Subject: Scatter: FEATURE IDEAS (1 already done)
Posted by kohait00 on Sun, 07 Mar 2010 11:31:50 GMT
View Forum Message <> Reply to Message

hi andrej

i more and more get warmer with the Scatter, really like it and the interface. had developped own diagram control, but yours is far better i will try to provide some ideas though, to maybe enhance functionality of Scatter and if i can also the solutions. then you decide..

1) attached value popup
when clicking into diagram, a popup appears, and remains static, while moving clicked to other locations. better to have the popup travel with mouse showing values, isn't it? best, to have it selectable, which behaviour to have. attached a solution, maybe you like it..it does not have it swicthable though, but it works.

2) zooming / moving
when zooming out extremely (in tab8 i.e) the scatter hangs somewhere. the zoom is always realizes/painted respect to the SetXYMin Values. better would be to realize zoom centered to the current mous position, which is not that easy, i know, had my struggles . also logrithmic stuff is hard to handle as hell. maybe the problem is that the frature of drawing funcitons stands in the way..

3) marks drawing
it would be cool to have the drawing reticle marks selected dynamicly according to zoomfactor also, they are already accrding to move position, but zooming zooms also the reticle.
i.e devisions -100 .. 50 .. 0 .. 50 .. 100 would step by step be changed to -75 .. -25 .. 25 .. 75 then -50 .. 25 .. 0 .. 25 .. 50
may be i can provide you a testcase with my diagram in this sense, it adapts the graph marks when moving and zooming

4) having graphs accessible via VectorMap, instead of Vector
i saw that one can access the data directly, but the data is organized in Vector< Vector<XY> >, which is ok, but when users dynamicly will want to access and modify same things, while other graphs are removed, the index is changed, and one needs to keep trak of indices in a map. wo why not using a map directly. the performance hit is neglectable anyway. this would hide away the index problem.

5) moving points of graphs
one of the goals in my diagram was to have it a sort of controling control, where one can drag some points of graph around, while some not, so i.e realizing parametrize of a eq filter bank or the like.

now these are some of the ideas i had in mind in my diagram. maybe you will find some of them usefull in case of help needed, ask..

File Attachments

1) Scatter.rar, downloaded 597 times

Subject: Re: Scatter: FEATURE IDEAS (1 already done)
Posted by koldo on Wed, 17 Mar 2010 21:43:10 GMT
View Forum Message <> Reply to Message

Hello Kohait00

Believe me. I have not seen your post before

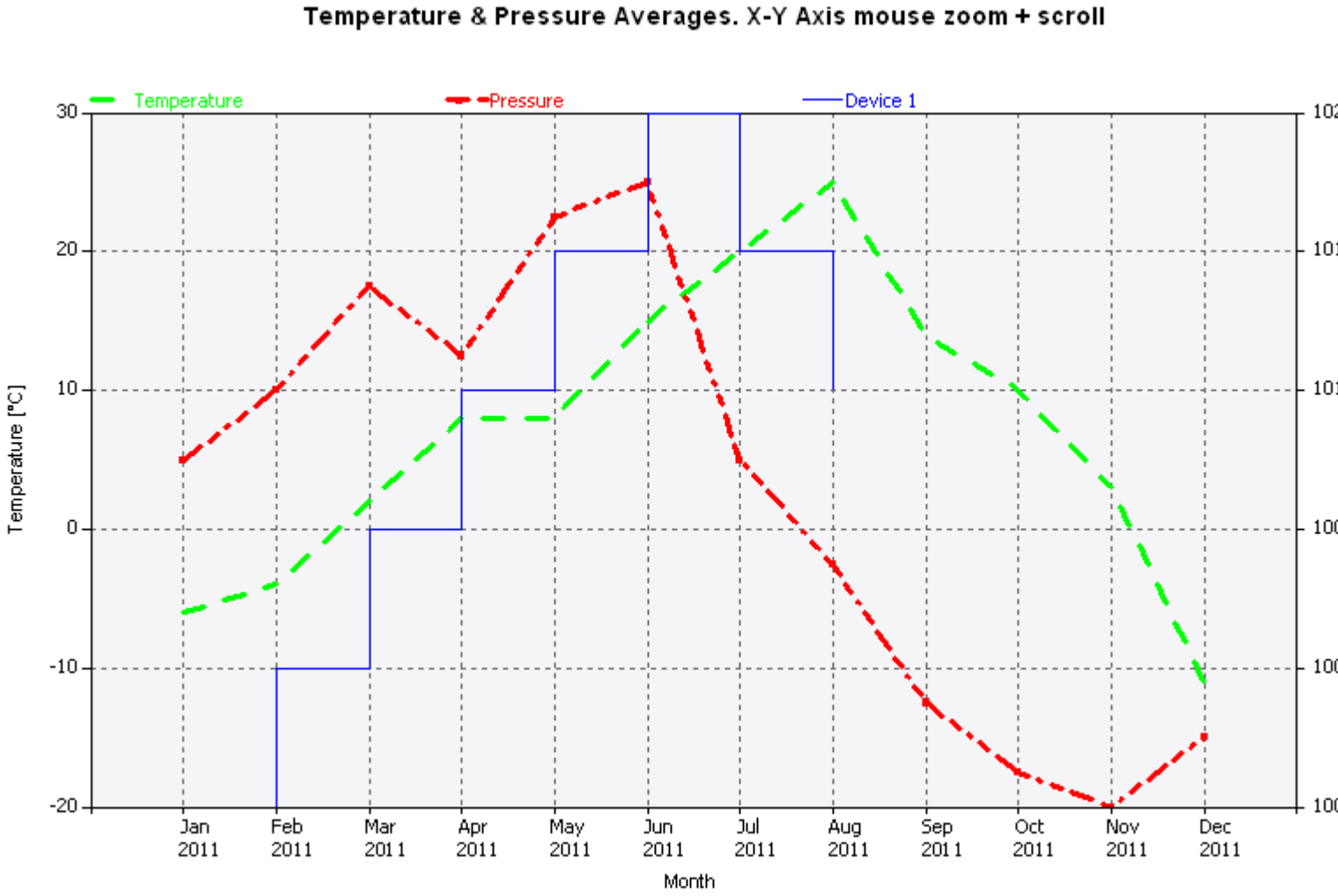
I have just uploaded the last Scatter version thanks to Andrei-Catalin. For now this is the last change from my side:

It includes line styles that can be easily defined by user, x axis legend can have more than one line, and precision has been improved (thanks to use of fround() instead of int() to round double into integer). There are also functions that limit the max and min zoom.

