Subject: New package MathTools
Posted by koldo on Sun, 27 Dec 2015 10:38:16 GMT
View Forum Message <> Reply to Message

Many times I see scientists and engineers using expensive over bloated tools like Matlab and Diadem for very simple issues. Even open source options like Scilab or Octave are big applications and some activities are not very evident, complex to be done.

New package MathTools includes simple math tools for scientists and engineers.

- Calculator is a text calculator that handles variables and some mathematical functions.
- Regression permits to paste numerical series and automatically searches for the equation that fits best.

These are utilities that I use frequently and I expect they could be useful for you.

Many improvements are possible always depending on the scarce time availability

If anybody may want to add new features, all of them and even new applications related with the subject will be acknowledged.

Subject: Re: New package MathTools
Posted by deep on Sun, 27 Dec 2015 13:22:56 GMT
View Forum Message <> Reply to Message

Hi Koldo,

I liked the concept.
I would like to add custom constants.

I tried to add it.

gridConstants.Add("DegToRad",M_PI/180.0);
Constants are not accessed for calculation from ArrayControl. It is internal to ScatterDraw/Equation.cpp

eval.constants.Add("DegToRad",M_PI/180.0);
'Upp::EvalExpr::constants': cannot access private member
    declared in class 'Upp::EvalExpr'

After looking at the code the I found constants are defined in ScatterDraw/Equation.cpp

Any hints on how have custom constants.

Subject: Re: New package MathTools
Posted by koldo on Sun, 27 Dec 2015 19:54:21 GMT

Easy solution. Now constants and functions are public :)

Subject: Re: New package MathTools
Posted by peterh on Mon, 22 Apr 2019 10:19:24 GMT

Hello I have a question about Mathtools:

It was initially my intention to tinker with it and add a "||" operator, which is really useful for electronics.
(Rp = R1*R2/(R1+R2) or Rp=1 /(1/R1 + 1/R2). If resistors are zero, then division by zero happens.)

When I input an expression in the calculator that divides by zero I get an exception.
When I restart the program, I cannot longer create variables.
This error persists even after I deleted Mathtools.json and even after rebooting.
I tried this on two different computers with the unmodified source and so I am pretty sure about this.

So the program must store some persistent data elsewhere. Where should I look?
I tried with the debugger but so far without success.

Subject: Re: New package MathTools
Posted by koldo on Tue, 23 Apr 2019 07:35:39 GMT

Thank you peterh

In fact MathTools had a problem that avoid to create new variables, by itself (nothing related with serialization) :roll:
I think all is fixed now. In addition , operator || is added. :)

Subject: Re: New package MathTools
Posted by peterh on Tue, 23 Apr 2019 08:47:39 GMT

I thank you very much! This makes the program very useful for me and it it will be the only calculator I use in future! :)

Is it possible for me to understand the problem?
Because the problem was persistent even after reboot, I conclude, the program stores some persistent information, on disk or in registry or in temp.
I did not find this in the source and temp should be deleted after reboot, so I currently dont understand the problem.

Thanks and regards,

Peter

Subject: Re: New package MathTools
Posted by koldo on Tue, 23 Apr 2019 09:43:09 GMT
View Forum Message <> Reply to Message

Hi Peterh

The problem was in calculator.cpp, EvalExpr2::Eval2(). When an assignment was entered, new variable is checked in line 140 if it is a constant in IsConstant(sid). Before the fix, IsConstant() always returned true. :roll:

Subject: Re: New package MathTools
Posted by peterh on Tue, 23 Apr 2019 10:11:50 GMT
View Forum Message <> Reply to Message

Thank you!
This program is beyond my prograqmming skills, but I nevertheless try to understand it. - partially
Anyway, variables worked before I had input 1/0. :roll:

Subject: Re: New package MathTools
Posted by koldo on Wed, 24 Apr 2019 05:29:19 GMT
View Forum Message <> Reply to Message

Mysteries may exist :twisted:
Either way, they've been resolved :)
If you have any idea for a new "math" "tool", we may include it.

