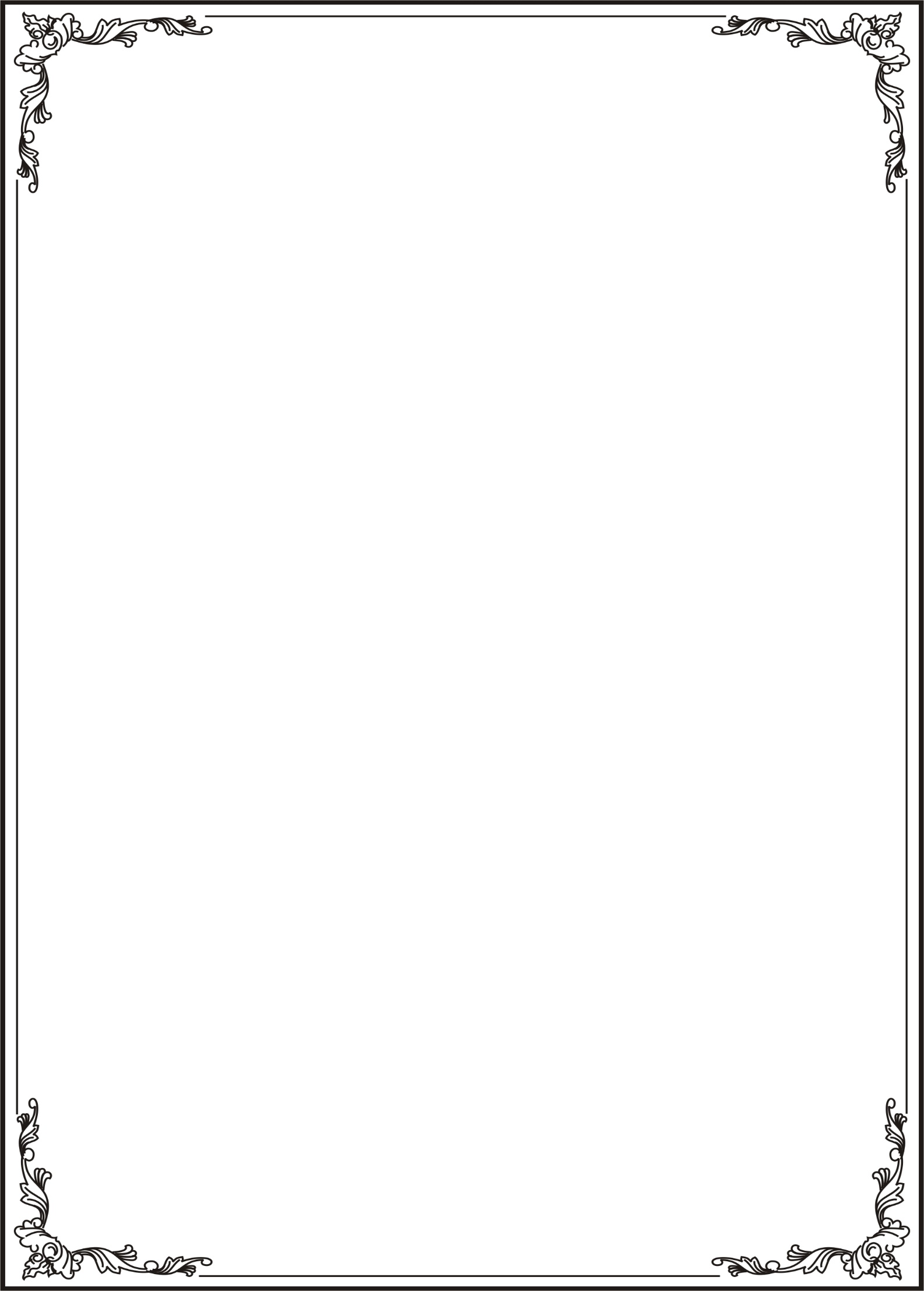
****

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP. HCM**

**KHOA VẬT LÝ – VẬT LÝ KỸ THUẬT**

**BỘ MÔN VẬT LÝ TIN HỌC**



-----🙞🙜🕮🙞🙜-----

**BÁO CÁO KHÓA LUẬN**

*Tham gia thực hiện:*

HUỲNH THỊ HẠ VY 1513229

*Ngày 8 tháng 3 năm 2019*

**Thuật toán mới cho Adaptive Multidimensional Integration**

(*Tạm dịch*: Thuật toán mới cho tích phân đa chiều thích nghi)

Mục đích chung của thuật toán mới được mô tả cho tích phân nhiều chiều. Thuật toán mới này đã được so sanh với nhiều thuật toán khác hiện đang được sử dụng. Và nó cho thấy được kết quả tốt hơn khi dùng để tính tích phân nhiều chiều (n >= 4).

