http://www.mapleprimes.com/questions/201654-How-To-Calculate-Hard-Integral

```
> restart; interface(version); Digits:=15;
```

Classic Worksheet Interface, Maple 18.00, Windows, Feb 10 2014, Build ID 922027

Digits := 15

```
> Int(Int((-12*y^2+1)/(4*y^2+1)^3 * ln(abs(Zeta(x+y*I))),y = 0 .. infinity),x =
  1/2 .. infinity) = `?`;
```

$$\int_{1/2}^{\infty}\int_{0}^{\infty}\frac{(-12\,y^2+1)\,\ln(\left|\zeta(x+y\,I)\right|)}{(4\,y^2+1)^3}\,dy\,dx = ?$$

```
> H:=proc(y)
  #option hfloat;
  evalf( Int('x -> (-12*y^2+1)/(4*y^2+1)^3 * ln(abs(Zeta(x+y*I))) ',
    1/2 .. infinity, method = _d01amc) );  #epsilon=1e-8
  end proc;

  H1:=proc(y, eps) # version with error control
  #option hfloat;
  evalf( Int('x -> (-12*y^2+1)/(4*y^2+1)^3 * ln(abs(Zeta(x+y*I))) ',
    1/2 .. infinity, method = _d01amc, epsilon=eps) );  # default =
  0.5*10^(1-Digits) = 0.5 * 1e-14
  end proc;
  ``;
  `task` = Int( 'H'(y), y=0 .. infinity);
```

H := **proc**(y)

evalf(Int('x → (−12∗y^2 + 1)∗ln(abs(ζ(x + y∗I))) / (4∗y^2 + 1)^3', 1 / 2 .. ∞, method = _d01amc))

end proc

H1 := **proc**(y, eps)

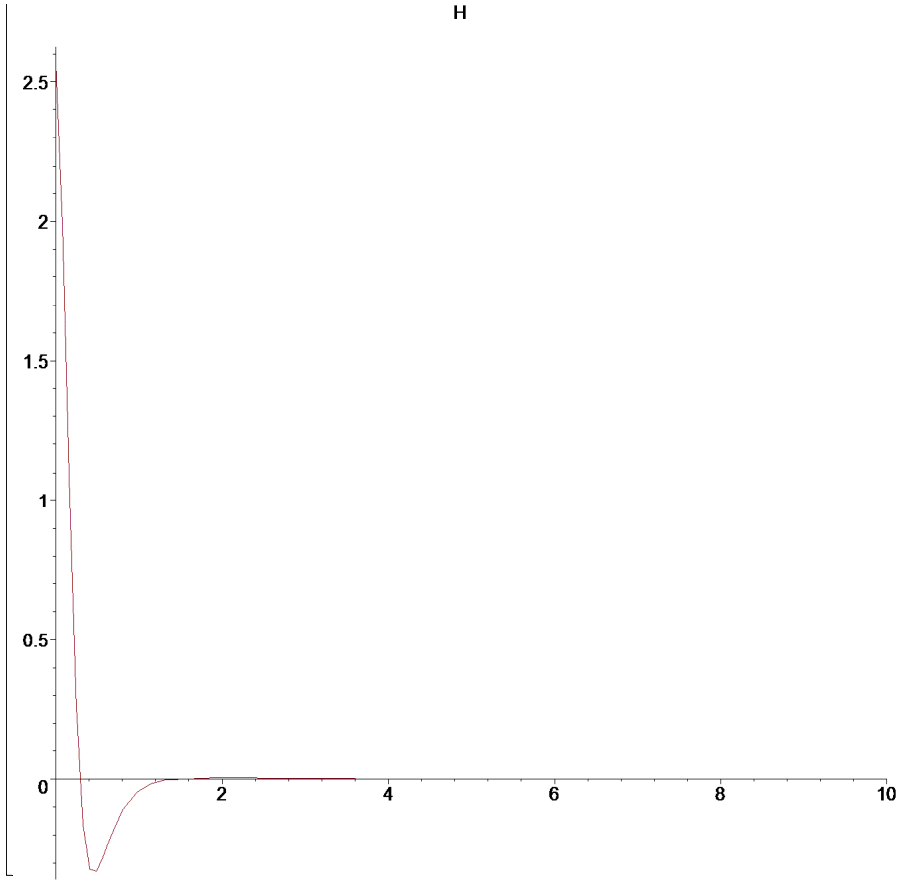
evalf(

Int('x → (−12∗y^2 + 1)∗ln(abs(ζ(x + y∗I))) / (4∗y^2 + 1)^3', 1 / 2 .. ∞, method = _d01amc, ε = eps))

