

[IntTest.mws](#)

- [A brief expository proof of the algorithm using the traditional bounding interval](#)
 - [Graphical evidence of inequality 1](#)
 - [The new tighter bounding interval](#)
 - [Graphical proofs of the Trapezoid Rule and Midpoint Rule inequalities for positive convex functions](#)
 - [The algorithm encoded in Maple with extensive comments](#)
 - [Example problems](#)
 - [Example 1: Evalhf used all the way](#)
 - [Example 2: Complicated antiderivative](#)
 - [Example 3: Maple cannot even approximate](#)
 - [Miscellaneous examples with no commentary](#)
-

Using the new bounding interval for the Integral Test to Approximate Series

Author: Carl Devore <devore@math.udel.edu>

Copyright (C) 2003, Carl DeVore. All rights reserved.

23 November 2003

Introduction :

New developments in Calculus, even at the elementary level, continue to this day. This worksheet has been updated to use a new substantially tighter bounding interval just published in November 2003. Here is an algorithm for approximating an infinite series by using a partial sum plus an error corrector derived from the Integral Test. This gives much more powerful error control than merely using partial sums to approximate the series. This technique can be used to quickly approximate some series that Maple cannot already approximate, and thus the power of Maple is extended significantly.

Keywords: Integral Test, Approximation of Series, Calculus Education

Audience:

1. Calculus students learning about the Integral Test who are interested in some of the finer computational aspects.

2. Numerical analysis students learning how to approximate series.
3. Computational professionals interested in learning efficient Maple techniques.

Unusual Maple commands used (info for those taking a beginning course in Maple):

error, evalhf, try, userinfo, fnormal, timelimit

Author contact:

The author, Carl Devore, would love to here from you, if you found this worksheet useful, or if you have any suggestions for its improvement. He can be reached via email or via paper mail at 189 Madison Dr, Newark DE 19711, USA. If you use this worksheet, please email me at <mailto://devore@math.udel.edu> , to discuss it, or just to say "hi". If you find this worksheet useful, please consider supporting the work of a poor graduate student who spends thousands of dollars per year on math books and computer equipment by sending me a small item from my Amazon.com wishlist http://www.amazon.com/exec/obidos/registry/4SWHEIGN0FN9/ref=wl_s_3/104-5062484-1726354 . You just have to select the items, and they are sent directly to me.

Copyright:

This worksheet and the Maple code it contains may not be distributed in any form other than by download from the Maple Applications Center without prior permission of the author. Instructors: Give your students a pointer to the worksheet's URL at the Maple Applications Center rather than putting the worksheet itself on your course web pages. In addition to helping me keep track of it, this ensures that you get the latest version.

A brief expository proof of the algorithm using the traditional bounding interval

(The student is encouraged to fill in the details.)

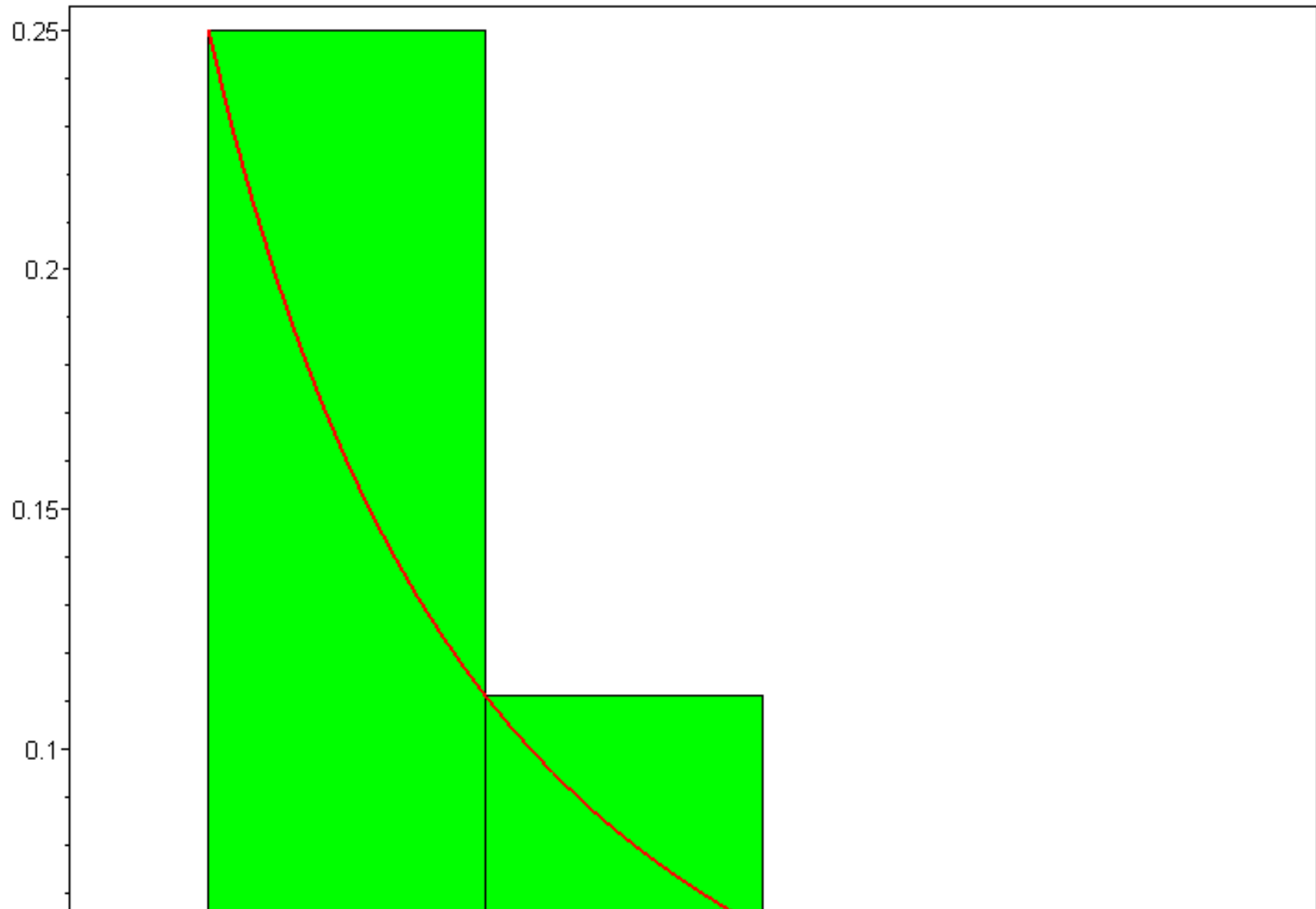
We suppose that we have a nonnegative nonincreasing function $A(x)$ such that $\sum_{n=b}^{\infty} A(n)$ converges by the Integral Test. It follows from a

