

Help for wqp.exe

Contents

Foundation:

Purpose of this program:

Start of program:

Start parameter:

Uniform default configuration:

Table sheets:

Lattices:

List:

The table "dim_offset":

The table "quantities":

Comment:

Algadd:

Sums1:

Sums2:

Graph1:

Graph2:

Config:

Menus:

File:

Cmd:

Mod:

Alg:

Help:

A:....:

S:....:

C or (C):

US:

1 Foundation

Subject of mathematical physics is the study of measurable resp. perceptible reality and to find a suitable model for it. Because measurable reality belongs to past, it is a priori finite and a mathematical model for it also must be a priori finite, i.e. the underlying numerical space and the number of operations on it must be finite.

Nevertheless both can increase without boundary when time increases without boundary (infinite potential).

One possibility for representation of this space are finite dimensional *numerical lattices*, i.e. sets of numbers defined on finite dimensional point lattices. This program handles such numerical lattices. All points are addressed by integer coordinates and the numbers resp. quantities assigned to the points can be complex rational. Subsequently by the term [lattice](#) always this kind of numerical lattice is meant.

2 Purpose of this program

Using this program you can view and edit one or several multidimensional lattices.

(In this version every lattice has up to 30 free (space-like) dimensions plus one "lattice-index" plus one time-like dimension, see below)

Furthermore you can study the results of numerical algorithms on these lattices. The aim is to find algorithms whose results correspond to experimental results better and better.

We have to make small steps: Up to now important physical equations are usually written as partial differential equations which work on continuous (a priori infinite) sets. If we try to find the natural finite basis of those equations, first we have to replace ordinary differential calculus by finite difference framework. This can soon lead to difficult combinatorics, especially in case of interactions across several dimensions. Often there are many possibilities for transfer a differential equation into finite difference framework. This means that there are also many possibilities for its implementation as algorithm (e.g. in [Algadd](#), see below). A numerical test of the chosen possibility gives additional information. So I hope this program can help us to find the right algorithm.

3 Start of program

You can start the program by double clicking on wqp.exe

3.1 Start parameter

It is possible to initially load a [*.aa1 file](#) by using its name as start parameter of the program. For example the call

wqp.exe test.aa1

will start the program and will cause it to initially load the file "test.aa1"

3.1.1 Uniform default configuration

The filename **wqpcfg.aa1** is reserved. This file contains the uniform default configuration (all entries marked by "u" in [Config](#)). If it exists in the current directory, uniform configuration data (but **not** other data like [lattice](#) quantities, [Algadd](#) definition etc.) will be read automatically from it after every input of another file as uniform default configuration. So also if another file is specified as [start parameter](#) of the program, the other file will determine all program data but not [uniform configuration data](#) which are determined by wqpcfg.aa1. Usually this is convenient to force uniform behavior of the program.

If you prefer to determine (like all other data) also [uniform configuration data](#) individually by the [*.aa1 file](#), use the item "read indiv." instead of "open" within the file menu. If you always want to use the individual configuration, simply remove wqpcfg.aa1 from the current directory (e.g. by using the corresponding item "Del Cfg File" in the [File menu](#)).

4 Table sheets

The table sheets provide tools for modification or check of lattice quantities and other program data:

4.1 Lattices

Here you can edit the [lattice\(s\)](#). Initially all lattice quantities are by default 0. This convention (that the quantity at every "untouched" coordinate is 0) allows the handling of the lattice(s) by a computer and is adequate because a lattice exists not a priori but has to be created by execution of branching algorithms on it. The shown tables allow numerical input and (as short cut) increment resp. decrement of the quantity under cursor by pressing the "p" resp. "m" key.