You have included a lot of ideas. Let's wait Andrei to know his opinion. I repeat that it is enough for me now, but new features could be interesting and if necessary I could help too.

File Attachments

1) dib.PNG, downloaded 1990 times



Subject: Re: Scatter: FEATURE IDEAS (1 already done)
Posted by andrei-catalin on Thu, 18 Mar 2010 19:36:42 GMT
View Forum Message <> Reply to Message

Certainly the number of options for a chart can be much higher. There are many aspects that could be customized. For now, Scatter can be considered only a starting point or a basic package. A list of new features that you want and concrete ideas would be formed. But I think that should be developed a new charting package, better structured and which to use painter.

Andrei

Subject: Re: Scatter: FEATURE IDEAS (1 already done)
Posted by tojocky on Fri, 19 Mar 2010 08:17:35 GMT
View Forum Message <> Reply to Message

I agree with you Andrei!
andrei-catalin wrote on Thu, 18 March 2010 21:36Certainly the number of options for a chart can be much higher. There are many aspects that could be customized. For now, Scatter can be considered only a starting point or a basic package. A list of new features that you want and concrete ideas would be formed. But I think that should be developed a new charting package, better structured and which to use painter.

Andrei

Subject: Re: Scatter: FEATURE IDEAS (1 already done)
Posted by koldo on Mon, 12 Apr 2010 16:57:15 GMT
View Forum Message <> Reply to Message

tojocky wrote on Fri, 19 March 2010 09:17I agree with you Andrei!
andrei-catalin wrote on Thu, 18 March 2010 21:36Certainly the number of options for a chart can be much higher. There are many aspects that could be customized. For now, Scatter can be considered only a starting point or a basic package. A list of new features that you want and concrete ideas would be formed. But I think that should be developed a new charting package, better structured and which to use painter.

Andrei

Hello all

I think Scatter works rather well for now with actual design. Of course there is a lot of space for improvements:

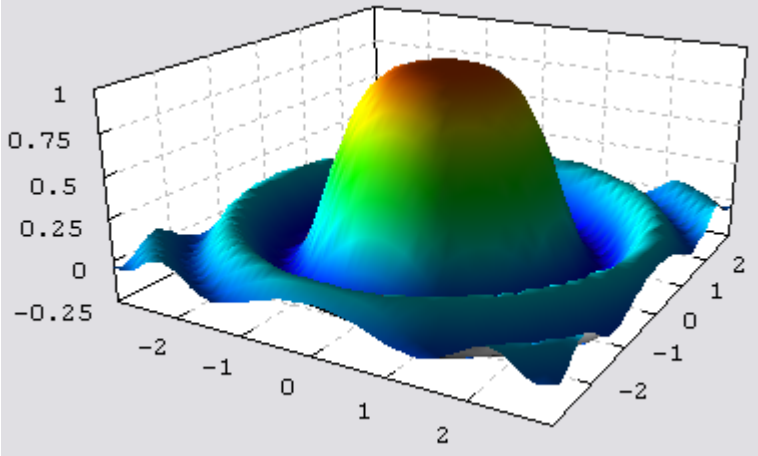
- Some of kohait00 ideas
- Add menus in Scatter class (zoom in/out, fit, ...)

- Implement new features in layout designer
- More graphs like 2D bars
- Documentation
- Perhaps to add a second name for Scatter class, like "Plot2D"

However, if I would have 20 hours for it I would begin a new Plot3D package to get things like this:

File Attachments

1) dib.PNG, downloaded 1969 times



Subject: easying the API
Posted by kohait00 on Tue, 11 May 2010 10:37:06 GMT
View Forum Message <> Reply to Message

hey all,

having had a look at the API again, i was wondering why all the Vector<> stuff hasnt been packed into a struct or something to have all info combined, and to let the data be publically accessible, so user can modify data directly, trigger a refresh and be happy, the interface easily would shrink to half of current, what do you think?

@koldo: how far is your Plot3D stuff

Subject: capsulated Scatter Vectors
Posted by kohait00 on Tue, 11 May 2010 13:07:24 GMT
View Forum Message <> Reply to Message

hey guys, trying to simplify the API, here comes the first shot

it simply combines all the Vectors used for a Graph unit into one struct.

i commented out what is not nessesary anymore so that you see what could be left out. tried not to touch what is not nessesary.
even the names have been kept the same, which could be accomodated to be better variable names when leaving the Graph Vectors public.

it is a current svn compare (with my previous movable Popup fix, see above).. so you can use svn diff.

comments please
cheers

File Attachments

1) Scatter.rar, downloaded 592 times

Subject: Re: easying the API
Posted by koldo on Tue, 11 May 2010 13:50:28 GMT
View Forum Message <> Reply to Message

Hello kohait00

Quote:having had a look at the API again, i was wondering why all the Vector<> stuff hasnt been packed into a struct or something to have all info combined

I agree. I did not touch it because I did not wanted to touch original andrei implementation and this is not an important issue in package development.

In fact I was thinking on other areas for radically improving speed with huge data sets.

Quote:@koldo: how far is your Plot3D stuff

Ah, it was not my intention to do it. If you have time you can do it .

Subject: Re: capsulated Scatter Vectors
Posted by koldo on Tue, 11 May 2010 14:12:42 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 11 May 2010 15:07hey guys, trying to simplify the API, here comes the first shot
it simply combines all the Vectors used for a Graph unit into one struct.

i commented out what is not nessesary anymore so that you see what could be left out. tried not to touch what is not nessesary.

even the names have been kept the same, which could be accomodated to be better variable names when leaving the Graph Vectors public.

it is a current svn compare (with my previous movable Popup fix, see above).. so you can use svn diff.

comments please
cheers

Hello kohait00

I do not feel very comfortable with the changes.