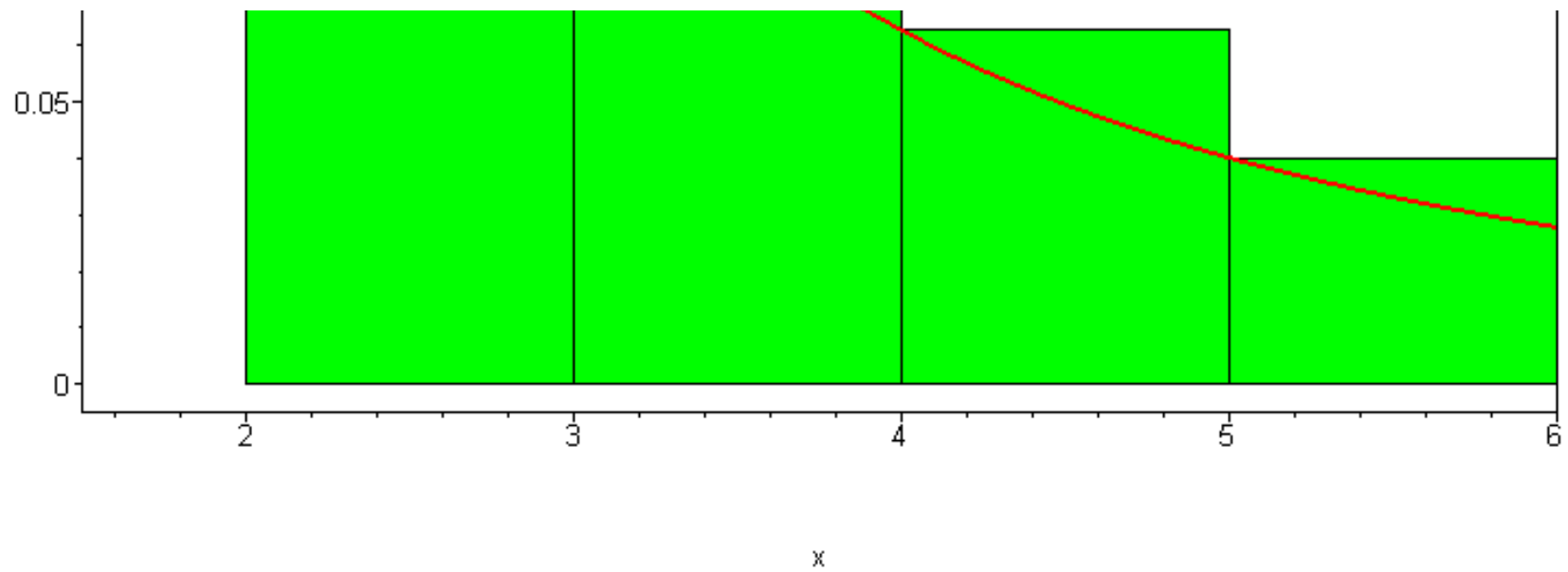
consideration of both left-endpoint and right-endpoint Riemann sums with $\Delta x = 1$ (you can find the details in the Integral Test section of most calculus textbooks) that for any integer $n > b$,

$$\int_{n+1}^{\infty} A(x) dx \leq \sum_{k=n+1}^{\infty} A(k) \leq \int_n^{\infty} A(x) dx \quad (1)$$

Graphical evidence of inequality 1

> `student[leftbox](1/x^2, x= 2..6, 4, shading= green, view= [1.5..6, DEFAULT], axes= boxed);`