4.1.1 The table "dim offset"

The table "dim offset" permits integer input in the column "offset" which specify the integer coordinates components of the lattice points. These coordinates (2 special and up to 30 free coordinates) specify one single point. If the cursor is in the "offset" column, you can choose by pressing "x" or "y" two coordinates, along which the lattice(s) can be displayed in the table "quantities" (see below) within a subrange. So you can select, visualize and edit every two dimensional subset of the lattice(s). At this the integer coordinates at "x" and "y" in the "offset" column specify the starting coordinates of table "quantities", i.e. the location of the right table within this two dimensional subspace. The free coordinates are simply numbered from 1 to 30, as name we will choose k1...k30, two additional coordinates (named l and n) have a special meaning:

The coordinate **l** is a **lattice-index** resp. name. It can be a powerful aid because it allows to combine an (nearly) arbitrary (!) number of multidimensional lattices, each having its individual lattice-index l. By combining many lattices you can study results of algorithms with total complexity far beyond reach of human brain (initially the situation may be not too complicated - so try to avoid too many different l).

The coordinate **l** has no quantitative meaning: every index l has to be understood as a name (of a lattice). If you combine two lattice, you can use l=0 in all coordinates of the first and l=1 in all coordinates of the second as well as l=-134652 in all coordinates of the first and l=999 in all coordinates for the second. Not the absolute value of l is relevant - but relevant is the fact, that the same index l indicate the same name resp. lattice and different l indicate different names resp. lattices. The reservation of l for this purpose can also

facilitate the discrete implementation of physical equations which connect different kinds of physical quantities (quantities with different names). For implementation of complex algorithms it can be useful to lattice-index different components of a quantity by different l . So for example you can assign indices $l=0,1,2,3,4,5$ to $E_x, E_y, E_z, B_x, B_y, B_z$ for discretization of the Maxwell Equations.

The coordinate n is intentioned as (global) time-like coordinate, i.e. every algorithm which uses quantities whose n is smaller or equal to n_0 ($n \leq n_0$) should only influence quantities whose n is greater than n_0 ($n > n_0$) (in [Algadd](#) handling is even more restrictive: per iteration of [algadd](#) only quantities with $n=n_{\text{last}}$ (maximal n) are used and influence only new quantities with $n=n_{\text{last}}+1$). If the [algadd](#) algorithm is so defined that its iterations of are conform to progress of time, this means from physical point of view that n_{last} always increases if some proper time increases. It represents the fastest possibility of any proper time.

The [Mod](#) menu provides means to move and copy all or some lattice quantities along n . Internally all lattices are organized in map which is ordered along $n=k_0$; all other coordinates ($k_1 \dots k_{30}, l$) have less priority.

4.1.2 The table "quantities"

The table "quantities" represents the [quantities](#) on the specified coordinates of the chosen lattice. It allows input of complex rational numbers (e.g. 1 ; $1i$; $1/2$, $1/2i = (1/2)i$; $1/12+5/16i = 1/12+(5/16)i \dots$) and of complex floating point numbers (e.g. $2E3=2000$; $1.2+3.01E1i = 1.2+30.1i \dots$). Note that the conversion of rational to floating point as irreversible due to possible loss of information (e.g. the conversion of $1/3$ to $0.3333\dots$). Consequent working with rational numbers can preserve exactness. Of course clearness may be lost in case of rational numbers with many digits. If there are more than 10 necessary digits for numerator or denominator of a rational number, the implemented arithmetic automatically converts it to floating point. This is also done to prevent overflow.

The possibility of using complex rational numbers (with imaginary part) is offered as bridge to current concepts and should make usage more convenient. A detailed combinatorial analysis with replacement of complex rational numbers by pairs of rational numbers is always possible by using additional free dimensions and/or additionally [indexed](#) lattices and remembering that complex (rational) numbers can be represented by (rational) 2×2 matrices.

4.2 List

In case of complex multidimensional lattices it may be difficult to estimate occupation of the lattices alone by looking at two-dimensional subspaces. Therefore you can get a complete (along n ordered) listing of all occupied (i.e. non-zero) lattice points in the List table sheet. Scrolling is possible by using the PgUp-, PgDown- and Arrow-Keys. If you click the right mouse button on the listing, you can specify a new n_{start} .