You have included inside source code a Scatter.rar file with a different version of some files. For example, for Scatter.cpp there are three version:
- The original
- The one in Scatter.rar
- The one in Scatter.rar inside Scatter.rar

In other side, there are a lot of internal changes. However there is no additional features for the user so I do not now if it is worthwhile.

For me it is important to know Andrei's opinion. Meanwhile I will include ProcessPopUp() that I think it is a nice feature.

Subject: Re: capsulated Scatter Vectors
Posted by kohait00 on Tue, 11 May 2010 14:33:28 GMT
View Forum Message <> Reply to Message

hey koldo, thx for replying

forget about the Scatter.rar in Scatter.rar. it was a remamanet from last change which i forgot to delete when making the new archive.

the topmost stuff is the current.

i agree with changing API not beeing a good idea. but during time i really got to apriciate a simple API, where not too much is beeing hidden from user, actually thinking, that the data comes from user, and user decides what to do with it, beeing the scatter graph only menas of visualizing them at some time. the Vector Containers as members are pretty cool capsulation of data handling on their own. so why IMHO unnessesarely exporting add/remove data extra.

the whole thing is driven by the idea of keeping controls simple. to use and understand, and concerning their visualisation, beeing still in control of all (Refresh() especially.

this was just an idea anyway, which i wanted to provide for anyone in need..i will continue to merge some of my diagram stuff, which is not that performant, but has some of the "complicated" features described initially.

@Plot3D, this thing is cool, but I have no time to do it one very fast ends up implementing own 3d engine, transform matrixes, viewpoints, world matrixes and all that (depending on rotation, translation and all of that). isnt there any lib already available for that (i mean computing 3d stuff in software, besides Mesa/DRI)?

another question: why the seperation between Parameters data points and Functional plots data points? couldnt they be managed all at in one? or is it a performance issue?

Subject: Re: capsulated Scatter Vectors
Posted by koldo on Tue, 11 May 2010 14:53:59 GMT
View Forum Message <> Reply to Message

Quote:@Plot3D, this thing is cool, but I have no time to do it one very fast ends up implementing own 3d engine, transform matrixes, viewpoints, world matrixes and all that (depending on rotation, translation and all of that). isnt there any lib already available for that (i mean computing 3d stuff in software, besides Mesa/DRI)?

Hello kohait00

If I would do it I would implement it in OpenGL. U++ has an OpenGL demo that can be a very basic base.

Subject: Re: capsulated Scatter Vectors
Posted by dolik.rce on Tue, 11 May 2010 15:17:16 GMT
View Forum Message <> Reply to Message

koldo wrote on Tue, 11 May 2010 16:12However there is no additional features for the user so I do not now if it is worthwhile.
Hi,

Just my two cents: I agree with kohait that having direct access to the data is IMHO a nice feature. Plots in PlotCtrl (which I still hope to finish during summer) actually has Vector as its public base class. I believe that it is faster and simpler and it also allows in some cases to apply some general algorithms (e.g. Sort) on the data without pushing them back and forth.

Also I believe that the API doesn't have to change, just extend. If you keep the old functions and just add the direct access to data, there should be no problems with backward compatibility.

Sorry if I didn't understand kohaits idea exactly, I just quickly scanned through the sources he posted...

Best regards,
Honza

Subject: Re: capsulated Scatter Vectors
Posted by kohait00 on Tue, 11 May 2010 20:16:51 GMT
View Forum Message <> Reply to Message

thanks guys for participating, its fun developing like that

here comes a version where i tried to remove the commented unnecessary stuff and have unified the usage of dthe Plot Data, merging the 2 cases of a given values plot and the drawing functions plot. it works just the same, and as benefit, the interface becomes more straightforward.

API:
i agree with keeping API compatible to old stuff. but the problem here is that the functions for adding data plots and function plots don't match syntactically.

porting is actually an easy thing to do, if accompishing to name the public members of the class well, this woule encourage (enforce) the users of Scatter to quickly port their app.

another way could be to hide the new implementatio in own namespace. Something like SlimScatter::Scatter, so a user later can decide about the porting point easily.. an to make the real Scatter a separate wrapper class, that keeps the old API but only translates it to the new Scatter, which would keep out all the compatibility thing from the new Scatter

something like

```
namespace SlimScatter
{
class Scatter
{
//new clean implementation
}
}

class Scatter : public SlimScatter::Scatter
{
//wrapping functions only
}
```

@kolo: OpenGL ofcorse is best maybe i take a look in there...

File Attachments

1) Scatter2.rar, downloaded 541 times

Subject: Re: capsulated Scatter Vectors
Posted by koldo on Wed, 12 May 2010 20:29:48 GMT
View Forum Message <> Reply to Message

Hello Kohait00

I have reviewed your code and it is difficult for me to find significant advances over the old one. In fact many times the old code from user point of view has less lines and follows more U++ style.

An area easy to implement and very advantageous would be to improve data entering. Now data sets have to be entered point by point so in many times data is duplicated.

So it would be great to let linking external data sets from:

- C arrays
- ArrayCtrl columns
- GridCtrl columns

What do you think about it?

Subject: Re: capsulated Scatter Vectors
Posted by kohait00 on Wed, 12 May 2010 21:07:41 GMT
View Forum Message <> Reply to Message

Quote:
the old code from user point of view has less lines and follows more U++ style.