Subject: Re: New package MathTools
Posted by peterh on Wed, 24 Apr 2019 08:31:28 GMT

Hi Koldo,

I got calculator.cpp and Mathtools.lay from the GIT website.
There where only minor modifications.

It compiled, but the behavior was unchanged.
I think GIT is not up to date yet.

Are the modifications today or tomorrow in nightly build?

Edit:

I downloaded nightly build and first test where very positive :)
I always wanted a calculator with parallel operator and now I have it.
Thank you so much!!!

Ok, at second look there is a small problem leftover:

0 || 0 should giv 0 as result and not division by zero error. 8)

All the best,

Peter

Subject: Re: New package MathTools
Posted by peterh on Wed, 24 Apr 2019 11:29:12 GMT

Results of stress-test:

1||0 = 0
0||0 Error Division by zero
-1||0 Error Division by zero
-1 || -1 Error Division by zero
1||1 = 0.5

If (arg1 * arg2) = 0 then the result is 0, else the result is arg1*arg2/(arg1+arg2). :roll:

And yes, there are negative resistors.

A negative resistor produces energy and gets cold when it does so. :d
Unfortunately it is not available as discrete component. It stops working when it gets cold, that is
the technological problem and causes resistor noise. :d :d

I am kidding :d

To be serious:
For example the input impedance of an amplifier with positive feedback can be negative.

Negative resistors can exist in active circuits only as a mathemathik intermediate result.
This means only two complementary resistor values should lead to division by zero e.g. -1||1 = infinite resistance = division by zero.

Subject: Re: New package MathTools
Posted by peterh on Sat, 27 Apr 2019 16:35:02 GMT
View Forum Message <> Reply to Message

Hi Koldo,

I modified the calculation a little bit:

```
 void ResParallel(const doubleUnit &d) {
//  if (val + d.val < 1e-100)
//   throw Exc(t_("Division by zero"));
  if (!(unit.IsEqual(d.unit) || IsNull(unit) || IsNull(d.unit)))
   throw Exc(t_("Units does not match in resistor parallel"));
  if (abs(val*d.val) < 1e-100)
   val = 0.0;
  else
   if (abs(val + d.val) < 1e-100)
    throw Exc(t_("Division by zero"));
   val = val*d.val/(val + d.val);
 }
```

It delivers now these results (which are correct):

1||1 = 0.5
1||0 = 0
0||0 = 0
-2||1 = 2
-1||2 = -2
-1||1 Error Division by zero

Please note, the "parallel" operator is not only useful for electronics.
There are many physic equations where this is useful.
For example, two elastic springs may have the elasticity konstants k1 and k2.

If the springs are in parallel then the total konstant is k1 + k2.

If the springs are in series then the total konstant is k1 || k2.

Some mathematicians might complain, that 0^2 / 0 = 0 is not allowed.
In physics this is allowed because the numbers have dimensions.

0^2 Ohm^2 is in a different dimension as 0^1 Ohm^1.
Ohm^2 and Ohm is as different as m^2 and m^1 and cannot been compared, added or subtracted.
The other point is: there is no absolute zero in physics, there is only zeropoint noise ;)

Therefore this calculation is perfectly legal in physics. ;)

Again, many thanks,

Peter

Subject: Re: New package MathTools
Posted by koldo on Sat, 27 Apr 2019 20:12:16 GMT
View Forum Message <> Reply to Message

Thank you Peterh

It is now included.

Just for you to know, you can do this. If text is not detected as part of the expresion, it is considered a comment and it is indented to the right as indicated in "Tab chars"

File Attachments
1) Captura.JPG, downloaded 742 times

Subject: Re: New package MathTools
Posted by peterh on Mon, 29 Apr 2019 06:46:09 GMT
View Forum Message <> Reply to Message

Thats a good feature.

Rp = Rx || R1
Rx is unknown. Follows:
Rx = Rp || -R1
Its now at my fingertips.
Thank you very much!