4.3 Comment

Here you can edit a comment to your specification (of [algadd](#) parameters, configuration etc.). If you save your specification (using the file menu), this comment will be also saved.

4.4 Algadd

The concept of this algorithm is simple and nevertheless very general. Among others it can be used to implement (superpositions of) generalized random walks. Generalized because \underline{p} is not restricted to $[0, 1]$ like an usual probability. It can also be a complex number, e.g. a probability amplitude. The second order finite difference along the location coordinate index k of a symmetric Bernoulli random walk is equal to the first order finite difference along \underline{p} like in the Schroedinger equation (cf. footnote on page 13 of <http://arxiv.org/abs/quant-ph/0207045>). This is also valid for linear combinations of these random walks (negative combinations can lead to finite differences) which can be also simulated by algadd.

By using the "algadd" algorithm you can add all *last* quantities of the lattices (all quantities whose n is maximal, i.e. $n=n_{\text{last}}$) to the next (new) lattice points (i.e. with $n=n_{\text{last}}+1$). Because of this convention every existing quantity remains untouched (for documentation of development).

(That's no restriction because parallel to criss-cross addition you can also add the original quantities to the same place in $n_{\text{last}}+1$, i.e. to the same other coordinates except $\text{coo}[0]=k_0=n_{\text{last}}+1$)

Every algadd algorithm is defined by one table which can be edited here.

A menu for making general changes appears after pressing the right mouse button.

Each column of the table represents one entry which specifies one addition. Every entry contains:

aanr:

You can combine several algadds, i.e. algadd algorithms with different definitions by data in several different tables. The index aanr in the left column is the index of the algadd table. You can change it by pressing 'Q' or 'W', by pressing the right mouse button and by using the "Alg" Menu. The other columns contain entries and the first of their cells the index of the entry. In the current version more than 200 entries (additions per iteration) are possible - there is much freedom for experiments.

p:

The *propagator* which is the factor by which every copied

(copy, because the original quantity remains unchanged so that development remains visible, see above)

non-zero quantity with $n=n_{\text{last}}$ is multiplied before addition. The name *propagator* for p is appropriate because it determines the extent of the propagation from the source to the destination lattice.

There is a relationship to the Feynman propagator, but p is elementary because it propagates along minimal dn (resp. dt) and every p represents with its entry exactly one component of the total propagation (which can have many components). Furthermore it does not propagate to $n < n_{\text{last}}$ resp. *past*. - according to the definition of the word *past*.

p is the only complex number in the table, all other numbers are integer. Of course only those entries with non-zero p have relevance (the other are "empty" and can be ignored).

If $0 < p < 1$ and if the sum over all p is 1, we may interpret p as probability in the usual sense. More detailed analysis has to consider that per perception more than one step is necessary. If e.g. 2 steps are necessary, probability can arise from multiplication of two (sums of) p .

ldest:

lattice-index of the destination (you can use it as index of the destination lattice).

l:

lattice-index of the source (you can use it as index of the source lattice), i.e. only if l of the source quantity is the same, then it is multiplied and added.

dn, dk1, ..., dk30:

Relative offset of the destiny coordinate. Because per iteration the last quantities (with $n=n_{\text{last}}$) are added to the "next" or "future" quantities (with $n=n_{\text{last}}+1$), the offset of n always is 1.

Example: Suppose, $n=3$ is the last row of the lattice with non-zero quantities and some entries of algadd, e.g. entries no. 0,1,2 have the following properties:

p	$1/2i$	$-1/2i$	1
-----	--------	---------	-----

ldest	1	0	0
l	0	1	0
dn	1	1	0
dk1	1	-1	0
dk2	0	0	0
dk3	0	0	0
dk4	0	0	0
...			