I totaly agree on that one. my changes have neither performance improvements nor beauty changes. the point its all about is, to define what is the core of the Scatter control. as of my point of view it is the data, not the API (which was to be read fine but was hiding away the data too much and IMHO bloatet the API a bit).

andrejs goal was to have a Scatter control with a neat API to post stuff to and forget about it. my goal is to be able to access the data of the graphs in tried and true containers' manner (Vector..) and thus public and to see the Scatter just as a wrap over the data, which, upon triggering of Refresh()/Paint() draws a representation of it. so its basicly a question of design goals.

the current solution could be improved in readability by changing the PData members' names to something more usefull, which i havent done so far (wanting to keep the visibility where the members came from and to make compare in svn easy not to change too much at once). also, the "more lines" stuff could be done better, by returning PData & instead of Scatter & AddSeries..

i will provide a cleaned up version to sea what i mean.

from code point of view, my current code leaves the control with almost only the code needed to draw the data. as api see the Vector<PData> API which is cool as well. another benefit is that functional plots are handled just the way normal plots are.. making code clearer again.

i hope this came clear..it was not my goal to enhance the api but simplify it by leaving stuff accessible instead of hiding it behind api. (this is what i learned as well that sometimes its better not to hide)

Quote:
So it would be great to let linking external data sets from:

- C arrays
- ArrayCtrl columns
- GridCtrl columns

this is a great idea.. could be done in some Additional AddSeries() functions..

BTW: OpenGL, i am starting to like it.. the only thing is to create the vertices from data, set the viewpoint and have the stuff rotate depending on mouse position as seen on a surface of a bounding sphere.

Subject: Re: capsulated Scatter Vectors
Posted by koldo on Thu, 13 May 2010 07:06:32 GMT
View Forum Message <> Reply to Message

Hello Kohait00

If the final design from user point of view has:

- The same or less lines of code
- Follows U++ style as much as possible
- With a cleaner interface and better access to data

I will try to help you to convince Andrei to approve the changes.

Do not forget that to set basic Scatter data there are three type of functions:

1. Simplest: Just assign

```
Scatter& Scatter::SetLegendWeight(const int& weight)
{
 legendWeight=weight;
```

```
 return *this;
}
```

2. Less simple: Check data and assign

```
Scatter& Scatter::H_Border(const int& poz_x)
{
 if(poz_x>=0)
  px=poz_x;
 return *this;
}
```

3. More complex: Do some actions and assign

```
Scatter &Scatter::SetMajorUnits(double ux, double uy)
{
 xMajorUnit=ux;
 yMajorUnit=uy;
 yMajorUnit2=yRange2*yMajorUnit/yRange;
 AdjustMinUnitX();
 AdjustMinUnitY();
 AdjustMinUnitY2();
 return *this;
}
```

(I like these "return *this")

This way many times to use = to assign internal data is not enough .

Subject: Re: capsulated Scatter Vectors
Posted by cbpporter on Thu, 13 May 2010 09:05:39 GMT
View Forum Message <> Reply to Message

(const int& weight)
Why?

Subject: Re: capsulated Scatter Vectors
Posted by andrei_natanael on Thu, 13 May 2010 13:31:57 GMT
View Forum Message <> Reply to Message

Why not 3 different version of Scatter? The version of Andrei Catalin will remain the same and Koldo version and Konstantin version will build up onto Catalin version. I think everyone try to achieve something different. Soon or later we'll see which idea stand. That's why we have bazaar and we're not forced to work on others code if we don't like his style or idea.
IMO a Scatter should follow MVC pattern, that would make easy to have a series of points in a

Vector and multiple views of it in Scatter (just an idea).
About API design... http://doc.trolltech.com/qq/qq13-apis.html

Andrei

Subject: Re: capsulated Scatter Vectors
Posted by andrei-catalin on Mon, 17 May 2010 17:17:59 GMT
View Forum Message <> Reply to Message

I agree with Natanael. Why not multiple versions?

Andrei-Catalin

Subject: Re: capsulated Scatter Vectors
Posted by kohait00 on Mon, 17 May 2010 19:31:02 GMT
View Forum Message <> Reply to Message

Quote:
Koldo version and Konstantin version will build up onto Catalin version.

i'd prefer to build on someones code, ofcorse. deriving and extending a class is best. but in this case, my code change implies making data components public. so i cant derive from andrej's code. on the other hand, it'd be great not to have to duplicate code in the Scatter versions, i.e. the drawing routines, which i pretty like.

nevermind, if it is possible and easy to do, we could split the code, crediting the sources to andrej ofcorse, and i could start to include some of the ideas described earlier.

PS: MVC pattern is nice and great to do tings..but i remember a wise man (mirek saying: simplicity is a virtue on it's own. i think the building pieces for own ideas (Ctrl's, Container's, plug-ins) are simple and structured enough to take advantage of their API where possible..without the need to invent own API just to hide a memeber that has a great API on its own anyway.

Subject: Re: capsulated Scatter Vectors
Posted by kohait00 on Wed, 07 Jul 2010 21:07:00 GMT
View Forum Message <> Reply to Message

hi guys, i managed to 'port' or better say integrate all the stuff that has been done to Scatter so far, so here comes a package again. i havent changed anything else besides this.

next step will be to change the public names and make a wrapper class that is API compatible to current Scatter, if that makes sense.

after that i will start to include the ideas described above..

cheers

File Attachments
1) Scatter2.rar, downloaded 511 times

Subject: My current Diagram
Posted by kohait00 on Wed, 07 Jul 2010 21:23:42 GMT
View Forum Message <> Reply to Message

to show what i mean, here is my current diagram, code is bit patchwork, because i was not too experienced both in coding and Upp , so this is beginners level

use DiagramTest

CMeter: a metering / progressbar / value bar setter with mouse, adjustable alignment and appearance (to some way)

CDiagram: base diagram class, supports movable points, zooming (STRG + left mouse), right mouse menu, moving (SHIFT + left mouse), autoscrolling of grid, auto scaling of grid (a bit different to Scatter) and maybe some more stuff. the code is not very cute though. thats why i prefer to switch / port to Scatter, i like the API much better.

