This worksheet shows, how to use Pari from Maple for numerical integration.
For details check the Pari manuals, at least Chap 3.9 in users.pdf.

AVt, Nov 2006 and Feb 2007

```
> restart; kernelopts(version);
```

Maple 10.06, IBM INTEL NT, Oct 2 2006 Build ID 255401

Settings to use Pari

```
> currentdir(): theDLL:=cat(%,`\\pari_mpl.dll`);
```

This stores the location of the DLL in a string, here it is supposed exist in the directory of this worksheet.
One can use an explicite path otherwise (care for inputting because of back slashes)

theDLL := "C:_Work\DevCpp\pari_240_mpl\release\pari_mpl.dll"

```
> pari := module()
    local ModuleLoad, ModuleUnload;
    global theDLL;
    export ModuleApply;
    ModuleLoad:=   define_external(`start_Pari`,MAPLE,LIB=theDLL);
    ModuleUnload:= define_external(`stop_Pari`, MAPLE,LIB=theDLL);
    ModuleApply:=  define_external(`eval_Pari`, MAPLE,LIB=theDLL);
  end module;
```

The reason for accessing the DLL through a module with the above syntax is, that Pari is to be initialized and closed, the above should guarantee it, even if Maple is exited, please check Maple's help for 'ModuleUnload'.

For doing this explicitely within a session one defines:

```
start_pari:=    define_external(`start_Pari`,MAPLE,LIB=theDLL):
stop_pari:=     define_external(`stop_Pari`,MAPLE,LIB=theDLL):
restart_pari := define_external(`restart_Pari`, MAPLE,LIB=theDLL):
```

The last command is convenient as I do not know of an easy way to clean up variables or do a save garbage collection.

Communication between Maple and Pari is through strings (even if this means different things in C and Maple).

If an error message is given concerning the string size for input ot output, then use the following command (`buffersize_pari(100000)` allows up to 100000 characters):

```
buffersize_pari := define_external(`buffersize_Pari`, MAPLE,LIB=theDLL):
```

If an error from Pari is shown complaining the stack size, then `pari("allocatemem(x=0)");` can
be used to double the stack size.

In case of 'numerical' results from Pari one has to modify the answers to numerics in Maple, the following is a short-hand for parsing Pari's result (and easier than coding a 'string replace' function in C):

```
pp:=proc(str)                 # parse pari result
   StringTools:-SubstituteAll(str," E","E");   # remove white space
   StringTools:-SubstituteAll(%,".E",".0E"); # to cover 0
   parse(%);                  # convert string to numeric value
end proc:
```

In case one knows the answer is numeric one use the following combined command:

```
value_pari:= pp@pari:
```

For rational numbers Pari returns the fraction, either use that in Maple or send it back to Pari to evaluate the expression to a floating point number:

```
evalf_pari := proc(str)
  value_pari(cat("1.0 * ", str)); # value_pari(cat(str, " + 0."));
end proc:
```

As a variant one could have used `value_pari(cat(str, " + 0."))` as well.

In case of any error by executing Pari its kernel is stopped, an error message is printed.

```
pari := module() local ModuleLoad, ModuleUnload; export ModuleApply; global theDLL;   end module
```

>
>
>

Let us work with Pari's standard precision of 28 Digits.

The syntax for (general) numerical integration over some interval a ... b in Pari is a bit special:

`intnum(x = a, b, expr)` or `intnum(X = [a,alpha], [b,beta], expr)`

where expr is a function expression and integration is understood w.r.t. to a variable x in expr and for the 2nd version `alpha` and `beta` are the types of (possible) singularities in a and b.

The manual says, how one cares for integrable algebraic singularities in the finite case or for simple exponential singularities at infinity::

The endpoints a and b are coded as follows. If a is not at ±infinity, it is either coded as a scalar
(real or complex), or as a two component vector [a, alpha], where the function is assumed to have a
singularity of the form (x - a)^(alpha + epsilon) at a, where epsilon indicates that powers of logarithms are neglected.

In particular, [a,alpha] with alpha >= 0 is equivalent to a. If a wrong singularity exponent is used, the result will lose a catastrophic number of decimals, for instance approximately half the number of digits will be correct if alpha = -1/2 is omitted.

