I don't think a sufficiently clever algorithm could save us, because inherent in the problem is that the roots are incredibly sensitive to the coefficients. It's nothing wrong with our algorithm per se.

Problem 8

Part (d)

Below, I've written a code to demonstrate that the algorithm we have made for calculating the terms in this series is accurate.

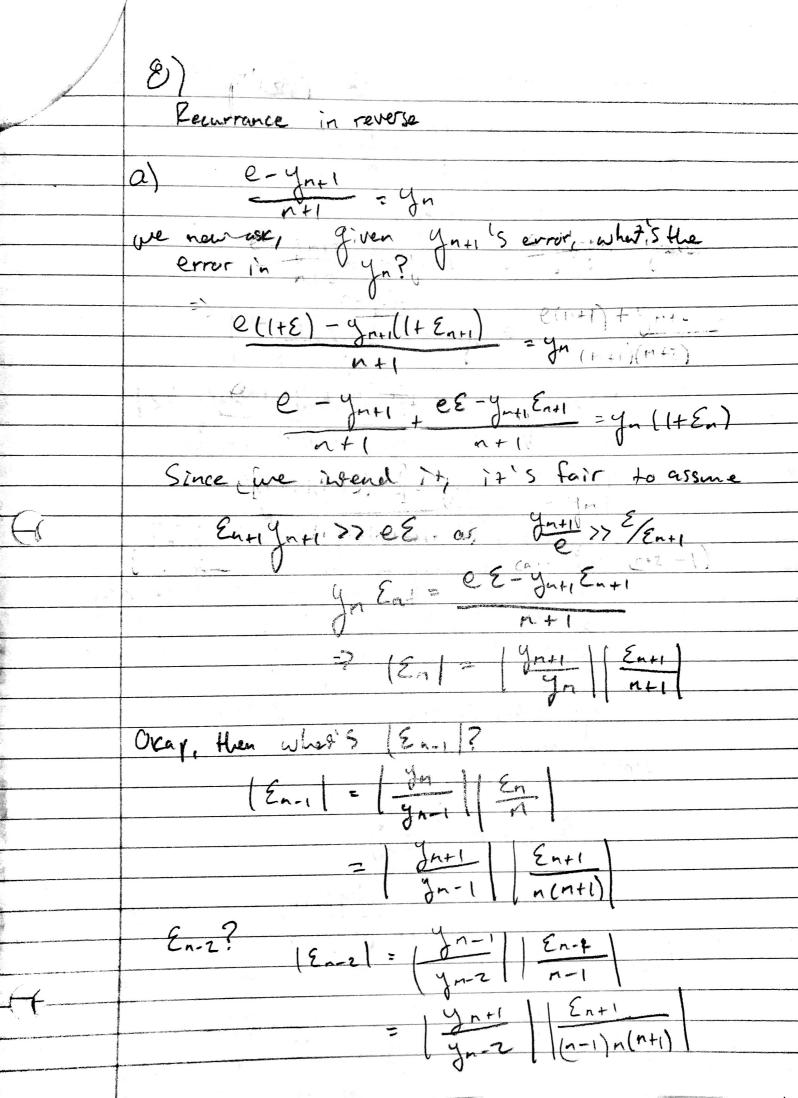
```
in[a] = y = 0.; (* Our "guess" *)
     n = 25; (* The higher value we start at. *)
     k = 20; (* The desired value *)
     For[i = n, i > k, i--,
      y = (E - y) / i;
     yk = y;
     Print[N[y]]
     0.123804
ln(x) = ykInt = NIntegrate[x^kExp[x], \{x, 0, 1\}]
Out[= 0.123804
```