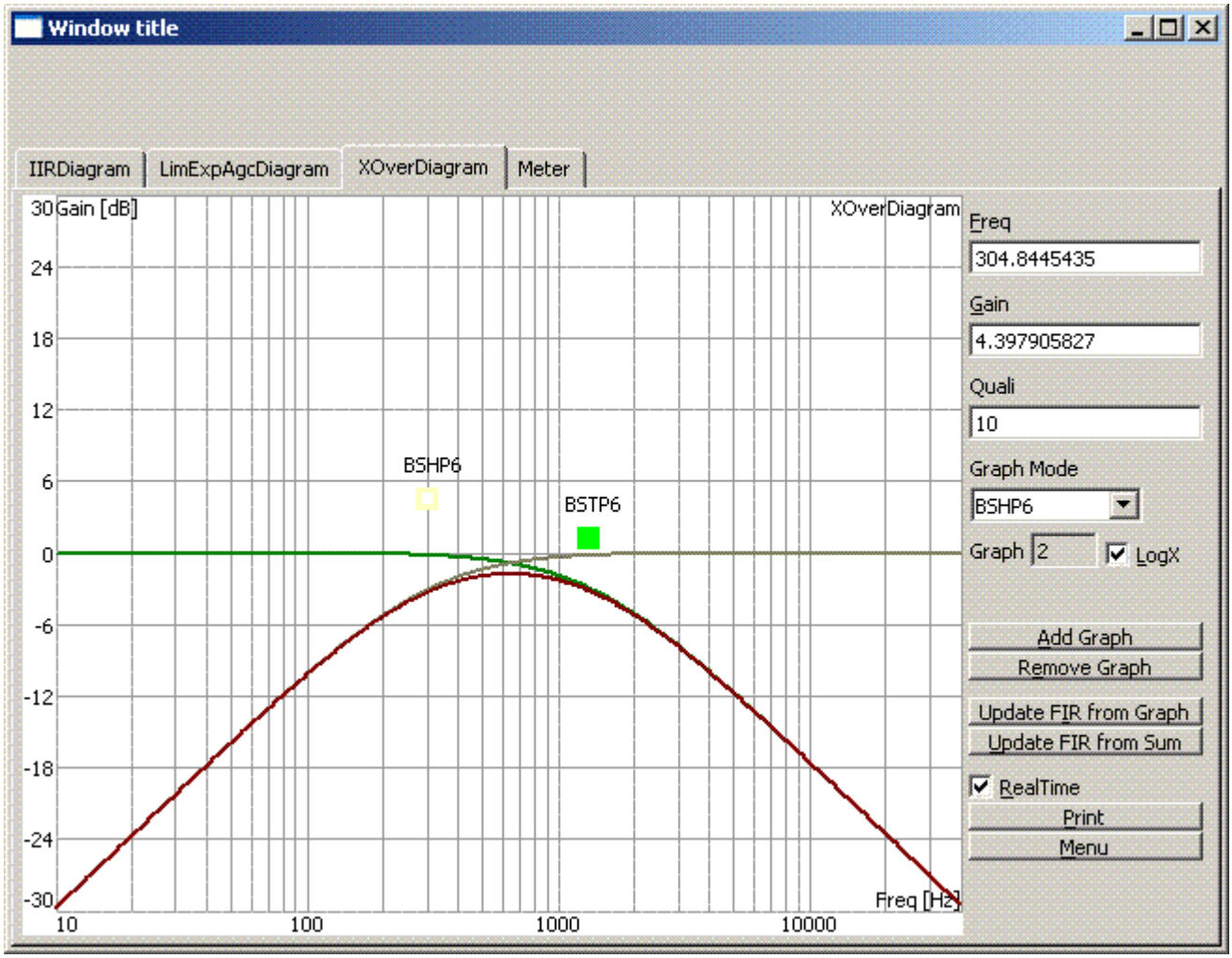
FilterDiagramBase: some special diagrams for parametrising equalizers, crossovers, limiters etc.. this is what CDiagram also can be used for, to have a visual representation *and* parametrising in 1 diagram, play around, inspect the code.

the most usable of them is probably CMeter.

File Attachments
1) DiagramTest.rar, downloaded 492 times

Subject: Re: My current Diagram
Posted by kohait00 on Wed, 07 Jul 2010 21:28:16 GMT
View Forum Message <> Reply to Message

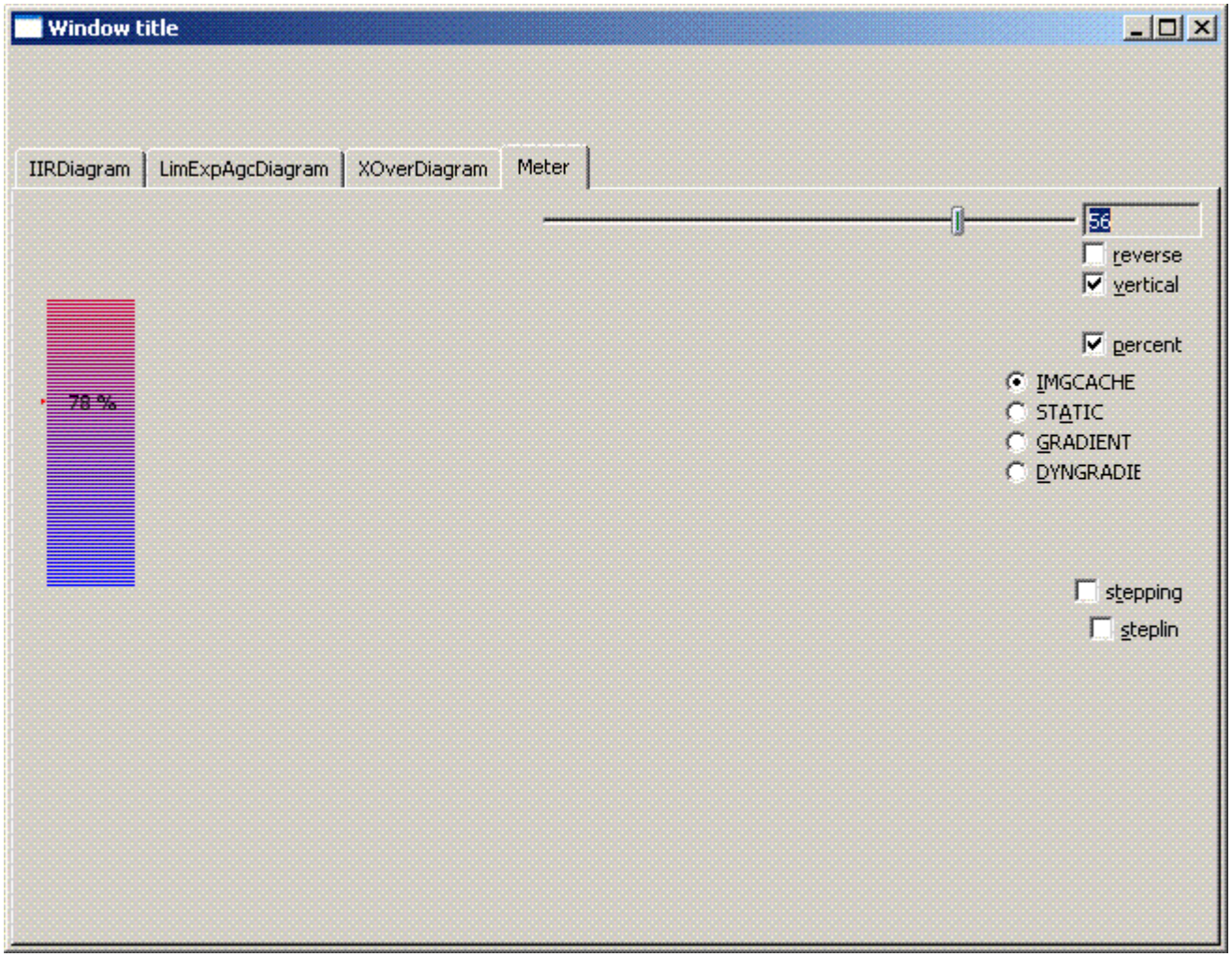
File Attachments
1) xover.GIF, downloaded 1556 times