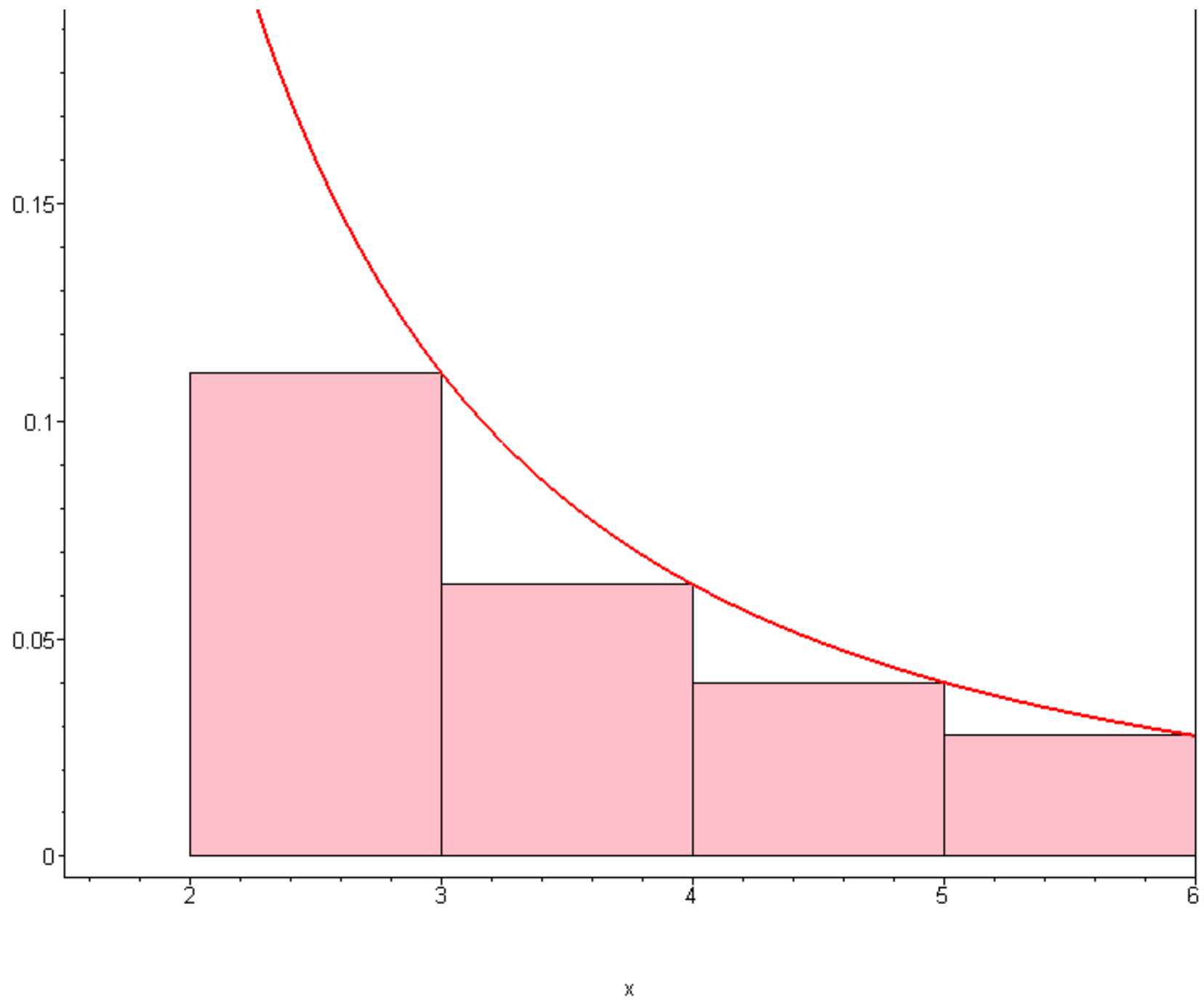
We see from the graph that

- > **$\text{Int}(1/x^2, x=3..infinity) \leq \text{Sum}(1/n^2, n=3..infinity);$**

$$\int_3^{\infty} \frac{1}{x^2} dx \leq \sum_{n=3}^{\infty} \frac{1}{n^2}$$

- > **`student[rightbox](1/x^2, x=2..6, 4, shading= pink, view= [1.5..6, DEFAULT], axes= boxed);`**





And we see from this graph that

$$> \text{Sum}(1/n^2, n= 3..\text{infinity}) <= \text{Int}(1/x^2, x= 2..\text{infinity});$$

$$\sum_{n=3}^{\infty} \frac{1}{n^2} \leq \int_2^{\infty} \frac{1}{x^2} dx$$

Continued exposition of the algorithm

If we add the partial sum $S_n := \sum_{k=b}^n A(k)$ to inequality (1), we get

$$\int_{n+1}^{\infty} A(x) dx + S_n \leq \sum_{k=b}^{\infty} A(k) \leq \int_n^{\infty} A(x) dx + S_n, \quad (2)$$

so that we have an interval that bounds the true sum of the series. If we approximate the sum as the midpoint of the interval, the error can be at most half the width of the interval. It remains to choose n . The width of the interval is the difference of the endpoints, which simplifies to

$$W := \int_n^{n+1} A(x) dx. \quad (3)$$

So, if we let E denote the maximum allowed error, we solve the equation

$$\int_n^{n+1} A(x) dx = 2E \quad (4)$$

for n . There likely will not be an integer solution, however, since A is necessarily a function decreasing to 0 (since the integral test was passed), there must be some real solution, and we can round it **up** to the next integer. Alternatively, we can just keep evaluating the above integral (by evaluating the antiderivative) until we get the first that works. Which approach is faster depends on the relative costs of evaluating A and its antiderivative and the complexity of solving equation (4) numerically.

The midpoint of the interval is the average of the endpoints, which is

$$M := S_n + \left(\frac{1}{2}\right) \left(\int_n^{\infty} A(x) dx + \int_{n+1}^{\infty} A(x) dx \right) .$$

Note that we only need to compute the antiderivative once and for all. The rest of the integral computations can be done with the symbolic antiderivative. If we need to resort to numerical integration, there is probably not much value in this algorithm (a fact that is not mentioned in some Maple companion books to popular calculus textbooks).

That midpoint value can also be expressed as

$$M := S_n + \int_{n+1}^{\infty} A(x) dx + \left(\frac{1}{2}\right) \int_n^{n+1} A(x) dx , \quad (5)$$

which may be a more convenient form since the latter integral may have already been computed while solving equation (4).

Letting F be the antiderivative of A , and letting $F(\infty)$ denote $\lim_{x \rightarrow \infty} F(x)$, the width (3) can be expressed as

$$W := F(n+1) - F(n) ,$$

and the midpoint (5) can be expressed as

$$M := S_n + F(\infty) - F(n+1) + \left(\frac{1}{2}\right)W$$

or

$$M := S_n - \frac{F(n) + F(n+1)}{2} + F(\infty) \quad .$$

The new tighter bounding interval

In article [1], Roger Nelsen shows that with the additional assumption that $A(x)$ is convex (also known as concave up, or having positive second derivative), inequality (2) can be substantially sharpened to

$$\frac{A(n+1)}{2} + \int_{n+1}^{\infty} A(x) dx + S_n \leq \sum_{k=b}^{\infty} A(k) \leq \int_{n+\frac{1}{2}}^{\infty} A(x) dx + S_n \quad . \quad (6)$$

The convexity assumption will hold for the vast majority of series to which the Integral Test applies.

The proof of this inequality begins with the observation that for a positive convex function, a Trapezoid Rule approximation to an integral is larger than the integral itself. This fact is obvious from the following plot, and does not need further proof. The Trapezoid Rule approximation to

$$\int_{n+1}^{\infty} A(x) dx \quad \text{with intervals of width 1 is } \frac{A(n+1)}{2} + \left(\sum_{k=n+2}^{\infty} A(k) \right) = -\frac{A(n+1)}{2} + \left(\sum_{k=n+1}^{\infty} A(k) \right) \quad . \quad \text{Hence we have the}$$

inequality

$$\int_{n+1}^{\infty} A(x) dx \leq -\frac{A(n+1)}{2} + \left(\sum_{k=n+1}^{\infty} A(k) \right).$$

After rearranging and adding S_n to both sides, we get the left side inequality in (6).