This means:

Every non-zero quantity with $n=3$ and $l=0$ is

- copied, the copy is multiplied by $(1/2i)$ and added to the quantity with $l=1$, $n=4$, with k_1 greater by 1 and with the same other coordinates.
- copied, the copy is multiplied by 1 and added to the quantity with $l=0$, $n=4$ and the same other coordinates.

and every non-zero quantity with $n=3$ and $l=1$ is

- copied, the copy is multiplied by $-(1/2i)$ and added to the quantity with $l=0$, $n=4$, with k_1 smaller by 1 and with the same other coordinates.

If you read this for the first time, this may seem difficult, but it's only a general approach which offers very much possibilities and often relatively simple possibilities are the most important.

For example suppose that the lattice 0 contains the quantity 1 in one point P and zero quantities anywhere else and the non empty algadd parameters are

p	1	1
ldest	0	0
l	0	0
dn	1	1
dk1	1	-1
dk2	0	0
dk3	0	0
...		

This will produce the Pascal triangle, beginning in P. If we replace p by $1/2$ we get the symmetric binomial distribution, beginning in P.

Similar algadd parameters

p	1	-1
ldest	0	0
l	0	0
dn	1	1
dk1	1	0
dk2	0	0
dk3	0	0
...		

will produce a first order finite difference along coordinate 1.

Another possibility for this is

p	-1/2	1/2
ldest	0	0
l	0	0
dn	1	1
dk1	1	-1
dk2	0	0
dk3	0	0
...		

The differences between these possibilities of discrete differentiation are neglected by analytical considerations. But all possibilities of analytical differentiation can be translated into discrete differentiation.

For example we can identify coordinates 1, 2, 3 with x, y, z and

p	-1/2	1/2	-1/2	1/2	-1/2	1/2
ldest	1	1	2	2	3	3
l	0	0	0	0	0	0
dn	1	1	1	1	1	1
dk1	1	-1	0	0	0	0
dk2	0	0	1	-1	0	0
dk3	0	0	0	0	1	-1
dk4	0	0	0	0	0	0
...						

will produce a three dimensional gradient of the last quantities with $l=0$. The components d/dx , d/dy , d/dz of the gradient are indexed by $l = 1, 2, 3$.

Similarly a possibility for a discrete rot operator (with dest. coordinates indexed by $l = 3, 4, 5$) is

p	-1/2	1/2	1/2	-1/2	1/2	-1/2	-1/2	1/2	-1/2	1/2	1/2	-1/2
ldest	4	4	5	5	3	3	5	5	3	3	4	4
l	0	0	0	0	1	1	1	1	2	2	2	2
dn	1	1	1	1	1	1	1	1	1	1	1	1
dk1	0	0	0	0	0	0	1	-1	0	0	1	-1
dk2	0	0	1	-1	0	0	0	0	1	-1	0	0
dk3	1	-1	0	0	1	-1	0	0	0	0	0	0
dk4	0	0	0	0	0	0	0	0	0	0	0	0
...												

Obviously the algadd algorithm model is rather flexible.

In the above examples the absolute propagator $|p|$ is arbitrary chosen. Its right choice is dependent on the units (dimensions) of the underlying quantities and has to be adapted. In case of analytical considerations the right value of $|p|$ is usually derived from experimental results, e.g. as physical constant. Discrete considerations can give deeper understanding of the realistic combinatorics. Particularly the investigation of dimensionless dependences and the accompanying proportionality factors can be interesting.

Because there is much freedom in designing the variables of the algorithm, it may suffice for an initial step towards a reality conform description. It can only suffice for further steps, if these variables (especially p) don't change with increasing n in physical reality, i.e. if they are always the same as in the initial step. If this is not the case it would be possible to expand the scheme as soon as we have more concrete information. For example we can define variables like p as functions of previous lattice quantities. After all we should not forget, that variables like p , which can be interpreted as probability, are mean average quantities. They are determined during a concrete experiment and it is reasonable to assume that free will has limited influence on that, if conservation laws are not violated (e.g. due to anti symmetrical influence).