The endpoints of integration can be ±1, which is coded as [±1] or as [[±1], alpha]. Here alpha codes
the behaviour of the function at ±infinity as follows.
• alpha = 0 (or no alpha at all, i.e. simply [±1]) assumes that the function to be integrated tends to zero, but not exponentially fast, and not oscillating such as sin(x)/x.
• alpha > 0 assumes that the function tends to zero exponentially fast approximately as exp(-alpha*x),
including reasonably oscillating functions such as exp(-x) sin(x). The precise choice of alpha, while
useful in extreme cases, is not critical, and may be off by a factor of 10 or more from the correct value.
• alpha < -1 assumes that the function tends to 0 slowly, like x^alpha. Here it is essential to give

the correct alpha, if possible, but on the other hand alpha < -2 is equivalent to alpha = 0, in other words to no alpha at all.

Two more codes are reserved for oscillating functions ...

Note.
If you do not like the code [±1] for ±1, you are welcome to set, e.g oo = [1] or INFINITY = [1], then using +oo, -oo, -INFINITY, etc. will have the expected behaviour.

For more and details please check the Pari manual.

```
> Digits:=28; # now set precision for Pari
  cat("default(realprecision,",  convert(Digits,string),  ")"): pari(%):
```

$$\text{Digits} := 28$$

Example 1: a simple integral

```
> F := unapply( exp(-x^2), x);
  #'Int(F(x), x=0 .. infinity)': '%'= value(%);
  'Int(F(x), x=-1 .. 1)': '%'= value(%);
  rhs(%): '%'= evalf(%);
```

$$F := x \to e^{(-x^2)}$$

$$\int_{-1}^{1} F(x)\,dx = \text{erf}(1)\sqrt{\pi}$$

$$\text{erf}(1)\sqrt{\pi} = 1.4936482656248540507989348 72$$

Define the according function expression in Pari (a function would work as well I think) and do it:

```
> 'pari("f(x) = exp(-x^2)")';
  eval(%): # to execute the command in Pari
  value_pari("intnum(x = -1, 1, f(x))");
```

$$\text{pari}(\text{"f(x) = exp(-x^2)"})$$

$$1.4936482656248540507989348 72$$

Example 2: an improper integral with removable singularity in 0, Maple knows the symbolic answer:

```
> F := unapply(1/(exp(x)-1) - exp(-x)/x, x); #plot( F(x),x=0 .. infinity);
  'Int(F(x), x=0 .. infinity)': '%'= value(%);
  rhs(%): '%'= evalf(%);
```

$$F := x \to \frac{1}{e^x - 1} - \frac{e^{(-x)}}{x}$$

$$\int_{0}^{\infty} F(x)\,dx = \gamma$$

$$\gamma = 0.5772156649015328606065120901$$

To denote infinity we use the suggestion above:

```
> pari("inf = [1]");
```

$$\text{"[1]"}$$

Define the according function expression in Pari and integrate (with beta = 1):

```
> convert(F(x), string): cat("f(x) = ", %, ""): 'pari(%)'; eval(%): # not as
  function
  'value_pari("intnum(x = 0, [inf,1], f(x))")': '%'=%;
```

pari("f(x) = 1/(exp(x)-1)-exp(-x)/x")

value_pari("intnum(x = 0, [inf,1], f(x))") = 0.57721566490153286060651209 02

The syntax for (general) numerical integration in Pari looks a bit strange:
intnum(x = a, b, expr, {tab}) or intnum(X = [a,alpha], [b,beta], expr, {tab})

Example 3: a well-known algebraic boundary singularity

```
> F := x -> 1/sqrt(1 - x^2); # plot(F(x), x=0 ..1);
  'Int(F(x), x=0 .. 1)': '%'= value(%);
  rhs(%): '%'= evalf(%);
  ``;
  convert(F(x), string): cat("f(x) = ", %, ""): 'pari(%)'; eval(%):
```

$$F := x \to \frac{1}{\sqrt{1 - x^2}}$$

$$\int_0^1 F(x)\,dx = \frac{\pi}{2}$$

$$\frac{\pi}{2} = 1.5707963267948966192313216 92$$

pari("f(x) = 1/(1-x^2)^(1/2)")