The right side inequality in (6) uses the fact that for a positive convex function, the Midpoint Rule approximation is less than the integral. This is not as obvious as the inequality for the Trapezoid Rule, but a proof is given in the plots section below. The Midpoint Rule approximation to

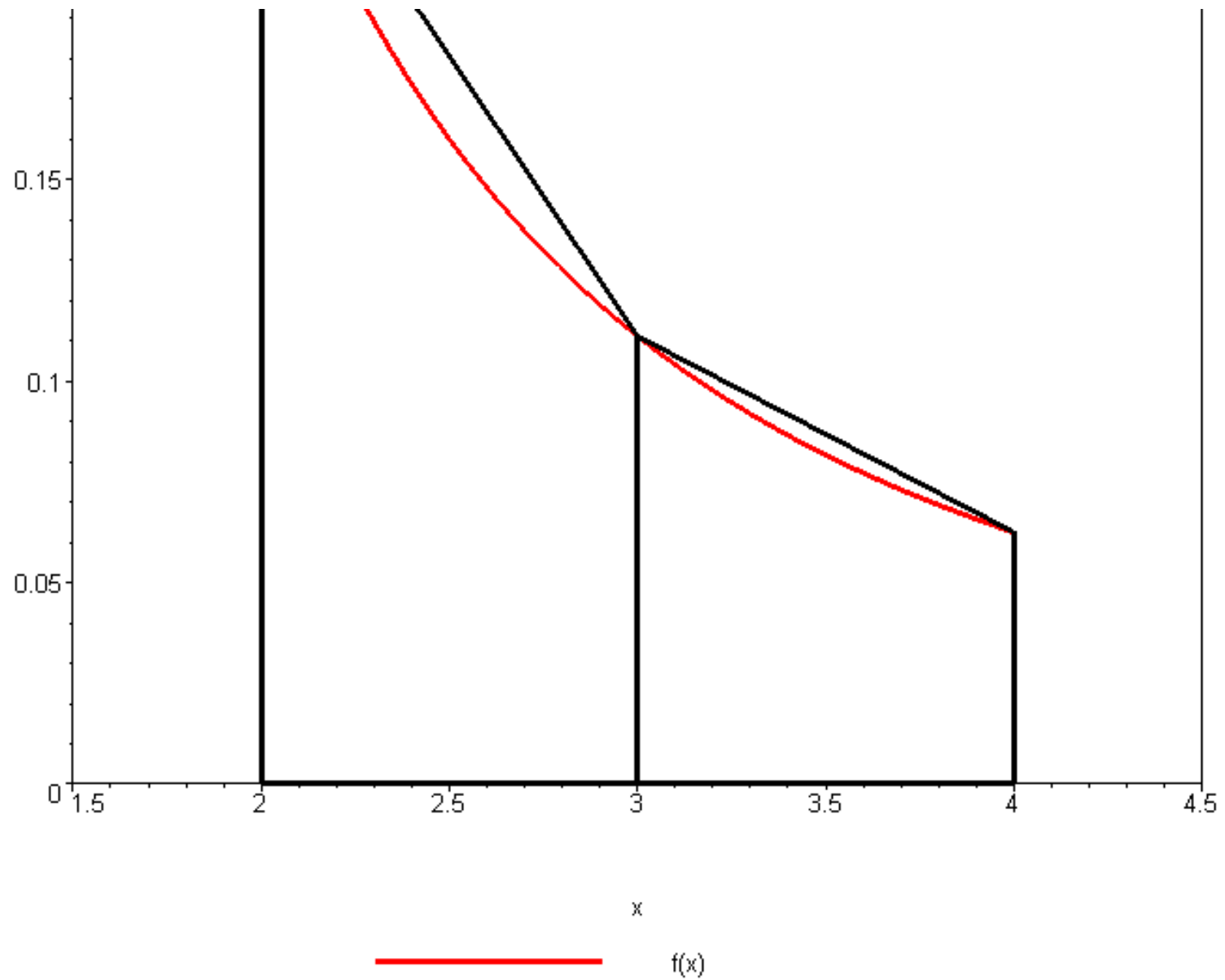
$\int_{n+\frac{1}{2}}^{\infty} A(x) dx$ using intervals of width 1 is simply $\sum_{k=n+1}^{\infty} A(k)$, and upon adding S_n , we have the right side inequality in (6).

Graphical proofs of the Trapezoid Rule and Midpoint Rule inequalities for positive convex functions

The following plot shows why the Trapezoid Rule approximation the integral of a positive convex function is greater than the intergal itself. Indeed, convexity can be characterized as the secant line always being above the curve.

> `plots[display](Student:-Calculus1:-ApproximateInt(1/x^2, x= 2..4, partition= [2,3], output= plot, method= trapezoid), view= [1.5..4.5, 0..0.28], title= '', axes= boxed, thickness= 3);`

