Romberg Integration

Recursive Composite Trapezoid Rules with Richardson Extrapolation

Maple Code

```
Romberg := proc(f,a,b,n)
    local R,i,j,k,h,s;
    h := evalf(b-a);
    R[0,0] := 0.5*h*(f(a)+f(b));
    for i to n do
        h := 0.5*h;
        s := 0.0;
        for k to 2^i-1 by 2 do
            s := s + f(a+k*h)
            end do;
        R[i,0] := 0.5*R[i-1,0] + s*h;
        for j to i do
            R[i,j] := R[i,j-1] + (R[i,j-1]-R[i-1,j-1])/(4.^j-1.);
            end do;
        end do;
    return(R);
    end proc:
```

Python Code

```
def romberg(f,a,b,n):
    """Romberg Integration:
        int(f(t), t=a..b) with n steps"""
    r=[] # the list must be defined before elements can be added
    h = (b-a)
    # Insert R[0,0]
    r.append([(h/2.0)*(f(a)+f(b))])
    for i in range(1,n+1):
        h = h/2.
        sum = 0
        for k in range(1,2**i,2):
            sum = sum + f(a+k*h)

        # Begin building the next row with R[i,0]
        rowi = [0.5*r[i-1][0] + sum*h]
        # Now calculate the rest of the row
        for j in range(1,i+1):
            rij = rowi[j-1] + (rowi[j-1]-r[i-1][j-1])/(4.**j-1.)
            # Add R[i,j] to rowi
            rowi.append(rij)

        # Add R[i,j] to r
        r.append(rowi)

    return r
```