4.5 Sums1

For all lattices and two selectable subspaces of it the sums over all lattice quantities q are displayed together with their count (=cnt) and their mean (sum q /cnt). The same is done for their absolute squares $|q|^2$ and for all quotients between the sums over both subspaces and the total sum. When specifying the subspaces or "ranges" 1 resp. 2 by entering the minima and maxima coordinates (Min 1,Max 1 resp. Min 2, Max 2) for every dimension in the left table, you can indicate "no lower limit" by typing "-" as Min, and "no upper limit" by typing "+" as Max.

By typing "X" in the left table (with "Range") you can choose the x-coordinate of both range 1 and range 2 ("x1 x2") for representation in the table sheet [Graph1](#) . By typing "A" you choose it for range 1 ("x1"), by typing "B" for range 2 ("x2"). By typing "Y" you can also define the y-coordinate for the 2D [Graph2](#) . By typing "C" you choose it for range 1 ("y2"), by typing "D" for range 2 ("y2").

You can define and activate several pairs of ranges (Range1, Range2) by changing the index of the defining table. This index is displayed in the first cell of the left column. Similarly like [aanr](#) you can change it by pressing 'Q' or 'W'.

After pressing the right mouse button a menu appears which offers some useful possibilities for quick initialization of the ranges.

4.6 Sums2

For every [lattice-index](#) l the sums over all lattice quantities q and separately also over those with $n=0, 1$ (selectable) and $n_{\text{last}}, n_{\text{last}}-1$ are displayed together with their count ($=\text{cnt}$) and their mean ($\text{sum } q/\text{cnt}$). The same is done for their absolute squares $|q|^2$.

4.7 Graph1

The Sums over the ranges defined in the Sums1 sheet are displayed along (in dependence of) the integer coordinate x specified in the left table of Sums1.

By clicking on the colored check boxes you can plot the graph for

$\text{cntAt} = \text{count at the coordinate(s)}$

(can be greater than 1, if sums1 specifies a multidimensional range)

$\text{Re}(q) = \text{real part of } q \text{ at } x$

$\text{Im}(q) = \text{imaginary part of } q \text{ at } x$

$|q|^2 = \text{squared absolute value of } q \text{ at } x$

$|q| = \text{absolute value of } q \text{ at } x$

for range 1 of left table in the Sums1 sheet (left 5 checkboxes) and

for range 2 of left table in the Sums1 sheet (right 5 checkboxes).

If option "Graph1_divmode" is set to 1, all these quantities are divided by the total fitting count cnt ; if this option is set to 2, the quantities v_1 of range1 are divided by those of range2 with convention $v_1/0:=0$.

If the option "Graph1_BarDoubleWidth" is true and only every second integer of coordinate x is non-zero, the widths of the bars are doubled due to better visibility. In this case a red "d" appears in the checkbox line.

If two Curves are plotted, a white Label with "T" (if normal mode) becomes visible between the checkboxes of range 1 and 2. If you click on it, it changes to "o" and a parameter-plot of both curves is generated, in which $x=y_1$ (y of the first curve) and $y=y_2$ (y of the second curve).

(The symbol "o" is used because this parameter plot option often is used to display circular curves.)

4.8 Graph2

Graph2 is the 2D variant of Graph1: The Sums over the ranges defined in the Sums1 sheet are displayed along (in dependence of) the two integer coordinates x and y specified in the left table of Sums1. Due to 2D visualization the quantity q is not represented by the y -coordinate of a plot but by the brightness of a color.