Without regarding the singularity the answer is disappointing, otherwise it is fine:

```
> 'value_pari("intnum(x = 0, 1, f(x))")': '%'=value(%);
  'value_pari("intnum(x = 0, [1,-1/2], f(x))")': '%'=value(%);
```

value_pari("intnum(x = 0, 1, f(x))") = 1.5707963267948966192244570 14

value_pari("intnum(x = 0, [1,-1/2], f(x))") = 1.5707963267948966192313216 92

The singularity can be determined using Maple's commands 'series' or 'asympt':

```
> `Example 3:`;
  series(F(x), x=1, 2): subs(x-1=X,%): convert(%,polynom);
  ``;
  `Example 2:`;
  MultiSeries[asympt]( 1/(exp(x)-1) - exp(-x)/x, x): convert(%,polynom);
```

Example 3:

$$\frac{\frac{-1}{2} I \sqrt{2}}{\sqrt{X}}$$

Example 2:

$$\frac{1-\frac{1}{x}}{e^{x}}$$

Example 4: the boundary singularity becomes worse

```
> F := x -> 1/(1 - x^2)^(8/9); ``;
  'Int(F(x), x=0 .. 1)': '%'= value(%);
  rhs(%): '%'= evalf(%);
```

$$F := x \to \frac{1}{(1-x^2)^{(8/9)}}$$

$$\int_0^1 F(x)\,dx = \frac{1}{2}B\left(\frac{1}{9},\frac{1}{2}\right)$$

$$\frac{1}{2}B\left(\frac{1}{9},\frac{1}{2}\right) = 5.158322793389998423788684375$$

```
> convert(F(x), string): cat("f(x) = ", %); pari(%):

  value_pari("intnum(x = 0, 1, f(x))");
  value_pari("intnum(x = 0, [1,-1/2], f(x))");
  st:=time():
  value_pari("intnum(x = 0, [1,-8/9], f(x))");
  `seconds needed` =time() - st;
```

"f(x) = 1/(1-x^2)^(8/9)"

5.158173470708001645

5.158322788812406322

5.158322793389998423788684376

seconds needed = 0.032

The answer may be useless if the wrong type of singularity is used, otherwise the answer is quick & exact.

Note that in this case a simple call for Maple's would do as well:

```
> Int(F(x), x= 0..1); evalf(%);
```

$$\int_0^1 \frac{1}{(1-x^2)^{(8/9)}}\,dx$$

5.158322793389998423788684376

Example 5: the boundary singularity is bad and we want higher precision

```
> Digits:=60;
  cat("default(realprecision,",  convert(Digits,string), ")"): pari(%);
```

Digits := 60

"60"

```
> F := x -> 1/(1 - x^2)^(11/12);
  'Int(F(x), x=0 .. 1)': '%'= value(%);
  rhs(%): '%'= evalf(%);
```

$$F := x \to \frac{1}{(1-x^2)^{\left(\frac{11}{12}\right)}}$$

$$\int_0^1 F(x)\,dx = \frac{1}{2}B\left(\frac{1}{12}, \frac{1}{2}\right)$$

$$\frac{1}{2}B\left(\frac{1}{12}, \frac{1}{2}\right) = 6.666475811899243400746756031781832857100379253427732516110 00$$

```
> convert(F(x), string): cat("f(x) = ", %); pari(%):
  st:=time():
  value_pari("intnum(x = 0, [1,-11/12], f(x))");
  `seconds needed` =time() - st;
```

"f(x) = 1/(1-x^2)^(11/12)"

6.666475811899243400746756031781832857100379253427732516109 99

seconds needed = 0.157

In this case a simple call for Maple's numerics fails, I canceled after considerable time:

```
> st:=time():
  Int(F(x), x= 0..1); evalf(%);
  `seconds needed` =time() - st;

  #timelimit(113, evalf(Int(F(x), x= 0..1))); # restrict CPU time to 113
  seconds
```

$$\int_0^1 \frac{1}{(1-x^2)^{\left(\frac{11}{12}\right)}}\,dx$$

```
Warning, computation interrupted
```

Example 6: an elliptic integral

This had to be done more in a more carefull, but similar way and and one can increase the working precision.

```
> restart_pari();
  Digits:=40;
  #cat("default(realprecision,",  convert(Digits,string), ")"): pari(%);
  cat("default(realprecision,",  convert( round(3/2*Digits),string), ")"):
  pari(%);
PARI 2.3.0 restarted and initialized.
```

