Code) để rèn luyện tư duy thuật toán và kỹ năng gỡ lỗi. Nhằm giải quyết khó khăn khi 100% học sinh có trình độ lập trình không đồng đều, giáo viên không yêu cầu học sinh viết mã lệnh từ đầu. Thay vào đó, giáo viên cung cấp

một đoạn mã Python bị khuyết công thức vật lí hoặc chứa các lỗi logic (ví dụ: mâu thuẫn biến t = [] và lệnh time.append()). Hoạt động gỡ lỗi (Debugging) này tạo ra một "giàn giáo nhận thức": giúp học sinh trung bình có điểm tựa để đọc hiểu luồng dữ liệu, đồng thời buộc học sinh khá giỏi phải vận dụng tư duy máy tính để phân rã từng dòng lệnh, đối chiếu với quy luật vật lí để "bắt bệnh" và sửa lỗi. Quá trình kiểm thử này giúp học sinh hình thành thói quen đánh giá chương trình và tối ưu hóa thuật toán - một mục tiêu cốt lõi của tư duy điện toán.

Giải pháp 2: Trực quan hóa dữ liệu động học bằng Matplotlib để củng cố năng lực trừu tượng hóa. Để phá vỡ tính trừu tượng của các công thức Động học, học sinh được yêu cầu thiết kế thuật toán gọi các hàm của thư viện Matplotlib (plt. plot(), plt.show()). Máy tính sẽ đảm nhận việc nội suy mảng dữ liệu tính toán khô khan thành một đồ thị Parabol trực quan. Sự thay đổi tức thời của hình dáng đồ thị khi học sinh thay đổi tham số (như gia tốc a âm hoặc dương) giúp các em ngay lập tức nhận dạng được "mẫu" (Pattern) của chuyển động, củng cố kỹ năng đối sánh giữa mô hình toán học và hiện tượng vật lí thực tế.

Giải pháp 3: Tổ chức không gian học tập công nghệ hỗ trợ bằng Rubric đa chiều và lớp học đảo ngược. Thay vì phụ thuộc hoàn toàn vào bài kiểm tra lý thuyết, giáo viên sử dụng Bộ tiêu chí đánh giá (Rubric) đo lường trực tiếp các thành tố của tư duy máy tính: Kỹ thuật lập trình Python, Tính đúng đắn của thuật toán Vật lí, Năng lực trực quan hóa và Kỹ năng gỡ lỗi (E). Song song đó, việc giao tài liệu cú pháp lệnh Python cho học sinh nghiên cứu trước tại nhà (Lớp học đảo ngược) giúp giải phóng quỹ thời gian trên lớp, tạo không gian tối đa cho học sinh thực hành lập trình nhóm, tranh luận kỹ thuật và tổ chức "Triển lãm công nghệ" phản biện chéo.

III. KẾT LUẬN

Nghiên cứu đã khẳng định việc áp dụng một số giải pháp phát triển tư duy máy tính cho học sinh thông qua dạy học STEM tích hợp Tin học - Vật lí là một hướng đi sư phạm đột phá, hiện thực hóa xuất sắc mục tiêu phát triển năng lực tư duy máy tính của Chương trình Giáo dục phổ thông 2018. Thông qua bài toán mô hình hóa kiến thức Động học, hệ thống 3 giải pháp sư phạm trọng tâm (ứng dụng Buggy Code, trực quan hóa bằng Matplotlib và sử dụng Rubric đa chiều kết hợp lớp học đảo ngược) đã khắc phục hoàn toàn những hạn chế lớn nhất hiện nay: quỹ thời gian thiết kế bài giảng của giáo viên và sự chênh lệch năng lực công nghệ của học sinh trong cùng một lớp học.

Kết quả thực tiễn chỉ ra rằng, khi được làm "kỹ sư kiểm thử" thông qua hệ thống mã nguồn có lỗi, học sinh không chỉ vượt qua rào cản sợ lập trình mà còn phát triển sâu sắc các thành tố của tư duy máy tính: từ phân rã vấn đề, trừu tượng hóa đến khả năng lập luận, gỡ lỗi và tối ưu hóa thuật toán. Các giải pháp được đề xuất trong bài viết không chỉ mang lại hiệu quả trực tiếp cho chủ đề Vật lí 10, mà còn cung cấp một khung lý luận và thực hành vững chắc để giáo viên tiếp tục mở rộng, ứng dụng sức mạnh của tư duy máy tính vào các môn khoa học khác trong tương lai.

TÀI LIỆU THAM KHẢO

Bộ Giáo dục và Đào tạo, Chương trình giáo dục phổ thông: Chương trình tổng thể. Hà Nội: Bộ Giáo dục và Đào tạo, 2018. Ban hành kèm theo Thông tư số 32/2018/TT-BGDĐT ngày 26 tháng 12 năm 2018 của Bộ trưởng Bộ Giáo dục và Đào tạo.

T. H. Nguyen (2020), Giáo dục STEM/STEAM: từ tầm nhìn chiến lược đến thực tiễn triển khai trong hệ thống giáo dục phổ thông tại Hoa Kỳ

L. H. Hoàng, "Giáo dục stem trong chương trình giáo dục phổ thông 2018: Định hướng và tổ chức thực hiện," Tạp chí Giáo dục, no. 516, pp. 1–6, 2021. Kì 2 - 12/2021

Lê Thanh Hà và Phan Thị Thanh Hội (2021), "Lược sử nghiên cứu giáo dục STEM ở một số nước trên Thế giới và Việt Nam", Tạp chí Khoa học - Trường Đại học Sư phạm Hà Nội: Khoa học Giáo dục, tập 2, số 66, trang 220–220.

Nguyễn Thị Anh Thi (2023), "Dạy học kĩ thuật lập trình cho sinh viên ngành kĩ thuật điện tử - viễn thông theo hướng phát triển tư duy điện toán", Luận án Tiến sĩ Khoa học giáo dục, Trường Đại học sư phạm Hà Nội.

J. VanderPlas (2016), "Python Data Science Handbook.: Essential Tools for Working with Data", O'Reilly Media.