Subject: Re: My current Diagram
Posted by kohait00 on Wed, 07 Jul 2010 21:31:10 GMT
View Forum Message <> Reply to Message

File Attachments

1) meter.GIF, downloaded 1571 times



Subject: Re: My current Diagram
Posted by kohait00 on Thu, 08 Jul 2010 09:47:44 GMT
View Forum Message <> Reply to Message

ah, it also supports zooming in a different way:
SHIFT + right mouse + moving left = zoom out, move right = zoom in

---

Subject: Scatter based on Callbacks
Posted by kohait00 on Wed, 15 Dec 2010 15:19:29 GMT
View Forum Message <> Reply to Message

hey all,

just came up with the idea of further parametrizing plot function, but this leads to usage of callbacks, instead of function pointers. what about that?

current case:

```
XY para4(double t) {return XY(0,-0.25+0.5*t);}
 scatter3.PlotParaFunction(para4,"VLine",LtRed,6,2);
```

this lets you calculate x and y separated, but what if one wants to specify some parameters, constants etc..?

here Callbacks can be a great deal.
example for a THISBACK, but also thinkable with general static fcuntions with arbitrary signature.
as long as Callback1<float> for t is yielded, it alright.
the additional parameters would be stored in CallbackAction anyway.

```
XY para4(double t, int a, float b) {return XY(a,-0.25*b+0.5*t);}
 scatter3.PlotParaFunction(THISBACK2(&para4, 12, 34.2) ),"VLine",LtRed,6,2);
```

i think Callbacks are much more powerfull here than simple function pointers, though the latter are faster, a neglactible bit..

Subject: Re: Scatter based on Callbacks
Posted by koldo on Wed, 15 Dec 2010 20:56:23 GMT
View Forum Message <> Reply to Message

Hello Kohait00

I do not understand well how to call the callback function from inside PlotParaFunction() with the fixed arguments and the variable (t).

There is of course other awful options like including a pointer to a function or a simple callback plus a Array of Values. However yours looks much better .

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Thu, 16 Dec 2010 05:58:27 GMT
View Forum Message <> Reply to Message

the good thing is, that you wont have to deal with the extra parameters at all since they are simply handled/stored by the CallbackAction (inside the Callback) itseld, Scatter wont even see it, it still thinks it has only a Callback1<float> and calls it like that. the Callback itself will complete the call supplying the additional parameters himself..

i'll try to make a quick demo in Scatter..

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Thu, 16 Dec 2010 06:51:29 GMT
View Forum Message <> Reply to Message

here comes a version, it includes only the very few changes needed. i didnt pay attention to make PlotFunction and PlotParamFunction

meanwhile i found out that parametrizing static functions is not yet supported for more than one param. but with THISBACK it works.

just take a look at the Test package.

i'll try to provide the extensions for Callback soon.

File Attachments
1) Scatter.rar, downloaded 492 times

Subject: Re: Scatter based on Callbacks
Posted by koldo on Thu, 16 Dec 2010 13:41:12 GMT
View Forum Message <> Reply to Message

Hello Kohait00

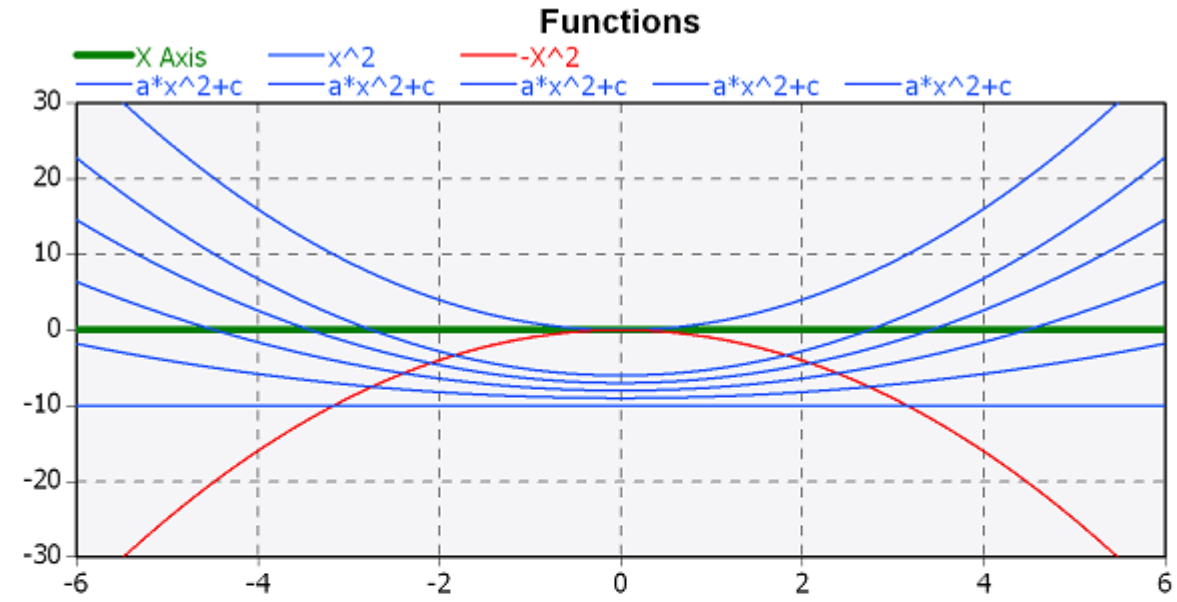
I have done some changes to your code so that it is compatible with existing one .