end proc

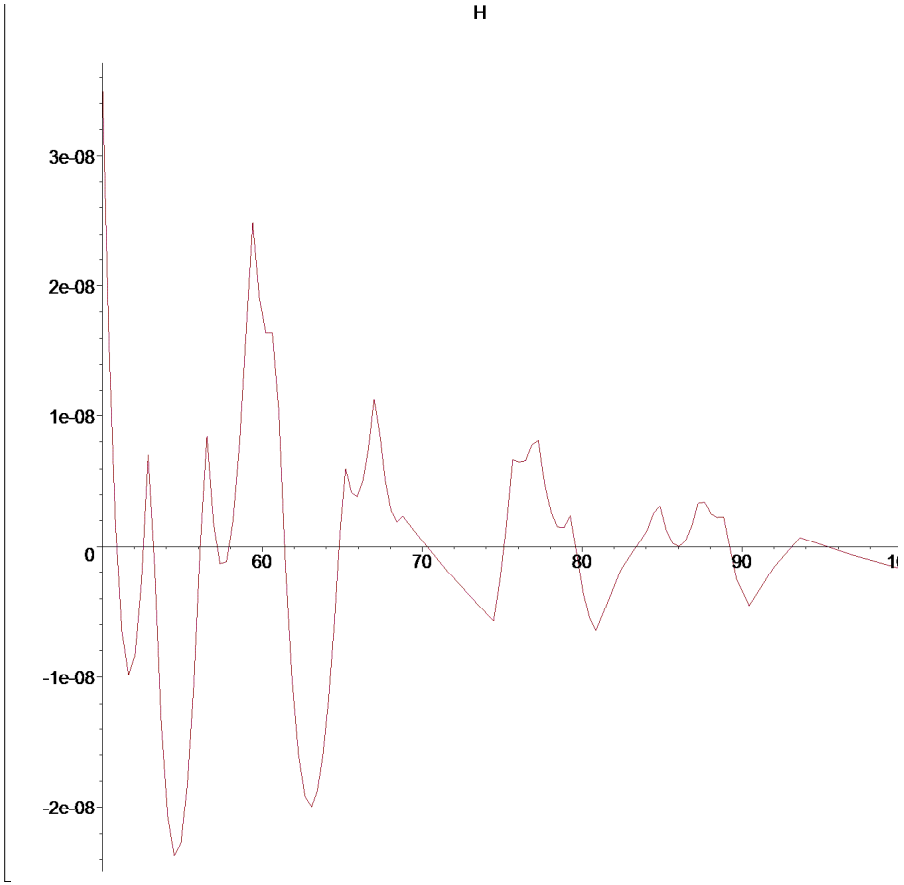
$$\text{task} = \int_{0}^{\infty} H(y)\,dy$$

```
> plot('H', 0 .. 10, numpoints=5, smartview=false, title="H");
```



It looks 'nice' for small values. But for larger values the graph is wild and it is not even clear, how the integrand decays!

```
> #plot(H, 50 .. 100, numpoints=5, smartview=false, title="H");
> # saved plot instead of plotting
```



Do the 'nice' part first: due to the 1st plot it should contribute almost everything for integration

```
> gc();
  Int( 'y -> H1(y, 1e-10)', 0 .. 2, method=_d01ajc, epsilon=1e-7);
  evalf(%);
  s0:=evalf[7](%);
```

$$\mathrm{Int}(y \to \mathrm{H1}(y, 0.1\ 10^{-9}), 0 .. 2, \mathrm{method} = \_d01ajc, \varepsilon = 0.1\ 10^{-6})$$

$$0.233222859264358$$

$$s0 := 0.2332229$$

So we have 7 'valid' decimals (after the zero dot). The next step is positive, has 2 leading zeros and relative error 1e-5, so adding them still gives 7 valid decimals

```
> gc();
  Int( 'y -> H1(y, 1e-10)', 2 .. 4, method=_d01ajc, epsilon=1e-5);
  res:=evalf(%);
  #res:=evalf[4](%);
  s:=s0 + res;
```

$$\mathrm{Int}(y \to \mathrm{H1}(y, 0.1\ 10^{-9}), 2 .. 4, \mathrm{method} = \_d01ajc, \varepsilon = 0.00001)$$

$$res := 0.00424735655729724$$

$$s := 0.237470256557297$$

Now continue in that style

```
> gc();
> for k from 2 to 10 do
    gc();
    res:=evalf( Int( 'y -> H1(y, 1e-10)', 2^k .. 2^(k+1), epsilon=1e-4) );
    if type(res, float) then
  #    res:=evalf[3](res);
      s:=s+res;
      print( INT(2^k .. 2^(k+1)) );
      print( 'res' = res, 's'=s);
    else
      break
    end if;
  end do:
```

$$\mathrm{INT}(4 .. 8)$$

$$res = 0.000425230639298627, s = 0.237895487196596$$

$$\mathrm{INT}(8 .. 16)$$

$$res = -0.0000342833799311755, s = 0.237861203816665$$

$$\mathrm{INT}(16 .. 32)$$

$$res = -0.482917001889248\ 10^{-5}, s = 0.237856374646646$$

$$\mathrm{INT}(32 .. 64)$$

$$res = -0.119801767606589\ 10^{-7}, s = 0.237856362666469$$

$$\mathrm{INT}(64 .. 128)$$

$$res = 0.266204355073591\ 10^{-7}, s = 0.237856389286905$$

$$\mathrm{INT}(128 .. 256)$$

$$res = -0.515014089529010\ 10^{-8}, s = 0.237856384136764$$

$$\mathrm{INT}(256 .. 512)$$

$$res = -0.459002717807084\ 10^{-9}, s = 0.237856383677761$$

$$\mathrm{INT}(512 .. 1024)$$

$$res = -0.943431768829678\ 10^{-10}, s = 0.237856383583418$$

$$\mathrm{INT}(1024 .. 2048)$$

$$res = 0.345914310978702\ 10^{-10}, s = 0.237856383618009$$

The final result should be correct to 6 decimals (assuming the partial integrals decay that way, I do not have estimates for the integrand)

```
> `task` = evalf[7](s);
```

$$task = 0.2378564$$

```
>
```