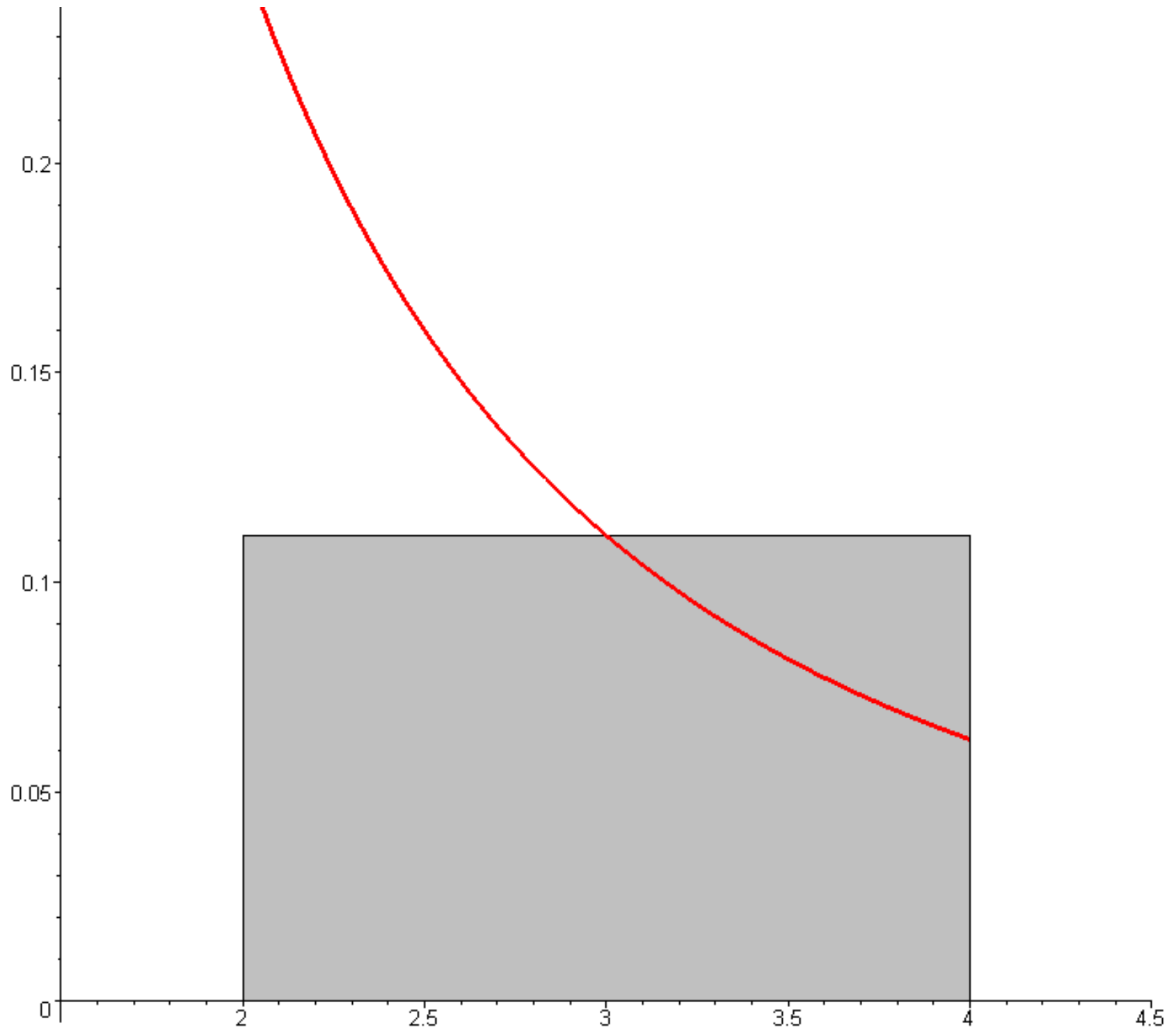




The Midpoint Rule approximation to the same integral can be shown as

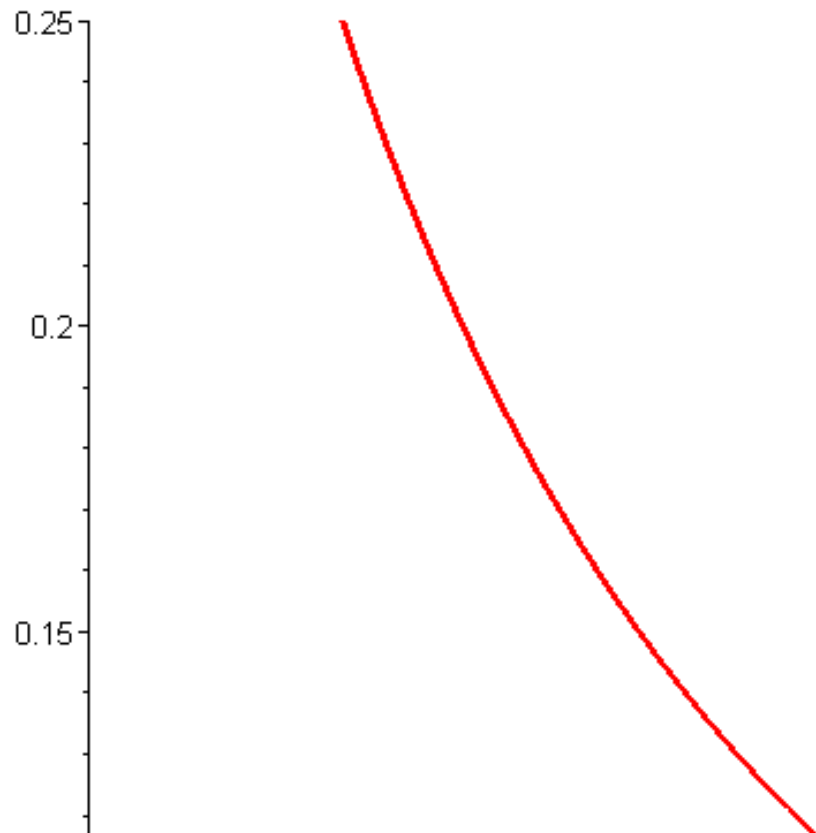
```
> student[middlebox](1/x^2, x= 2..4, 1, view= [1.5..4.5, DEFAULT], thickness= 3, shading= grey);
```