1. **Giới thiệu**

Một trong những vấn đề quan trong mà các nhà Vật lý phải đối mặt đó là tính toán các tích phân nhiều chiều và đôi khi **poorly behaved integrands**. Trong lý thuyết hạt cơ bản, tích phân lớn hơn bốn chiều được sử dụng thường xuyên để tính toán biên độ tán xạ theo lý thuyết nhiễu loạn của Feynman. Lưu ý, thuật toán mới này được dùng để tính tích phân nhiều chiều. Thuật toán kết hợp sự lặp và sơ đồ Monte Carlo thích nghi.

Các đặc điểm của thuật toán:

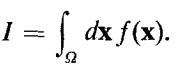
1. Phương sai chính xác cho tính phân thì được tính toán dễ dàng.
2. Hàm tính tích phân **không cần liên tục** khi sử dụng thuật toán mới này. Đặc biệt, các hàm bước nhảy (step function) cũng không có vấn đề gì. Vì thế để tính tích phân trên miền siêu thể tích (hypervolume) là việc đơn giản.
3. Tốc độ hội tụ không phụ thuộc vào số chiều của tích phân.
4. Đây là thuật toán thích nghi. Nó tự động tập trung đánh giá tích phân tại những vùng có tích phân quan trọng nhất.

Đặc điểm (a) và (c) là phổ biến trong tất cả phương pháp Monte Carlo. **Với (d) là một tính năng quan trọng nhất trong thuật toán này**. Vấn đề chính trong tính phân nhiều chiều là việc tăng số lượng mẫu theo cấp số nhân khi tăng số chiều. Do đó, mục đích chung của thuật toán để tích phân nhiều chiều là thích nghi (adaptive).

Trong phần 2 chúng tôi tóm tắt tính năng chung của tích phân Monte Carlo và công việc trước đây khi sử dụng phương pháp này. Phần 3 chúng tôi mô tả thuật toán mới dựa trên một biến. Phần 4 nói về cách sử dụng phương pháp này trong không gian n chiều. Sau cùng, phần 5 chúng tôi sẽ so sánh hiệu suất của phương pháp này so với các phương pháp khác hiện nay. Phương pháp này đã chứng minh rằng nó hiệu quả hơn rất nhiều khi sử dụng tính tích phân nhiều chiều (n >= 4).

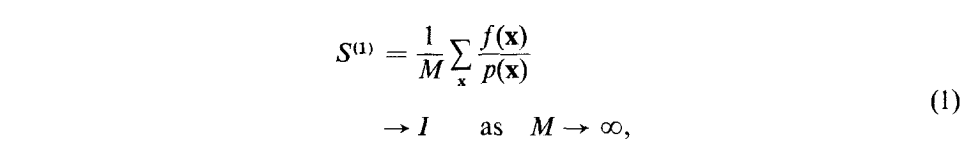
1. Tích phân Monte Carlo

Xét tích phân của một hàm n biến **x** (x1, …, xn) trên không gian mẫu ‟Ω”.

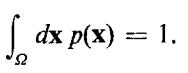


I =

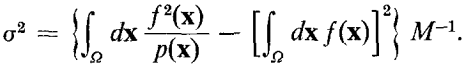
Nếu các điểm M (**x**) được chọn ngẫu nhiên từ một phân bố các điểm trong không gian mẫu Ω với xác suất p(**x**). Dễ dàng thấy được tích phân xấp xĩ bằng:



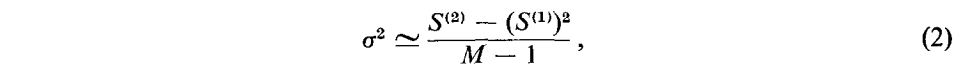
Trong đó hàm mật độ xác suất được chuẩn hóa thành:



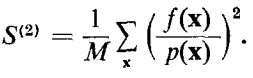
Đại lượng S(1) được kì vọng sẽ dao động về giá trị đúng của tích phân. Tương ứng với mỗi tập hợp M điểm ngẫu nhiên là 1 đại lượng S(1) khác nhau. Phương sai của dao động này là:



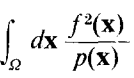
Đối với M lớn thì phương sai sẽ là:



Tại



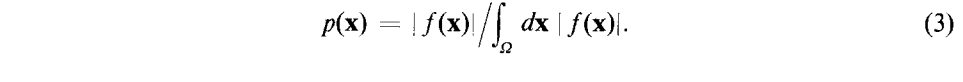
Độ lệch chuẩn σ chỉ ra kết quả chính xác của S(1) như là một kết quả gần đúng của I. Lưu ý rằng ước tính của phương sai chính xác khi tích phân:



hữu hạn nhưng nếu tích phân không hữu hạn thì S(1) vẫn xấp xĩ giá trị trung bình của I.

Có một số kỹ thuật để giảm phương sai σ(2) với M không đổi. Hai phương pháp phổ biến nhất được sử dụng trong tích phân nhiều chiều là:

1. *Lấy mẫu quan trọng.* Ở đây **mật độ p(x) được thay đổi để giảm σ(2).** Như đã biết, phương sai được tối ưu khi



/

Do đó**, khi sử dụng gieo điểm trọng tích phân sẽ được tính tập trung tại nơi tích phân có độ lớn lớn nhất** (dù nó có flat ở đó hay không).

1. *Lấy mẫu phân tầng.* Để giảm phương sai, thể tích tích phân có thể được chia thành N khối thể tích nhỏ hơn với các kích thước khác nhau. Sau đó tích phân Monte-carlo được thực hiên trong mỗi khối thể tích nhỏ. Với mỗi khối thể tích nhỏ này sử dụng M/N điểm ngẫu nhiên. Khi kích thướt của khối thể tích nhỏ bị thay đổi thì phương sai sẽ được tính toán lại và được tối ưu khi phương sai của mỗi khối thể tích nhỏ này là như nhau (σ(2)/N). Do đó, khi sử dụng gieo điểm phân tầng (stratified sampling), tích phân sẽ được tập trung tại nơi có sai số lớn nhất, tức là nơi tích phân vừa lớn và vừa dao động mạnh.

Các phương pháp giảm phương sai không phù hợp trong trường hợp tổng quát vì nó đòi hỏi phải có thông tin chi tiết về cách hành xử của hàm dưới dấu tích phân trước khi xử lý. Tuy nhiên Sheppeyl(1) đã nghĩ ra thuật toán lặp, sử dụng thông tin được tạo ra từ lần tính tích phân Monte Carlo trước để tính cho lần sau nhầm để giảm phương sai trong các lần tính tích phân tiếp theo. Ông sử dụng phương pháp lấy mẫu phân tầng. Ban đầu thuật toán chia siêu khối (hypercube) của tích phân n chiều thành Nn siêu khối nhỏ như nhau. Bằng cách sử dụng một lưới hình chữ nhật đồng nhất. Mỗi siêu khối nhỏ (hypercube) sẽ có hai điểm để đóng góp vào để tính tích phân Monte Carlo và phương sai. Các phương sai từ các siêu khối nhỏ sau đó được sử dụng để định nghĩa lại không gian lưới mới dọc theo mỗi trục. Và Không gian lưới mới này sẽ được sử dụng cho lần lặp tiếp theo. Và phải giữ cho tổng số siêu khối nhỏ không đổi. Như vậy qua một số vòng lặp thì các siêu khối nhỏ có thể tập trung dần dần về nơi có phương sai ban đầu lớn nhất và phương sai được giảm.

Thuật toán này được sử dụng rộng rãi trong ngành Vật lý Lý thuyết và rất thành công cho nhiều ứng dụng tính toán hai chiều trở lên. Sức mạnh lớn nhất của phương pháp là khả năng thích nghi của tích phân đang được xem xét. Tuy nhiên khả năng thích nghi của nó được quyết định bởi số lượng lưới dọc theo mỗi trục (N). Và N bị giới hạn bởi tổng số tích phân (M) được cho phép ở mỗi vòng lặp.