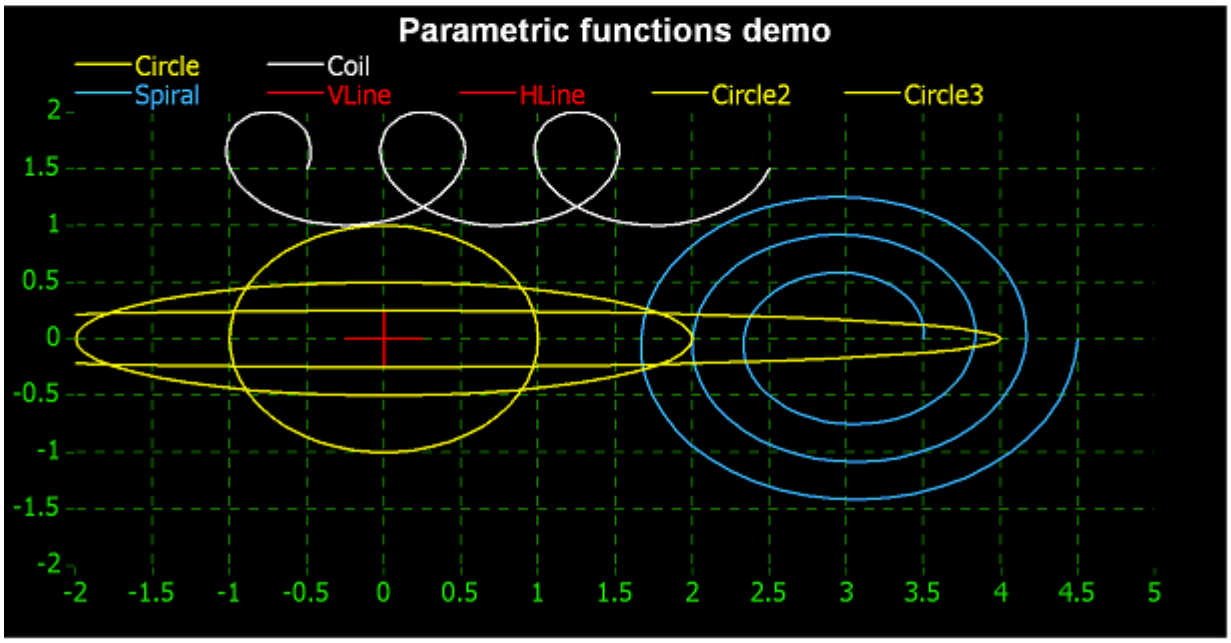
File Attachments
1) Scatter.7z, downloaded 484 times

Subject: Re: Scatter based on Callbacks
Posted by koldo on Thu, 16 Dec 2010 13:41:50 GMT
View Forum Message <> Reply to Message

Some screenshots:

File Attachments
1) dib.PNG, downloaded 1440 times





Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Thu, 16 Dec 2010 14:55:08 GMT
View Forum Message <> Reply to Message

cool

i'm on extending regrouping Callback code blocks, to have them bit more overviewable. i'll post it later, with that, even static parametrized function callbacks are possible.

Subject: Re: Scatter based on Callbacks
Posted by koldo on Thu, 16 Dec 2010 16:15:26 GMT
View Forum Message <> Reply to Message

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Fri, 17 Dec 2010 10:30:06 GMT
View Forum Message <> Reply to Message

i've posted the callbacks changes
http://www.ultimatepp.org/forum/index.php?t=msg&goto=302 32&#msg_30232

with them, it's possible to use

```
scatter3.PlotParaFunction(STDBACK2(para1p, 4.0, 0.25),"Circle",Yellow,6,50);
```

para1p beeing a non thiscall with 2 additional parameters

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Fri, 17 Dec 2010 12:28:34 GMT
View Forum Message <> Reply to Message

i'd do this instead for the old api, not to have to hassle with it in the code, simply map old api to new api like that..

```
inline static void PlotFuncCB(double& y, double x, double (*f)(double)) { y = f(x); }
void PlotFunction(double (*f)(double),const String& legend="", const class::Color&
fcolor=Green,const int& weight=6)
{
 PlotFunction(callback1(&PlotFuncCB, f), legend, fcolor, weight);
}

inline static void PlotParamFuncCB(XY& xy, double t, XY (*pf)(double)) { xy = pf(t); }
void PlotParaFunction(XY (*pf)(double),const String& legend="", const class::Color&
fcolor=Green,const int& weight=6,const int& Np=100)
{
 PlotParaFunction(callback1(&PlotParamFuncCB, pf), legend, fcolor, weight, Np);
}
```

File Attachments

1) Scatter.rar, downloaded 482 times

Subject: Re: Scatter based on Callbacks
Posted by koldo on Sun, 19 Dec 2010 20:10:44 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Fri, 17 December 2010 13:28i'd do this instead for the old api, not to have to hassle with it in the code, simply map old api to new api like that..

```
 inline static void PlotFuncCB(double& y, double x, double (*f)(double)) { y = f(x); }
 void PlotFunction(double (*f)(double),const String& legend="", const class::Color&
fcolor=Green,const int& weight=6)
 {
  PlotFunction(callback1(&PlotFuncCB, f), legend, fcolor, weight);
 }

 inline static void PlotParamFuncCB(XY& xy, double t, XY (*pf)(double)) { xy = pf(t); }
 void PlotParaFunction(XY (*pf)(double),const String& legend="", const class::Color&
fcolor=Green,const int& weight=6,const int& Np=100)
 {
  PlotParaFunction(callback1(&PlotParamFuncCB, pf), legend, fcolor, weight, Np);
 }
```

Hello Kohait00

I agree. It looks better.