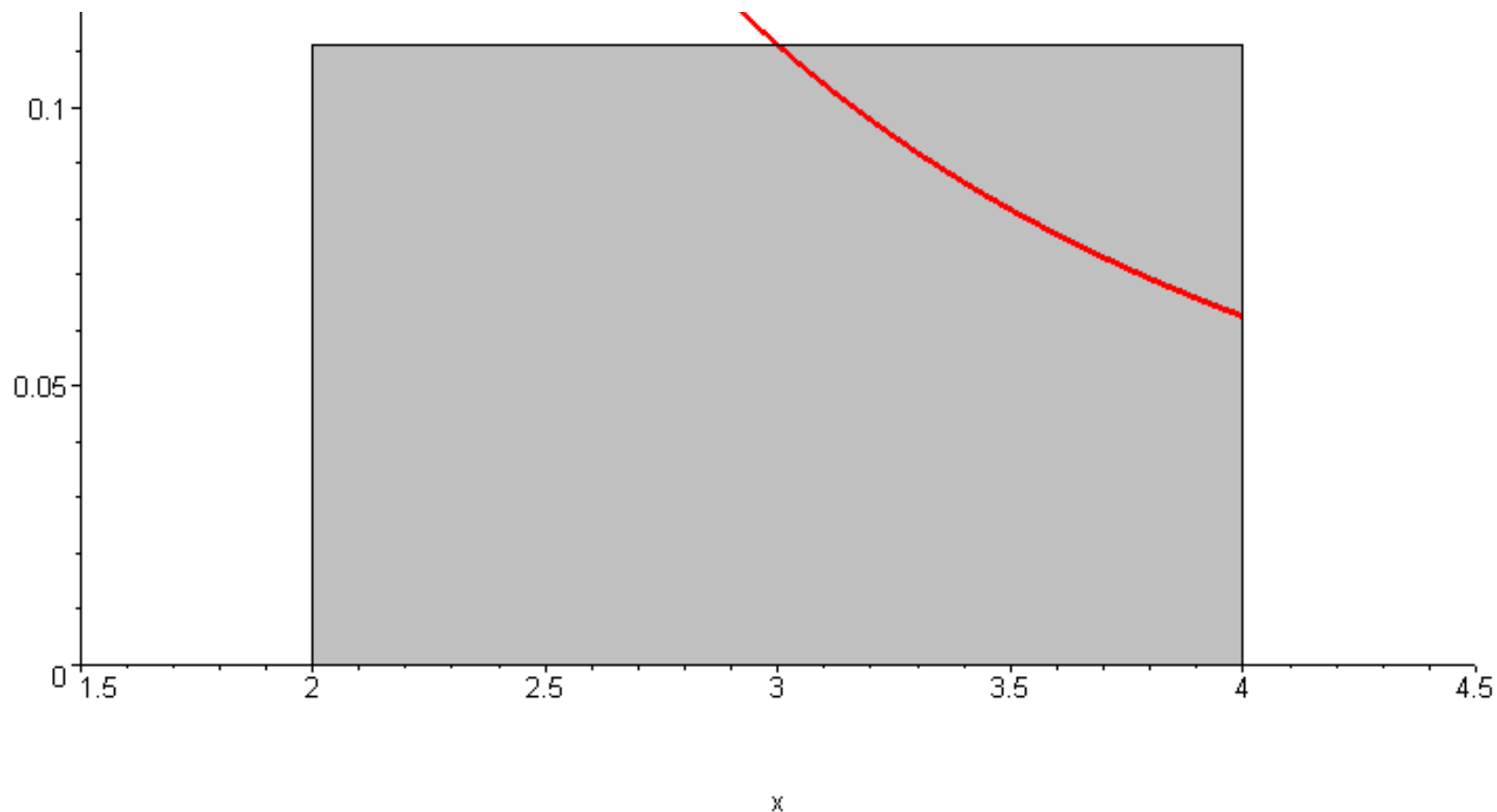




If we rotate the line segment through the point $(3, 1/9)$ clockwise until it is tangent to the curve, the grey rectangle becomes a trapezoid, but its area is not changed.

```
> plots[display](
  plots[display](
    seq(
      plots[polygonplot]([[2,0], [2,5/27-(2/27)*k/10], [4,1/27+(2/27)*k/10], [4,0]], color= grey)
      ,k= 0..10
    )
    ,insequence= true
  )
  ,plot(1/x^2, x= 2..4, thickness= 3)
  ,view= [1.5..4.5, 0..0.25]
);
```





Play the animation to the line rotate without the area changing.

When the line is in tangent position, it is clear that the grey area is less than the integral. Indeed, another characterisation of convexity is the tangent line always being below the curve.

For interval (6), the width is

$$W := F(n+1) - F\left(n + \frac{1}{2}\right) - \frac{A(n+1)}{2},$$

and the midpoint is

$$M := S_n + \frac{A(n+1)}{4} - \left(\frac{1}{2}\right) \left(F(n+1) + F\left(n + \frac{1}{2}\right) \right) + F(\infty) .$$

You should fill in the details.

Reference:

[1] Roger B. Nelsen <nelsen@lclark.edu>, An Improved Remainder Estimate for Use with the Integral Test, *College Mathematics Journal* **34:5** (November 2003), 397-399.

The algorithm encoded in Maple with extensive comments

```
> restart;

> IntSum:= proc(A::procedure, b::integer, E::And(positive,numeric))
# Note how we enforce the type of the arguments.
option `Copyright (C) 2003, Carl James DeVore, Jr. All rights reserved.`;
local
  F #antiderivative of A
  ,Finf #limit at infinity of F
  ,DoItHF #proc that does the numerical computation in the evalhf environment
  ,DoIt #proc that does numerical evaluation if evalhf can't be used
  ,x, t
;
# Compute symbolic antiderivative: The following technique with a variable
# limit of integration is a more robust way to get the antiderivative because
# it makes more use of the assumption.
assume(x>b);
F:= int(A(t), t= b..x);
# Note below how I check whether the int command actually did anything useful:
# if the expression still begins with "int" in it, then it didn't work.
if op(0,F)=int then
  # Try less robust "pure antiderivative" method
  x:= 'x'; #Remove assumptions
  F:= int(A(x), x);
  if op(0,F)=int then
    error "Don't know how to integrate"
```

```

fi
fi;
# Make antiderivative a procedure:
F:= unapply(F, x);
Finf:= evalf(limit(F(x), x= infinity));

# Using evalhf, we just accumulate the partial sum until the error condition is met.
# This is quicker than precomputing the necessary value of n.
DoItHF:= proc(A,b,F,E)
  local
    S #partial sum
    #Note that you can give variables brief yet meaningful names by using ``
, `F(n+1)`, n, `F(n+1/2)`, `2E`, `A(n+1)`
;
S:= 0;
`2E`:= 2*E;
`A(n+1)`:= A(b);
for n from b do
  S:= S+`A(n+1)`;
  `F(n+1)`:= F(n+1);
  `F(n+1/2)`:= F(n+1/2);
  `A(n+1)`:= A(n+1);
  if `F(n+1)` - `F(n+1/2)` - `A(n+1)/2 < `2E` then break fi
od;
userinfo(1, 'IntSum', `Used`, n, `terms`);
S + `A(n+1)/4 - (`F(n+1)` + `F(n+1/2)`)/2
# F(infinity) will be added outside evalhf
end proc;

# If we can't use evalhf, then it pays to use a bit of finesse to find a good value of N.
# Then we can use "add" to get the partial sum without looping. Outside of evalhf,
# add is far faster than looping.
DoIt:= proc(A, b)
  # I need to declare the variables above because I might want to pass them to
# evalhf. evalhf can't use scoped variables.
local ans, S, N, n, k, Width, err, oldDigits;

err:= evalf(2*E);
assume(N, integer); additionally(N >= b);
Width:= unapply(simplify(F(N+1)-F(N+1/2)-A(N+1)/2), N);