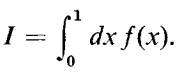
M = 2Nn

Hạn chế này cho thấy một khuyết điểm lớn khi có số chiều lớn ( M < 106 khi n = 9 và N <= 4), như sẽ được minh họa trong phần 5.

1. **Thuật toán trong một chiều**

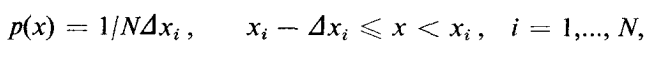
Hạn chế từ phương pháp của Sheppey có thể tránh được bằng cách sử dụng phương pháp lấy mẫu quan trọng thay vì phương pháp lấy mẫu phân tầng để giảm phương sai. Phương pháp **lấy mẫu quan trọng** dường như cho thấy kém hiệu quả hơn phương pháp lấy mẫu phân tầng. Tuy nhiên trong thực tế khả năng thích nghi **vượt trội** hơn khi được thể hiện ở **bài toàn có số chiều lớn**.

Giống như Sheppey, thuật toán được mô tả ở đây là sự lặp đi lặp lại. Để minh họa hãy xem tích phân một chiều:

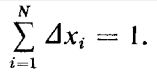


Ban đầu, khởi tạo M điểm với mật độ xác suất như nhau (p(x) = 1). Sau đó tính toán ước tính của tích phân và phương sai. Phương trình (1) và (2), điểm M dùng để tính tích phân cũng có thể được sử dụng để định nghĩa lại mật độ xác suất tốt hơn cho vòng lặp tiếp theo. Trong phương pháp này phương sai sẽ được giảm dần qua các vòng lặp.

Có một số kỹ thuật để tạo ra dãy số giả ngẫu nhiên với phân bố đều. Sẽ khó khăn hơn khi muốn tạo ra dãy số ngẫu nhiên với một phân bố bất kỳ có mật độ p(x).Vậy p(x) là một hàm bước nhảy với N bước. Xác suất của một số ngẫu nhiên được chọn từ một bước nhảy bất kỳ thì được xác định là một hằng số 1/N cho tất cả các bước (0 = x0 < … XN =1, Δxi = xi – xi-1):

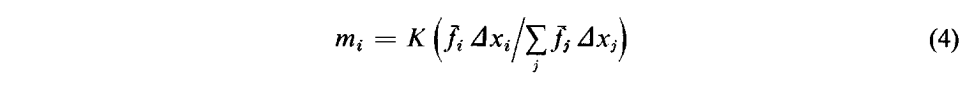


Tại

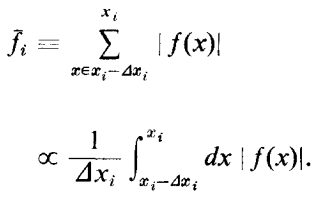


Phân phối xác suất được điều chỉnh theo các tích phân từng phần cụ thể bằng cách thay đổi kích thước của các số gia Δxi. Trên thực tế thì N bị giới hạn bởi bộ nhớ máy tính và thường thì N là một hằng số (N = 50 đến 100).

Cho M điểm tính tích phân, phân bố xác suất. Mật độ số gia được điều chỉnh bằng cách chia nhỏ mỗi số gia thành mi + 1 số gia nhỏ hơn, tại

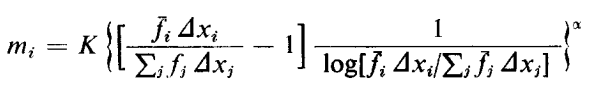


Và



Mỗi đoạn sẽ bị chia nhỏ thành k+1 đoạn con (thường thì k = 1000). Sự đóng góp của mỗi đoạn cho hàm trọng sẽ tăng tỉ lệ với sự đóng góp của nó cho tích phân , như yêu cầu trong phương trình (3). Do cần tổng số đoạn vẫn giữ nguyên là N, các đoạn con sẽ được dom lại thành các đoạn mới sao cho số đoạn con là như nhau và không đổi (để giữ mật độ tỉ đối giữa các đoạn không đổi giữa các đoạn). Hệ quả là ta phải thay đổi kích thước của các đoạn nhưng giữ tổng số đoạn không đổi sao cho đoạn có kích thước nhỏ nhất ở nơi lớn nhất. Lưới mới này sẽ được dùng và sẽ được tinh chỉnh trong mỗi lần lặp cho tới khi ta có được lưới tối ưu (tức ).

