Algorithm template with Function

Roy

October 21, 2020

```
Def calculate_definite_integral_of_f(f, initial_step_size):
    """
    This algorithm calculates the definite integral of a function from 0
     to 1, adaptevily, by choosing smaller steps near problematic points.
    """
    x = 0.0
    h = initial_step_size
    accumulator = 0.0
    While x < 1.0:
        If x + h > 1.0:
            h = 1.0 - x #At end of unit interval, adjust last step to end
             at 1.
        If error_too_big_in_quadrature_of_f_over_range(f, [x, x + h]):
            h = make_h_smaller(h)
        Else :
            accumulator += quadrature_of_f_over_range(f, [x, x + h])
            x += h
            If error_too_small_in_quadrature_of_over_range(f, [x, x + h]):
                h = make_h_larger(h) #Avoid wasting time on tiny steps.
    return accumulator
```