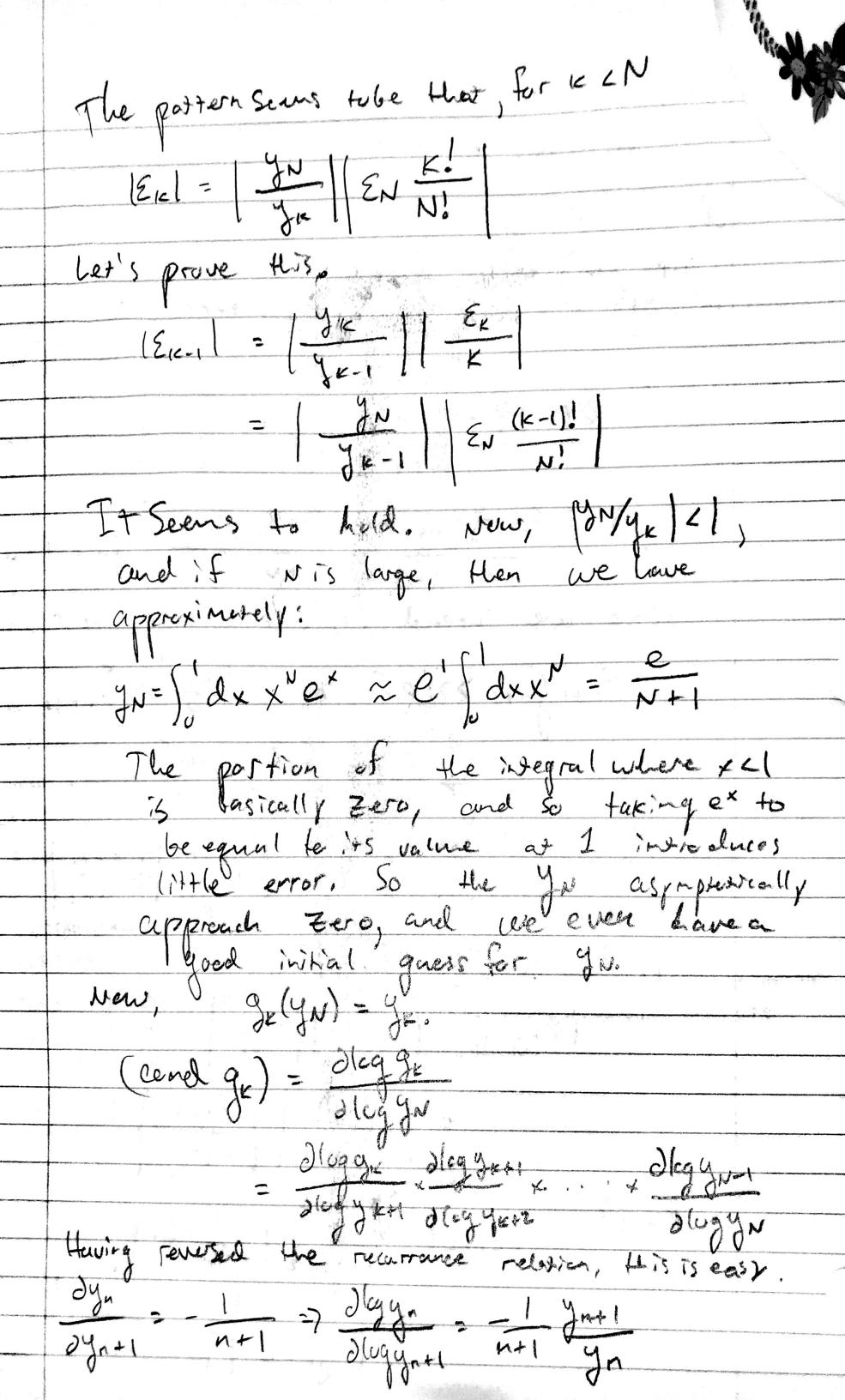
At least by inspection, they appear the same, so it appears our algorithm has worked. Let's subtract the two to find what the relative error is, assuming that the integral is exact to machine precision.

```
in[*]:= Log2[Abs[ykInt - yk]]
Out[\circ] = -25.9144
```

The error is less than 2^{-25} , and our goal was only to get it to better than 2^{-23} , so I would say we have met and exceeded our tolerances. It helps to use reasonable upper bounds on everything all the time. Printing these side by side, we can visually see the slight difference.

```
In[-]:= NumberForm[yk, 16]
       NumberForm[ykInt, 16]
Out - WNumberForm=
       0.1238038465745379
Out | JaNumberForm=
       0.1238038307625704
```





So we see that : (Cond gk) = (-1) N-K yN K! YK N! we could have backed this own of our previous expression for Ek(EN). Let ExLE be the required tolerance. Let He initial EN = 14N, and so EK = [JN K!] Well, yk > yv, so taking yv -> yn > yx Ext gn Kel Also, Ju & P (Since exx X x on (e,1) So to ensure machine precision, we need to Start at what ever N satisfies, givenow &, Ex < ex! CE There's no inverse factorial, so it's hard to write an exact expression,

C) &= eps. K=20. E+20! (eps = 2⁻²³ (single precision)

[N+1)!

The have to, we cannot stirling's approximation for n!

we would want to impose a lower boundon

n! the bound is: $E_{k} \leftarrow \frac{e_{k}!}{(N+1)!} \leftarrow \frac{e_{k}!}{\sqrt{2\pi}} + \frac{$ For K=20, what N makes this less than
the much ne epsilon? It's easingt to
Solve His in leg space: -23 log 2 > log (20!) + (N+2) - 12N+13 - (N+3) log (N+1)
- 1 (cg (27)) (I numerically Solved this) Remaine up, we find that N = 25 Not antholy so for from 20! Nice. What about double precision? we just let -23 leg2 - ? = 521 cg? => To doubte precision account, we need to go from an N=31