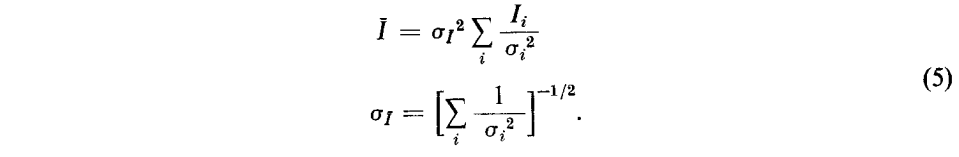
Trong thực tế, cách tốt nhất để làm giảm sự phân chia của thuật toán một cách nhanh chóng và mất ổn định khi thay đổi lưới từ vòng lặp này sang vòng lặp khác, ta sử dụng [5]



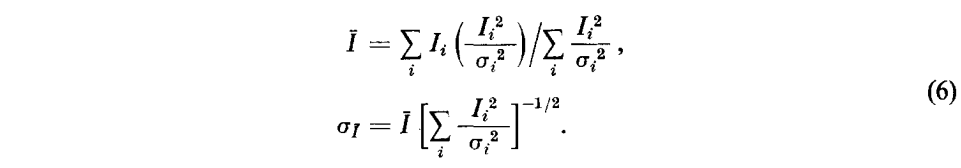
Thay cho (4). Tham số α xác định tốc độ hội tụ và thường nằm trong khoảng (1,2).

Các giá trị phải được xóa bỏ sau mỗi vòng lặp bởi vì bộ nhớ máy tính có giới hạn.

Tuy nhiên, các ước tính và sai số của tích phân có thể sử dụng cho tất cả các đánh giá của tích phân.



Ii và σi2 là tích phân và độ lệch chuẩn trong vòng lặp i bằng cách sử dụng hai phương trình (1) và (2). Khi tích phân có đỉnh cao và hẹp, Ii và σi2 thì ước tính tích phân đôi khi không chính xác trong những vòng lặp đầu tiên (trước khi tích phân có sự thích nghi). Và cách tốt nhất là thay thế phương trình (5) bằng

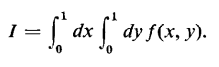


Trong mọi trường hợp Χ2 không được vượt quá số lần lặp. Thuật toán có thể không còn đúng khi thực hiện nó.

Số lặp và số lần tính tích phân cần cho mỗi lần lặp rõ ràng đều phụ thuộc vào độ phức tạp của hàm lấy tích phân và độ chính xác tìm được. Nói chung, tốt nhât để sử dụng như là một số đánh giá tính phân ở mỗi vòng lặp

1. **Thuật toán cho n chiều**

Thuật toán được mô tả bên dưới được tổng quá hóa với mục đích xử lý tích phân nhiều chiều. Để minh họa cho sự thay đổi



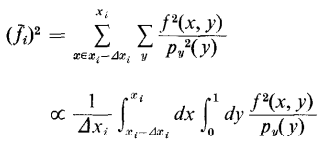
Hàm mật độ xác suất được tách rời để phù hợp với bộ nhớ lưu trữ có giới hạn của máy tính.



Trong trường hợp này mật độ tối ưu được thể hiện với một kết quả tương tự cho Py(y).



Do đó thuật toán một chiều có thể áp dụng dọc theo một trục. Nhưng (phương trình (4)) được định nghĩa bằng



Theo trục X và tương tự choh trục Y. Việc khái quát hóa cho số chiều bất kì là hiển nhiên.

1. Các ví dụ số (Numerical)

Chúng ta đã viết chương trình cho thuật toán bằng chương trình Fortran IV được gọi là VEGAS. Sau đó, lấy hiệu suất của chúng đem so sánh với các một số chương trình tính tích phân nhiều chiều khác. Các bảng I – V thể hiện kết quả các sự kiện so sánh này.

Tích phân trong bảng I và II là một khối cầu đối xứng Gaussian. Nó được đặt tại trọng tâm của miền lấy tích phân.