1

Digits := 40

"0"

```
> kappa:='kappa':
  F := x -> 1/((1-x^2)^(1/2)*(1-kappa^2*x^2)^(1/2)); ``;
  'Int(F(x), x=0 .. 1)': '%'= value(%);
```

$$F := x \to \frac{1}{\sqrt{1-x^2}\sqrt{1-\kappa^2 x^2}}$$

$$\int_0^1 F(x)\,dx = \text{EllipticK}(\kappa)$$

The singularities are $1, -1, -\frac{1}{k}, \frac{1}{k}$. Let us choose a somewhat arbitrary value

```
> kappa:=RandomTools[Generate](float(range=1 .. 2, digits=16));
  kappa:=RandomTools[Generate](float(range=1 .. 2));
  kappa_float:=%:
```

$$\kappa := 1.646213512988971$$

$$\kappa := 1.32582269041177566252297003245809033906 6$$

```
> #kappa:=1.22590085916034269683911071437741536363 7;
> 'EllipticK(kappa)': '%'= evalf(%);
```

$$\text{EllipticK}(\kappa) = 1.44612417627053293668085828380456278323 8 - 1.35744594880566659588906456571065751322 1\,I$$

Now split the integration at the singularity as $\int_0^{\frac{1}{k}} F(x)\,dx + \int_{\frac{1}{k}}^1 F(x)\,dx$ to compare it with Maple's exact answer through the special function EllipticK:

```
> convert(kappa, string): cat("kappa = ", %); pari(%):
  convert(F(x), string): cat("f(x) = ", %); pari(%):
  ``;
  st:=time():
  value_pari("intnum(x = 0, [1/kappa,-1/2], f(x))")
  + value_pari("intnum(x = [1/kappa,-1/2], [1,-1/2], f(x))");
  `seconds needed` =time() - st;
```

"kappa = 1.32582269041177566252297003245809033906 6"

"f(x) = 1/(1-x^2)^(1/2)/(1-1.75780580641071913329503182669217619812 4*x^2)^(1/2)"

$$1.44612417627053293667444305678 - 1.35744594880566665958740191116\,I$$

$$\text{seconds needed} = 0.125$$

This is not so bad, but far from convincing (for an automatic routine, Pari allows some tweaks for that), but Maple does not find an answer with reasonable time:

```
> 'Int(F(x), x=0..1/kappa) + Int(F(x), x=1/kappa ..1)';
  timelimit(23, evalf(%)); # restrict CPU time to 23 seconds
```

$$\int_0^{\frac{1}{\kappa}} F(x)\,dx + \int_{\frac{1}{\kappa}}^1 F(x)\,dx$$

```
Error, (in convert/real_rat) time expired
```

To make it work, one can use rationals:

```
> kappa:= convert(kappa, rational);
  'F(x)': '%'=%;
```

$$\kappa := \frac{23767137129129035634}{17926331553239150389}$$

$$F(x) = \frac{1}{\sqrt{1-x^2}\sqrt{1-\frac{5648768073148239778566275597668417819 56\,x^2}{32135336295665757013754304114255885 1321}}}$$

The both Maple and Pari find the correct value:

```
>
> st:=time():
  'Int(F(x), x=0..1/kappa) + Int(F(x), x=1/kappa ..1)': '%'=evalf(%);
  `seconds needed` =time() - st;
```

$$\int_0^{\frac{1}{\kappa}} F(x)\,dx + \int_{\frac{1}{\kappa}}^{1} F(x)\,dx =$$

1.4461241762705329366808582838045627832 41 − 1.3574459488056665958890645657106575132 22 I

seconds needed = 8.922

```
> 'EllipticK(kappa)': '%'= evalf(%);
```

$$\text{EllipticK}(\kappa) =$$

1.4461241762705329366808582838045627832 40 − 1.3574459488056665958890645657106575132 21 I

```
> convert(kappa, string): cat("kappa = ", %); pari(%):
  convert(F(x), string): cat("f(x) = ", %); pari(%):
  ``;
  st:=time():
  value_pari(
    "intnum(x = 0, [1/kappa,-1/2], f(x)) + intnum(x = [1/kappa,-1/2], [1,-1/2],
  f(x))"
  );
  `seconds needed` =time() - st;
```

"kappa = 23767137129129035634/17926331553239150389"

"f(x) = 1/(1-x^2)^(1/2)/(1-5648768073148239778566275597668417819 56/32135336295665757013754304114\
2558851321*x^2)^(1/2)"

1.4461241762705329366808582838 − 1.3574459488056665958890645657 I

seconds needed = 0.188

```
>
```