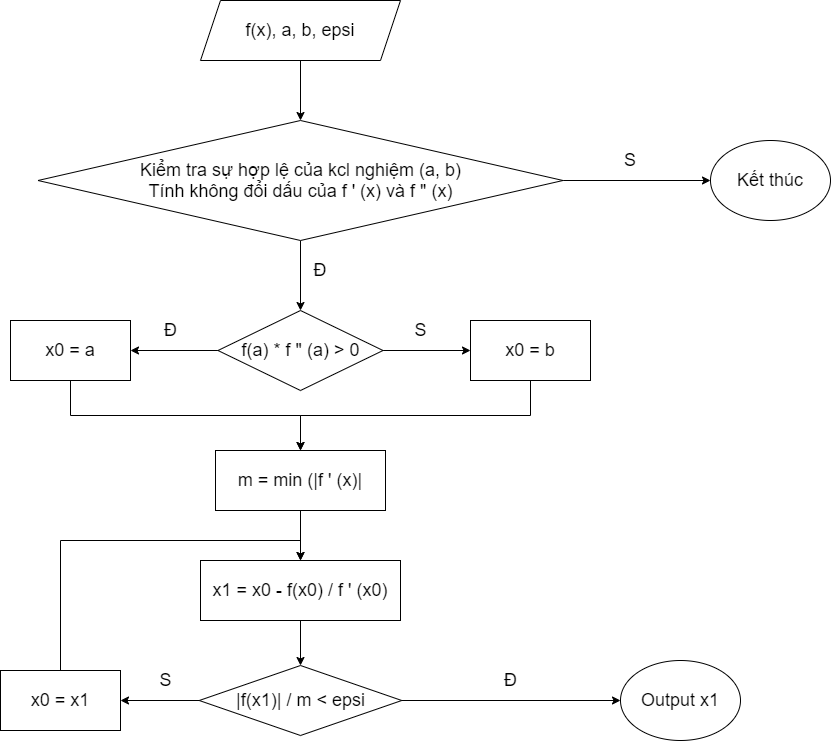
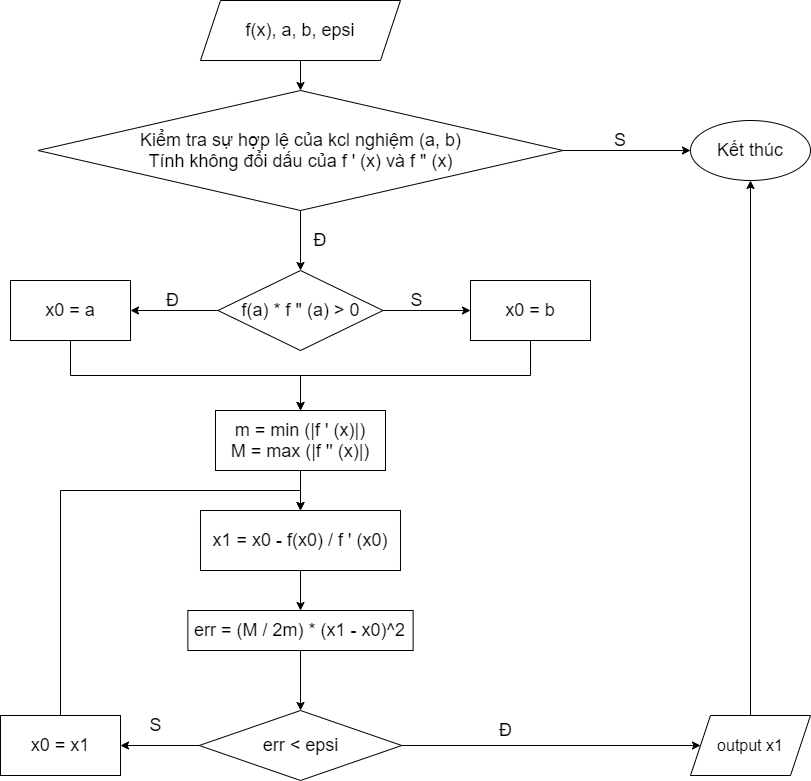
1. Thuật toán tổng quát
2. Thuật toán theo công thức:

|x – x\*| ; với m = min(|f’(x)|), [a, b]



1. Thuật toán theo công thức:

|xn – x\*| |xn – xn-1|2 ; với M = max(|f ’’(x)|), m = min(|f’(x)|) ; x [a, b]