```

```

# Don't need much accuracy on the fsolve
oldDigits:= Digits;
Digits:= 5;
# Note that unapply is used so that the simplification is only done once.
try
  # fsolve can be slow. Allow a max of .5 seconds.
  ans:= timelimit(.5, fsolve(Width(n)=err, n= b..infinity))
catch:
  userinfo(1, IntSum, `fsolve failed. Using cruder technique.`)
end try;
Digits:= oldDigits;
if not ans::numeric then
  # fsolve didn't work. Need to use cruder technique.
  for k do
    N:= b+2^k-1;
    # The abs is used to deal with spurious small imaginary parts that
    # sometimes occur when complicated theoretically real expressions are evalf'd.
    if abs(evalf(Width(N))) < err then break fi
  od
else
  N:= ceil(ans)
fi;
userinfo(1, IntSum, `Using`, N-b+1, `terms`);
# Might be able to use evalhf for A even if it won't work for the antiderivative.
if Digits > evalhf(Digits) then
  if fnormal(evalf(E)) > 0 then
    userinfo(1, IntTest, `Extra accuracy is probably wasted. You can speed up`);
    userinfo(1, IntTest, `computations by setting Digits to`, evalhf(Digits));
    userinfo(1, IntTest, `or less.`)
  fi;
  # Note that I put the evalf inside the add. There is some chance that this will take
  # a little longer, but it will save a tremendous amount of memory if the partial sums
  # cannot be simplified.
  S:= add(evalf(A(n)), n= b..N)
else
  # "try" evalhf on the add. (See extended comment on try-catch blocks below.)
  # Note that all variables must be passed to the evalhf'd proc.
  try
    S:= evalhf(proc(A,b,N) local n; add(A(n), n= b..N) end (A,b,N));

```



```

    # If there is a problem with the evalhf, then the statement below is skipped.
    userinfo(1, IntSum, `Evalhf successful for the partial sum.`)
  catch:
    userinfo(1, IntSum, `evalhf failed on the partial sum. Using slower method.`);
    S:= add(evalf(A(n)), n= b..N)
  end try
fi;
# Add the corrector.
S + evalf(A(N+1)/4 - (F(N+1)+F(N+1/2))/2) + Finf;
end proc;

if Digits > evalhf(Digits) then # Can't use hardware floating point.
  if fnormal(evalf(E)) > 0 then
    userinfo(1, IntTest, `Extra accuracy is probably wasted. You can speed up`);
    userinfo(1, IntTest, `computations by setting Digits to`, evalhf(Digits));
    userinfo(1, IntTest, `or less.`)
  fi;
  DoIt(A,b,F,E)
else
  # We "try" to use hardware floating point computation with evalhf. This, if
  # it works at all, is far faster than evalf. It is substantially trickier
  # to use than evalf, and the functions A for which it works are severely
  # restricted, but the time savings are worth it. If you do serious numerical
  # computations with Maple, then you need to learn how to use evalhf.
  # Note that the final limit must be done outside of evalhf.
  try
    evalhf(DoItHF(A,b,F,E)) + Finf
    # If there are any errors in the "try" block, then the "catch:" block
    # is executed.
  catch:
    userinfo(1, IntSum, `evalhf failed. Using slower technique.`);
    DoIt(A,b,F,E)
  end try
fi
end proc:

```

Example problems

In order to access the extra information provided by the **userinfo** commands in the above procedure, we issue the following command:

> **infolevel[IntSum]:= 1:**

Example 1: Evalhf used all the way

Verify that $\sum_{n=1}^{\infty} \ln\left(1 + \frac{1}{n^2}\right)$ converges, and then approximate this series to 6 decimal places.

Solution:

> **A:= n-> ln(1+1/n^2);**

$$A := n \rightarrow \ln\left(1 + \frac{1}{n^2}\right)$$

The function is clearly positive and decreasing over the interval of interest. Apply the integral test.

> **int(A(n), n= 1..infinity);**

$$-\ln(2) + \frac{\pi}{2}$$

The test proves that the series converges.

> **IntSum(n-> ln(1+1/n^2), 1, .5e-6);**

IntSum: Used 63 terms

1.301846244

Let's compare that with Maple's approximation. First, let's see if Maple can do the sum symbolically.

> **Sum(ln(1+1/n^2), n= 1..infinity);**

$$\sum_{n=1}^{\infty} \ln\left(1 + \frac{1}{n^2}\right)$$

> **value(%);**

$$\sum_{n=1}^{\infty} \ln\left(1 + \frac{1}{n^2}\right)$$

Maple, can't do it symbolically.

> **evalf(%);**

1.301846399

We see that we have agreement to 6 decimal places. Let's see how bad it is to just use the partial sum:

> **add(evalf(ln(1+1/n^2)), n= 1..63);**

1.286099349

We only get 1 decimal place. Even if we take a very large number of terms...

> **evalhf(proc(A) local n; add(A(n), n= 1..100000) end(A));**

1.30183639865396428

...we only get 4 decimal places. Note the speed with which evalhf computed 100,000 terms above.

Example 2: Complicated antiderivative

Approximate $\sum_{n=1}^{\infty} \frac{1}{\sqrt{n^3+1}}$ to six decimal places.