When clicking on the radio button you can generate the graph for

$\text{cntAt} = \text{count at the coordinate(s)}$

(can be greater than 1, if sums1 specifies a multidimensional range)

$\text{Re}(q) = \text{real part of quantity at } x, y$

$\text{Im}(q) = \text{real part of quantity at } x, y$

$|q|^2 = \text{squared absolute value of } q \text{ at } x, y$

$|q| = \text{absolute value of } q \text{ at } x, y$

for range 1 of left table in the Sums1 sheet (left 5 checkboxes) and

for range 2 of left table in the Sums1 sheet (right 5 checkboxes).

The same can be done for the quantities divided by the total fitting count cnt if option "Graph2_divide_y_by_count" is activated in the [Config](#) sheet.

If option "Graph2_BoxDoubleWidth" is true and only every second integer is non-zero, the widths of the boxes are doubled due to better visibility. In this case a red "d" appears in the checkbox line.

By clicking on the button "set ny:..." in the upper left corner you can activate another plot mode in which multiple 1D plots (one for maximal ny different quantities of y) are possible. Blue plots are activated/deactivated by clicking on one of the entries in the left table. By double clicking red plots are activated. By pressing the right mouse button within the table a menu for simultaneous selection / unselection of all plots is activated. If in the configuration "Graph2a_max_dypix_per_plot" and "Graph2a_max_dypix_percent" are greater than 0, the plots are shifted along y and so better distinguishable; in this case the y scaling is valid only for plot 1.

4.9 Config

Using this sheet you can check and adjust all configuration parameters to your own needs. If the file "wqpcfg.aa1" exists in the current directory, all parameters marked by "u" (in the r-column) are initially read from it. They define the [uniform default configuration](#).

5 Menus

Using the menus you can save data, start algorithms etc:

5.1 File menu

For writing / reading all data (lattice quantities, algorithm parameters, configuration) to / from an file with name *.aa1 in ASCII format. Also for default initialization (cf. [wqpcfg.aa1](#)) and for determination of warning level.

It is possible to specify an initial filename by using it as [start parameter](#) of the program.

5.2 Cmd menu

Especially for global commands on the lattices like copying, moving along [n](#), deletion etc... There is also the possibility to delete those last (n=nlast) lattice points which have the smallest |q|. This can be necessary to branching algorithms without too much memory consumption.

5.3 Mod menu

For global modifications on the lattices like initialization, conversion etc...

5.4 Alg menu

For starting the algorithms, especially for starting of the algadds 0...15, i.e. the algadd algorithms defined by tables 0...15.

Due to quick handling of different definitions of [Algadd](#) you can start algadd 0, 1, 2 directly. For start of the rest of the algadds you first change aanr (the table index) and then use "Start algadd". Note that start is not possible (grey menu items) if the corresponding algadd table contains only zero entries.

If the item "algadd with del" is chosen, after every iteration of algadd all entries of the lattices are deleted except those with maximal [n=nlast](#). This is useful if there is not enough memory in the computer hardware. If a graph along n (defined by Sums1) is wished, the deletion may be restricted to those entries outside Sums1 (option "AA del outside S1"). This is also valid for option "AA Tracking Mode" which additionally automatically deletes the smallest "DelSmallPercent_Default" (->config) percent iteratively every time when more than "Algadd_TM_DelSmallestAfterKB" (->config) are allocated. This can be convenient to keep track of longer memory wasting developments.

The menu also provides means for copying and changing algadd definitions (tables).

5.5 Help menu

For start of help.

5.6 A:...

Shows index of the current algadd table index. By clicking on it you can change it.

5.7 S:...

Shows index of the current Sums1 table index. By clicking on it you can change it.

5.8 C or (C)

C indicates that [wqpcfg.aa1](#) exists in the current directory, (C) indicates that it does not. If the file exists, its contents are used as uniform default configuration, if not, the configuration is used which is saved in individual .aa1 files.

5.9 US

This menu activates the user form with code in the unit wqpus1.cpp. If you want to write own code, e.g. for implementation of own special algorithms, it is recommendable that you place your code in this separate file to ensure well defined program structure also in case of further versions.