Subject: Re: Scatter based on Callbacks
Posted by koldo on Mon, 20 Dec 2010 12:35:11 GMT
View Forum Message <> Reply to Message

Hello Kohait00

New code is not compatible with old one. See the callback functions:

```
double funct1(double x)   {return (x*x);} //Old
void funct1(double& y, double x) {y = (x*x);} //New
```

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Mon, 20 Dec 2010 12:44:40 GMT

View Forum Message <> Reply to Message

dont quite get you here..the provided test application is working with both or isnt it? here comes a fresh rar file, maybe i made some mistake in packing.

this one is my current test package, where both old and new api work..

make sure to use the Callback improvement for this ..see in anther thread ..

File Attachments

1) bazaar.rar, downloaded 499 times

Subject: Re: Scatter based on Callbacks
Posted by koldo on Tue, 21 Dec 2010 16:19:20 GMT
View Forum Message <> Reply to Message

Hello Kohait

The code in bazaar.rar is the same reported:

double funct1(double x) {return (x*x);} //Old
void funct1(double& y, double x) {y = (x*x);} //New
So the problem remains .

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Tue, 21 Dec 2010 16:56:35 GMT
View Forum Message <> Reply to Message

are you sure you have opened the right archive?

this is inside the above bazaar.rar

```
//old syntax
double ofunct1(double x){return (0.5*x*x-5);}
double ofunct2(double x){return (-0.5*x*x-5);}
double ofunct3(double x){return 3;}

//new additional syntax
void funct1(double& y, double x){y = (x*x);}
void funct2(double& y, double x){y = (-x*x);}
void funct3(double& y, double x){y = 0;}
void funct1p(double& y, double x, double a, double c){y = a*(x*x)+c;}
void funct3p(double& y, double x, double m){y = m*x;}
```

```
class Tab2 : public WithTab2<ParentCtrl> {
public:

 typedef Tab2 CLASSNAME;

 Tab2();
};

Tab2::Tab2()
{
 CtrlLayout(*this);
 HSizePos().VSizePos();
 scatter2.SetRange(12,60);
 scatter2.SetXYMin(-6,-30);
 scatter2.SetMajorUnits(2,10);

 scatter2.PlotFunction(&ofunct3,"X Axis",Cyan,18);
 scatter2.PlotFunction(&ofunct1,"x^2", Color(28,85,255),6);
 scatter2.PlotFunction(&ofunct2,"-X^2",Cyan);

 scatter2.PlotFunction(STDBACK(funct3),"X Axis",Green,18);
 scatter2.PlotFunction(STDBACK(funct1),"x^2", Color(28,85,255),6);
 scatter2.PlotFunction(STDBACK(funct2),"-X^2",LtRed);

 scatter2.PlotFunction(STDBACK1(funct3p, 3.0),"X Axis",Green,18);

 for(int i = 0; i < 5; i++)
  scatter2.PlotFunction(STDBACK2(funct1p, 2.0*double(i)/10.0, -10.0+double(i)),"a*x^2+c",
Color(28,85,255),6);
}
```

Subject: Re: Scatter based on Callbacks
Posted by koldo on Thu, 23 Dec 2010 15:19:36 GMT
View Forum Message <> Reply to Message

Hello Kohait00

You are right.

The only problem now is that you include a new Callback so that to apply the changes the new one would have to replace the existing one.

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Thu, 23 Dec 2010 16:05:03 GMT

View Forum Message <> Reply to Message

i'm kinda waiting for them to be included donno, mirek needs to review the thing before commiting, and it's quite huge, takes time.

meanwhile one could spare the new callbacks out, it works with 1 additional parameter (2 wasnt supported before).

using the normal

callback1(&function, param);

instead of the STDBACK()

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Sun, 26 Dec 2010 09:32:40 GMT
View Forum Message <> Reply to Message

looks like mirek wont be including the callback extensions. sad but true nevertheless, the callback1 can be used for it, callback2 and more are out, then ..

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Wed, 22 Jun 2011 12:37:33 GMT
View Forum Message <> Reply to Message

hi koldo..

the Callback changes have been included some time ago..so i've been bold enough to commit the stuff, hope you dont mind (trying to get rid of outstanding changes where possible)

Subject: Re: Scatter based on Callbacks
Posted by koldo on Wed, 22 Jun 2011 21:28:06 GMT
View Forum Message <> Reply to Message

Hello Kohait

You know I do not like the new interface but, as the compatibility with actual format remains, there is no problem for me.

Subject: Re: Scatter based on Callbacks
Posted by kohait00 on Wed, 22 Jun 2011 22:04:30 GMT
View Forum Message <> Reply to Message

the new interface offers some advantages, i.e. using THISBACKs, where implicit parametrizing in the scope of a class instance can be used (invoking a method which has access to instance data).

ofcorse 'double& y' is not cool, but...

nevermind, i like your package

Subject: Re: Scatter based on Callbacks
Posted by koldo on Thu, 23 Jun 2011 14:05:13 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Thu, 23 June 2011 00:04the new interface offers some advantages, i.e. using THISBACKs, where implicit parametrizing in the scope of a class instance can be used (invoking a method which has access to instance data).

ofcorse 'double& y' is not cool, but...

nevermind, i like your package
It is not mine