Solution:

This series converges by the Limit Comparison Test comparing with $\sum_{n=1}^{\infty} \frac{1}{\left(\frac{3}{2}\right)^n}$. The terms clearly determine a continuous, positive decreasing

function. So applying the Integral Test in reverse, the corresponding integral converges even if we don't know what the integral is. As long as Maple knows the antiderivative and its limit at ∞ , we can still approximate the sum by the method of this worksheet. The antiderivative is complicated:

> **A:= n-> 1/sqrt(n^3+1);**

$$A := n \rightarrow \frac{1}{\sqrt{n^3+1}}$$

> **int(A(x), x);**

$$\frac{2 \left(\frac{3}{2} - \frac{1}{2} I \sqrt{3} \right) \sqrt{\frac{1+x}{\frac{3}{2} - \frac{1}{2} I \sqrt{3}}} \sqrt{\frac{x - \frac{1}{2} - \frac{1}{2} I \sqrt{3}}{-\frac{3}{2} - \frac{1}{2} I \sqrt{3}}} \sqrt{\frac{x - \frac{1}{2} + \frac{1}{2} I \sqrt{3}}{-\frac{3}{2} + \frac{1}{2} I \sqrt{3}}} \operatorname{EllipticF} \left(\sqrt{\frac{1+x}{\frac{3}{2} - \frac{1}{2} I \sqrt{3}}}, \sqrt{\frac{-\frac{3}{2} + \frac{1}{2} I \sqrt{3}}{-\frac{3}{2} - \frac{1}{2} I \sqrt{3}}} \right)}{\sqrt{x^3+1}}$$

The EllipticF cannot be processed by evalhf. Can Maple do the infinite integral?

> **int(A(x), x= 1..infinity);**

$$\frac{1}{3} 3^{\left(\frac{3}{4}\right)} \text{EllipticF}\left(\frac{2\sqrt{2} 3^{\left(\frac{1}{4}\right)}}{2+\sqrt{3}}, \frac{\sqrt{2}\sqrt{3}}{4} + \frac{\sqrt{2}}{4}\right)$$

> **IntSum(A, 1, .5e-6);**

IntSum: evalhf failed. Using slower technique.

DoIt: fsolve failed. Using cruder technique.

DoIt: Using 128 terms

DoIt: Evalhf successful for the partial sum.

2.294115538

> **Sum(A(n), n= 1..infinity);**

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt{n^3+1}}$$

> **value(%);**

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt{n^3+1}}$$

> **evalf(%);**

We got 6 digits.

Example 3: Maple cannot even approximate

Verify that $\sum_{n=2}^{\infty} \frac{1}{n \ln(n)^2}$ converges and approximate it to 6 decimal places.

Solution:

Clearly the terms determine a continuous positive decreasing function.

> **A:= n-> 1/(n*ln(n)^2);**

$$A := n \rightarrow \frac{1}{n \ln(n)^2}$$

> **int(A(x), x= 2..infinity);**

$$\frac{1}{\ln(2)}$$

> **evalf(sum(A(n), n= 2..infinity));**

$$\sum_{n=2}^{\infty} \frac{1}{n \ln(n)^2}$$

Maple can't do it either exactly or approximately . But IntSum can do it very quickly.

> **IntSum(A, 2, .5e-6);**

IntSum: Used 93 terms

2.109742639

Even if we do not ask Maple for much accuracy...

> **evalf[3](sum(A(n), n= 3..infinity));**

$$\sum_{n=3}^{\infty} \frac{1}{n \ln(n)^2}$$

...it still can't do it. Maple's procedures for numerical summation are not as sophisticated as its procedures for numerical integration.

>

Miscellaneous examples with no commentary

> **A:= n-> sin(1/n^2)/n^2;**

$$A := n \rightarrow \frac{\sin\left(\frac{1}{n^2}\right)}{n^2}$$

> **int(A(x), x= 1..infinity);**

$$\frac{1}{2} \text{FresnelS}\left(\frac{\sqrt{2}}{\sqrt{\pi}}\right) \sqrt{2} \sqrt{\pi}$$

> **sum(A(n), n= 1..infinity);**

$$\sum_{n=1}^{\infty} \frac{\sin\left(\frac{1}{n^2}\right)}{n^2}$$

> **evalf(%);**

0.9231167068

> **IntSum(A, 1, .5e-6);**

IntSum: evalhf failed. Using slower technique.

DoIt: Using 13 terms

DoIt: Evalhf successful for the partial sum.

0.9231165827

> **A:= n-> sin(1/n)*ln(n)/n^2;**

$$A := n \rightarrow \frac{\sin\left(\frac{1}{n}\right) \ln(n)}{n^2}$$

> **int(A(x), x= 1..infinity);**

$-\text{Ci}(1) + \gamma$

> **sum(A(n), n= 1..infinity);**

$$\sum_{n=1}^{\infty} \frac{\sin\left(\frac{1}{n}\right) \ln(n)}{n^2}$$

> **evalf(%);**

$$\sum_{n=1}^{\infty} \frac{\sin\left(\frac{1}{n}\right) \ln(n)}{n^2}$$

> **IntSum(A, 1, .5e-6);**

IntSum: evalhf failed. Using slower technique.

DoIt: Using 33 terms

DoIt: Evalhf successful for the partial sum.

0.1934138051

> **A:= n-> 2^(-n^2);**

$$A := n \rightarrow 2^{(-n^2)}$$

> **sum(A(n), n= 0..infinity);**

$$\sum_{n=0}^{\infty} 2^{(-n^2)}$$

> **evalf(%);**

1.564468414

> **IntSum(A, 0, .5e-8);**

IntSum: Used 5 terms

1.564468414

> **A:= n-> arctan(n^2)*n/(1+n^4);**

$$A := n \rightarrow \frac{\arctan(n^2) n}{1 + n^4}$$

> **int(A(x), x= 1..infinity);**

$$\frac{3 \pi^2}{64}$$

> **sum(A(n), n= 1..infinity);**

$$\sum_{n=1}^{\infty} \frac{\arctan(n^2) n}{1 + n^4}$$

> **evalf(%);**

0.6632792947

> **IntSum(A, 1, .5e-6);**

IntSum: Used 27 terms

0.6632791472

> **A:= n-> 1/(n^4+1)^(1/3);**

$$A := n \rightarrow \frac{1}{(1+n^4)^{\left(\frac{1}{3}\right)}}$$

> **int(A(n), n= 0..infinity);**

$$\frac{1}{4}B\left(\frac{1}{12}, \frac{1}{4}\right)$$

> **sum(A(n), n= 0..infinity);**

$$\sum_{n=0}^{\infty} \frac{1}{(1+n^4)^{\left(\frac{1}{3}\right)}}$$

> **evalf(%);**

4.385442773

> **IntSum(A, 0, .5e-6);**

IntSum: evalhf failed. Using slower technique.

DoIt: fsolve failed. Using cruder technique.

DoIt: Using 256 terms

DoIt: Evalhf successful for the partial sum.

4.385442706

>

>