Algorithm 3: Secant method and Inverse linear interpolation

|  |
| --- |
| Method introduction: |
| The secant method is defined by the recurrence relation  As can be seen from the recurrence relation, the secant method requires two initial values, x0 and x1, which should ideally be chosen to lie close to the root. |
| Algorithm Design |
| F=@(x) …  X(1)=…;  X(2)=…;  For i=3:Maxiteration  X(i)=…%see in method introduction  end |
| Matlab code |
| function [x, y] = MySecant(fun, a0, b0, tol, max)  % This is the code for Secant method.  % Input:  % [a0, b0] Initial guess  % fun function  % tol Allowable tolerance in successive iterates  % max Maximum number of iterations  % Output:  % x Vector of approximations to zero  % y Vector of function values, fun(x)  % Preallocate vectors.  x = zeros(max, 1);  y = zeros(max, 1);  % Set an intial interval.  x(1) = a0; x(2) = b0;  y(1) = feval(fun, x(1)); y(2) = feval(fun, x(2));  % Secant search  for i = 2 : max  x(i+1) = x(i) - y(i)\*(x(i) - x(i-1))/(y(i) - y(i-1));  y(i+1) = feval(fun, x(i+1));  if y(i) == 0  fprintf('Exact solution found\n');  break;  end  if (abs(x(i+1) - x(i)) < tol)  fprintf('Secant method has converged\n');  break;  end  iter = i+2;  end  if (iter > (max+1))  fprintf('Zero not found to desired tolerance within the maximum number of iterations\n');  end  % Output results  k = 1:iter;  fprintf(' iter x y\n');  disp([k' x(1:iter) y(1:iter)]); |
| Examples and Result |
| Remarks |
| 此处写该方法程序设计的一些注意事项，也可以空白 |
|  |