1. Thuật toán chi tiết. Giả mã

|  |
| --- |
| 1. Hàm tính f(x) //Nhập hàm cần khảo sát ở hàm này  input: x  output: f(x)  Function f:  return x2 + x  2. Hàm tính f‘(x)  Input: f(x), x  Output: df(x)  Function df:  return (f(x + e-7) - f(x – e-7)) / (2 \* e-7)  3. Hàm tính giá trị đạo hàm cấp 2 của hàm số f(x)  Input: f(x), x  Output: dff(x)  Function dff:  return (df(x + e-7) - df(x – e-7)) / (2 \* e-7)  4. Hàm khảo sát tính đơn điệu của hàm số, chính là khảo sát sự không đổi dấu của hàm f’(x) và f”(x)  Input: f(x), a, b  Output: 1 nếu hàm input đơn điệu; 0 nếu hàm input không đơn điệu  Function check:  eta = 0.0001  if f(a) = f(b):  return 0  x0 = a  sign = -1  temp = df(f, a)  if df(f, x0) > 0:  sign = 1  while (x0 <= b):  x1 = x0 + sign \* eta \* df(f, x0)  if df(f, x1) \* temp < 0:  return 0  x0 = x1  return 1  5. Phương pháp Newton tiếp tuyến  5.1. Thuật toán theo công thức 1:  |x – x\*| ; với m = min(|f’(x)|), [a, b]  Input: f(x), a, b, epsi  Ouput: nghiệm gần đúng x  Function newton1:  if f(a) \* f(b) < 0 và check(f, a, b) = 1 và check(df, a, b) = 1:  if f(a) \* dff(f, a) > 0:  x0 = a  else:  x0 = b  if |df(f, a)| > |df(f, b)|:  m = |df(f, b)|  else:  m = |df(f, a)|  do:  x1 = x0 – f(x0) / df(f, x0)  x0 = x1  while (|f(f, x1)| / m > epsi)  return x1  5.2. Thuật toán theo công thức sai số 2:  |xn – x\*| |xn – xn-1|2 ; với M = max(|f ’’(x)|), m = min(|f’(x)|) ; x [a, b]  Input: f(x), a, b, epsi  Ouput: nghiệm gần đúng x  Function newton2:  if f(a) \* f(b) < 0 và check(f, a, b) = 1 và check(df, a, b) = 1:  if f(a) \* dff(f, a) > 0:  x0 = a  else:  x0 = b  if |df(f, a)| > |df(f, b)|:  m = |df(f, b)|  else:  m = |df(f, a)|  if |dff(f,a)| > |dff(f,b)|:  M = |dff(f,a)|  else:  M = |dff(f,b)|  do:  x1 = x0 – f(x0) / df(f, x0)  err = (M / 2m) \* |x1 – x0|^2  x0 = x1  while (err > epsi)  return x1 |

1. Ưu và nhược điểm của phương pháp Newton tiếp tuyến
2. Ưu điểm

* Dễ lập trình tính toán trên máy tính
* Tốc độ hội tụ nhanh do tốc độ hội tụ là bậc 2

1. Nhược điểm

* Yêu cầu điều kiện đầu vào cao: khoảng cách li nghiệm (a, b); tính không đổi dấu của hàm f’(x) và f’’(x)
* Độ phức tạp tính toán lớn hơn so với phương pháp chia đôi
* Quan trọng còn phải chọn đúng điểm Fourier

\*Chú ý: Điều kiện đầu vào cần có tính không đổi dấu của hàm f’(x) và f’’(x) hay là tính đơn điệu của hàm số f(x) và f’(x) trên kcl [a, b]

1. **Tóm tắt phương pháp.**
2. Điều kiện thỏa mãn phương pháp:

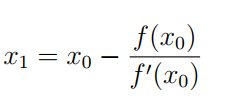
* (a, b) là khoảng cách li nghiệm
* f’(x) và f’’(x) liên tục và có dấu không đổi trên đoạn [a, b]: Dùng thuật toán Gradient Descent để xác định tính liên tục và tính không đổi dấu trên đoạn [a, b]

1. Chọn điểm Fourier:

x0 được gọi là điểm Fourier nếu f(x0) \* f’’(x0) > 0

* Nếu f(a) \* f’’(a) > 0:
* Đúng: điểm Fourier: x0 = a
* Sai: điểm Fourier: x0 = b

1. Áp dụng công thức lặp để tìm nghiệm gần đúng



1. Áp dụng công thức sai số: để tìm ra nghiệm
2. |x – x\*| ; với m = min(|f’(x)|), [a, b]
3. |xn – x\*| |xn – xn-1|2 ;

với M = max(|f ’’(x)|), m = min(|f’(x)|) ; x [a, b]